

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

В.Е. Бондаренко, О.В. Шутенко, Д.Н. Баклай

**МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕХНИЧЕСКОЙ
ДИАГНОСТИКИ ОБЪЕКТОВ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ**

Учебное пособие
для студентов специальности
«Электроэнергетика, электротехника и электромеханика»,
в том числе для иностранных студентов

Часть вторая

**ДИАГНОСТИКА СОСТОЯНИЯ ОБОРУДОВАНИЯ
ЭЛЕКТРИЧЕСКИХ СЕТЕЙ НА ОСНОВЕ ЛОГИЧЕСКИХ
МЕТОДОВ РАСПОЗНАВАНИЯ, ТЕОРИИ ГРАФОВ,
НЕЧЕТКОЙ ЛОГИКИ И НЕЙРОННЫХ СЕТЕЙ**

Рекомендовано ученым советом
Национального технического университета
«Харьковский политехнический институт»

Харьков
НТУ «ХПИ»
2018

УДК 621.311.1(075)

Б 81

Рецензенты:

Ю.Ф. Павленко, д-р техн. наук, проф., гл. н.с., ННЦ «Институт метрології»;

О.О. Мирошник, д-р техн. наук, проф. ХНТУСХ им. П.Василенко

Рекомендовано ученым советом
Национального технического университета
«Харьковский политехнический институт»
как учебное пособие для студентов специальности
«Электроэнергетика, электротехника и электромеханика»,
протокол № 4 от 28 апреля 2017 г.

У другій частині навчального посібника розглянуто основні методи діагностики стану об'єктів електричних мереж, засновані на використанні алгебри логіки, теорії графів, нечіткої логіки і нейронних мереж. Наведено практичні приклади діагностики стану високовольтного обладнання на основі викладених методів.

Призначено як для студентів та магістрів спеціальності № 141 «Електроенергетика, електротехніка та електромеханіка» за спеціалізацією «Техніка високих напруг та діагностика об'єктів електричних мереж», так і для здобувачів, аспірантів і дослідників, що працюють у сфері діагностики стану високовольтного електроенергетичного обладнання.

Бондаренко В. Е.

Б 81 Математические основы технической диагностики объектов электрических сетей : учеб. пособ. : в 2-х ч. Часть II / В.Е. Бондаренко, О.В. Шутенко, Д.Н. Баклай. – Харьков : НТУ «ХПИ», 2018. – 262 с.

ISBN

Во второй части учебного пособия рассмотрены основные методы диагностики состояния объектов электрических сетей, основанные на использовании алгебры логики, теории графов, нечеткой логики и нейронных сетей. Приведены практические примеры диагностики состояния высоковольтного оборудования на основе изложенных методов.

Предназначено как для студентов и магистров специальности № 141 «Электроэнергетика, электротехника и электромеханика» по специализации «Техника высоких напряжений и диагностика объектов электрических сетей», так и для соискателей, аспирантов и исследователей, работающих в области диагностики состояния высоковольтного электроэнергетического оборудования.

Ил. 103. Табл. 21 Библиогр. 85 назв.

УДК 621.311.1(075)

ISBN

© В.Е. Бондаренко, О.В. Шутенко, Д.Н. Баклай 2018

© НТУ «ХПИ», 2018

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ.....	5
ВВЕДЕНИЕ.....	6
1. ЛОГИЧЕСКИЕ МЕТОДЫ РАСПОЗНАВАНИЯ.....	7
1.1. Основные понятия алгебры логики	7
1.2. Булевы функции	10
1.3. Базис булевой функции и изображающие числа.....	11
1.4. Использование булевых функций для построения диагностических устройств	12
1.5. Использование булевых функций для задач распознавания. Метод сокращенного базиса.....	14
1.6. Построение диагностических тестов на основе булевых матриц.....	16
Вопросы для самоконтроля	31
Литература к разделу.....	32
2. МЕТОДЫ РАСПОЗНАВАНИЯ, ПОСТРОЕННЫЕ НА ТЕОРИИ ГРАФОВ.....	33
2.1. Алгоритм определения состояния объекта, представленного графом.....	34
2.2. Построение тестов для систем, функциональная модель которых может быть представлена ориентированным графом.....	38
2.3. Оценка технического состояния опорно-стержневых изоляторов.....	52
Вопросы для самоконтроля.....	61
Литература к разделу.....	62
3. РАСПОЗНАВАНИЕ СОСТОЯНИЯ ЭЛЕКТРОЭНЕРГЕТИЧЕСКОГО ОБОРУДОВАНИЯ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ.....	63
3.1. Возможности применения теории нечетких множеств для описания различных видов неопределенности	63

3.2. Нечеткие множества	65
3.3. Нечеткая и лингвистическая переменные.....	75
3.4. Алгоритм формирования нечеткого вывода.....	75
3.5. Алгоритмы нечеткого вывода.....	82
3.6. Интерпретация результатов хроматографического анализа растворенных в масле газов с использованием аппарата нечеткой логики.....	87
Вопросы для самоконтроля	113
Литература к разделу.....	113
4. РАСПОЗНАВАНИЕ СОСТОЯНИЯ ЭЛЕКТРОЭНЕРГЕТИЧЕСКОГО ОБОРУДОВАНИЯ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ.....	
4.1. Биологический нейрон и его кибернетическая модель.....	115
4.2. Биологические нейронные сети.....	118
4.3. Искусственный нейрон.....	120
4.4. Персептроны.....	139
4.5. Многослойный персептрон.....	141
4.6. Обучение нейронных сетей.....	151
4.7. Нейронные сети с алгоритмом самообучения.....	171
4.8. Сети с обратными связями.....	176
4.9. Нейронные сети ассоциативной памяти.....	182
4.10. Адаптивные резонансные нейронные сети.....	187
4.11. Нейронная сеть встречного распространения.....	191
4.12. Нечеткие нейронные сети.....	194
4.13. Прогнозирование значений показателей качества трансформаторных масел с помощью нейро-нечеткой системы <i>ANFIS</i>	217
4.14. Оценка возможностей нечетких нейронных сетей для интерпретации результатов хроматографического анализа растворенных в масле газов.....	227
Вопросы для самоконтроля.....	249
Литература к разделу.....	251
Предметный указатель	257

ПРЕДИСЛОВИЕ

Курс «Математические основы технической диагностики» входит в число дисциплин для подготовки специалистов и магистров по специальности № 141 «Электроэнергетика, электротехника и электромеханика» по специализации «Техника высоких напряжений и диагностика объектов электрических сетей» и предусматривает в процессе изучения формирование у студентов фундаментальных представлений о современных методах распознавания состояния сложных технических объектов и систем.

Во второй части пособия изложены основные методы распознавания состояния высоковольтного электроэнергетического оборудования на основе алгебры логики и теории графов. Изложены основные положения теории нечетких множеств, рассмотрены примеры диагностики высоковольтного оборудования с использованием аппарата нечеткой логики. Также в пособии подробно описаны принципы работы нейронных сетей, их конфигурации и алгоритмы обучения. Приведены примеры использования нейронных сетей для решения задач прогнозирования и диагностики состояния реального высоковольтного оборудования.

Содержание учебного пособия соответствует программе курса «Математические основы технической диагностики», читаемого в Национальном техническом университете «Харьковский политехнический институт» студентам специальности 8.05070102 «Электрические системы и сети».

ВВЕДЕНИЕ

Электроэнергетические системы представляют собой сложную часть человеко-машинной технической системы производства, передачи, распределения и потребления электроэнергии. Поскольку элементы подсистемы передачи электроэнергии распределены в пространстве и подвержены влиянию большого числа случайных и неслучайных факторов, проблема надежности электроэнергетических систем становится одной из основных как в теоретическом, так и в практическом аспектах.

Надежность всей электроэнергетической системы определяется надежностью ее отдельных элементов (генерирующих агрегатов, линий электропередачи, коммутационной аппаратуры, устройств защиты, автоматики и др.), схемы (степени резервирования), режима (запасов статической и динамической устойчивости), а также жизнеспособности или живучести системы, то есть ее способности выдерживать системные аварии цепочечного характера без катастрофических последствий. Фактически надежность электроэнергетической системы будет определяться как на стадии ее проектирования и монтажа, так и в процессе эксплуатации.

Одним из способов обеспечения и повышения эксплуатационной надежности высоковольтного оборудования, особенно в условиях его крайней изношенности, является усовершенствование процедур принятия решения при распознавании состояния данного оборудования. В связи с этим подготовка высококвалифицированных специалистов в области диагностики состояния энергетического оборудования является задачей государственной важности. Важнейшим направлением подготовки таких специалистов является овладение ими современными методами и алгоритмами диагностики оборудования. В данной части пособия приведены основные подходы распознавания состояния оборудования электрических систем и сетей на основе использования основных положений алгебры логики, теории графов, нечеткой логики и нейронных сетей.

РАЗДЕЛ 1

ЛОГИЧЕСКИЕ МЕТОДЫ РАСПОЗНАВАНИЯ

Логические методы основаны на установлении логических связей между признаками и состояниями объектов [1], поэтому будут рассмотрены только простые (качественные) признаки, для которых возможны лишь два значения (например, 0 и 1). Точно так же и состояния технической системы (диагнозы) в рассматриваемых методах могут иметь только два значения (наличие и отсутствие). Два значения признака или состояния системы могут быть выражены любыми двумя символами («да» – «нет», «ложь» – «истина», 0–1).

Переменные величины или функции, принимающие только два значения (0 и 1), называются *логическими*, или *булевыми*. Исследованием таких переменных и функций занимается *математическая логика*, имеющая обширные приложения во многих технических проблемах (релейные системы, теория ЭВМ и автоматов и др.). Применительно к задачам распознавания (диагностика) методы математической логики стали использоваться после работ Р. Ледли [2]. Детерминистское описание с помощью двоичных переменных, характерное для логических методов распознавания, является приближенной моделью реальной ситуации. Однако во многих задачах логические методы пригодны для начальных этапов распознавания. Весьма перспективны методы математической логики для второго направления технической диагностики – поиска и локализации неисправностей технических систем.

1.1. Основные понятия алгебры логики

Напомним вкратце некоторые необходимые сведения из булевой алгебры. Логической величиной (или высказыванием) [1] называется величина, которая может принимать только одно из двух значений: 0 или 1, «ложь» или «истина». Логические переменные обычно обозначаются заглавными буквами латинского алфавита.

Логической суммой двух логических переменных A и B (или *дизъюнкцией*) называют логическую величину C :

$$A \vee B = C, \quad (1.1)$$

где \vee – знак логического сложения (дизъюнкции). Часто для логического сложения используется также знак $+$.

Величина C является истинной ($C = 1$), если истинно хотя бы одно из высказываний A и B или оба вместе. Таким образом, для дизъюнкции:

$$\begin{aligned} 1 \vee 1 &= 1 & 1 + 1 &= 1 \\ 0 \vee 1 &= 1 & 0 + 1 &= 1 \\ 1 \vee 0 &= 1 & 1 + 0 &= 1 \\ 0 \vee 0 &= 0 & 0 + 0 &= 0 \end{aligned} \quad (1.2)$$

Логическое суммирование при словесном выражении соответствует союзу «или». Слово «или» может служить и обозначением операции дизъюнкции.

Логическим произведением двух логических величин A и B (или *конъюнкцией*) называют логическую величину C :

$$A \wedge B = C, \quad (1.3)$$

где \wedge – знак логического умножения (конъюнкции).

Для логического умножения используются и обычные знаки умножения. Величина C является истинной только в том случае, когда истинными оказываются высказывания A и B . Таким образом, для конъюнкции:

$$\begin{aligned} 1 \wedge 1 &= 1 & 1 \times 1 &= 1 \\ 0 \wedge 1 &= 0 & 0 \times 1 &= 0 \\ 1 \wedge 0 &= 0 & 1 \times 0 &= 0 \\ 0 \wedge 0 &= 0 & 0 \times 0 &= 0. \end{aligned} \quad (1.4)$$

Логическое произведение в словесном выражении соответствует союзу «и». Слово «и» может служить и обозначением операции конъюнкции.

В булевой алгебре часто используется операция *отрицания* высказывания A . Она обозначается \bar{A} и читается «не A ». Естественно, что истинность и ложность высказываний A и \bar{A} противоположны.

Операции «и», «или» и «не» (конъюнкция, дизъюнкция, отрицание) позволяют составить различные комбинации высказываний, которые называются булевыми функциями и сами, разумеется, являются логическими величинами. Простейшие, наиболее употребительные булевы функции получили название операций *импликации* и *эквивалентности*. Импликация двух высказываний обозначается следующим образом:

$$A \rightarrow B, \quad (1.5)$$

где \rightarrow – знак импликации (иногда используется знак \supset). Соотношение (1.5) читается так: « A влечет B » или «если A , то B ».

Импликация (следование) представляет собой операцию, результат которой C является логической величиной:

$$(A \rightarrow B) = C. \quad (1.6)$$

Импликация может быть выражена с помощью двух основных операций в такой форме:

$$A \rightarrow B = \bar{A} \vee B. \quad (1.7)$$

Таким образом, импликация представляет собой простейшую булеву функцию высказываний A и B . Если импликация [высказывание (1.5)] является истинной, то при истинном A должно быть истинным B (A влечет B). Если A оказывается ложным, то при истинности импликации высказывание B может быть как истинным, так и ложным. Отметим, что из условия (1.5) не следует условие $B \rightarrow A$, то есть высказывания A и B – неравноправны.

Рассмотрим еще *эквивалентность* (или тождественность) двух высказываний, обозначаемую так:

$$A \equiv B, \quad (1.8)$$

где \equiv – знак эквивалентности.

Условие (1.8) представляет собой логическую величину C :

$$(A \equiv B) = C, \quad (1.9)$$

которую можно выразить с помощью элементарных операций:

$$C = A \wedge B \vee \bar{A} \wedge \bar{B} = A \cdot B + \bar{A} \cdot \bar{B}. \quad (1.10)$$

Следовательно, если эквивалентность истинна ($C = 1$), то величины A и B обе или истинны, или ложны. Истинность операций приведена в табл. 1.1.

Таблица 1.1 – Таблица истинности логических операций

A	B	$A \vee B$	$A \wedge B$	$A \rightarrow B$	$A \equiv B$
0	0	0	0	1	1
1	0	1	0	0	0
0	1	1	0	1	0
1	1	1	1	1	1

Отметим еще некоторые простейшие булевы выражения, которые остаются истинными независимо от истинности или ложности входящих в них высказываний. Например, $C = A \vee \bar{A}$, $C = (A \vee \bar{A}) \vee B$. Такие выражения называются *тавтологиями*.

1.2. Булевы функции

Булевой функцией называется логическая величина [1], значение которой зависит от других логических величин:

$$E = f(A, B, C, \dots). \quad (1.11)$$

В этом равенстве E является булевой функцией логических переменных A, B, C, \dots . Как и для обычных функций, функциональная связь f выражает последовательность операций, совершаемых над переменными. Во многих случаях для сокращения записи и наглядности используется операция импликации. Например, выражение $F = \bar{A} \vee B \wedge \bar{C} \vee D$ записывается в виде $F = (A \rightarrow B) \wedge (C \rightarrow D)$.

В практических задачах часто приходится упрощать выражение булевых функций, пользуясь следующими правилами:

- *абсорбции*

$$A + A = A; \quad A \cdot A = A;$$

- *коммутативности*

$$A + B = B + A; \quad A \cdot B = B \cdot A;$$

- *ассоциативности*

$$(A + B) + C = A + (B + C) = A + B + C; \quad (A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C;$$

- *дистрибутивности* умножения относительно сложения

$$A \cdot (B + C) = A \cdot B + A \cdot C;$$

- *дистрибутивности* сложения относительно умножения

$$A + B \cdot C = (A + B) \cdot (A + C);$$

- *отрицания* (правило Моргана)

$$\overline{A \cdot B} = \overline{A} + \overline{B}; \quad \overline{A + B} = \overline{A} \cdot \overline{B};$$

- *правила поглощения*

$$A \cdot B + A \cdot \overline{B} = A; \quad (A + B) \cdot (A + \overline{B}) = A;$$

$$A + A \cdot B = A; \quad A \cdot (A + B) = A.$$

1.3. Базис булевой функции и изображающие числа

Для задач диагностики целесообразно ввести некоторые понятия, связанные с булевыми функциями [2, 3].

Базисом булевой функции будем называть совокупность всех возможных значений ее аргументов (область задания функции). Если булева функция содержит n логических переменных, то базис состоит из 2^n чисел (0 или 1). Записать базис можно разными способами, но условимся придерживаться правил, которые поясним примерами. Для функции трех аргументов $f(A, B, C)$ нормальный базис запишем в такой форме:

$$\begin{array}{l} A \quad 01 \ 01 \ 01 \ 01 \\ B \quad 00 \ 11 \ 00 \ 11 \\ C \quad 00 \ 00 \ 11 \ 11. \end{array} \quad (1.12)$$

Аргументы идут в порядке следования, в первой строке имеются перестановки чисел 0 и 1, во второй – перестановки пар чисел, в третьей – четверок чисел. Число одинаковых цифр в перестановке равно 2^{i-1} ,

где i – номер строки.

Каждую строку базиса можно рассматривать как двоичное число, которое называется *изображающим числом* аргумента и обозначается знаком – «#». Каждый столбец базиса также представляет собой двоичное число, равное номеру столбца (от 0 до 7).

Изображающее число переменной A в базисе (A, B, C)

$$\#A = 01010101.$$

Для изображающих чисел справедливы операции «и», «или» и «не», совершаемые поразрядно. Например,

$$\#A \wedge \#B = (01010101) \wedge (00110011) = 00010001;$$

$$\#A \vee \#B = (01010101) \vee (00110011) = 01110111.$$

Изображающее число булевой функции образуется с помощью соответствующих операций над изображающими числами аргументов. Например, если

$$E = A \vee B \wedge C = A + B \cdot C, \quad (1.13)$$

то, учитывая (1.12), найдем:

$$\begin{aligned} \#E &= 01010101 \vee (00110011) \cdot (00001111) = 01010101 + 00000011 = \\ &= 01010111. \end{aligned}$$

Операция отрицания («не») для изображающего числа означает замену 0 на 1 и наоборот. Например, если $\#A = 01010101$, то $\#\bar{A} = 10101010$. Метод изображающих чисел удобно использовать для проверки тождественности булевых функций.

1.4. Использование булевых функций для построения диагностических устройств

Диагностические устройства представляют собой приборы, моделирующие связи признаков и состояний [1]. Они позволяют автоматически вводить двоичные признаки включением тумблеров и получать сведения о возможных состояниях системы, например, с помощью световых сигналов (загорания лампочек). Связь признаков и состояний систем выражается булевой функцией, которую будем называть булевой диагностической функцией.

Диагностические устройства можно рассматривать как реализацию условий истинности булевой диагностической функции.

Пусть имеются простые (двоичные) признаки k_1, k_2, k_3, \dots , с помощью которых различаются состояния системы. Наличие признака обозначается числом 1, отсутствие признака – числом 0. Таким образом,

$$k_j = \begin{cases} 1 - \text{наличие признака;} \\ 0 - \text{отсутствие признака.} \end{cases} \quad (1.14)$$

Часто наличие или отсутствие признака k_j будем обозначать следующим образом: наличие признака $k_j (k_j = 1)$; отсутствие признака $\bar{k}_j (\bar{k}_j = 0)$. Состояния системы обозначаются D_1, D_2, D_3, \dots , причем наличие состояния соответствует числу 1, а отсутствие – числу 0:

$$D_i = \begin{cases} 1 - \text{наличие } i\text{-го состояния;} \\ 0 - \text{отсутствие } i\text{-го состояния.} \end{cases} \quad (1.15)$$

Рассмотрим в качестве примера систему, имеющую два состояния: исправное D_1 и неисправное D_2 . Пусть состояние системы в рассматриваемом случае описывается четырьмя признаками k_1, k_2, k_3 и k_4 , причем неисправное состояние возникает при наборе признаков, указанных в табл. 1.2.

Таблица 1.2 – Таблица неисправностей

k_1	k_2	k_3	k_4
1	0	0	0
1	1	0	0
0	0	0	1

Во всех остальных случаях система находится в исправном состоянии ($D_1 = 1$). Булева диагностическая функция:

$$F = (\bar{k}_1 \vee k_2 \vee k_3 \vee k_4) \vee (\bar{k}_1 \vee \bar{k}_2 \vee k_3 \vee k_4) \wedge (k_1 \vee k_2 \vee k_3 \vee \bar{k}_4) \equiv D_1 \wedge \bar{D}_2. \quad (1.16)$$

Выражение в первой скобке соответствует первой строке таблицы. Оно составлено в виде логической суммы событий, противоположных тем, которые указаны в первой строке. Естественно, что скобка обращается в

ноль только в том случае, когда появляется набор признаков, содержащихся в первой строке таблицы. Но неисправность возникает в случае, если реализуется любой из трех наборов признаков, то есть любая из строк таблицы. Поэтому выражения, соответствующие трем наборам признаков, соединены знаком логического умножения (конъюнкции). Так как любая часть равенства обращается в нуль при указанных в таблице наборах признаков, а во всех остальных случаях равна 1, то условие истинности $F(F=1)$ соответствует исправному состоянию системы и отсутствию неисправного состояния ($D_1 \wedge \bar{D}_2$). Булева диагностическая функция легко реализуется в диагностических устройствах с помощью логических элементов типа «или» и «и».

1.5. Использование булевых функций для задач распознавания. Метод сокращенного базиса

Задача распознавания при использовании булевых функций формулируется следующим образом [1]: известны логические связи признаков и состояний в виде булевой диагностической функции $F(k_1, k_2, \dots, k_m, D_1, D_2, \dots, D_n)$ и задана булева функция признаков $G(k_1, \dots, k_n)$. Требуется найти такую булеву функцию состояний (диагнозов) $f(D_1, \dots, D_n)$, для которой выполняется условие:

$$G \rightarrow f, \quad (1.17)$$

при

$$F(k_1, k_2, \dots, k_m, D_1, D_2, \dots, D_n) = 1. \quad (1.18)$$

Иными словами, по функции признаков следует определить функцию состояний при условии истинности диагностической функции. Например, пусть в случае двух признаков и двух состояний логические связи таковы:

- 1) при состоянии D_1 появляется признак k_1 ;
- 2) обнаружение признака k_2 свидетельствует об отсутствии состояния D_1 ;
- 3) при состоянии D_2 появляются оба признака.

На основании этих сведений булева диагностическая функция

$$F(k_1, k_2, D_1, D_2) = (D_1 \rightarrow k_1) \wedge (k_2 \rightarrow \bar{D}_1) \wedge (D_2 \rightarrow k_1 \wedge k_2). \quad (1.19)$$

Пример. Пусть при обследовании обнаружено наличие признака k_1 и отсутствие признака k_2 . Тогда булева функция признаков

$$G = k_1 \wedge \bar{k}_2. \quad (1.20)$$

Требуется найти булеву функцию состояний. Воспользуемся общим методом отыскания булевой функции состояния – методом сокращенного базиса. Для этого сначала выписываем полный базис:

номер столбца	0123	4567	89 10 11	12	13	14	15
k_1	0101	0101	01 0 1	0	1	0	1
k_2	0011	0011	00 1 1	0	0	1	1
D_1	0000	1111	00 0 0	1	1	1	1
D_2	0000	0000	11 1 1	1	1	1	1

Далее находим изображающее число для булевой диагностической функции

$D_1 \rightarrow k_1 = \bar{D}_1 \vee k_1$	1111	0101	1111	0101
$k_2 \rightarrow \bar{D}_1 = \bar{k}_2 \vee \bar{D}_1$	1111	1100	1111	1100
$\bar{D}_2 \vee (k_1 \wedge k_2)$	1111	1111	0001	0001
$\# F$	1111	0100	0001	0000

В изображающем числе для функции F , полученном в соответствии с формулой (1.19), единицы стоят в столбцах, отражающих возможные (непротиворечащие F) сочетания признаков и диагнозов. По булевой функции признаков G $k_1 = 1; k_2 = 0$. Такое сочетание признаков при $F = 1$ имеется только в столбцах 1 и 5, для которых $D_1 = 0, D_2 = 0$ и $D_1 = 1, D_2 = 0$. Столбцы 9 и 13, имеющие сочетание $k_1 = 1, k_2 = 0$ [равенство (1.20)], исключаются из рассмотрения, так как в соответствующих столбцах $F = 0$ они противоречат диагностической функции, [то есть условиям (1.19)]. Таким образом, булева функция диагнозов (столбцы 1 и 5):

$$f = (\bar{D}_1 \wedge \bar{D}_2) \vee (D_1 \wedge \bar{D}_2) = \bar{D}_2. \quad (1.21)$$

Преобразование равенства (1.21) легко найти с помощью изображающих чисел. Так как рассмотрение относится к двум булевым величинам, то базис:

$$D_1 \quad 0101$$

$$D_2 \quad 0011.$$

Далее находим:

$$\# \bar{D}_1 \wedge \bar{D}_2 \quad 1000$$

$$\# \bar{D}_1 \wedge D_2 \quad 0100$$

$$\# (\bar{D}_1 \wedge \bar{D}_2) \vee (D_1 \wedge \bar{D}_2) \quad 1100 = \# \bar{D}_2$$

Наличие признака k_1 и отсутствие признака k_2 приводит к отрицанию диагноза D_2 и не позволяет вывести заключение о состоянии D_1 .

Рассмотренный пример позволяет достаточно ясно наметить процедуру отыскания булевой функции состояний. Логические методы позволяют выявить состояния, не противоречащие имеющимся техническим сведениям о связях состояний и признаков. К числу логических методов распознавания могут быть отнесены методы теории графов (см. раздел 2), лингвистические и другие методы.

В общем случае функциям G и F не противоречат несколько возможных состояний, поэтому имеющихся сведений недостаточно для однозначного решения. В подобной ситуации для выбора решения используется метод Байеса или другие методы распознавания.

1.6. Построение диагностических тестов на основе булевых матриц

Теоретический анализ реальных технических систем всегда предполагает определенную идеализацию, при которой выделяются некоторые существенные (для технической диагностики) черты реальных систем и отбрасываются второстепенные, то есть реальная техническая система заменяется некоторой *моделью* [4]. Эта модель должна быть достаточно абстрактной, чтобы ее можно было применять для анализа целого класса

технических систем. В то же время модель должна позволять учитывать все существенные особенности конкретных систем и способов поиска в них отказавших элементов. Одной из первых моделей объектов технической диагностики, которая охватывает обширный класс реальных технических систем, явилась функциональная модель, предложенная Брюле, Джонсоном и Клетским [5].

При построении этой модели предполагается, что систему, рассматриваемую как объект диагностики, можно подразделить на некоторое число в общем случае связанных между собой функциональных элементов. Под функциональным элементом (или просто элементом) понимается часть системы, которая может находиться в одном из двух несовместимых состояний (работоспособна, отказала) и в работоспособном состоянии отвечает требуемой реакцией на определенную совокупность воздействий, в число которых могут входить реакции других элементов.

Воздействия, которые необходимо приложить к работоспособному элементу для получения требуемой (допустимой) реакции, будем называть допустимыми. Реакцию отказавшего элемента назовем недопустимой. В рассматриваемой модели предполагается, что требуемая реакция любого элемента может быть получена только в том случае, если все приложенные к этому элементу воздействия являются допустимыми и элемент работоспособен, а реакция отказавшего элемента не зависит от приложенных к нему воздействий.

Для того чтобы полностью задать функциональную модель системы [4], необходимо:

а) перечислить все возможные для данной системы комбинации одновременно отказавших элементов, то есть задать множество возможных состояний системы;

б) указать, какие комбинации допустимых воздействий необходимо приложить к каждому элементу для получения допустимой реакции;

в) задать схему объекта, на которой указаны все элементы и связи между ними, причем для любой пары связанных элементов должно выполняться следующее условие: если элемент b_i связан с элементом b_j , то допустимая реакция элемента b_i является допустимым воздействием для элемента b_j и, наоборот, недопустимая реакция элемента b_i является недопустимым воздействием для элемента b_j .

При графическом изображении схемы объекта каждый элемент обозначается прямоугольником с некоторым количеством входящих стрелок (входов) и одной выходящей стрелкой (выходом), обозначающей реакцию элемента. Количество входов элемента равно количеству допустимых воздействий, которые необходимо приложить к этому элементу для получения допустимой реакции. Связи между элементами обозначаются линиями, соединяющими стрелки между собой так, чтобы направление стрелок совпадало. При этом выход любого элемента может быть соединен с любым числом входов, тогда как вход любого элемента может быть соединен только с одним выходом. Входы, которые не соединены ни с одним выходом, называются внешними. Эти входы обозначают внешние воздействия, которые подаются на систему.

Пример графического изображения схемы объекта изображен на рис. 1.1. Эта схема содержит пять элементов, каждый из которых имеет два входа и выход.

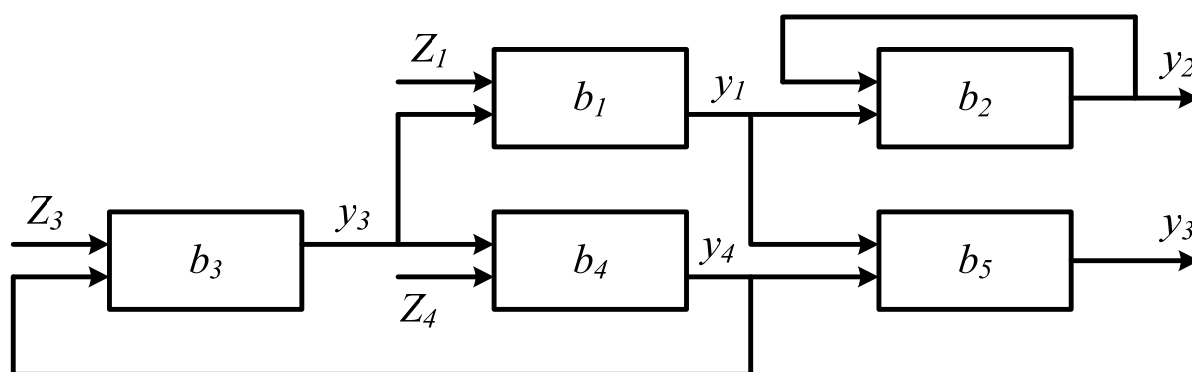


Рисунок 1.1 – Схема объекта диагностики

Внешние воздействия обозначены символами Z_1 , Z_3 , Z_4 , а реакции элементов – символами y_1 , y_2 , y_3 , y_4 , y_5 . Реакции y_1 , y_2 , y_3 , y_4 являются воздействиями для соответствующих элементов. В частности, реакция элемента b_3 является воздействием для элементов b_1 и b_4 , а реакция элемента b_2 является воздействием для этого же элемента.

Располагая графическим изображением схемы объекта, можно не только проследить связи между элементами, но и определить все допустимые воздействия, которые необходимо приложить к работоспособному элементу для получения допустимой реакции. Так, например, для получения допустимой реакции элемента b_5 (рис. 1.1) необходимо приложить к

нему допустимые воздействия y_1, y_4 , а для получения допустимой реакции элемента b_2 необходимо приложить к нему допустимое воздействие Y_1 .

Для того чтобы задать множество возможных состояний системы, достаточно перечислить все возможные комбинации одновременно отказавших элементов. Если система подразделена на N элементов, то каждую такую комбинацию можно представить N -мерным вектором состояния s , v -я ($1 \leq v \leq N$) компонента которого равна 1, если в данной комбинации v -й элемент работоспособен, и равна 0, если в данной комбинации v -й элемент отказал. Например, одновременный отказ элементов b_1 и b_2 в схеме объекта, изображенной на рис. 1.1, представляется вектором состояния $s = (00111)$. N -мерный вектор $s_0 = (11111)$ соответствует такому состоянию системы, в котором все ее элементы работоспособны. В дальнейшем предполагается, что вектор s_0 всегда входит в множество $S = \{s\}$ возможных состояний системы.

Функциональная модель однозначно определяет проверки, которые могут выполняться с целью поиска всех отказавших элементов или, иначе, с целью диагностики состояния системы. Каждая проверка заключается в подаче допустимых воздействий на элементы и контроле ответных реакций. Например, если к элементам b_1 и b_4 (см. рис. 1.1) приложены допустимые воздействия Z_1, y_3, Z_4 и реакции y_2, y_5 являются допустимыми, то элементы b_1, b_2, b_4 и b_5 работоспособны. Появление недопустимой реакции Y_2 означает, что отказал либо элемент b_1 , либо b_2 , либо оба элемента b_1 и b_2 одновременно. Таким образом, для того чтобы задать любую возможную проверку, необходимо указать, какие допустимые воздействия прикладываются и какие реакции контролируются при выполнении данной проверки.

Очевидно, что каждая проверка устанавливает работоспособность или отказ группы из l ($l \geq 1$) элементов. Остальные $N-l$ элементов остаются при этом непроверенными, то есть каждая проверка разбивает множество элементов системы на два подмножества. Поскольку из N элементов можно образовать $2^N - 1$ различных непустых подмножеств, то число возможных различных проверок для системы из N элементов равно $2^N - 1$. Однако в большинстве практических случаев на число и вид возможных проверок накладывается ограничение. Это объясняется тем, что реализация некоторых проверок практически неосуществима или же требует внесения изме-

нений в конструкцию реальных систем. Например, для того, чтобы к элементу b_4 (см. рис. 1.1) можно было прикладывать недопустимое воздействие при условии, что элемент b_3 отказал, необходимо предусмотреть в реальной системе возможность разрыва физической связи между этими элементами.

Рассмотрим некоторую проверку π , которая предусматривает контроль ответной реакции k ($k > 1$) элементов на определенную совокупность допустимых воздействий. Так как в рассматриваемой модели реакция каждого элемента может быть либо допустимой, либо недопустимой, то для k элементов возможно 2^k различных комбинаций реакций этих элементов. Если каждую комбинацию реакций рассматривать как возможный исход проверки π , то можно полагать, что данная проверка имеет 2^k исходов.

Располагая множеством проверок $\Pi = \{\pi\}$, можно для заданной функциональной модели построить так называемую таблицу неисправностей, которая содержит все исходные данные, необходимые для построения диагностических тестов. Для построения таблицы неисправностей необходимо сначала для каждого состояния s из множества S и каждой проверки π из множества Π определить по функциональной модели исход этой проверки в предположении, что система находится в заданном состоянии s . Затем найденные исходы проверки заносятся в таблицу, строки которой соответствуют состояниям из множества S , а столбцы – проверкам из множества Π . Полученная таблица и представляет собой таблицу неисправностей.

Рассмотрим пример построения таблицы неисправностей для модели, схема которой изображена на рис. 1.1. При этом будем считать, что возможен отказ любого одного, и только одного, элемента системы, а каждая возможная проверка предполагает контроль ответной реакции одного из элементов на приложенные к системе допустимые внешние воздействия Z_1, Z_3, Z_4 . В этом случае множество Π содержит пять проверок (по числу выходов элементов).

Обозначим через π_i ($1 \leq i \leq 5$) проверку, предусматривающую контроль реакции i -го элемента. Очевидно, что каждая проверка π_i может иметь только два исхода: положительный (**1**), если реакция i -го элемента допустима, и отрицательный (**0**), если реакция i -го элемента недопустима.

Пусть все элементы системы работоспособны. Очевидно, что в этом случае исходы всех «проверок» положительны. Предположим теперь, что отказал элемент b_3 . Тогда реакция этого элемента будет недопустимой. Поскольку эта реакция является воздействием для элементов b_1 и b_4 , то реакция этих элементов, а следовательно, и реакции элементов b_2, b_5 также будут недопустимыми. Таким образом, отказ элемента b_3 вызывает отрицательные исходы всех проверок. Так как реакция элемента b_4 является воздействием для b_3 , то отказ b_4 также вызывает отрицательные исходы всех проверок. При отказе элемента b_1 отрицательные исходы будут иметь проверки π_1, π_2, π_5 . При отказе элемента b_2 отрицательный исход будет иметь только проверка π_2 , а при отказе элемента b_5 отрицательный исход будет иметь только проверка π_5 . Итак, мы определили исходы всех проверок для всех возможных состояний. Занося эти исходы в графы, получим таблицу неисправностей для заданной схемы объекта (табл. 1.3).

Таблица 1.3 – Таблица неисправностей объекта из рис. 1.1

s	π_1	π_2	π_3	π_4	π_5
1 1 1 1 1	1	1	1	1	1
0 1 1 1 1	0	0	1	1	0
1 0 1 1 1	1	0	1	1	1
1 1 0 1 1	0	0	0	0	0
1 1 1 0 1	0	0	0	0	0
1 1 1 1 0	1	1	1	1	0

Из построенной таблицы видно, что отказ элемента b_3 проявляется так же, как и отказ элемента b_4 . Поэтому посредством выполнения всех возможных проверок нельзя определить, какой из этих элементов отказал. Такие отказы будем называть *неразличимыми*. Неразличимым отказам соответствуют тождественные строки таблицы неисправностей.

Для устранения неразличимых отказов необходимо или расширить множество Π , вводя в него дополнительные проверки, или изменить структуру системы. Например, если в схеме объекта, изображенной на рис. 1.1, предусмотреть возможность разрыва обратной связи, идущей от элемента b_4 к элементу b_3 , то отказы этих элементов становятся различимыми без

введения дополнительных проверок. Однако при этом на элемент b_3 необходимо подавать извне допустимое воздействие Y_4 .

Функциональные модели, в которых каждая возможная проверка предполагает контроль реакции одного из элементов на приложенные к системе допустимые внешние воздействия, обладают рядом особых свойств. В частности, если в этих моделях допускается отказ любого одного, и только одного элемента, то схему объекта можно представить в виде ориентированного графа. Кроме того, если в схемах объектов моделей отсутствуют обратные связи (то есть изменение реакции любого элемента не приводит к появлению недопустимого воздействия на его входе, поступающего с выхода какого-либо другого элемента), то отказы любого другого элемента являются различимыми [6]. Отметим также, что в случае произвольного числа отказавших элементов для функциональных моделей без обратных связей существует такая последовательность проверок, которая позволяет выявить любую комбинацию отказавших элементов [7].

Рассмотренная выше функциональная модель применялась, главным образом, для диагностики состояния электронных систем, осуществляющих преобразование непрерывных сигналов. Однако для целого ряда технических систем предположения, на которых основана эта функциональная модель, не выполняются. В частности, в некоторых системах недопустимая реакция одного или нескольких элементов не влечет за собой недопустимую реакцию других элементов. Такое явление, например, имеет место в системах с резервированием, в комбинационных схемах и конечных автоматах.

При построении таблиц неисправностей системы иногда предполагают, что для каждого возможного состояния известна функциональная связь между внешними воздействиями и реакцией отдельных элементов или системы в целом. Если считать, что каждая проверка состоит в определении реакции системы или ее элементов на заданную совокупность воздействий, то знание этой связи позволяет найти исход данной проверки для любого состояния системы и тем самым – построить таблицу неисправностей. Следует отметить, что каждая конкретная таблица неисправностей содержит исходные данные, необходимые при построении диагностического теста либо программы диагностики конкретной технической системы. Если же рассматривать такие задачи технической диагностики

как разработку методов построения диагностических тестов и оптимальных программ диагностики, то можно не учитывать специфику отдельных систем и полагать, что любая таблица, которая задает связь между множеством состояний системы, множеством возможных проверок и множеством исходов этих проверок, является таблицей неисправностей. При таком подходе таблицу неисправностей можно рассматривать как некоторую достаточно абс-трактную модель объектов диагностики, охватывающую практически все реальные технические системы.

1.6.1. Тесты для проверки работоспособности и локализации неисправностей

Релейная схема с p реле и N контактами при допущении обрывов и коротких замыканий контактов в произвольных комбинациях будет иметь $3^N - 1$ различных неисправностей и 2^p состояний схемы. Таким образом, уже для небольших p и N полная проверка схемы становится практически невозможной. В связи с этим возникает необходимость построения экономичных тестов, позволяющих по сравнительно небольшому числу испытаний (проверок) определять место и характер неисправности.

В дальнейшем с двумя основными задачами диагностики технических устройств (контроль работоспособности и поиск неисправностей) мы будем связывать построение тестов для проверки работоспособности и тестов для локализации неисправностей.

Согласно построенной выше модели для произвольного объекта диагностики задано:

а) множество попарно различных состояний объекта $S = \{s_i\}, i = \overline{1, n}$. Каждому состоянию объекта диагностики соответствует либо одна неисправность, либо некоторая комбинация неисправностей, причем различным состояниям объекта соответствуют различные неисправности;

б) множество попарно различных проверок $\Pi = \{\pi_j\}, j = \overline{1, m}$. и множество $A = \{a\}$ результатов проверок;

в) множество Γ всех возможных троек $\{s_i, \pi_j, a_{ij}\}$, где $s_i \in S, \pi_j \in \Pi, a_{ij} \in A$, причем каждой паре $\{s_i, \pi_j\}$ в тройке Γ поставлен в соответствие один, и только один, элемент из множества A . На содержательном уровне это условие означает, что при данном состоянии s_i диагностируемого объекта результат данной проверки π_j определяется однозначно.

При таких допущениях диагностика объекта сводится к определению его состояния по результатам, полученным при поочередном выполнении проверок из Π (предполагается, что за время, необходимое для выполнения всех проверок из Π , объект диагностики не может перейти из одного состояния в другое).

Пусть задана таблица состояний диагностируемого объекта, строки которой индексируются всеми возможными состояниями, а столбцы – всеми возможными проверками. Состояния $s_i = (a_{i1}, \dots, a_{ij}, \dots, a_{im})$ и $s_r = (a_{r1}, \dots, a_{rj}, \dots, a_{rm})$ будем называть различимыми на множестве Π (обозначается $s_i Ts_r$), если существует, по крайней мере, одна такая проверка $\pi_j \in \Pi$, что $a_{ij} \neq a_{rj}$. В противном случае состояния s_i и s_r называются неразличимыми. Из этого определения следует, что если два произвольных состояния неразличимы, то по результатам всех проверок из Π нельзя установить, в каком из двух данных состояний находится диагностируемый объект. Для различимости этих состояний необходимо произвести расширение множества возможных проверок, доопределяющее данные состояния. При последующем изложении будем предполагать, что все строки таблицы состояний попарно различны, то есть неразличимые состояния отсутствуют.

Обозначим через R множество неупорядоченных пар различимых состояний из S , причем пары (s_i, s_r) исключаются.

Множество $T \in \Pi$ назовем диагностическим тестом (относительно Γ , S , R), если, каковы бы ни были пары, $(s_i, s_r) \in R$, $s_i Ts_r$ на множестве T .

Таким образом, диагностический тест определяется множеством проверок $T \in \Pi$, на котором все состояния диагностируемого объекта различимы. Иными словами, для того, чтобы провести диагностику объекта, достаточно лишь произвести проверки, входящие в тест.

Тест T назовем *элементарным*, если всякое подмножество $T' \in T$ не является диагностическим тестом. Тест, содержащий минимальное число проверок, будем называть *минимальным* диагностическим тестом. В дальнейшем нас будут интересовать минимальные тесты или тесты, близкие к минимальным.

При фиксированном Γ понятие теста зависит от множеств S , R . Если при определении теста в множество S включается состояние s_0 – «объект работоспособен», а в множество R лишь пары вида (s_0, s_i) , $i = 1, 2, \dots, n$, то проведение проверок, входящих в такой тест, позволяет обнаружить нали-

чие любой неисправности; то есть определяемый таким образом тест будет тестом для проверки работоспособности.

Если же в множество S состояние s_0 не включается, а множество R содержит все возможные пары различных состояний, то тест, определяемый относительно Γ, S, R , позволяет локализовать любую неисправность, то есть будет являться тестом для локализации неисправности.

1.6.2. Задача построения тестов и ее интерпретации

Перейдем теперь непосредственно к формулированию задачи построения тестов вообще и построения минимальных тестов в частности [4]. Рассмотрим множество $E = \{e_k\}$, $k = \overline{0, 2^m - 1}$ всех двоичных наборов длины m , то есть $e_k = (\sigma_1^k, \sigma_2^k, \dots, \sigma_m^k)$, $\sigma_j^k \in \{0, 1\}$ и число элементов в наборе e_k равно числу проверок в множестве $\Pi = \{\pi_j\}$, $j = \overline{1, \dots, \delta}$.

Будем считать, что набор e_k определяет подмножество проверок $\Pi_k \in \Pi$, если для всех j ($j = \overline{1, \dots, \delta}$) соблюдаются следующие условия:

$$\begin{aligned} \text{если } \sigma_j^k = 1, \text{ то } \pi_j \in \Pi_k, \\ \text{если } \sigma_j^k = 0, \text{ то } \pi_j \notin \Pi_k. \end{aligned} \tag{1.22}$$

Пусть, например, $\Pi = \{\pi_1, \pi_2, \pi_3, \pi_4\}$ и $e_k = 0101$. Тогда $\Pi = \{\pi_2, \pi_4\}$. Очевидно, что $e_0 = 00, \dots, 0$ определяет пустое множество проверок, а набор $e_{2^m-1} = 11, \dots, 1$ определяет все множество Π .

Набор e_k определяет тест, если Π_k есть тест относительно Γ, S, R . Пусть далее $(s_i, s_r) \in R$, где $s_i = (a_{i1}, \dots, a_{ij}, \dots, a_{im})$ и $s_r = (a_{r1}, \dots, a_{rj}, \dots, a_{rm})$. Построим двоичный набор $e_{ir} = (\sigma_1^{ir}, \dots, \sigma_j^{ir}, \dots, \sigma_m^{ir})$, такой что для всех $j = \overline{1, \dots, m}$:

$$\begin{aligned} 0, \text{ если } a_{ij} = a_{rj}; \\ 1, \text{ если } a_{ij} \neq a_{rj}. \end{aligned} \tag{1.23}$$

Набор e_{ir} определяет подмножество $\Pi_{ir} \in \Pi$ проверок, на котором состояния s_i и s_r различимы. Выполнив указанное построение для всех пар из множества R , получим некоторое множество E_0 двоичных наборов.

Располагая множеством E_0 , построим матрицу M , в которой каждая строка представляет собой набор из E_0 , причем число строк в матрице M равно числу наборов в множестве E_0 и порядок строк безразличен.

Число столбцов в матрице M равно числу проверок в множестве Π , и номер каждого столбца совпадает с номером проверки из Π . Так как все пары состояний из R различимы, то матрица M не содержит строк, состоящих только из нулей. Построенную таким образом матрицу будем называть булевой матрицей.

В терминах булевых матриц можно сформулировать следующим образом две основные задачи построения диагностических тестов, которые в последующем называются первой и второй задачами соответственно.

1. Задача построения минимального диагностического теста: для данной булевой матрицы найти минимальное множество столбцов, так чтобы каждая строка имела значение, равное 1, по крайней мере, в одном из столбцов матрицы.

2. Задача построения всех элементарных диагностических тестов: для данной булевой матрицы найти множество P всех множеств столбцов так, чтобы для любого элемента P_i множества P нашелся в каждой строке, по крайней мере, один элемент, равный 1, в столбце, принадлежащем P_i , и так, чтобы вычеркивание любого столбца из P_i приводило к потере указанного свойства.

Все решения первой задачи могут быть получены из решения второй задачи. Пример булевой матрицы приведен в табл. 1.4.

Первая задача для табл. 1.4 имеет три решения:

1) $\{\pi_1, \pi_2, \pi_4\}$; 2) $\{\pi_1, \pi_4, \pi_6\}$; 3) $\{\pi_2, \pi_4, \pi_5\}$.

Решение второй задачи в дополнение к перечисленным трем имеет еще три решения:

4) $\{\pi_2, \pi_3, \pi_5, \pi_6\}$; 5) $\{\pi_1, \pi_2, \pi_3, \pi_5\}$; 6) $\{\pi_3, \pi_4, \pi_5, \pi_6\}$.

Таблица 1.4 – Пример булевой матрицы

	π_1	π_2	π_3	π_4	π_5	π_6
1	1	0	0	0	1	0
2	1	1	1	0	0	0
3	1	0	0	1	0	1
4	0	1	0	0	0	0
5	0	1	0	0	0	1
6	0	0	1	1	0	0
7	0	0	0	1	1	0

Вторая задача допускает следующую алгебрологическую интерпретацию. Каждый столбец булевой матрицы представим булевой переменной, а каждую строку – булевой суммой (дизъюнкцией) этих переменных (в зависимости от того, равна переменная 1 или 0, она входит или не входит в указанную дизъюнкцию). Содержательно это означает, что элементарный тест, являющийся решением задачи, должен содержать, по крайней мере, одну проверку, по которой пара состояний $(s_i, s_r) \in R$, соответствующая данной строке булевой матрицы, различима. Указанную дизъюнкцию запишем для каждой строки булевой матрицы. Очевидно, что для того чтобы удовлетворить определению теста, необходимо образовать произведение (конъюнкцию) полученных дизъюнкций, поскольку сконструированная таким образом булева функция вида $\&\vee$ будет истинна тогда, и только тогда, когда одновременно все пары состояний, принадлежащие множеству R , будут различимы. Применяя дистрибутивный закон (свойство согласованности двух бинарных операций, определённых на одном и том же множестве), а также известные преобразования алгебры логики, получаем:

$$\pi_k \& \pi_k = \pi_k \text{ и } \pi_k \vee Q \& \pi_k = \pi_k, \quad (1.24)$$

где Q – произвольная конъюнкция.

Преобразуем полученное выражение к виду $\vee\&$ (дизъюнкция конъюнкций), причем полученная булева сумма не будет содержать лишних слагаемых. Теперь мы можем сформулировать следующее предложение.

Элементы, входящие в одно слагаемое (дизъюнкцию) выражения $\vee\&$, порождают множество, которое является элементарным диагностическим тестом.

Действительно, каждое слагаемое выражения $\vee\&$ обуславливает истинность исходной булевой функции вида $\&\vee$ (попарную различимость всех состояний диагностируемого объекта), поскольку оно имеет общий с каждым сомножителем в выражении $\&\vee$ по крайней мере один элемент. Проиллюстрируем смысл введенных преобразований на примере булевой матрицы (см. табл. 1.4). Имеем:

$$\begin{aligned}
f &= (\pi_1 \vee \pi_5) \cdot (\pi_1 \vee \pi_2 \vee \pi_3) \cdot (\pi_1 \vee \pi_4) \cdot (\pi_2 \vee \pi_4) \times \\
&\times (\pi_2 \vee \pi_6) \cdot (\pi_3 \vee \pi_4) \cdot (\pi_4 \vee \pi_5) = \pi_1 \pi_2 \pi_4 \vee \pi_1 \pi_4 \pi_6 \vee \\
&\vee \pi_2 \pi_4 \pi_5 \vee \pi_2 \pi_3 \pi_5 \pi_6 \vee \pi_1 \pi_2 \pi_3 \pi_5 \vee \pi_3 \pi_4 \pi_5 \pi_6.
\end{aligned}
\tag{1.25}$$

В последнем выражении каждое произведение порождает элементарный тест, причем в этом выражении содержатся все возможные элементарные тесты. Отметим, что преобразование $\&\vee$ к $\vee\&$ чрезвычайно трудно уже для сравнительно небольшого числа переменных. Таким образом, решение второй задачи заключается в нахождении эффективных способов преобразования булевых функций из произведения сумм в сумму произведений.

Поскольку все минимальные диагностические тесты (их может быть несколько) находятся среди элементарных тестов, решение первой задачи сводится к выбору после преобразования $\&\vee$ к $\vee\&$ элементарных произведений минимальной длины. Ввиду трудоемкости указанного преобразования при последующем изложении мы будем интересоваться другими способами построения минимальных диагностических тестов, которые исключают необходимость подобного преобразования.

Рассмотрим диагностируемый объект, состоящий из пяти функциональных элементов (b_1, b_2, b_3, b_4, b_5), соединенных по схеме рис. 1.2, y_1, y_2, y_3, y_4, y_5 – выходы функциональных элементов (направление внешних воздействий Z_1 и Z_2 указано стрелками).

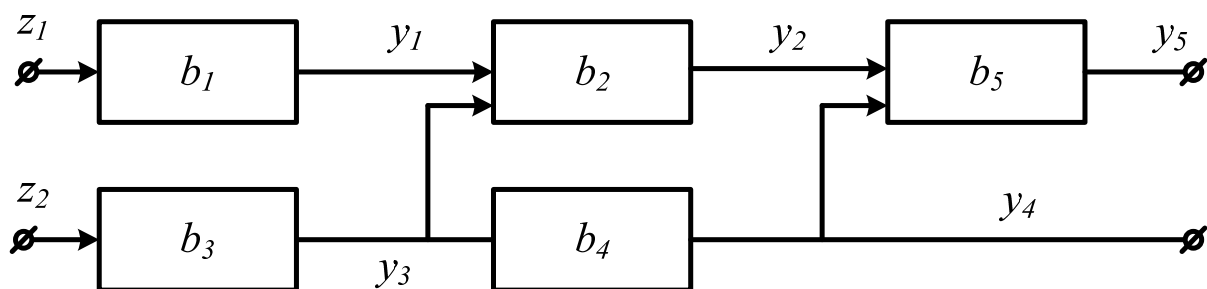


Рисунок 1.2 – Схема объекта диагностики

Будем предполагать, что ненормальная работа объекта диагностики вызывается наличием только одной неисправности, которая локализуется внутри функционального элемента. Тогда множество возможных состояний диагностируемого объекта будет состоять из пяти элементов:

$$S = \{s_i\}, i = \overline{1, 5};$$

$s_1(b_1)$ – неисправен элемент b_1 ;

$s_2(b_2)$ – неисправен элемент b_2 ;

.....

$s_5(b_5)$ – неисправен элемент b_5 .

Предположим, что неисправность функционального элемента приводит к выходу из зоны допуска его выходного сигнала, по которому мы осуществляем проверку состояния данного функционального элемента. Считаем, что выход из зоны допуска входа функционального элемента приводит к выходу из зоны допуска его выхода. При этих предположениях множество возможных проверок определится пятью элементами:

$$\Pi = \{\pi_i\}, i = \overline{1, 5};$$

$\pi_1(y_1)$ – проверка состояния функционального элемента b_1 по его выходу y_1 ; $\pi_5(y_5)$ – проверка состояния функционального элемента b_5 по его выходу y_5 .

Множество результатов проверок будет состоять из двух элементов $A = \{0, 1\}$:

0 – выход функционального элемента не в норме;

1 – выход функционального элемента в норме.

Тогда таблица состояний (табл. 1.5) диагностируемого объекта запишется в следующем виде.

Таблица 1.5 – Таблица неисправностей объекта из рис. 1.2

	π_1	π_2	π_3	π_4	π_5
$s_1(b_1)$	0	0	1	1	0
$s_2(b_2)$	1	0	1	1	0
$s_3(b_3)$	1	0	0	0	0
$s_4(b_4)$	1	1	1	0	0
$s_5(b_5)$	1	1	1	1	0

Просматривая таблицу, можно убедиться, что проверка π_5 не несет никакой информации о состоянии диагностируемого объекта, и, следовательно, она может быть исключена из дальнейшего рассмотрения. В ре-

результате попарного сравнения состояний получим следующую булеву матрицу (табл. 1.6).

Таблица 1.6 – Булева матрица для объекта из рис. 1.2

	π_1	π_2	π_3	π_4
(s_1, s_2)	1	0	0	0
(s_1, s_3)	1	0	1	1
(s_1, s_4)	1	1	0	1
(s_1, s_5)	1	1	0	0
(s_2, s_3)	0	0	1	1
(s_2, s_4)	0	1	0	1
(s_2, s_5)	0	1	0	0
(s_3, s_4)	0	1	1	0
(s_3, s_5)	0	1	1	1
(s_4, s_5)	0	0	0	1

Выражение $\&\vee$ будет иметь следующий вид:

$$f = \pi_1 \cdot (\pi_1 \vee \pi_3 \vee \pi_4) \cdot (\pi_1 \vee \pi_2 \vee \pi_4) \cdot (\pi_1 \vee \pi_2) \cdot (\pi_3 \vee \pi_4) \times \\ \times (\pi_2 \vee \pi_4) \cdot \pi_2 \cdot (\pi_2 \vee \pi_3) \cdot (\pi_2 \vee \pi_3 \vee \pi_4) \cdot \pi_4. \quad (1.26)$$

По виду самой функции непосредственно устанавливаем, что минимальным подмножеством проверок, обеспечивающим попарную различимость всех состояний, является диагностический тест:

$$T_1 = (\pi_1, \pi_2, \pi_4). \quad (1.27)$$

Первая задача построения минимальных диагностических тестов допускает следующую известную интерпретацию [4]: элементами множества Π булевой матрицы служат простые импликанты (импликантой $g(a, b, \dots)$ функции $f(a, b, \dots)$ называется такая функция, для которой справедливо $(\bar{g} \vee f) = 1$) некоторой булевой функции, элементами множества R – элементы булева пространства аргументов данной функции, на которых функция принимает значение «1» (элемент булевой матрицы $C_{ij} = 1$, если i -й элемент булева пространства входит в интервал, соответствующий j -му импликанту, и $C_{ij} = 0$); в противном случае задача состоит в получении всех минимальных дизъюнктивных нормальных форм (ДНФ) булевой функции.

Построению нетривиальных алгоритмов минимизации булевых функций в классе ДНФ посвящены исследования многих авторов, наиболее подробно эти методы изложены в [4]. Все подобные алгоритмы подразделяются на два этапа:

1) однозначная часть процесса – построение некоторой сокращенной ДНФ;

2) неоднозначная часть – построение тупиковой ДНФ (ДНФ, из которой нельзя исключить никакую элементарную конъюнкцию).

Общим недостатком таких методов является то, что при построении тупиковых форм не производится анализа удаляемых конъюнкций на вхождение в некоторые минимальные формы. Поэтому после получения тупиковой формы невозможно выяснить, является ли построенная тупиковая форма минимальной. Приходится строить все возможные тупиковые формы и выбирать среди этих форм минимальные. Очевидно, что такие алгоритмы также связаны с просмотром значительного числа вариантов.

Учитывая сделанные замечания, иногда целесообразно удовлетвориться построением не минимального теста, а теста, достаточно хорошего в том смысле, что его длина не очень сильно превышает длину минимального теста.

В электроэнергетике логические методы распознавания нашли широкое применение при диагностике систем автоматики, телемеханики и связи. Так, в работе [8] приводятся тесты для обнаружения коротких замыканий в логических схемах и для контроля исправности электрического монтажа.

Вопросы для самоконтроля

1. Перечислить и дать пояснение основным логическим операциям.
2. Дать определение булевой функции. Основные правила преобразования булевых функций.
3. Базис булевой функции и изображающие числа.
4. Булева диагностическая функция и таблица неисправностей.
5. Диагностические устройства на основе булевых функций.
6. Метод сокращенного базиса.
7. Модель объектов диагностики и принципы ее построения.

8. Построение таблицы неисправностей.
9. Диагностические тесты для проверки работоспособности.
10. Минимальный диагностический тест.
11. Построение диагностических тестов.
12. Основные проблемы построения минимальных диагностических тестов.

Литература к разделу 1

Основная

1. Биргер И.А. Техническая диагностика / И.А. Биргер. – М. : Машиностроение, 1978. – 240 с.
2. Ледли Р. Программирование и использование вычислительных машин / Р. Ледли. – М. : Мир, 1966. – 642 с.
3. Папернов А.А. Логические основы цифровых машин и программирования / А.А. Папернов. – М. : Наука, 1968. – 591 с.
4. Введение в техническую диагностику / Верзаков Г.Ф., Киншт Н.В., Рабинович В.И., Тимонен Л.С. – М. : Энергия, 1968. – 224 с.

Дополнительная

5. Брюле Д.Д. Отыскание неисправностей в технических устройствах / Д.Д. Брюле, Р.А. Джонсон, Е.Д. Клетский // Зарубежная радиоэлектроника. – 1961. – № 7. – С. 123–137.
6. Согомоян Е.С. Контроль работоспособности и поиск неисправностей в функционально связанных системах / Е.С. Согомоян // Автоматика и телемеханика. – 1964. – Т. 25. – № 6. – С. 980–990.
7. Карибский В.В. Анализ систем для контроля работоспособности и диагностики неисправностей / В.В. Карибский // Автоматика и телемеханика. – 1965. – № 2. – С. 308–314.
8. Сапожников В.В. Основы технической диагностики : учеб. пособ. для студ. вузов ж.д. транспорта / В.В. Сапожников, Вл.В. Сапожников. – М. : Маршрут, 2004. – 318 с.

РАЗДЕЛ 2 МЕТОДЫ РАСПОЗНАВАНИЯ, ПОСТРОЕННЫЕ НА ТЕОРИИ ГРАФОВ

Для описания многих видов абстрактных данных в теории искусственного интеллекта и технической диагностике очень широко используется терминология, заимствованная из теории графов [1]. Следующие определения приведены здесь для того, чтобы показать, как эти заимствованные термины трактуются при описании структурированных объектов, что несколько отличается от их трактовки в математической сфере. Все определения сформулированы в предположении, что существуют два вида примитивов – узлы и связи. Узлы представляют собой исходящие и целевые пункты для связей и обычно каким-либо образом промаркированы. Связи также могут быть промаркированы, но это не обязательно. Все зависит от того, имеем ли мы дело со связями одного или разных видов. В общепринятой терминологии теории графов узлы называются «вершинами», а связи – «ребрами графа», или «дугами». На рис. 2.1 показаны разные типы графов.

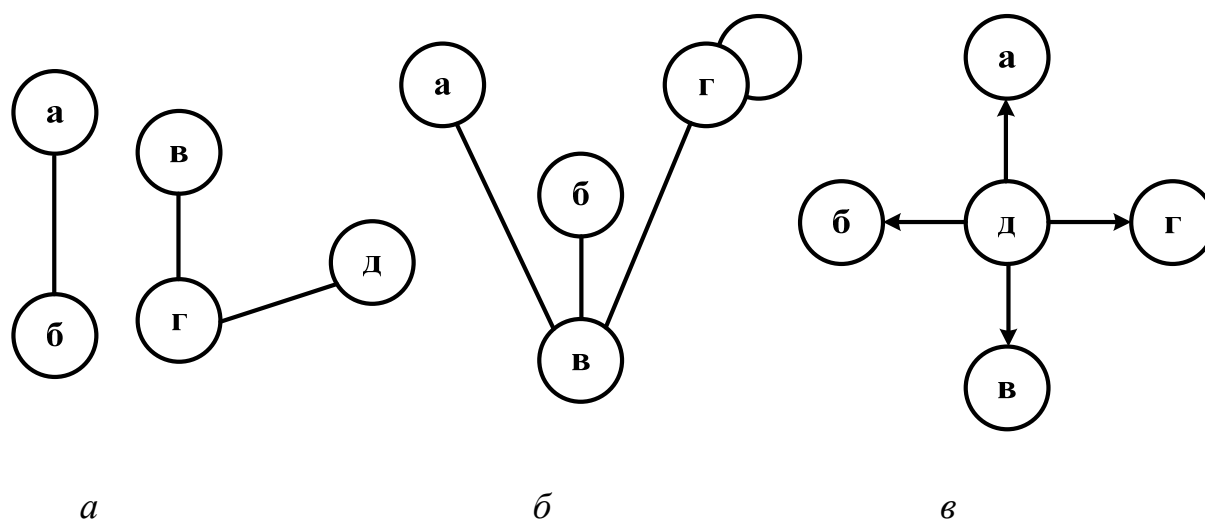


Рисунок 2.1 – Некоторые виды графов:
а – обыкновенный граф; б – связный граф с петлей;
в – обыкновенный ориентированный граф – дерево

Следует отметить, что граф не обязательно должен быть связным. Если задаться условием, что петли не допускаются, то есть в каждой паре должны присутствовать разные узлы, то такой граф называется *обыкновенным*. Если на графе не допускаются не только петли, но и циклы (то есть последовательность связей, в которой начальный и конечный узлы совпадают), то такой граф называется *лесом*.

Иными словами, *дерево* – это связный лес. Обычно один из узлов дерева является его *корнем*, например, узел *б* на графе, представленном на рис. 2.1, *в*. Остальные узлы образуют ветвящуюся структуру «наследников» корневого узла, в которой отсутствуют циклы. Узлы, не имеющие наследников, являются терминальными, или «листьями» дерева, а остальные узлы называются промежуточными (нетерминальными).

В теории графов сетью называется взвешенный ориентированный граф, то есть граф, в котором каждой связи сопоставлено определенное число. Обычно этими числами оценивается «стоимость» пути вдоль этой связи, или длина связи, как на карте дорог. В каждом конкретном случае применения графа как формального средства описания проблемы эти числа могут трактоваться по-своему.

Обыкновенные графы используются для представления взаимоотношений между объектами в пространстве или во времени. Можно использовать их и для представления более абстрактных причинно-следственных связей как, например, связей между различными видами патологий в медицине. Доступ к такой информации связан в той или иной мере с использованием специальных средств прослеживания путей на графе, для которых разработаны самые различные алгоритмы. Для представления иерархических классификаций и сетей применяются *деревья*.

2.1. Алгоритм определения состояния объекта, представленного графом

Результаты любой элементарной проверки могут быть использованы как признаки разбиения множества E технических состояний объекта или подмножеств этого множества на классы. Любой алгоритм диагноза можно представить некоторым ориентированным графом [2]. Ограничимся случаем, когда граф, представляющий алгоритм диагноза, является *деревом*.

Дерево имеет вершины двух типов: вершины, из которых исходит хотя бы одна дуга, и вершины, из которых не выходит ни одной дуги.

Пример дерева дан на рис. 2.2, где вершины первого типа представлены зачерненными кружками, а вершины второго типа – светлыми. В дереве имеется единственная вершина первого типа, в которую не заходит ни одна дуга. Эта вершина называется начальной, или корнем дерева (t_0, E) .

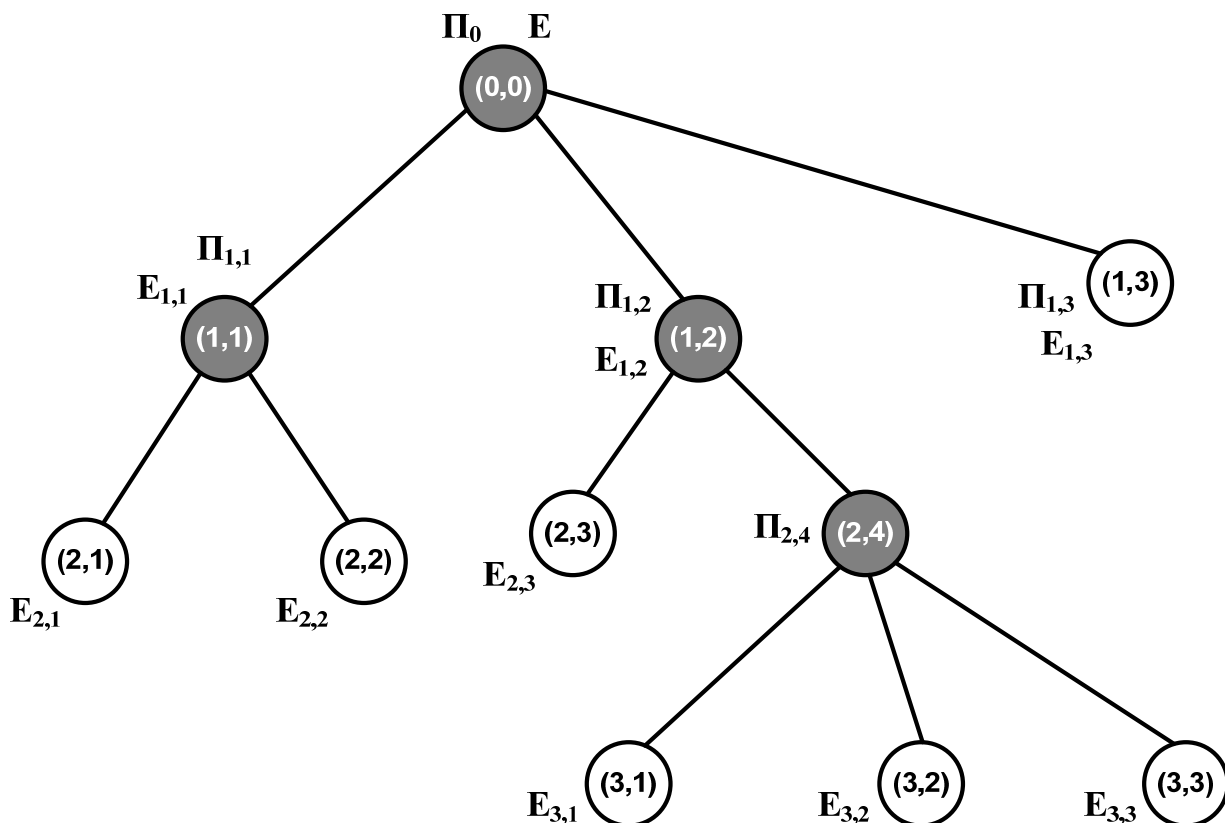


Рисунок 2.2 – Дерево алгоритма диагноза

Вершины, из которых не исходит ни одной дуги, называются конечными, или висячими ($E_{2,1}, E_{2,2}, E_{2,3}, E_{3,1}$). Остальные вершины дерева называются внутренними ($E_{1,1}, E_{1,2}, E_{2,4}$). В каждую вершину дерева, кроме его корня, заходит только одна дуга. В дереве нет контуров.

Рангом вершины дерева называется число дуг пути, начинающегося в начальной вершине и заканчивающегося в рассматриваемой вершине.

Минимальным рангом r_0 дерева является ранг, для которого существует хотя бы одна висящая вершина и не существует ни одной висящей вершины ранга, меньше r_0 .

Максимальным рангом r_m дерева является ранг, для которого существует хотя бы одна вершина ранга r_m и не существует ни одной вершины ранга, больше r_m .

Будем внутренние и висящие вершины дерева нумеровать парой чисел в скобках (r, σ) , где r – есть ранг вершины, а σ – ее порядковый номер среди всех вершин одного и того же ранга (например, слева направо). Начальную вершину будем обозначать парой $(0,0)$. Начальной и внутренней вершине сопоставим элементарные проверки множества Π . Дугам дерева, исходящим из некоторой его вершины, сопоставим возможные результаты проверки, представляемой этой вершиной. Начальной вершине поставим в соответствие множество E возможных технических состояний объекта, а внутренним и висящим вершинам – подмножества технических состояний, получаемые как классы разбиений по результатам соответствующих элементарных проверок. Элементарную проверку и подмножество технических состояний, сопоставляемые вершине (r, σ) дерева, будем обозначать символами $t_{r\sigma}$ и $E_{r\sigma}$ соответственно. Множество элементарных проверок обозначим символом Π . Из правил построения дерева следует, что объединение множества технических состояний есть множества E возможных технических состояний. Каждому пути соответствует последовательность элементарных проверок.

Рассмотрим, как дерево представляет алгоритм диагноза. Начальной вершине соответствует элементарная проверка π_0 множество E всех возможных технических состояний. Элементарная проверка имеет три возможные результата и тем самым разбивает множества E на три подмножества $E_{1,1}$, $E_{1,2}$, $E_{1,3}$ не различаемых этой проверкой технических состояний. Первые два из этих подмножеств соответствуют внутренним вершинам $(1,1)$ и $(1,2)$ и подлежат дальнейшим разбиениям элементарными проверками $\pi_{1,1}$ и $\pi_{1,2}$. Третье подмножество соответствует висящей вершине, и поэтому разбиение его на подмножества алгоритмом не предусмотрено. Аналогично можно рассмотреть любую другую вершину дерева. Как только в процессе элементарных проверок будет достигнута висящая вершина, алгоритм диагноза прекращается. Фактическое техническое состояние объекта принадлежит подмножеству, соответствующему достигнутой висящей вершине. Каждой конкретной реализации алгоритма диагноза соответствует единственный путь. Например, если фактическое техническое

состояние принадлежит подмножеству $E_{3,2}$, то последовательность реализации элементарных проверок будет $\pi_{1,2}, \pi_{2,4}$.

Рассмотрим некоторый ненулевой ранг дерева. В общем случае дерево может иметь несколько внутренних вершин. Это значит, что возможны две разные последовательности реализации элементарных проверок. Обе эти проверки могут являться как одной и той же элементарной проверкой π_j множества Π , так и разными элементарными проверками последнего. Если для каждого ранга дерева выполняется условие, состоящее в том, что всем внутренним вершинам этого ранга сопоставлена одна и та же элементарная проверка из множества Π , то алгоритм диагноза называется *безусловным*. Это соответствует заданию одной фиксированной последовательности реализации элементарных проверок из множества Π , не зависящей от фактического технического состояния объекта. Таким образом, выбор или назначение очередной элементарной проверки в последовательности их реализации не зависит от результатов предыдущих уже реализованных элементарных проверок. Если же в дереве найдется хотя бы один ранг с несколькими внутренними вершинами, которым сопоставимы разные элементарные проверки из множества Π , то алгоритм диагноза называется *условным*. В условных алгоритмах выбор или назначение некоторых или всех (кроме π_0) элементарных проверок производится с учетом результатов предыдущей, уже реализованной элементарной проверки.

Достоинства безусловных алгоритмов – простота проверок: требуется хранить лишь состав элементарных проверок и единственную последовательность их реализации. Для условных требуется хранить, кроме состава элементарных проверок, все признаки безусловных и условных переходов от данной элементарной проверки к следующей, то есть хранить не одну, а несколько последовательностей реализации элементарных проверок.

Безусловные алгоритмы диагноза можно классифицировать по характеру их остановки. Если выдача результатов диагноза предусмотрена только после реализации всех элементарных проверок, то алгоритм является алгоритмом с *безусловной остановкой*. Таким алгоритмам соответствуют деревья, у которых все висящие вершины имеют один и тот же ранг.

Во многих случаях фактическое техническое состояние объекта может быть определено с требуемой глубиной диагноза прежде, чем будут реализованы все элементарные проверки алгоритма диагноза. Безусловные алгоритмы, у которых предусмотрена возможность выдачи результатов диагноза после реализации каждой элементарной проверки, называются алгоритмами с *условной остановкой*. Характерная особенность – не менее двух висящих вершин, ранги которых различны. Все условные алгоритмы являются алгоритмами с условной остановкой. Все указанные типы алгоритмов применяются в системах тестового диагноза. В системах функционального диагноза применяются алгоритмы с условной остановкой в связи с необходимостью немедленного определения неисправного состояния объекта.

2.2. Построение тестов для систем, функциональная модель которых может быть представлена ориентированным графом

В данном разделе, наряду с задачей построения тестов, рассмотрим следующие задачи анализа технических систем:

– определение перечня неразличимых отказов, местонахождение которых нельзя установить выполнением всех возможных для данной системы проверок;

– определение числа и вида изменений структуры системы, введение которых необходимо для локализации всех возможных отказов.

Рассмотрим решение этих задач для систем, функциональная модель которых построена при следующих предположениях [3]:

1) в системе, состоящей из N элементов, возможен отказ любого только одного элемента;

2) каждая возможная для данной системы проверка предполагает контроль реакции одного из элементов на приложенные к системе допустимые внешние воздействия;

3) j -му ($1 \leq j \leq N$) элементу системы поставлена в соответствие проверка π_j , предусматривающая контроль реакции этого элемента и имеющая два исхода: *положительный*, если реакция элемента является допустимой, или *отрицательный*, если реакция элемента является недопустимой; отказ j -го элемента всегда вызывает отрицательный исход проверки π_j ;

4) заданы направленные связи между элементами, и для любой пары элементов выполняется следующее условие: если i -й ($1 \leq i \leq N$) элемент системы связан с j -м, то отрицательный исход проверки π_j влечет отрицательный исход проверки π_i .

Функциональную модель любой системы, удовлетворяющей данным предположениям, можно представить в виде ориентированного графа с петлями у каждой вершины – граф [3]. Вершины графа соответствуют элементам системы, а дуги – связям между элементами. Нумерация вершин графа должна совпадать с нумерацией элементов системы. Поскольку каждому элементу поставлена в соответствие только одна проверка, то, очевидно, номер вершины графа однозначно определяет номер проверки. Предполагается также, что каждая вершина графа, как и каждый элемент системы, может находиться в одном из двух несовместимых состояний (работоспособна, отказала). Таким образом, для задания функциональной модели системы, для которой выполняются предположения 1–4, достаточно задать граф этой системы.

На рис. 2.3 изображен граф системы, функциональная модель которой рассматривалась как пример на рисунке 1.1.

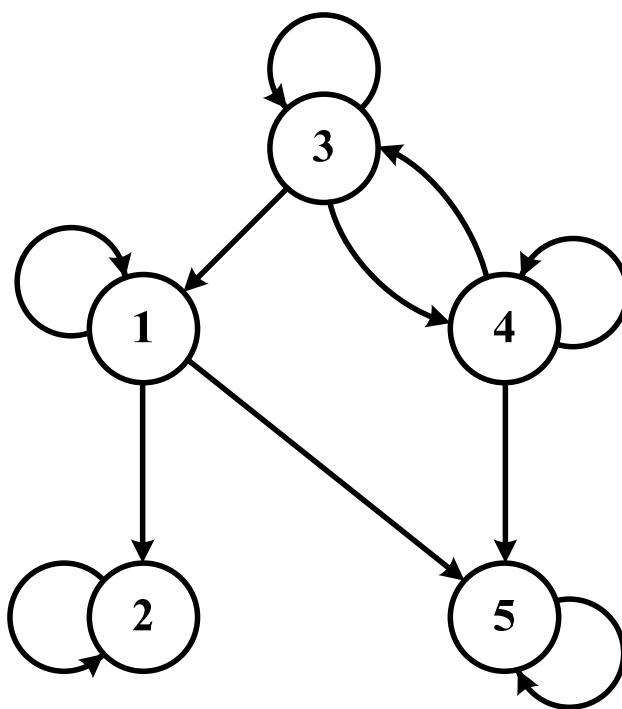


Рисунок 2.3 – Граф системы

В отличие от схемы объекта, представленной на рис. 2.1, на графе системы не указаны внешние воздействия и реакции элементов, хотя связи

между элементами полностью совпадают. Учитывая предположения 3 и 4, можно по графу системы определить исходы всех проверок при заданном состоянии вершин. Так, например, если вершина 1 отказала, то по предположению 3 исход проверки π_1 является отрицательным, и по предположению 4 исходы проверок π_2, π_5 также являются отрицательными. Поскольку от вершины 1 к вершинам 3 и 5 не существует путей, совпадающих с ориентацией дуг, то проверки π_3, π_4 в данном случае будут иметь положительный исход.

Допустим теперь, что ориентированный граф с петлями у каждой вершины представляет собой функциональную модель некоторой реальной системы. Тогда можно говорить о задачах анализа графов и о задачах построения тестов для графов, подразумевая под этим задачи анализа и задачи построения тестов для систем, которые удовлетворяют предположениям 1–4.

Рассмотрим методы решения этих задач для графов, заданных матрицей смежности.

Матрица смежности [3] $B = \|b_{ij}\|$ графа G , содержащего N вершин, – это квадратная матрица с N строками и N столбцами, в которой число b_{ij} , стоящее на пересечении i -й строки и j -го столбца, равно 1, если в графе имеется дуга, идущая из i -й вершины в j -ю, и равно 0, если такой дуги нет. Поскольку каждая вершина графа G имеет петлю, то все числа, стоящие на главной диагонали матрицы B , равны 1, то есть b_{ii} ($1 \leq i \leq N$). Например, матрица смежности графа, изображенного на рис. 2.3, имеет вид:

$$B = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left\| \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} \right\| \end{matrix} \quad (2.1)$$

Для анализа графа G , заданного матрицей смежности B , построим матрицу путей этого графа, то есть квадратную матрицу $D = \|d_{ij}\|$ с N строками и N столбцами, в которой число d_{ij} , стоящее на пересечении i -й строки и j -го столбца, равно 1, если в графе существует путь от i -й верши-

ны к j -й, и равно 0, если такого пути нет. Легко проверить, что для произвольной матрицы путей выполняется равенство

$$(D \dot{\times} D) \dot{+} D = D, \quad (2.2)$$

где операции $\dot{\times}$ и $\dot{+}$ являются операциями булевого умножения и булевого сложения матриц. Эти операции определяются следующим образом.

Пусть заданы квадратные матрицы $B = \|b_{ij}\|$ и $D = \|d_{ij}\|$ размером $N \times N$, в которых числа b_{ij} и d_{ij} принимают значения 0 или 1. Рассмотрим числа f_{ij} и c_{ij} ($1 \leq i, j \leq N$), которые равны:

$$f_{ij} = b_{ij} \dot{+} d_{ij}; \quad (2.3)$$

$$c_{ij} = (b_{i1} \dot{\times} d_{1j}) \dot{+} (b_{i2} \dot{\times} d_{2j}) \dot{+} \dots \dot{+} (b_{iN} \dot{\times} d_{Nj}), \quad (2.4)$$

где операции булевого сложения $\dot{+}$ и булевого умножения $\dot{\times}$ определяются так:

$$\begin{array}{ll} 1 \dot{+} 1 = 1 & 1 \dot{\times} 1 = 1 \\ 1 \dot{+} 0 = 1 & 1 \dot{\times} 0 = 0 \\ 0 \dot{+} 0 = 0 & 0 \dot{\times} 0 = 0 \end{array}$$

Если теперь из чисел f_{ij} и c_{ij} построить квадратные матрицы $F = \|f_{ij}\|$ и $C = \|c_{ij}\|$, то $B \dot{+} D = F$, $B \dot{\times} D = C$.

Для построения матрицы путей можно воспользоваться следующими рекуррентными соотношениями (последовательность $a_1, a_2, \dots, a_n, \dots$ задана рекуррентно, если существует формула, выражающая n -й член этой последовательности через один или несколько предыдущих членов; такая формула называется рекуррентным соотношением), которые следуют из равенства (2.2):

$$B_0 = B; \quad B_k = (B_{k-1} \dot{\times} B_{k-1}) \dot{+} B_{k-1}, \quad k = 1, 2, \dots$$

Определяя последовательно матрицы B_1, B_2, \dots, B_k , заметим, что для некоторого $k + 1$ будет выполняться равенство $B_{k+1} = B_k$. Тогда B_k есть матрица путей исходного графа. Например, для графа, заданного матрицей смежности (2.1), имеем:

$$D = B_2 = \begin{matrix} & 1 & 2 & 3 & 4 & 5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left\| \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} \right\| \end{matrix}.$$

Допустим, что в некотором графе G существует путь от i -й вершины к j -й, и, следовательно, число, стоящее на пересечении i -й строки и j -го столбца матрицы путей для этого графа равно 1. Тогда, согласно предположениям 3 и 4, отказ i -й вершины графа влечет отрицательный исход проверки π_j . Отсюда следует, что номера столбцов матрицы путей можно отождествить с номерами проверок, а числа d_{ij} рассматривать как обозначения исходов этих проверок (при отрицательном исходе $d_{ij} = 1$, при положительном $d_{ij} = 0$). Таким образом, рассматривая i -ю строку матрицы путей, можно определить те проверки, которые будут иметь отрицательный исход при отказе i -й вершины графа, или наоборот, зная исходы всех проверок, можно определить те вершины, отказ которых влечет эти исходы. Например, если для графа с матрицей смежности (2.1) все проверки имеют отрицательный исход, то отказала либо 3-я, либо 4-я вершины графа.

Отметим, что заменив в матрице путей все $d_{ij} = 1$ на $d_{ij} = 0$, а все $d_{ij} = 0$ на $d_{ij} = 1$, получим матрицу $\bar{D} = \|\bar{d}_{ij}\|$, совпадающую с таблицей неисправностей системы, функциональная модель которой представлена данным графом.

2.2.1. Определение перечня неразличимых отказов

Пусть задана матрица смежности некоторого графа G . Для определения перечня неразличимых отказов достаточно построить матрицу путей исходного графа и выписать все подмножества вершин, которым соответствуют попарно тождественные строки (или столбцы) этой матрицы. Неразличимыми являются отказы тех вершин, которые входят в одно подмножество. Например, для графа, изображенного на рис. 2.3, отказы третьей и четвертой вершин являются неразличимыми. Очевидно, что каждому

выписанному подмножеству вершин X соответствует сильно связанный подграф графа G , то есть для любых двух вершин i и j из X в графе G существуют пути, идущие от i к j и от j к i .

Наличие неразличимых отказов можно установить непосредственно по матрице смежности исходного графа, не прибегая к построению матрицы путей. Для этого достаточно воспользоваться следующим приемом.

В матрице смежности зачеркиваются строка, содержащая только одну единицу, и столбец, номер которого совпадает с номером зачеркнутой строки (можно также зачеркивать столбец, содержащий только одну единицу, и строку, номер которой совпадает с номером зачеркнутого столбца). Повторяя эту операцию зачеркивания, можно прийти к двум результатам:

а) все строки и столбцы матрицы смежности зачеркнуты; в этом случае неразличимых отказов нет;

б) все оставшиеся строки и столбцы матрицы смежности содержат не менее двух единиц; в этом случае неразличимые отказы имеются.

2.2.2. Построение тестов

Проверяющим тестом T_{π} , или тестом для проверки работоспособности, назовем такой набор проверок, выполнение которых необходимо для обнаружения факта отказа любой вершины графа (то есть отказа любого элемента системы).

Локализирующим тестом T_{λ} назовем такой набор проверок, выполнение которых необходимо для определения местонахождения (номера) любой отказавшей вершины.

Тест T , полученный объединением тестов T_{π} и T_{λ} , назовем *диагностическим тестом для данного графа*.

Для построения тестов выполним следующие преобразования исходного графа G .

1. Пусть G содержит одно или несколько подмножеств вершин, отказы которых неразличимы. К каждому такому подмножеству X применим операцию стягивания, которая состоит в том, что все дуги, соединяющие вершины из X , опускаются, а сами вершины объединяются в одну новую вершину с петлей. Полученный граф обозначим через G' . Если в матрице путей исходного графа G оставить вершины, отказы которых различимы, а также по одной вершине из каждого подмножества (зачеркнув строки и

столбцы, соответствующие всем остальным вершинам из этих подмножеств), то полученная матрица D_0 будет матрицей путей графа G' . Новым вершинам графа G' соответствуют строки и столбцы тех вершин из подмножеств X , которые остались в D_0 . Например, граф G' , полученный стягиванием вершин 3 и 4 графа с матрицей (2.1), имеет матрицу путей

$$D_0 = \begin{matrix} & 1 & 2 & 3' & 5 \\ \begin{matrix} 1 \\ 2 \\ 3' \\ 5 \end{matrix} & \left\| \begin{matrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix} \right\| \end{matrix},$$

в которой 3-я строка и 3-й столбец соответствуют новой вершине 3'. Если в исходном графе G отсутствуют контуры, проходящие не менее чем через две вершины, то матрица D_0 совпадает с матрицей путей исходного графа.

2. Пусть G' имеет не менее двух вершин. Найдем те дуги графа G' , удаление которых не изменяет его матрицу путей (удаление дуги, соединяющей вершину i с j ($i \neq j$), не изменяет матрицу D_0 , если в G' существует путь от i к j , не проходящий через эту дугу). Удалив эти дуги из G' , получим приведенный граф G_0 , матрица путей которого равна D_0 .

На рис. 2.4 изображен приведенный граф, полученный преобразованием исходного графа с матрицей смежности (2.1).

Матрица смежности приведенного графа имеет вид:

$$B_0 = \begin{matrix} & 1 & 2 & 3' & 5 \\ \begin{matrix} 1 \\ 2 \\ 3' \\ 5 \end{matrix} & \left\| \begin{matrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \right\| \end{matrix}.$$

Матрица путей D_0 произвольного приведенного графа, содержащего не менее двух вершин, обладает следующими свойствами:

– все строки (столбцы) матрицы D_0 попарно различны (в приведенном графе отсутствуют контуры, проходящие не менее чем через две вершины);

– в D_0 имеется, по крайней мере, одна вершина, называемая входом, которой соответствует столбец, содержащий только одну единицу, и, по крайней мере, одна вершина, называемая выходом, которой соответствует строка, содержащая только одну единицу (если приведенный граф содержит изолированную вершину, то эта вершина является одновременно как входом, так и выходом; в дальнейшем изолированные вершины будем относить к выходам);

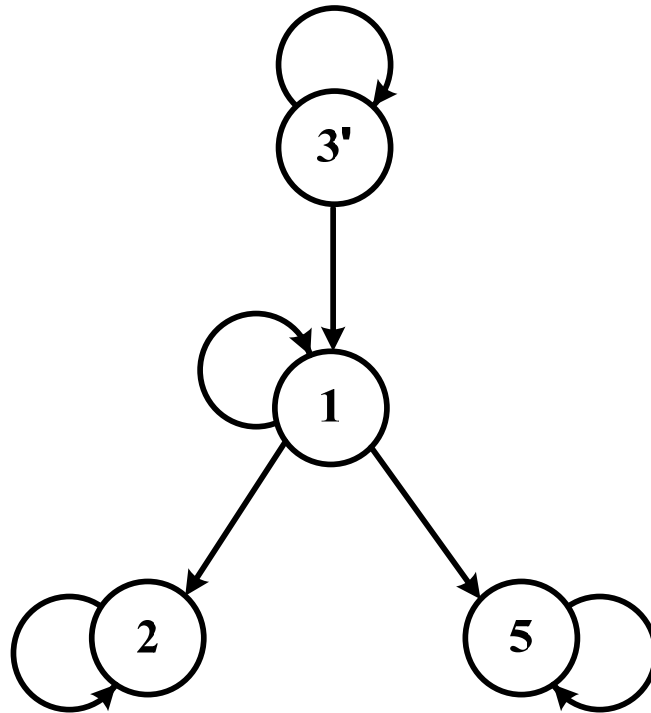


Рисунок 2.4 – Приведенный граф системы

– для D_0 выполняется соотношение (2.2);

– строки и столбцы матрицы D_0 всегда можно переставить так, чтобы все числа d_{ij} ($1 \leq j \leq i$), стоящие ниже главной диагонали, были равны нулю. Эта перестановка производится следующим образом. Из матрицы D_0 вычеркиваем все строки и столбцы, соответствующие вершинам типа «выход». Находим все вершины типа «вход», которые назовем вершинами 1-го ранга, и зачеркнем строки и столбцы, соответствующие этим вершинам. В полученной матрице найдем все вершины, которым соответствуют столбцы, содержащие только одну единицу. Эти вершины назовем вершинами второго ранга. Зачеркнув строки и столбцы, соответствующие вер-

шинам второго ранга, аналогично находим все вершины 3-го ранга и так далее, пока все строки и столбцы матрицы D_0 не будут зачеркнуты. Вершинам типа «выход» присваивается наивысший ранг. Упорядочив теперь столбцы и строки матрицы D_0 в соответствии с рангом вершин, получим требуемую перестановку. Так, например, матрица (2.5) содержит одну вершину 1-го ранга, одну вершину 2-го ранга и две вершины 3-го ранга. Выполнив перестановки строк и столбцов этой матрицы, получим:

$$D_0 = \begin{matrix} & 3' & 1 & 2 & 5 \\ \begin{matrix} 3' \\ 1 \\ 2 \\ 5 \end{matrix} & \left\| \begin{matrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \right\| & \end{matrix}. \quad (2.5)$$

Пусть приведенный граф G_0 содержит новые вершины. Припишем каждой новой вершине состояние (работоспособна, отказала). Будем считать, что новая вершина, полученная стягиванием подмножества X , работоспособна, если работоспособны все вершины из X . Отказ любой вершины из X эквивалентен отказу новой вершины. Поскольку вершинам из X соответствуют тождественные столбцы матрицы путей исходного графа G , то новой вершине, полученной стягиванием X , следует приписать номер той проверки, которая ранее была приписана одной из вершин этого подмножества. Например, новой вершине $3'$ приведенного графа с матрицей смежности (2.1) можно приписать проверку с номером 3 (π_3) или проверку с номером 4 (π_4).

Рассмотрим приведенный граф G_0 , который получен из исходного графа G . Будем считать, что строки и столбцы матрицы путей графа G_0 упорядочены в соответствии с рангом вершины. Так как номер вершины графа G_0 определяет номер проверки, то для построения проверяющего, локализирующего или диагностического тестов для графа G достаточно перечислить номера вершин графа G_0 , входящие в соответствующий тест.

Число проверок, входящих в тест, назовем *длиной теста*. Допустим, что для графа G построена совокупность однородных тестов (либо прове-

ряющих, либо локализирующих, либо диагностических). Любой тест, принадлежащий этой совокупности и имеющий минимальную длину по сравнению со всеми остальными тестами из этой же совокупности, назовем минимальным.

Перейдем к построению минимальных тестов. Для того чтобы проверяющий тест T_π для исходного графа G был минимальным, необходимо и достаточно, чтобы в этот тест вошли все вершины типа «выход» приведенного графа G_0 . Действительно, через каждую вершину графа G_0 проходит путь к одной из вершин типа «выход». Следовательно, выполнив проверки, номера которых совпадают с номерами вершин типа «выход», можно обнаружить отказ любой вершины исходного графа G . С другой стороны, если в проверяющий тест не войдет хотя бы одна вершина типа «выход» приведенного графа, то ее отказ невозможно обнаружить выполнением любых проверок, номера которых совпадают с номерами всех остальных вершин этого графа. Например, тест $T_\pi = \{2, 3', 5\}$ является минимальным проверяющим тестом для графа с матрицей смежности (2.1). Данный тест предусматривает выполнение проверок π_2 и π_5 .

Для построения минимального локализирующего теста для исходного графа G следует в матрице путей D_0 приведенного графа G_0 выбрать такой набор, содержащий минимальное число вершин, что после вычеркивания из D_0 столбцов, соответствующих вершинам, которые не вошли в данный набор, все строки полученной матрицы останутся попарно различными.

Нетрудно показать, что лишь одна из вершин типа «выход» приведенного графа G_0 иногда может не входить в минимальный локализирующий тест. Кроме того, если какая-либо из вершин графа G_0 имеет только одну исходящую из нее дугу, то эта вершина обязательно входит в минимальный локализирующий тест. Например, вершина $3'$ приведенного графа с матрицей смежности (2.1) входит во все минимальные локализирующие тесты $T_{\pi 1} = \{1, 2, 3'\}$, $T_{\pi 2} = \{1, 3', 5\}$, $T_{\pi 3} = \{2, 3', 5\}$ для исходного графа с матрицей смежности (2.1). При выполнении этих тестов можно производить либо проверку π_3 , либо проверку π_4 , так как новой вершине $3'$ может быть приписана любая из этих проверок.

Для построения минимального диагностического теста для исходного графа G необходимо дополнить приведенный граф G_0 изолированной вершиной без петли (этой вершине будут соответствовать строка и столбец, содержащие только нули). При этом следует учесть, что в минимальный диагностический тест должны входить все вершины типа «выход», а также все вершины приведенного графа G_0 , которые имеют только одну исходящую дугу. Например, тест $T = \{2, 3', 5\}$ является минимальным диагностическим тестом для графа с матрицей смежности (2.1).

Для произвольного исходного графа G длина минимального диагностического теста или равна, или на единицу больше длины минимального локализирующего теста.

2.2.3. Определение оптимального числа и вида изменений структуры системы

Рассмотрим систему, которая представлена графом G . Предположим, что каждой дуге графа G (за исключением петель) соответствует реальная физическая связь между элементами системы. Если G не содержит сильно связных подграфов, то, как было отмечено выше, отказы всех элементов системы различимы. С другой стороны, если G содержит сильно связные подграфы, то разорвав в системе те связи, которым соответствуют дуги, образующие сильно связные подграфы, можно осуществить поиск любого отказавшего элемента. Так, например, если в системе, представленной графом (см. рис. 2.3) с матрицей смежности (2.1), разорвать связь, которой соответствует дуга, идущая от 4-й вершины к 3-й, то отказы элементов, поставленных в соответствие этим вершинам, будут различимы. Такой же результат даст разрыв связи, которой соответствует дуга, идущая от 3-й вершины к 4-й.

Пусть задана матрица смежности B графа G и для каждой дуги, идущей от i -й вершины к j -й, известна стоимость c_{ij} ($c_{ij} > 0$) разрыва этой дуги. Тогда для графа G можно построить матрицу стоимостей $C = \|c_{ij}\|$ ($1 \leq i \leq N$) ($1 \leq j \leq N$), в которой $c_{ij} = 0$, если в графе G нет дуги, идущей от i -й вершины к j -й; и $c_{ii} = 0$, ибо разрыв петель, приписанных вершинам графа, не влияет на различимость отказов вершин.

Рассмотрим теперь следующую задачу. Необходимо найти те дуги графа G , разрыв которых устраняет все сильно связные подграфы данного

графа, причем суммарная стоимость всех разрывов должна быть минимальной. Заметим, что для устранения всех сильно связных подграфов необходимо и достаточно разорвать в графе все ориентированные контуры, проходящие не менее чем через две вершины.

Известно [3], что для устранения всех контуров, проходящих не менее чем через две вершины, достаточно удалить из G все дуги, которым соответствуют числа $b_{ij} = 1$, расположенные ниже главной диагонали матрицы B . Стоимость удаления (разрыва) этих дуг равна:

$$C(\{1, 2, \dots, N\}) = \sum_{i=2}^N \sum_{j=1}^{i-1} c_{ij}, \quad (2.6)$$

где N – число вершин графа G .

Очевидно, что $C(\{1, 2, \dots, N\})$ зависит от порядка расположения строк и столбцов в матрицах B и C , и, следовательно, для решения данной задачи необходимо найти такое упорядочение строк и столбцов матрицы C , чтобы сумма всех чисел c_{ij} , расположенных ниже главной диагонали матрицы C , была минимальной.

Пусть X – подмножество, содержащее k ($2 \leq k \leq N$) вершин графа G . Построим подграф $G(X)$, образованный всеми вершинами из X и дугами, соединяющими эти вершины в исходном графе G . Обозначим через $C(X)$ минимальную стоимость разрыва в подграфе $G(X)$ всех ориентированных контуров, проходящих не менее чем через две вершины. Пусть далее вершина i принадлежит X . Обозначим через X/i подмножество, полученное из X при выбрасывании из него вершины i . Тогда стоимость разрыва всех дуг, идущих от i к вершинам из X/i , равна:

$$c_i(X \setminus i) = \sum_{j \in X} c_{ij}. \quad (2.7)$$

Для вычисления $C(X)$ можно воспользоваться следующими рекуррентными соотношениями:

$$\begin{aligned} k = 1: C(X) &= 0; \\ k \geq 2: C(X) &= \min\{C(X \setminus i) + c_i(X \setminus i)\}. \end{aligned} \quad (2.8)$$

Оптимальное упорядочение строк и столбцов матриц B и C находится в два этапа. На первом этапе, начиная с $k = 2$, рекурсивно (рекурсия – определение, описание, изображение какого-либо объекта или процесса

внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя) по формулам (2.8) вычисляется $C(X)$ для всех возможных подмножеств X , содержащих k ($k \leq N$) вершин. Одновременно с вычислением $C(X)$ фиксируется та вершина из X , для которой достигается минимум в формулах (2.8). На втором этапе определяется само оптимальное упорядочение.

Для этого сначала находится вершина l_n ($1 \leq l_n \leq N$), зафиксированная при вычислении $C(X)$ для исходного графа G , то есть $X = X^* = \{1, 2, \dots, N\}$. Затем находится вершина l_{n-1} ($1 \leq l_{n-1} \leq N$), зафиксированная при вычислении $C[X^* \setminus l_n]$, и т.д. Последней определяется вершина l_1 ($1 \leq l_1 \leq N$), зафиксированная при вычислении $C[(\dots ((X^* \setminus l_n) \setminus l_{n-1}) \dots \setminus l_2)]$. Отметим, что иногда целесообразно фиксировать не вершины из X , а упорядочение вершин в подмножестве X , для которого достигается минимум в формулах (2.8). В этом случае оптимальное упорядочение будет получено после вычисления $C(X)$ для всего множества вершин, то есть $X = X^*$. Если теперь столбцы и строки матриц B и C расположить в порядке, который определяется перестановкой $(l_1, \dots, l_{n-1}, l_n)$, то получим оптимальное упорядочение. Следовательно, для того чтобы суммарная стоимость разрывов была минимальной, необходимо и достаточно удалить те дуги графа G , которым соответствуют числа $b_{ij} = 1$ в матрице B , упорядоченной согласно с перестановкой $(l_1, \dots, l_{n-1}, l_n)$.

Рассмотрим пример, иллюстрирующий описанную методику определения оптимального упорядочения. Пусть для графа G с матрицей смежности

$$B = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left\| \begin{array}{ccccc} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{array} \right\| \end{matrix}$$

задана матрица стоимостей

$$C = \begin{matrix} & 1 & 2 & 3 & 4 & 5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left\| \begin{matrix} 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 3 & 5 & 0 \\ 6 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 4 & 5 & 0 & 0 \end{matrix} \right\| \end{matrix}.$$

Необходимо найти такое упорядочение вершин графа G , чтобы стоимость разрыва всех ориентированных контуров была минимальной.

Результаты вычислений по формулам (2.8) сведены в табл. 2.1.

Таблица 2.1 – Результаты оптимизации

№ п/п	X	$C(X)$	Оптимальное упорядочение	Номер	X	$C(X)$	Оптимальное упорядочение
1	1; 2	0	1; 2	14	1; 3; 4	0	3; 1; 4
2	1; 3	0	3; 1	15	1; 3; 5	2	5; 3; 1
3	1; 4	0	1; 4	16	1; 4; 5	0	1; 4; 5
4	1; 5	0	1; 5	17	2; 3; 4	0	2; 4; 5
5	2; 3	0	2; 3	18	2; 3; 5	2	5; 2; 3
6	2; 4	0	2; 4	19	2; 4; 5	3	5; 2; 4
7	2; 5	0	5; 2	20	3; 4; 5	2	4; 5; 3
8	3; 4	0	3; 4	21	1; 2; 3; 4	2	2; 3; 1; 4
9	3; 5	2	5; 3	22	1; 2; 3; 5	4	5; 2; 3; 1
10	4; 5	0	4; 5	23	1; 2; 4; 5	3	1; 5; 2; 4
11	1; 2; 3	2	2; 3; 1	24	1; 3; 4; 5	5	3; 1; 4; 5
12	1; 2; 4	0	1; 2; 4	25	2; 3; 4; 5	5	5; 2; 3; 4
13	1; 2; 5	0	1; 5; 2	26	1; 2; 3; 4; 5	7	5; 2; 3; 1; 4

В данном примере на каждом шаге вычислений определялось оптимальное упорядочение для всех подмножеств вершин графа G . Как это видно из таблицы, на первых 10 шагах вычислений определялась минимальная стоимость $C(X)$ для всех возможных подмножеств, содержащих 2 вершины. На следующих 10 шагах определялась минимальная стоимость $C(X)$ для всех возможных подмножеств, содержащих 3 вершины, и так далее.

В частности, минимальная стоимость $C(X)$, вычисленная на 22 шаге

для подмножества $\{1, 2, 3, 4, 5\}$, была найдена из следующих соотношений:

$$c_1 = C(2; 3; 5) + c_1(2; 3; 5) = 2 + 2 = 4;$$

$$c_2 = C(1; 3; 5) + c_2(1; 3; 5) = 2 + 3 = 5;$$

$$c_3 = C(1; 2; 5) + c_3(1; 2; 5) = 0 + 8 = 8;$$

$$c_4 = C(1; 2; 3) + c_4(1; 2; 3) = 2 + 9 = 11.$$

Поскольку минимальное значение стоимости, равное 4, достигается для первой вершины и оптимальное упорядочение для подмножества $\{2; 3; 5\}$, найденное на 18 шаге, есть $(5; 2; 3)$, то оптимальное упорядочение для рассматриваемого подмножества есть $(5; 2; 3; 1)$.

Оптимальному упорядочению вершин графа соответствует перестановка $(5; 2; 3; 1; 4)$, найденная на 26 шаге. Для этой перестановки минимальная стоимость разрыва контуров равна 7, тогда как для исходного упорядочения стоимость разрыва контуров равна 15. Матрица смежности, упорядоченная в соответствии с перестановкой $(5; 2; 3; 1; 4)$, имеет вид:

$$B = \begin{matrix} & \begin{matrix} 5 & 2 & 3 & 1 & 4 \end{matrix} \\ \begin{matrix} 5 \\ 2 \\ 3 \\ 1 \\ 4 \end{matrix} & \left\| \begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{array} \right\| \end{matrix}.$$

Числа $b_{ij} = 1$, стоящие ниже главной диагонали в матрице B^0 , определяют те дуги графа G , которые необходимо разорвать для устранения всех ориентированных контуров, проходящих не менее чем через две вершины.

В заключение отметим, что рассмотренные в данном разделе методы преобразования графов и методы построения тестов достаточно хорошо алгоритмируются и могут быть использованы для программирования электронных цифровых вычислительных машин.

2.3. Оценка технического состояния опорно-стержневых изоляторов

В качестве примера использования методов распознавания, построенных на теории графов, рассмотрим метод оценки состояния опорно-

стержневых изоляторов, разработанный в Винницком национальном техническом университете [4]. Большое количество видов дефектов опорно-стержневых изоляторов (ОСИ), разные условия и скорость их развития требуют применения нескольких методов контроля технического состояния. Использование отдельно любого из методов контроля технического состояния ОСИ не обеспечивает нужной производительности контроля с необходимым уровнем условной вероятности выявленной неисправности.

Для создания комплексного метода проведена оценка эффективности методов, которые целесообразно использовать для контроля изоляторов на основе достоверности выявления дефектов и их опасности [5].

К эффективным методам контроля изоляторов относятся:

- визуальный контроль – ВК (Π_1);
- контроль средствами ИКТ – ТВК (Π_2);
- ультразвуковая структурометрия и дефектометрия – УЗК (Π_3);
- испытание силовыми методами, в том числе с регистрацией сигналов акустической эмиссии – АЭ (Π_4).

Возможные дефекты изоляторов и эффективные методы их контроля приведены в табл. 2.2.

Таблица 2.2 – Возможные дефекты изоляторов и эффективные методы их контроля

Состояние	Вид дефекта	Методы контроля
S_2	Видимые трещины фарфора или фланцев	ВК, УЗК, ТВК, АЭ
S_3	Внутренние трещины и зоны растрескивания	ТВК, УЗК, АЭ
S_4	Наличие зоны открытой микроскопической пористости	ТВК, УЗК
S_5	Обгоревшая и засоренная макроскопическая пористость	УЗК, АЭ
S_6	Дефекты поверхности фарфора или цементного шва	ВК
S_7	Дефекты, вызванные механической усталостью и т. п.	АЭ

Авторы выполнили формализацию комплексного метода контроля ОСИ матрицей неисправностей (табл. 2.3), то есть таблицей функций исправного S_1 и дефектного состояний изолятора в зависимости от возможности определения состояния изолятора разными методами контроля.

Авторы ввели ряд предположений:

- 1) состояние изолятора характеризует более критический дефект в приоритетах по состояниям: 2, 3, 4, 5, 6, 7;
- 2) дефектные изоляторы разных видов распределены равномерно в общем количестве, то есть отсутствуют явным образом дефектные партии;
- 3) количество ОСИ, которые эксплуатируются в энергосистеме, постоянное.

Таблица 2.3 – Таблица функций исправного и дефектного состояний изолятора в зависимости от возможности методов контроля

Состояние	Метод			
	П ₁	П ₂	П ₃	П ₄
S_1	0	0	0	0
S_2	1	1	1	1
S_3	0	1	1	1
S_4	0	1	1	0
S_5	0	0	1	1
S_6	1	0	0	0
S_7	0	0	0	1

Примечание. В таблице обозначено: 1 – метод позволяет указать дефект данного вида; 0 – не позволяет.

Для идентификации конкретного состояния ОСИ была построена таблица перекрытия поиска дефектов (табл. 2.4). Таблица 2.4 является формализацией логической функции различения двух состояний разными методами на основе дизъюнкции данных таблицы состояний по формуле:

$$S_{mj}S_{nj} = S_{mj} \vee S_{nj}, \quad (2.9)$$

где S_{mj} , S_{nj} – значения в таблице состояний, которые отвечают j методу контроля для состояний m и n соответственно.

Согласно данным табл. 2.3 и 2.4, происходят перекрытия всех строк дефектных состояний, то есть таблицы есть логически полными для дефектных состояний.

Таблица 2.4 – Таблица перекрытия поиска дефектов

Состояние	Метод			
	П ₁	П ₂	П ₃	П ₄
S_1S_2	1	1	1	1
S_1S_3	0	1	1	1
S_1S_4	0	1	1	0
S_1S_5	0	0	1	1
S_1S_6	1	0	0	0
S_1S_7	0	0	0	1
S_2S_3	1	1	1	1
S_2S_4	1	1	1	1
S_2S_5	1	1	1	1
S_2S_6	1	1	1	1
S_2S_7	1	1	1	1
S_3S_4	0	1	1	1
S_3S_5	0	1	1	1
S_3S_6	1	1	1	1
S_3S_7	0	1	1	1
S_4S_5	0	1	1	1
S_4S_6	1	1	1	0
S_4S_7	0	1	1	1
S_5S_6	1	0	1	1
S_5S_7	0	0	1	1
S_6S_7	1	0	0	1

Кроме того, существует ряд дефектных состояний – «исключений», которые могут различаться между собой одним методом контроля:

– визуальный контроль (П₁) позволяет различать между собой визуальные трещины (S_2) от дефектов поверхности и выкрошивание цементного шва (S_6);

– ультразвуковой контроль (П₂) позволяет различать между собой состояния S_3 и S_4 ; S_2 и S_3 ; S_2 и S_4 ; S_3 и S_5 ; S_4 и S_5 , и S_3 , S_4 , S_5 между собой.

Авторами разработан универсальный для данного класса задач алгоритм построения деревьев решений. При разработке алгоритма построения дерева решений использовалось обозначение «полного» дерева, которое показано на рис. 2.5. Уровень вершины k обозначается сверху вниз, номер вершины g – слева направо для вершин данного уровня, номер метода j считался по анализируемым методам в следующем порядке: ВК, ТВК, УЗК и АЭ. Входной ветвью вершины считали входящие состояния в вершину,

которые представляются в двоичном виде: 1; 0. Входными ветвями верхней вершины являются 1. Текущим ответвлением считалась совокупность всех вершин и ветвей от текущей вершины или ветви к первой (верхней) вершине. Строка состояний в левой L_i и в правой R_i исходных ветвях состояний рассчитывалась как произведение соответствующих элементов строке состояний входных ветвей текущей вершины дерева V_i на строку матрицы неисправностей S_{ij} и на инверсную строку матрицы неисправностей соответственно для метода j в текущей вершине по формулам:

$$L_i = S_{ij}V_i; \quad (2.10)$$

$$R_i = \bar{S}_{ij}V_i. \quad (2.11)$$

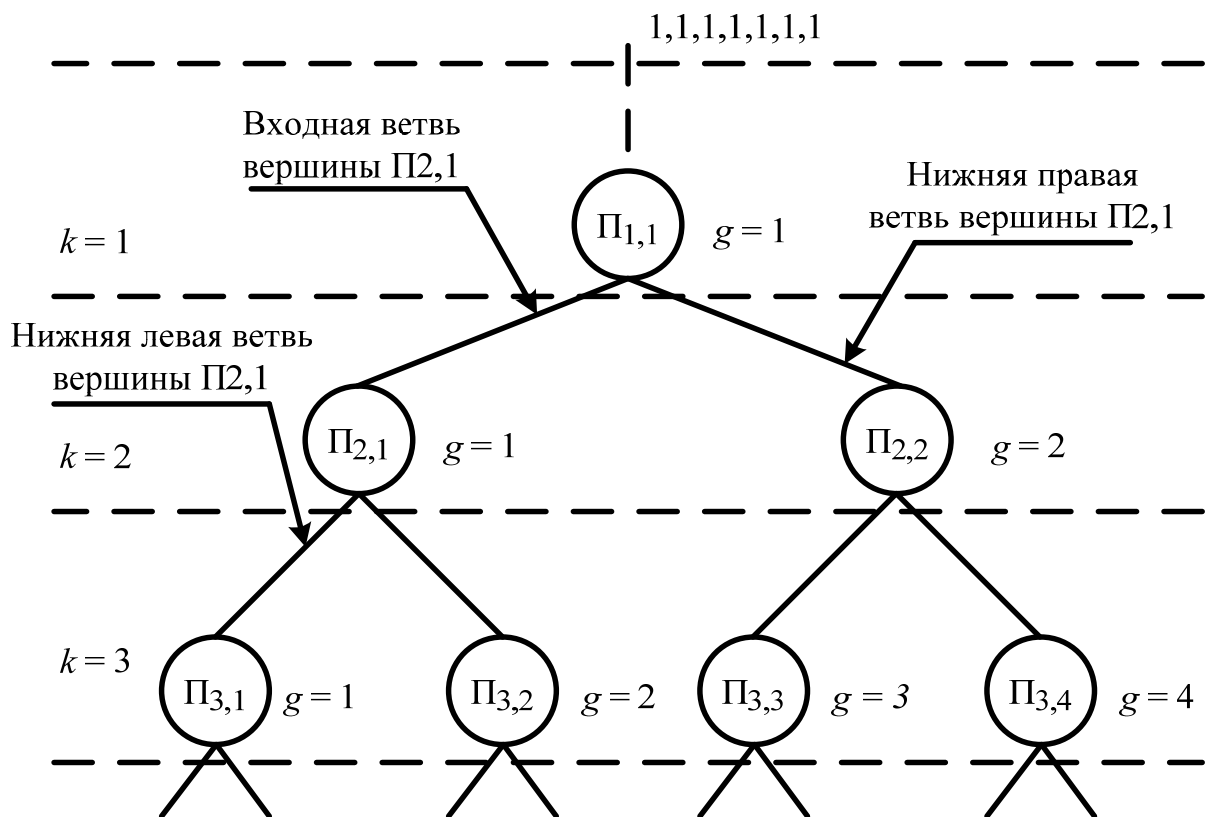


Рисунок 2.5 – Полное дерево решений

Построение одного логического дерева выполнялось с помощью рекурсивной процедуры (рекурсивная процедура – это процедура, вызывающая сама себя) формирования любой вершины дерева и перехода по ветвям дерева.

Формирование вершины дерева выполняется в следующей последовательности.

1. Проверка, не является ли входная ветвь конечной, то есть такой, в которой осталось или одно состояние, или «исключение» при условии выполнения метода, который их идентифицирует в данном ответвлении – пути от данной ветви к первой (верхней) вершине дерева.

2. Выбор метода контроля ОСИ (значение вершины) и формирование станом в нижних (исходных) ветвях осуществляется по следующим принципам:

– метод не использовался в текущем ответвлении – выше по дереву, по ветвям от вершины дерева, которое рассматривается, к первой вершине;

– метод не дает «нулевых» ветвей, то есть и в правую, и в левую ветви переходит хотя бы одно состояние.

3. Если ветви не конечные, то происходит формирование состояний в нижних ветвях вершины как произведение строк матрицы неисправностей и состояний во входных ветвях.

Идея построения всего дерева следующая:

1. Выполняется построение вершин по левым ветвям от верхней вершины к крайней нижней левой вершине при условии, что ветвь не конечная.

2. В случае, если левая ветвь есть конечной по условию одного состояния в ветвях или «исключений», выполняется переход на правую ветвь текущей вершины.

3. При построении вершины, в которой левая и правая нижние ветви являются конечными, входная ветвь данной вершины обозначается как конечная, и выполняется переход на верхний уровень к вершине в текущем ответвлении.

4. Процедура выполняется к моменту окончания левой и правой ветвей первой вершины.

На основе первого построенного дерева формируется таблица вершин (табл. 2.5), строки которой отвечают уровню от верхней вершины дерева вниз, столбцы – номеру вершины слева направо на данном уровне, а значение элементов таблицы – номеру метода контроля ОСИ.

Также формируется таблица прохождения с аналогичной структурой, но со значениями элементов, которые отвечают количеству изменений

значений в вершине дерева при неизменной конфигурации других ответвлений и значений вершин верхних уровней дерева. На начальном этапе элементы, которые отвечают «ненулевым» вершинам, в дереве обозначаются 1.

Таблица 2.5 – Пример таблицы вершин дерева

1							
0	2						
0	0	3	3				
0	0	0	0	0	0	0	4

Построение всех возможных деревьев решений, основанное на минимизации количества выполнения процедур для построения следующего дерева заключается в том, что следующее дерево формируется не путем построения дерева сначала, а изменением значений лишь одной вершины, если ее ветви конечны, или одной вершины в верхних уровнях дерева и вершин, которые построены вниз от текущей вершины, в зависимости от уровня вниз по дереву, на котором происходит изменение значения вершины.

Принцип построения всех деревьев следующий:

1) после построения первого дерева преобразование в таблицу результатов и формирование таблицы вершин дерева решений и таблицы прохождения;

2) в таблице прохождения выбирается крайний правый нижний «ненулевой» и «не выполненный» (для которого не выполнены все прохождения) элемент, который обозначается k_{\max} , g_{\max} ;

3) для вершины выполняется поиск следующего за установленным в таблице вершин «успешного» метода контроля, который удовлетворяет условиям построения (выбор метода контроля ОСИ), при этом при анализе каждого следующего метода значение текущего элемента в таблице прохождения увеличивается на 1;

4) если «успешный» метод найден, то новое дерево заносится в массив построенных деревьев;

5) иначе, если «успешный» метод не найден, текущий элемент в таблице прохождения обозначается как «выполненный», происходит переход

к крайней правой нижней «ненулевой» и «не выполненной» вершине, которая находится левее текущей по дереву, и выполняется шаг 3;

б) если «успешный» метод найден, то это значение заносится в таблицу вершин, а значения элементов, которые находятся правее и ниже текущей вершины, значения элементов в таблице прохождения, в том числе обозначенных «выполненными», устанавливаются 1; значения соответствующих элементов в таблице вершин устанавливаются равными значениям первого дерева. Новое дерево преобразуется в массив построенных деревьев. На следующем этапе процедура 2–6 повторяется;

7) при условии достижения крайней левой «ненулевой» и «не выполненной» вершины на текущем уровне, когда все элементы уровня есть «выполненными», осуществляется переход на крайний правый элемент верхнего уровня и повторяются шаги 3, 6;

8) в случае нахождения следующего «успешного» значения вершины на верхних уровнях вершины нижних уровней в ответвлениях данной вершины строятся заново, значение «ненулевых» вершин в таблице прохождения меняется на 1, а таблицы вершин – на параметры первого дерева;

9) выполнение процедуры заканчивается, если значение верхней вершины дерева в таблице прохождения приобретает значение «выполнено».

В результате расчетов авторами получен массив деревьев решений с 63 деревьями для разъединителей 35–110 кВ и с 6 деревьями для другого оборудования. При этом рассматриваются два варианта массива: для ОСИ в составе разъединителей 35–110 кВ, где возможно использование силовых испытаний (АЭ), и для ОСИ в составе другого оборудования.

В многоэтажных конструкциях (оборудование 330 кВ и выше) и в шинных сопротивлениях любого класса напряжения дефект S_7 свидетельствует о том, что наличие в ОСИ дефектов, вызванных механической усталостью фарфора, или наличие сосредоточенных дефектов в ствольной части ОСИ существующими методами выявить невозможно. Матрица неисправностей в этом случае приобретает вид (табл. 2.6). Схемы проведения комплексного контроля ОСИ существенно отличаются друг от друга, при этом важной диагностической информацией является как выявление дефектов и их идентификация по виду, так и определение бездефектного состояния ОСИ.

Таблица 2.6 – Матрица неисправностей для многоэтажных конструкций

Состояние	Метод		
	П ₁	П ₂	П ₃
S ₁	0	0	0
S ₂	1	1	1
S ₃	0	1	1
S ₄	0	1	1
S ₅	0	0	1
S ₆	1	0	0

Для поиска оптимального из построенных деревьев решений в качестве основного критерия оптимальности использовалась вероятность ошибки 2-го рода при использовании комплексного контроля по плану, который отвечает этому или другому дереву решений. Для выбора наиболее оптимального среди полученных оптимальных деревьев введены дополнительные критерии: цена контроля изолятора, продолжительность проверки, вероятность ошибки 1-го рода (ошибочная отбраковка). Полученные оптимальные деревья для ОСИ разных конструкций приведены на рис. 2.6 и 2.7.

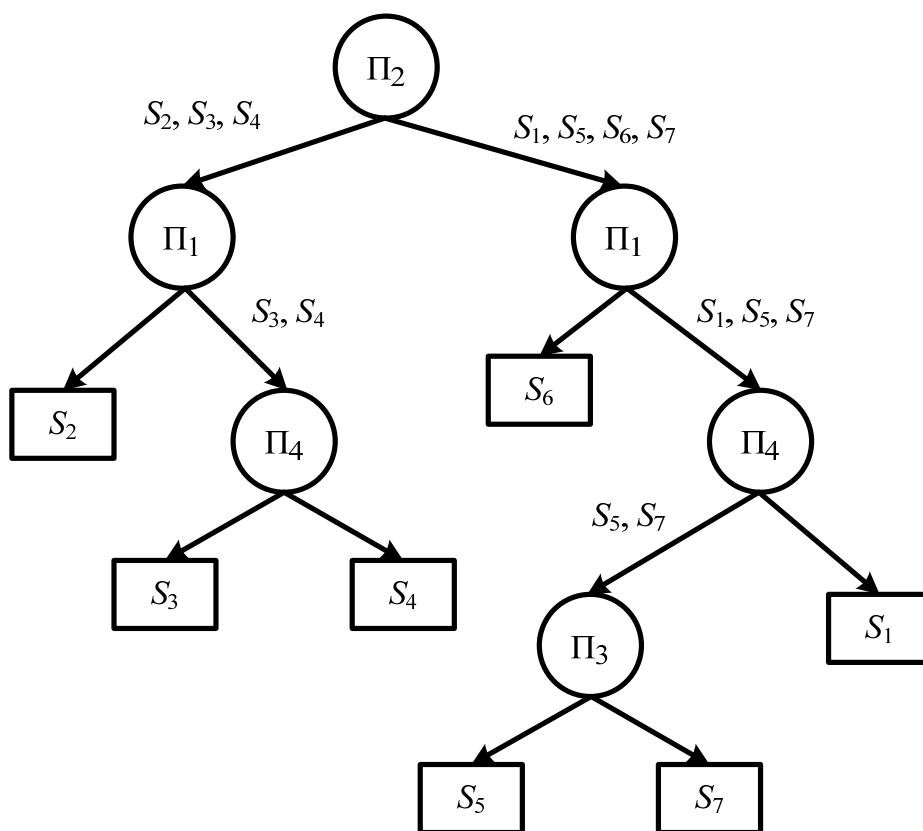


Рисунок 2.6 – Оптимальное дерево решений для ОСИ в составе разъединителей 35–110 кВ

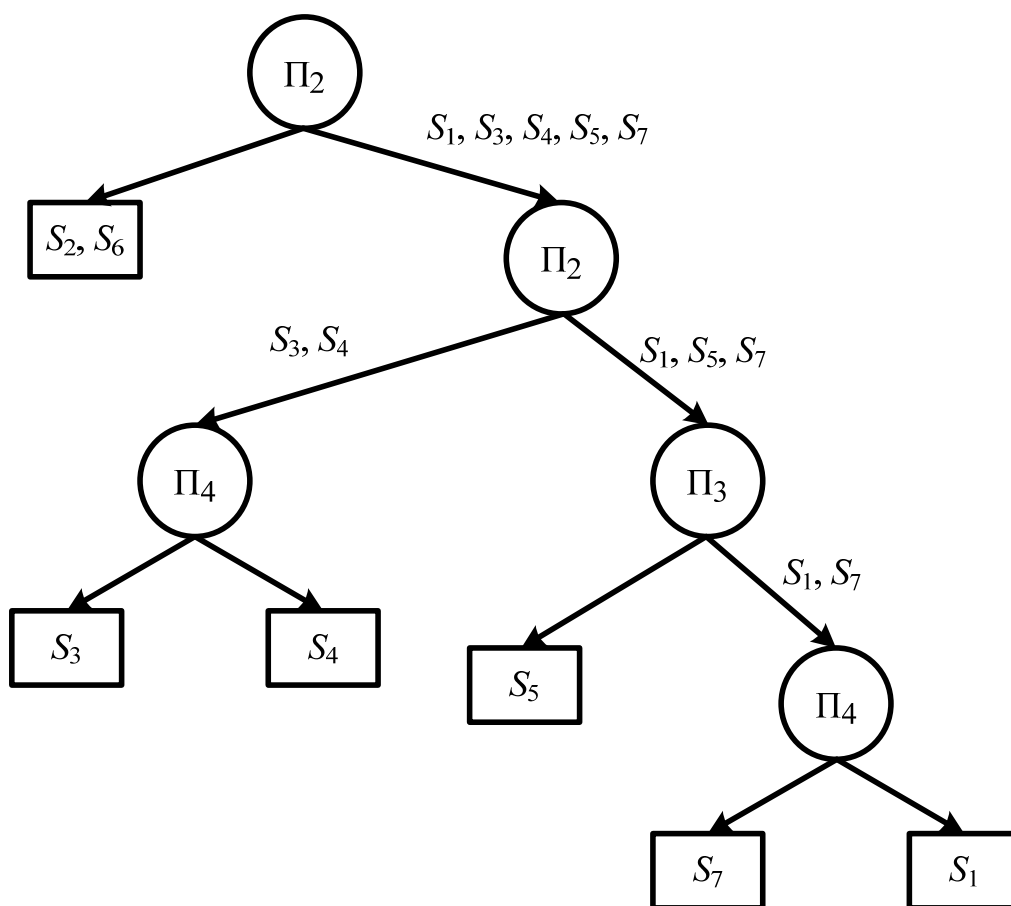


Рисунок 2.7 – Оптимальное дерево решений для ОСИ в составе многоэтажных конструкций

Таким образом, с использованием методов распознавания, основанных на теории графов, авторами получены оптимальные деревья решений для вариантов контроля ОСИ в составе разъединителей 35–110 кВ и в составе другого оборудования.

Вопросы для самоконтроля

1. Виды и структура графов. Дайте пояснения терминам: «вершины», «дуги», «лес», «корни».
2. Ранг дерева, максимальный и минимальный ранги.
3. Представление алгоритма диагноза с помощью дерева.
4. Основные допущения при построении диагностических тестов на основе ориентированного графа.
5. Матрица смежности и матрица путей.
6. Определение перечня неразличимых отказов.

7. Построение диагностических тестов на основе ориентированного графа.
8. Определение оптимального числа и вида изменений структуры системы.
9. Оценка технического состояния опорно-стержневых изоляторов с помощью моделей, построенных на основе графов.

Литература к разделу 2

Основная

1. Зыков А.А. Основы теории графов / А.А. Зыков. – М. : Наука, 1987. – 381 с;
2. Маркович З.П. Использование граф-модели для решения задач технической диагностики. В кн. «Кибернетика и диагностика» / З.П. Маркович. – Рига : Зинантне, 1966. – Вып. 2. – С. 49–62.
3. Введение в техническую диагностику / Верзаков Г.Ф., Киншт Н.В., Рабинович В.И., Тимонен Л.С. – М. : Энергия, 1968. – 224 с.

Дополнительная

4. Лабзун М.П. Комплексний метод оцінки технічного стану опорно-стрижневих ізоляторів / М.П. Лабзун, В.М. Кутін // Електромеханічні і енергозберігаючі системи. – Вінниця : ВНТУ. – 2011. – Вип. 3(15). – С. 140–143.
5. Лабзун М.П. Методи та засоби діагностування опорно-стрижневих ізоляторів : монографія / М.П. Лабзун, О.Є. Рубаненко, В.М. Кутін. – Вінниця : ВНТУ, 2009. – 272 с.
6. Кутін В.М. Діагностування електричних розподільчих мереж / В.М. Кутін, О.І. Кульматицький. – Київ : Техніка, 1993. – 160 с.
7. Нормы испытания электрооборудования: СОУ-Н ЕЕ 20.302:2007. – Офиц. изд. – Киев : ГРІФРЕ: М-во топлива и энергетики Украины, 2007. – 262 с.
8. Р. Беллман Прикладные задачи динамического программирования / Р. Беллман, С. Дрейфу. – М. : Наука, 1965. – 460 с.

РАЗДЕЛ 3

РАСПОЗНАВАНИЕ СОСТОЯНИЯ ЭЛЕКТРОЭНЕРГЕТИЧЕСКОГО ОБОРУДОВАНИЯ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

3.1. Возможности применения теории нечетких множеств для описания различных видов неопределенности

При диагностике сложных электроэнергетических объектов характерно наличие одновременно разнородной информации:

- 1) точечных замеров и значений параметров;
- 2) допустимых интервалов их изменения;
- 3) статистических законов распределения для отдельных величин;
- 4) лингвистических критериев и ограничений, полученных от специалистов-экспертов и т.д.

Наличие в сложной системе диагностики одновременно различных видов *неопределенности* делает необходимым использование для принятия решений теории нечетких множеств, которая позволяет адекватно учесть имеющиеся виды неопределенности [1].

Соответственно и вся информация о диагностических признаках, режимах работы электрических сетей, областях допустимости, предпочтительности одних методов диагностики перед другими, о рисках принятия решений и т.д. должна быть преобразована к единой форме и представлена в виде функций принадлежности. Такой подход позволяет свести воедино всю имеющуюся неоднородную информацию: детерминированную, статистическую, лингвистическую и интервальную.

Применение для работы с неопределенными величинами аппарата теории вероятности приводит к тому, что фактически неопределенность, независимо от ее природы, отождествляется со случайностью, между тем как основным источником неопределенности во многих процессах принятия решений является нечеткость или расплывчатость (*fuzziness*) [1, 2].

В отличие от *случайности*, которая связана с неопределенностью, касающейся принадлежности или непринадлежности некоторого объекта к

нерасплывчатому множеству, понятие «*нечеткость*» относится к классам, в которых могут быть различные градации степени принадлежности, промежуточные между полной принадлежностью и непринадлежностью объектов к данному классу.

Вопрос выбора адекватного формального языка является очень важным, поэтому следует отметить преимущества описания процесса принятия решений в сложной многоуровневой иерархической системе на основе теории нечетких множеств. Этот язык дает возможность адекватно отражать сущность самого процесса принятия решений и нечетких условий для многоуровневой системы, оперировать с нечеткими ограничениями и целями, а также задавать их с помощью лингвистических переменных.

Во многих задачах контроля и управления сложной системой нет необходимости в получении оптимального четкого решения для каждого момента времени, так как затраты на накопление информации и жесткое устранение невязок (ошибок) в системе могут превышать достигаемый при этом эффект. Чаще всего конкретное содержание задачи требует обеспечения заданного уровня нечеткости решения.

Реальные задачи содержат в себе нечеткие условия и некоторую нечеткость цели в связи с тем, что их постановку осуществляет человек. Если искусственное введение четких ограничений и целей при рассмотрении одноуровневых одноцелевых систем позволяет получать неплохие детерминированные модели, то для иерархических систем необходимо рассматривать работу любой подсистемы с точки зрения ее связей с подсистемами на всех уровнях управления.

Учет фактора неопределенности при решении задач во многом изменяет методы принятия решения; меняется принцип представления исходных данных и параметров модели, становятся неоднозначными понятия решения задачи и оптимальности решения.

Наличие неопределенности может быть учтено непосредственно в моделях соответствующего типа с представлением недетерминированных параметров как случайных величин с известными вероятностными характеристиками, как нечетких величин с заданными функциями принадлежности или как интервальных величин с фиксированными интервалами изменения и нахождения решения задачи с помощью методов стохастического, нечеткого или интервального программирования.

Возможно также и прямое построение зоны неопределенности без непосредственного учета характеристик недетерминированных параметров модели. В этом случае решается ряд детерминированных задач и получается некоторый набор вариантов, оптимальных при конкретных значениях случайных (или нечетких) параметров.

3.2. Нечеткие множества

Пусть E – универсальное множество, x – элемент E , а R – некоторое свойство. Обычное (четкое) подмножество A универсального множества E , элементы которого удовлетворяют свойству R , определяется как множество упорядоченных пар [3]:

$$A = \{\mu_A(x)/x\}, \quad (3.1)$$

где $\mu_A(x)$ – характеристическая функция, принимающая значение 1, если x удовлетворяет свойству R , и 0 – в противном случае.

Нечеткое подмножество отличается от обычного тем, что для элементов x из E нет однозначного ответа: «да – нет» относительно свойства R . В связи с этим нечеткое подмножество A универсального множества E определяется как множество упорядоченных пар

$$A = \{\mu_A(x)/x\}, \quad (3.2)$$

где $\mu_A(x)$ – характеристическая функция принадлежности (или просто функция принадлежности), принимающая значения в некотором вполне упорядоченном множестве M (например, $M = |0, 1|$).

Функция принадлежности указывает *степень* (или уровень) принадлежности элемента x подмножеству A . Множество M называют *множеством принадлежностей*. Если $M = \{0, 1\}$, то нечеткое подмножество A может рассматриваться как обычное или четкое множество.

Пример записи нечеткого множества

Пусть $E = \{x_1, x_2, x_3, x_4, x_5\}$, $M = |0,1|$; A – нечеткое множество, для которого:

$$\mu_A(x_1) = 0,3; \mu_A(x_2) = 0; \mu_A(x_3) = 1; \mu_A(x_4) = 0,5; \mu_A(x_5) = 0,9.$$

Тогда A можно представить в виде:

$$A = \{0,3/ x_1; 0/ x_2; 1/ x_3; 0,5/ x_4; 0,9/ x_5\}$$

или

$$A = 0,3/ x_1 + 0/ x_2 + 1/ x_3 + 0,5/ x_4 + 0,9/ x_5,$$

где «+» не является обозначением операции сложения, а имеет смысл объединения.

3.2.1. Основные характеристики нечетких множеств

Пусть $M = |0,1|$ и A – нечеткое множество с элементами из универсального множества E и множеством принадлежностей M .

Величина $\sup_{x \in E} \mu_A(x)$ называется *высотой* [3] нечеткого множества A .

Здесь аббревиатура \sup (супремум) – это точная верхняя граница множества. Нечеткое множество A *нормально*, если его высота равна 1, то есть верхняя граница его функции принадлежности равна 1 ($\sup_{x \in E} \mu_A(x) = 1$).

При $\sup_{x \in E} \mu_A(x) < 1$ нечеткое множество называется *субнормальным*.

Нечеткое множество *пусто*, если $\forall x \in E \mu_A(x) = 0$. Непустое субнормальное множество

$$\frac{\mu_A(x)}{\sup \mu_A(x)}$$

Нечеткое множество *унимодально*, $\mu_A(x) = 1$ только на одном x из E . *Носителем* нечеткого множества A является обычное подмножество со свойством $\mu_A(x) > 0$, то есть *носитель* $A = X\{x/ \mu_A(x) > 0\} \forall x \in E$.

Элементы $x \in E$, для которых $\mu_A(x) = 0,5$, называются *точками перехода* множества A .

Пример записи нечеткого множества [3]

Пусть $E = \{\text{Запорожец, Жигули, Мерседес, ...}\}$ – множество марок автомобилей, а $E' = [0, \infty]$ – универсальное множество «стоимость», тогда на E' мы можем определить нечеткие множества типа: «для бедных», «для среднего класса», «престижные», с функциями принадлежности, представленными на рисунке 3.1.

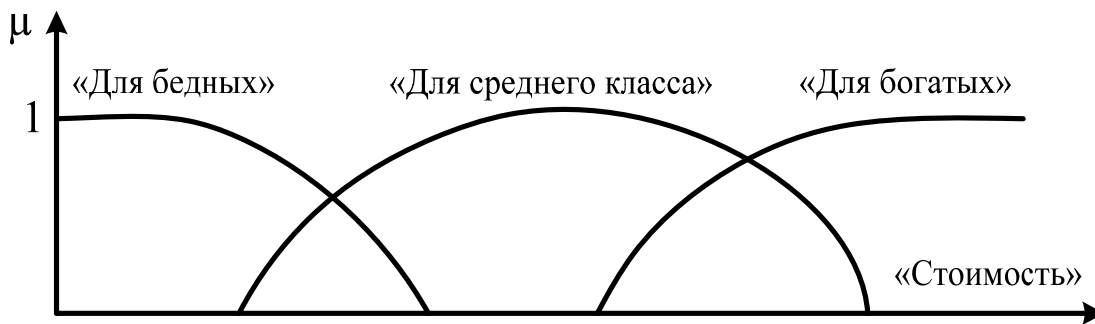


Рисунок 3.1 – Функции принадлежности для множества марок автомобилей

Имея эти функции и зная стоимости автомобилей из E в данный момент времени, мы тем самым определим на E нечеткие множества с этими же названиями. Так, например, нечеткое множество «для бедных», заданное на универсальном множестве $E = \{Запорожец, Жигули, Мерседес, \dots\}$ выглядит следующим образом.

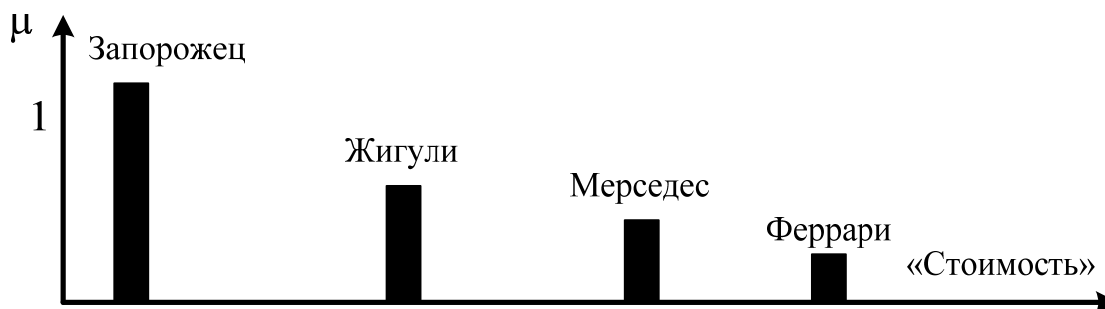


Рисунок 3.2 – Вид нечеткого множества "для бедных", заданное на универсальном множестве $E = \{Запорожец, Жигули, Мерседес, \dots\}$

Аналогично можно определить нечеткие множества «скоростные», «средние», «тихоходные» и т. д.

3.2.2. Методы построения функций принадлежности нечетких множеств

В приведенном выше примере использованы *прямые* методы, когда эксперт либо просто задает для каждого $x \in E$ значение $\mu_A(x)$, либо определяет функцию совместимости. Как правило, прямые методы задания функции принадлежности используются для измеримых понятий, таких как скорость, время, расстояние, давление, температура и т.д., или когда выделяются полярные значения [3].

Во многих задачах при характеристике объекта можно выделить набор признаков и для каждого из них определить полярные значения, соответствующие значениям функции принадлежности, 0 или 1.

Например, в задаче распознавания лиц можно выделить следующие шкалы (табл. 3.1).

Для конкретного лица A эксперт, исходя из приведенной шкалы, задает $\mu_A(x) \in [0, 1]$, формируя векторную функцию принадлежности $\{\mu_A(x_1), \mu_A(x_2), \dots, \mu_A(x_9)\}$.

При *прямых методах* используются также групповые прямые методы, когда, например, группе экспертов предъявляют конкретное лицо и каждый должен дать один из двух ответов: «этот человек лысый» или «этот человек не лысый», тогда количество утвердительных ответов, деленное на общее число экспертов, дает значение $\mu_{\text{«лысый»}}$ (данного лица). (В этом примере можно действовать через функцию совместимости, но тогда придется считать число волосинок на голове у каждого из предъявленных эксперту лиц).

Таблица 3.1 – Характерные шкалы для распознавания лиц

Код	Признак	0	1
x_1	высота лба	низкий	широкий
x_2	профиль носа	курносый	горбатый
x_3	длина носа	короткий	длинный
x_4	разрез глаз	узкие	широкие
x_5	цвет глаз	светлые	темные
x_6	форма подбородка	остроконечный	квадратный
x_7	толщина губ	тонкие	толстые
x_8	цвет лица	темный	светлый
x_9	очертание лица	овальное	квадратное

Косвенные методы [3] для определения значений функции принадлежности используются в случаях, когда нет элементарных измеримых свойств, через которые определяется интересующее нас нечеткое множество. Как правило, это методы попарных сравнений. Если бы значения функций принадлежности были нам известны, например, $\mu_A(x_i) = w_i$, $i =$

$= 1, 2, \dots, n$, то попарные сравнения можно представить матрицей отношений $A = \{a_{ij}\}$, где $a_{ij} = w_i/w_j$ (операция деления).

На практике эксперт сам формирует матрицу A , при этом предполагается, что диагональные элементы равны 1, а для элементов симметричных относительно диагонали $a_{ij} = 1/a_{ji}$ то есть если один элемент оценивается в α раз сильнее, чем другой, то этот последний должен быть в $1/\alpha$ раз сильнее, чем первый. В общем случае задача сводится к поиску вектора w , удовлетворяющего уравнению вида $A_w = \lambda_{\max} w$, где λ_{\max} – наибольшее собственное значение матрицы A . Поскольку матрица A положительна по построению, решение данной задачи существует и является положительным.

3.2.3. Операции над нечеткими множествами

Включение

Пусть A и B – нечеткие множества на универсальном множестве E .

Говорят, что A содержится в B , если $\forall x \in E \mu_A(x) \leq \mu_B(x)$.

Обозначение: $A \subset B$.

Иногда используют термин «доминирование», то есть в случае, когда $A \subset B$, говорят, что B доминирует над A .

Равенство

A и B равны, если $\forall x \in E \mu_A(x) = \mu_B(x)$.

Обозначение: $A = B$.

Дополнение

Пусть $M = [0,1]$, A и B – нечеткие множества, заданные на E . A и B дополняют друг друга, если $\forall x \in E \mu_A(x) = 1 - \mu_B(x)$.

Обозначение: $B = \bar{A}$ или $A = \bar{B}$.

Очевидно, что $\overline{(\bar{A})} = A$. (Дополнение определено для $M = [0,1]$, но очевидно, что его можно определить для любого упорядоченного M).

Пересечение

$A \cap B$ – наибольшее нечеткое подмножество, содержащееся одновременно в A и B . $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$.

Объединение

$A \cup B$ – наименьшее нечеткое подмножество, включающее как A , так и B , с функцией принадлежности:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

Разность

$A - B = A \cap \bar{B}$ с функцией принадлежности:

$$\mu_{A-B}(x) = \mu_A \cap \bar{B}(x) \min(\mu_A(x), 1 - \mu_B(x)).$$

Дизъюнктивная сумма

$A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (A \cap \bar{B})$ с функцией принадлежности:

$$\mu_{A-B}(x) = \max \left\{ \left[\min \{ \mu_A(x), 1 - \mu_B(x) \} \right] \left[\min \{ 1 - \mu_A(x), \mu_B(x) \} \right] \right\}$$

Для нечетких множеств можно строить визуальное представление. Рассмотрим прямоугольную систему координат, на оси ординат которой откладываются значения $\mu_A(x)$, на оси абсцисс в произвольном порядке расположены элементы X (мы уже использовали такое представление в примерах нечетких множеств). Если X по своей природе упорядочено, то этот порядок желательно сохранить в расположении элементов на оси абсцисс. Такое представление делает наглядными простые логические операции над нечеткими множествами (рис. 3.3).

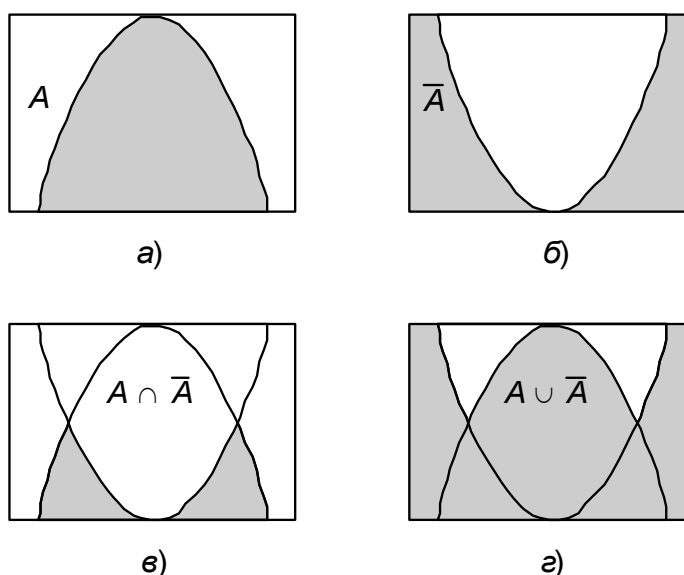


Рисунок 3.3 – Графическая интерпретация логических операций:

a – нечеткое множество A ; b – \bar{A} ; $в$ – $A \cap \bar{A}$; $г$ – $A \cup \bar{A}$

На верхней части рисунка заштрихованная часть соответствует нечеткому множеству A и, если говорить точно, изображает область значений A и всех нечетких множеств, содержащихся в A . На нижней – даны \bar{A} , $A \cap \bar{A}$, $A \cup \bar{A}$.

Замечание. Введенные выше операции над нечеткими множествами основаны на использовании операций \max и \min . В теории нечетких множеств разрабатываются вопросы построения обобщенных, параметризованных операторов пересечения, объединения и дополнения, позволяющих учесть разнообразные смысловые оттенки соответствующих им связей «И», «ИЛИ», «НЕ».

Один из подходов к операторам пересечения и объединения заключается в их определении в классе треугольных норм и конорм.

Треугольной нормой (t -нормой) называется двухместная действительная функция $T: [0,1] \times [0,1] \rightarrow [0,1]$, удовлетворяющая следующим условиям:

- 1) $T(0,0) = 0$; $T(\mu_A, 1) = \mu_A$; $T(1, \mu_A) = \mu_A$ – ограниченность;
- 2) $T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D)$, если $\mu_A \leq \mu_C$, $\mu_B \leq \mu_D$ – монотонность;
- 3) $T(\mu_A, \mu_B) = T(\mu_B, \mu_A)$ – коммутативность;
- 4) $T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C)$ – ассоциативность.

Простым случаем треугольных норм являются:

$$\min(\mu_A, \mu_B); \text{ произведение } \mu_A \cdot \mu_B; \max(0, \mu_A + \mu_B - 1).$$

Треугольной конормой (t -конормой) называется двухместная действительная функция $S: [0,1] \times [0,1] \rightarrow [0,1]$ со свойствами:

- 1) $S(1, 1) = 1$; $S(\mu_A, 0) = \mu_A$; $S(0, \mu_A) = \mu_A$ – ограниченность;
- 2) $S(\mu_A, \mu_B) \geq S(\mu_C, \mu_D)$, если $\mu_A \geq \mu_C$, $\mu_B \geq \mu_D$ – монотонность;
- 3) $S(\mu_A, \mu_B) = S(\mu_B, \mu_A)$ – коммутативность;
- 4) $S(\mu_A, S(\mu_B, \mu_C)) = S(S(\mu_A, \mu_B), \mu_C)$ – ассоциативность.

Примеры t -конорм:

$$\max(\mu_A, \mu_B); \mu_A + \mu_B - \mu_A \cdot \mu_B; \min(1, \mu_A + \mu_B).$$

3.2.4 Нечеткие отношения

Пусть $E = E_1 \times E_2 \times \dots \times E_n$ – прямое произведение универсальных множеств и M – некоторое множество принадлежностей (например $M = [0,1]$). Нечеткое парное отношение определяется как нечеткое подмножество R на E , принимающее свои значения в M . В случае $n = 2$ и $M = [0,1]$ нечетким отношением R между множествами $X = E_1$ и $Y = E_2$ будет называться функция $R: (X, Y) \rightarrow [0,1]$, которая ставит в соответствие каждой паре элементов $(x, y) \in X \times Y$ величину $\mu_R(x, y) \in [0,1]$.

Обозначение: нечеткое отношение на $X \times Y$ запишется в виде: $x \in X, y \in Y: xRy$. В случае, когда $X = Y$, то есть X и Y совпадают, нечеткое отношение $R : X \times X \rightarrow [0,1]$ называется *нечетким отношением на множестве X* .

Операции над нечеткими отношениями

Объединение двух отношений R_1 и R_2

Объединение двух отношений обозначается $R_1 \cup R_2$ и определяется выражением:

$$\mu_{R_1 \cup R_2}(x, y) = \mu_{R_1}(x, y) \vee \mu_{R_2}(x, y).$$

Пересечение двух отношений

Пересечение двух отношений R_1 и R_2 обозначается $R_1 \cap R_2$ и определяется выражением:

$$\mu_{R_1 \cap R_2}(x, y) = \mu_{R_1}(x, y) \wedge \mu_{R_2}(x, y).$$

Алгебраическое произведение двух отношений

Алгебраическое произведение двух отношений R_1 и R_2 обозначается $R_1 \cdot R_2$ и определяется выражением:

$$\mu_{R_1 \cdot R_2}(x, y) = \mu_{R_1}(x, y) \cdot \mu_{R_2}(x, y).$$

Алгебраическая сумма двух отношений

Алгебраическая сумма двух отношений R_1 и R_2 обозначается $R_1 \hat{+} R_2$ и определяется выражением:

$$\mu_{R_1 \hat{+} R_2}(x, y) = \mu_{R_1}(x, y) + \mu_{R_2}(x, y) - \mu_{R_1}(x, y) \cdot \mu_{R_2}(x, y).$$

Для введенных операций справедливы следующие свойства дистрибутивности:

$$R_1 \cap (R_2 \cup R_3) = (R_1 \cap R_2) \cup (R_1 \cap R_3),$$

$$R_1 \cup (R_2 \cap R_3) = (R_1 \cup R_2) \cap (R_1 \cup R_3),$$

$$R_1 \cdot (R_2 \cup R_3) = (R_1 \cdot R_2) \cup (R_1 \cdot R_3),$$

$$R_1 \cdot (R_2 \cap R_3) = (R_1 \cdot R_2) \cap (R_1 \cdot R_3),$$

$$R_1 \hat{+} (R_2 \cup R_3) = (R_1 \hat{+} R_2) \cup (R_1 \hat{+} R_3),$$

$$R_1 \hat{+} (R_2 \cap R_3) = (R_1 \hat{+} R_2) \cap (R_1 \hat{+} R_3).$$

Дополнение отношения

Дополнение отношения R обозначается \bar{R} и определяется функцией принадлежности:

$$\mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y).$$

Дизъюнктивная сумма двух отношений

Дизъюнктивная сумма двух отношений R_1 и R_2 обозначается $R_1 \oplus R_2$ и определяется выражением:

$$R_1 \oplus R_2 = (R_1 \cap \bar{R}_2) \cup (\bar{R}_1 \cap R_2).$$

Обычное отношение, ближайшее к нечеткому

Пусть R – нечеткое отношение с функцией принадлежности $\mu_R(x, y)$. Обычное отношение, ближайшее к нечеткому, обозначается \underline{R} и определяется выражением:

$$\mu_{\underline{R}}(x, y) = \begin{cases} 0, & \text{если } \mu_R(x, y) < 0,5, \\ 1, & \text{если } \mu_R(x, y) > 0,5, \\ 0 \text{ или } 1, & \text{если } \mu_R(x, y) = 0,5. \end{cases}$$

По договоренности принимают $\mu_{\underline{R}}(x, y) = 0$ при $\mu_R(x, y) = 0,5$.

Композиция (свертка) двух нечетких отношений

Пусть R_1 – нечеткое отношение $R_1: (X \times Y) \rightarrow [0,1]$ между X и Y , и R_2 – нечеткое отношение $R_2: (Y \times Z) \rightarrow [0,1]$ между Y и Z . Нечеткое отношение между X и Z , обозначаемое $R_2 \otimes R_1$, определенное через R_1 и R_2 выражением

$$\mu_{R_1 \otimes R_2}(x, z) = \bigcup_y [\mu_{R_1}(x, y) \cap \mu_{R_2}(y, z)]$$

называется (max-min)-композицией ((max-min)-сверткой) отношений R_1 и R_2 .

3.2.5 Нечеткая импликация

Пусть имеем обычные («четкие») высказывания вида: $p = \langle x \text{ есть } A \rangle$ и $q = \langle y \text{ есть } B \rangle$. Импликацией (в обычной, «четкой» логике) «если p , то q » называется предложение, которое ложно тогда и только тогда, когда p истинно, а q ложно.

Такой вид отношений между высказываниями обычно обозначается как $p \rightarrow q$ и понимается как утверждение « p влечет за собой q ».

Если перейти к бинарной (булевой) алгебре логики, где с понятием «истина» сопоставляется 1, а с понятием «ложь» – 0, то импликацию можно представить логической формулой

$$p \rightarrow q = \bar{p} \vee q = \overline{(p \wedge \bar{q})},$$

или таблицей истинности:

p	q	$p \rightarrow q$
1	1	1
0	1	1
0	0	1
1	0	0

Более полная трактовка понятия импликации означает, что истинность $p \rightarrow q$ – это то же, что истинность утверждения «степень истинности q не меньше, чем степень истинности p », то есть

$$p \rightarrow q = \begin{cases} 1, & \text{если } \tau(p) \leq \tau(q), \\ 0 & \text{в противоположном случае,} \end{cases}$$

где $\tau(p)$ и $\tau(q)$ – указанные «степени истинности» (в четкой логике принимающие значения лишь 0 или 1).

Нечеткая импликация определяется обычно следующим образом.

Будем считать, что заданы универсальные множества X и Y , содержащие конечное число элементов. Под способом определения нечеткой импликации «**если** A , **то** B », где A и B – нечеткие множества на X и Y соответственно (например: «**если температура – большая, то скорость близка к нулю**»), будем понимать способ задания нечеткого отношения R на $X \times Y$, соответствующего данному высказыванию.

Такое отношение можно задать по-разному, поэтому для математического представления нечеткой импликации предложено большое число различных формул, некоторые из которых вместе с фамилиями предложивших их авторов приведены ниже:

- 1) Ларсен (Larsen) – $\mu_R(x, y) = \mu_A(x)\mu_B(y)$;
- 2) Лукасевич (Lukasiewicz): – $\mu_R(x, y) = \min\{1, 1 - \mu_A(x) + \mu_B(y)\}$;
- 3) Мамдани (Mamdani): – $\mu_R(x, y) = \min\{\mu_A(x), \mu_B(y)\}$;
- 4) Брауэра: – $\mu_R(x, y) = \begin{cases} 1, & \text{если } \mu_A(x) \leq \mu_B(y), \\ \mu_B(y) & \text{в противоположном случае;} \end{cases}$
- 5) Гёдель (классическая нечеткая импликация):

$$\mu_R(x, y) = \max\{1 - \mu_A(x), \mu_B(y)\}.$$

В общем, какого-либо преимущества одной формулы над другими нет, поэтому все они (а к настоящему времени известно несколько десятков подобных формул) имеют право на существование.

3.3. Нечеткая и лингвистическая переменные

Понятие нечеткой и лингвистической переменных используется при описании объектов и явлений с помощью нечетких множеств [3].

Нечеткая переменная характеризуется тройкой $\langle \alpha, X, A \rangle$,

где α – наименование переменной;

X – универсальное множество (область определения α);

A – нечеткое множество на X , описывающее ограничения (то есть $\mu_A(x)$) на значения нечеткой переменной α .

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$,

где β – наименование лингвистической переменной;

T – множество ее значений (терм-множество), представляющих собой наименования нечетких переменных, областью определения каждой из которых является множество X . Множество T называется базовым терм-множеством лингвистической переменной;

G – *синтаксическая процедура*, позволяющая оперировать элементами терм-множества T , в частности, генерировать новые термы (значения). Множество $T \cup G(T)$, где $G(T)$ – множество сгенерированных термов, называется расширенным терм-множеством лингвистической переменной;

M – *семантическая процедура*, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G , в нечеткую переменную, то есть сформировать соответствующее нечеткое множество.

Чтобы избежать большого количества символов, символ β используют для названия самой переменной, а также для всех ее значений.

Пользуются одним и тем же символом для обозначения нечеткого множества и его названия, например терм «*молодой*», являющийся значением лингвистической переменной $\beta =$ «*возраст*», одновременно есть и нечеткое множество M («*молодой*»). Присвоение нескольких значений символам предполагает, что контекст позволяет разрешить возможные неопределенности.

3.4. Алгоритм формирования нечеткого вывода

Процедура принятия решения с помощью теории нечетких множеств состоит из трёх этапов: *фаззификации*, *вычисления правил* и *дефаззификации* (рис. 3.4).

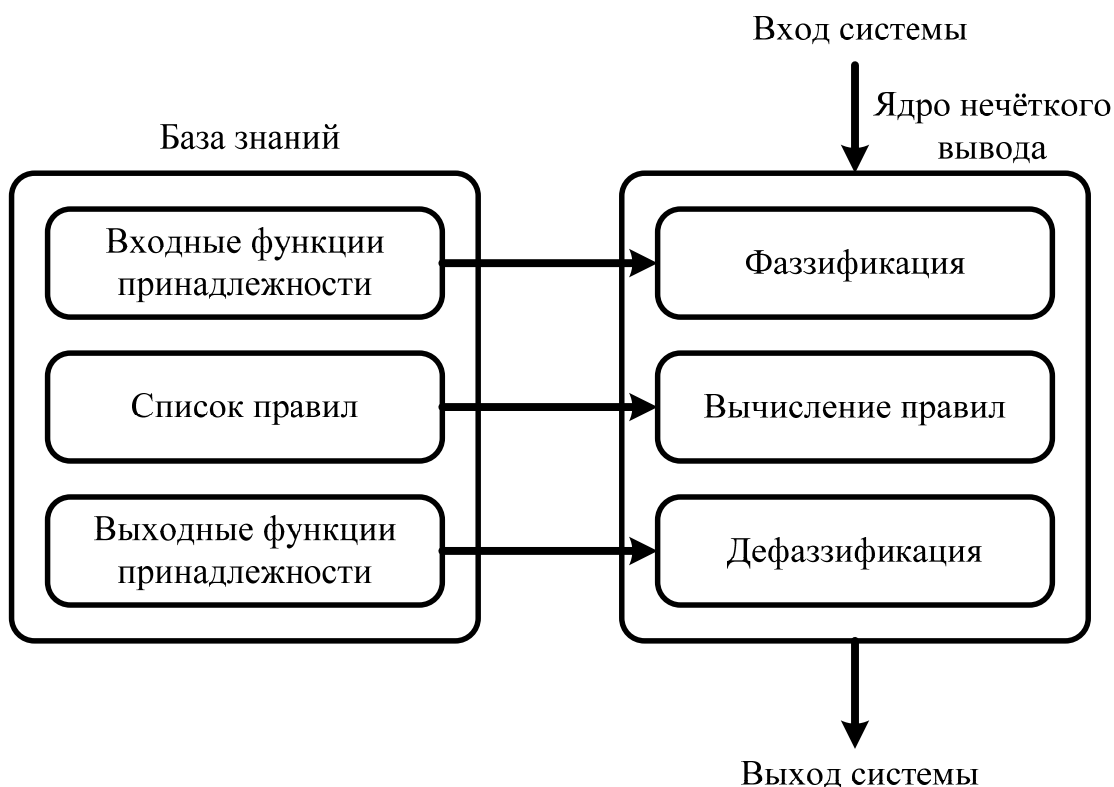


Рисунок 3.4 – Структура алгоритма системы нечёткого вывода

Фаззификация. Алгоритм работы системы основывается на использовании лингвистических переменных (ЛП) [4]. Поэтому для построения системы с нечёткой логикой (НЛ) необходимо первоначально выделить в задаче те ЛП, которые влияют на выходной результат системы.

Далее ЛП должна быть разделена на несколько лингвистических термов (ЛТ), характеризующих особенности состояния данной ЛП. Обычно, ЛП состоит из трёх – пяти ЛТ. Каждый ЛТ имеет характеристическую функцию, называемую функцией принадлежности (ФП). Эта функция характеризует степень достоверности понятия, соответствующего ЛТ, в зависимости от текущего значения величины измеряемого параметра. Набор ФП должен перекрывать возможный диапазон измеряемой величины. ФП, в принципе, могут иметь произвольную форму, но на практике, особенно для задач управления, часто используют ФП, имеющие треугольную или трапецеидальную форму, так как их проще реализовать и они поддерживаются большинством процессов, которые имеют специальный набор команд для НЛ.

Таким образом, на этапе фаззификации производится приём входного чёткого значения для системы с НЛ, а также поочерёдное считывание из

базы данных всех **ФП**, которые имеют отношение к данному входному значению, и устанавливается значение достоверности измеренного значения каждому **ЛТ** по функциям принадлежности.

Результатом работы блока фаззификации является массив нечётких выходов, которые представляют собой набор пар: **ЛТ** и его значение достоверности. Следующий этап – вычисление правил.

Вычисление правил. Для вычисления правил используются нечёткие входы, полученные от блока фаззификации, и правила, находящиеся в базе данных. Система с НЛ должна иметь базу правил, которая, по сути, представляет собой эмпирические знания эксперта о механизме управления. Правило имеет вид ЕСЛИ [], ТО []. В левой части правил перебираются возможные описания ситуации на входе системы, а в правой части указывается, какой **ЛТ** в этом случае описывает правильную реакцию системы.

Дефаззификация. Это заключительный шаг в алгоритме НЛ, на котором происходит преобразование нечёткой информации, содержащейся в виде значений достоверности **ЛТ**, в чётко определённое значение. Дефаззификация производится по фигуре, полученной путём сложения всех **ФП** термов выходной **ЛП**, причём от **ФП** берётся лишь нижняя часть (до вычисленного значения её достоверности). Существует много алгоритмов дефаззификации. Чаще всего используются следующие из них: алгоритмы *Mamdani, Tsukamoto, Sugeno, Larsen* [4].

В настоящее время разработано множество стратегий вычисления массива нечётких выводов. Самые распространённые из них – центроидный метод, метод первого максимума, метод среднего максимума, высотная дефаззификация, метод произвольного максимума [4].

Если базу знаний организуют два нечётких правила вида [4]:

$$\begin{aligned} P_1 : \text{ЕСЛИ } [x \in A_1] \text{ И } [y \in B_1], \text{ ТО } [z \in C_1], \\ P_2 : \text{ЕСЛИ } [x \in A_2] \text{ И } [y \in B_2], \text{ ТО } [z \in C_2], \end{aligned} \quad (3.3)$$

где $x, y(z)$ – имена входных (выходных) лингвистических переменных; $A_1, A_2, B_1, B_2, C_1, C_2$ – некоторые нечёткие множества, описываемые своими функциями принадлежности (рис. 3.5), то дефаззификация на основе алгоритма *Mamdani* производится в следующем порядке.

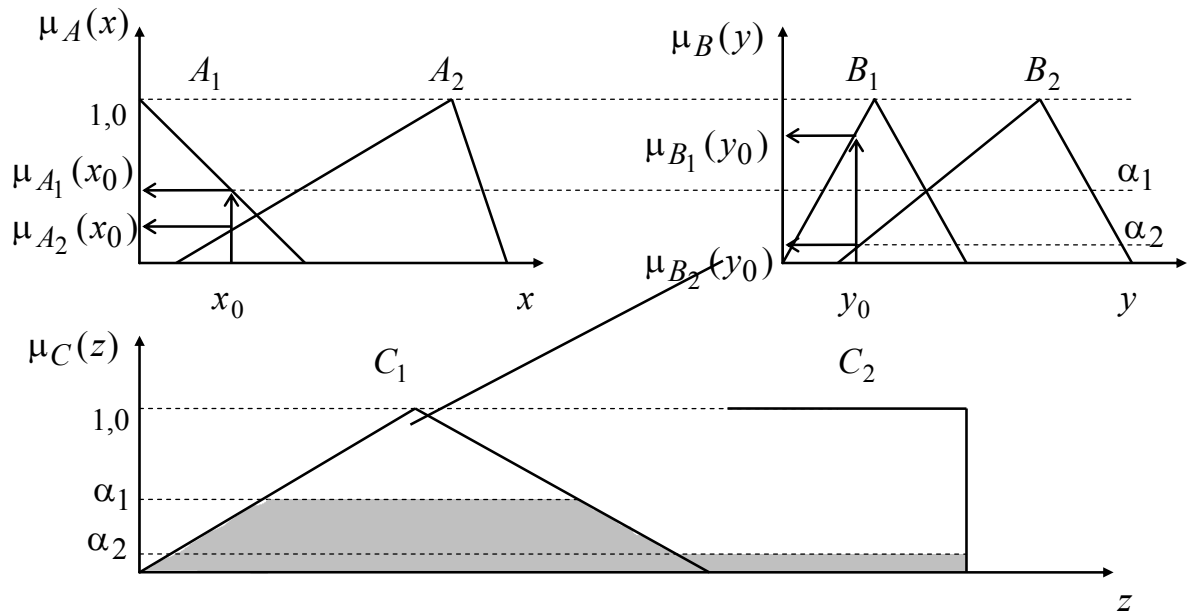


Рисунок 3.5 – Пример входных и выходных функций принадлежности

Фаззификация (приведение к нечёткости). Определяются степени истинности для предпосылок каждого правила (рис. 3.4):

$$\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{B_1}(y_0), \mu_{B_2}(y_0).$$

Вычисление правил (логический вывод). Находятся уровни отсечения для предпосылок каждого из правил с использованием операции минимум:

$$\begin{aligned} \alpha_1 &= \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0), \\ \alpha_2 &= \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0). \end{aligned} \quad (3.4)$$

Затем находятся усечённые функции принадлежности:

$$\mu_{C'_1}(z) = (\alpha_1 \wedge \mu_{C_1}(z)), \quad \mu_{C'_2}(z) = (\alpha_2 \wedge \mu_{C_2}(z)). \quad (3.5)$$

Производится объединение найденных усечённых функций принадлежности с использованием операции максимум. Это приводит к получению итогового нечёткого подмножества для переменной выхода с функцией принадлежности:

$$\mu_{\Sigma}(z) = \mu_C(z) = \mu_{C'_1}(z) \vee \mu_{C'_2}(z) = (\alpha_1 \wedge \mu_{C_1}(z)) \vee (\alpha_2 \wedge \mu_{C_2}(z)). \quad (3.6)$$

Дефаззификация (приведение к чёткости). Осуществляется с помощью центроидного метода, называемого ещё методом центра тяжести

$$z_0 = \frac{\int_{-\infty}^{+\infty} \mu_{\Sigma}(z) \cdot z \, dz}{\int_{-\infty}^{+\infty} \mu_{\Sigma}(z) \, dz}. \quad (3.7)$$

Существуют прямые методы определения функции принадлежности, когда эксперт просто задает значение $\mu_A(x)$ для каждого $x \in E$. При прямых методах используются также групповые прямые методы, когда опрос проводится у множества экспертов с целью получения положительного или отрицательного ответа, затем положительное число ответов делится на общее число экспертов и получается $\mu(x)$. Эти методы применяются для величин, которые легко измеряются (скорость, расстояние, время, температура и т. д.).

Кроме указанного в **НЛ**, для задания функций принадлежности используются также типовые формы приведенных ниже функций (рис. 3.6):

- треугольная (**trimf**);
- трапецеидальная (**trapmf**);
- гауссова (**gaussmf**);
- двойная гауссова (**gauss2mf**);
- обобщенная колоколообразная (**gbellmf**);
- сигмоидальная (**sigmf**);
- двойная сигмоидальная (**dsigmf**);
- произведение двух сигмоидальных функций (**psigmf**);
- Z-функция;
- S-функция;
- Pi-функция.

Конкретный вид данных функций определяется значениями параметров, входящих в их аналитические представления, например:

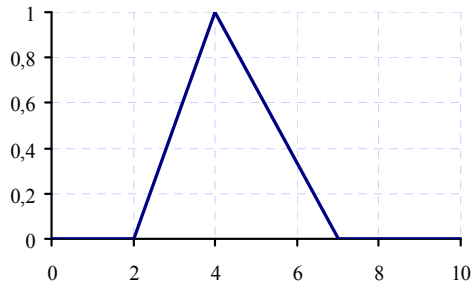
$$\text{trimf}(x, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right),$$

$$\text{trapmf}(x, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right),$$

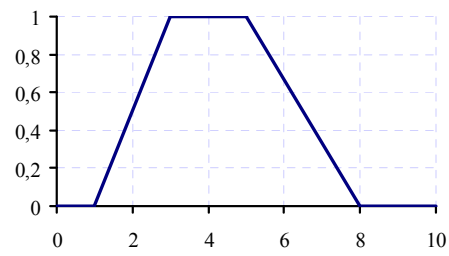
$$\text{gaussmf}(x, \sigma, c) = e^{-\left(\frac{x-c}{\sigma}\right)^2},$$

$$\text{gbellmf}(x, a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}},$$

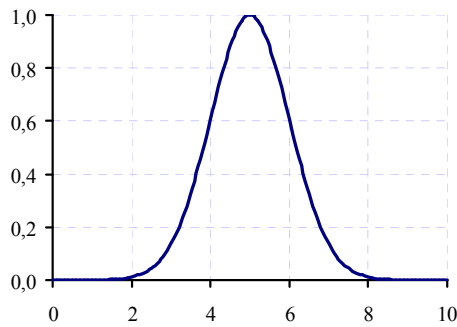
$$\text{sigmf}(x, a, c) = \frac{1}{1 + \exp(-a(x-c))}.$$



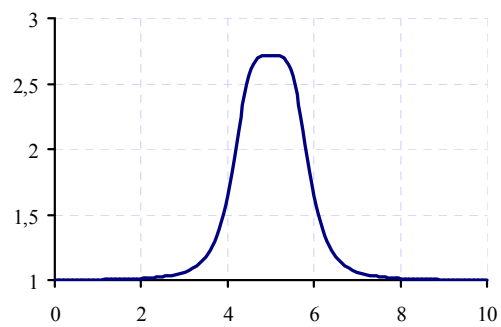
a



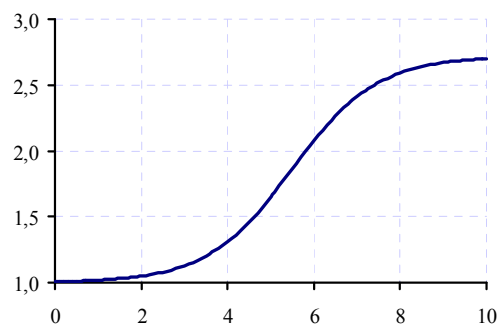
б



в



г



д

Рисунок 3.6 – Типовые функции принадлежности нечетких множеств:
a – треугольная (**trimf**); *б* – трапецеидальная (**trapmf**); *в* – гауссова (**gaussmf**);
г – обобщенная колоколообразная (**gbellmf**); *д* – сигмоидальная (**sigmf**).

Используемый в различного рода экспертных и управляющих системах механизм нечетких выводов в своей основе имеет *базу знаний*, формируемую специалистами предметной области в виде совокупности нечетких предикатных правил вида:

Π_1 : если x есть A_1 , то y есть B_1 ,

Π_2 : если x есть A_2 , то y есть B_2 ,

...

Π_n : если x есть A_n , то y есть B_n ,

где x – входная переменная (имя для известных значений данных); y – переменная вывода (имя для значения данных, которое будет вычислено); A и B – функции принадлежности, определенные соответственно на x и y .

Общий логический вывод осуществляется за следующие четыре этапа.

1. *Нечеткость* (введение нечеткости, фаззификация). Функции принадлежности, определенные на входных переменных, применяются к их фактическим значениям для определения степени истинности каждой предпосылки каждого правила.

2. *Логический вывод*. Вычисленное значение истинности для предпосылок каждого правила применяется к заключениям каждого правила. Это приводит к одному нечеткому подмножеству, которое будет назначено каждой переменной вывода для каждого правила. В качестве правил логического вывода обычно используются только операции \min (минимум) или prod (умножение). В логическом выводе \min функция принадлежности вывода «отсекается» по высоте, соответствующей вычисленной степени истинности предпосылки правила (нечеткая логика «И»). В логическом выводе prod функция принадлежности вывода масштабируется при помощи вычисленной степени истинности предпосылки правила.

3. *Композиция*. Нечеткие подмножества, назначенные для каждой переменной вывода (во всех правилах), объединяются вместе, чтобы сформировать одно нечеткое подмножество для каждой переменной вывода. При подобном объединении обычно используются операции \max (максимум) или sum (сумма). При композиции \max комбинированный вывод нечеткого подмножества конструируется как поточечный максимум по всем нечетким подмножествам (нечеткая логика «ИЛИ»). При композиции

sum комбинированный вывод нечеткого подмножества конструируется как поточечная сумма по всем нечетким подмножествам, назначенным переменной вывода правилами логического вывода.

4. В заключение (дополнительно) – *приведение к четкости* (дефаззификация), которое используется, когда полезно преобразовать нечеткий набор выводов в четкое число.

3.5. Алгоритмы нечеткого вывода

3.5.1. Алгоритм Sugeno

Sugeno и *Takagi* использовали набор правил в следующей форме:

Π_1 : если x есть A_1 , то y есть B_1 , тогда $z_1 = a_1x + b_1y$,

Π_2 : если x есть A_2 , то y есть B_2 , тогда $z_2 = a_2x + b_2y$.

Представление алгоритма.

Первый этап – как в алгоритме *Mamdani*.

На втором этапе находятся $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$

и индивидуальные выходы правил:

$$\begin{aligned} z_1^* &= a_1x_0 + b_1y_0, \\ z_2^* &= a_2x_0 + b_2y_0. \end{aligned} \tag{3.8}$$

На третьем этапе определяется четкое значение переменной вывода:

$$z_0 = \frac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}. \tag{3.9}$$

Алгоритм иллюстрируется на рис. 3.7. Приведенное представление относится к алгоритму *Sugeno* 1-го порядка. Если правила записаны в форме

Π_1 : если x есть A_1 , то y есть B_1 , тогда $z_1 = c_1$,

Π_2 : если x есть A_2 , то y есть B_2 , тогда $z_2 = c_2$,

то говорят, что задан алгоритм *Sugeno* 0-го порядка (рис. 3.8).

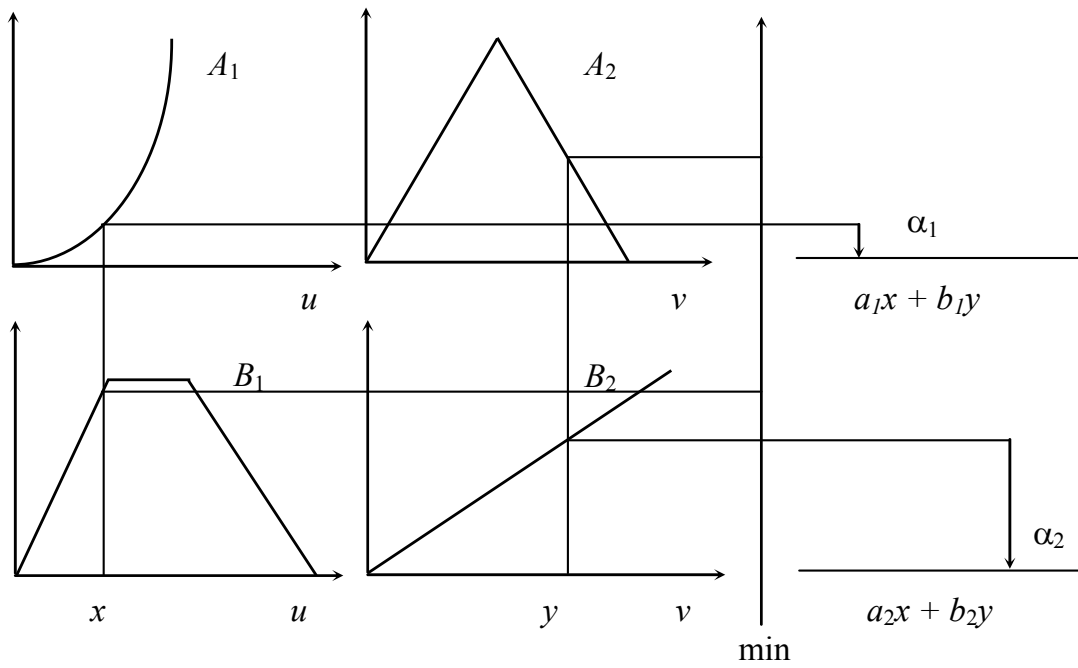


Рисунок 3.7 – Иллюстрация к алгоритму *Sugeno*

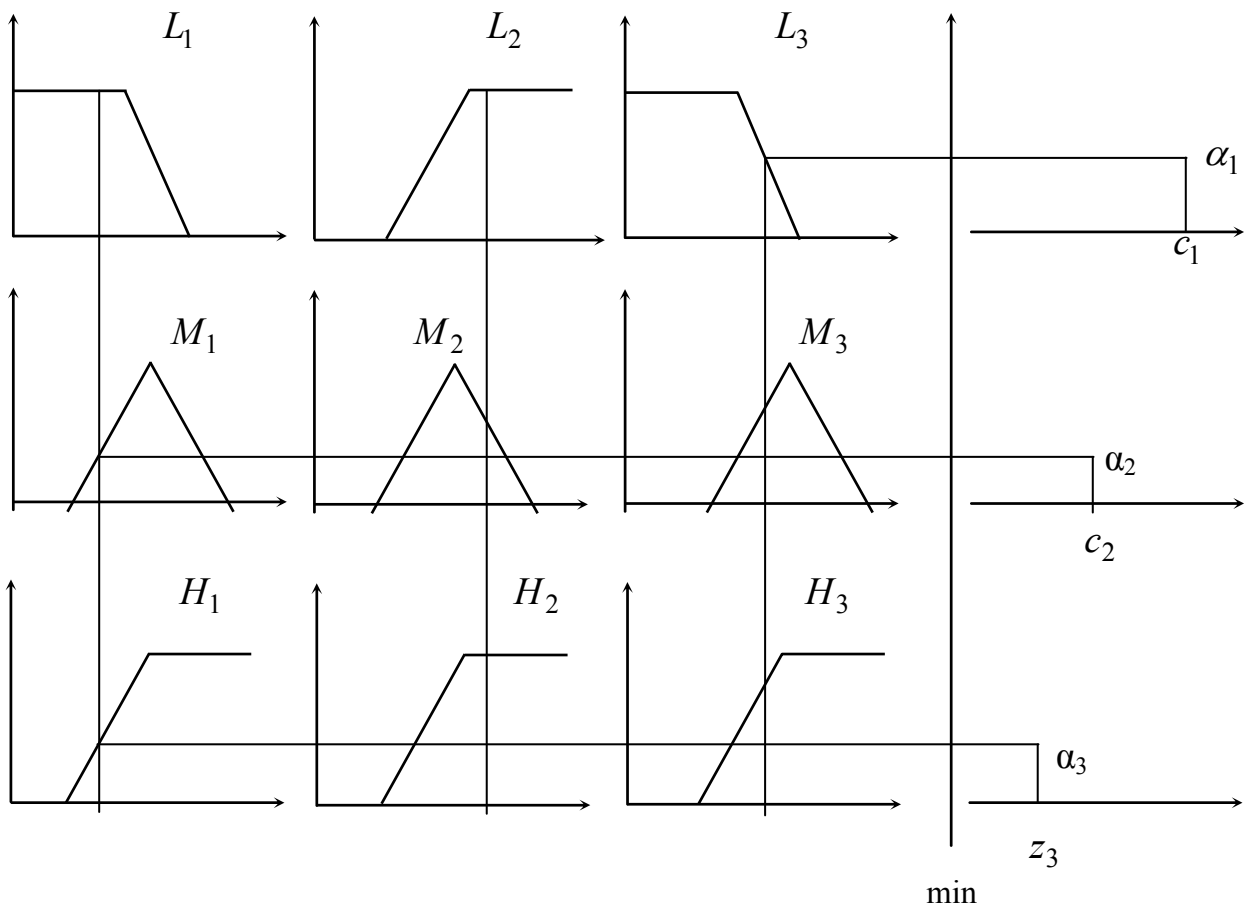


Рисунок 3.8 – Алгоритм *Sugeno* 0-го порядка

3.5.2. Алгоритм Tsukamoto

1. Введение нечеткости (как и в алгоритме *Mamdani*). Отличие: предполагается, что $\mu_{C_1}(z)$ и $\mu_{C_2}(z)$ – монотонные функции.

2. Нечеткий вывод. Сначала находятся уровни «отсечения» α_1 и α_2 (как в алгоритме *Mamdani*), а затем решением уравнений $\alpha_1 = \mu_{C_1}(z_1)$, $\alpha_2 = \mu_{C_2}(z_2)$ определяются четкие значения z_1 и z_2 для каждого исходного правила.

3. Определяется четкое значение переменной вывода (как взвешенное среднее)

$$z_0 = \frac{\alpha_1 \cdot z_1 + \alpha_2 \cdot z_2}{\alpha_1 + \alpha_2},$$

$$z_0 = \frac{\sum_i \alpha_i \cdot z_i}{\sum_i \alpha_i}.$$

Если z_i много, то применяется дискретный вариант центроидного метода.

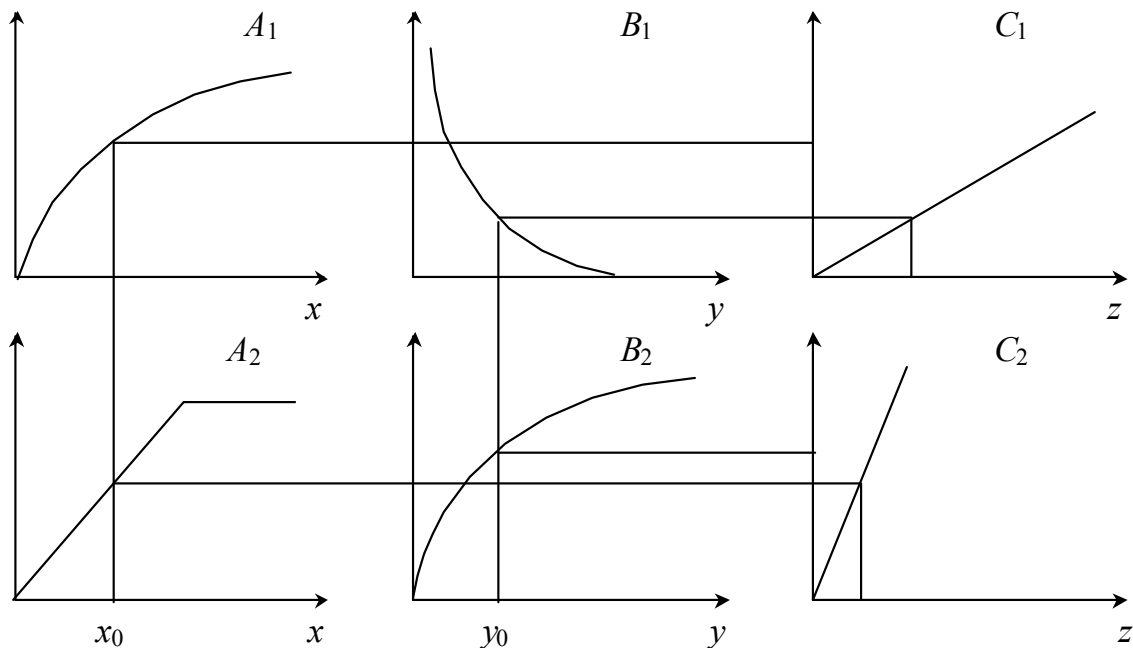


Рисунок 3.9 – Иллюстрация к алгоритму *Tsukamoto*

3.5.3. Алгоритм Larsen

1. Введение нечеткости.

2. Нечеткий вывод (α_1, α_2)

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0),$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0).$$

Затем определяются частные нечеткие подмножества

$$\mu_{C_1}^*(z) = \alpha_1 \cdot \mu_{C_1}(z); \mu_{C_2}^*(z) = \alpha_2 \cdot \mu_{C_2}(z).$$

3. Находится итоговое нечеткое подмножество

$$\mu_{\Sigma}(z) = \mu_C(z) = (\alpha_1 \cdot \mu_{C_1}(z) \vee (\alpha_2 \cdot \mu_{C_2}(z))).$$

В общем случае при n -правил

$$\mu_{\Sigma}(z) = \mu_C(z) = \bigvee_{i=1}^n (\alpha_i \cdot \mu_{C_i}(z)).$$

4. Затем производится приведение к четкости с помощью центроидного метода

$$z_0 = \frac{\int_{-\infty}^{+\infty} \mu_{\Sigma}(z) \cdot z dz}{\int_{-\infty}^{+\infty} \mu_{\Sigma}(z) dz}.$$

Алгоритм *Larsen* представлен на рис. 3.10.

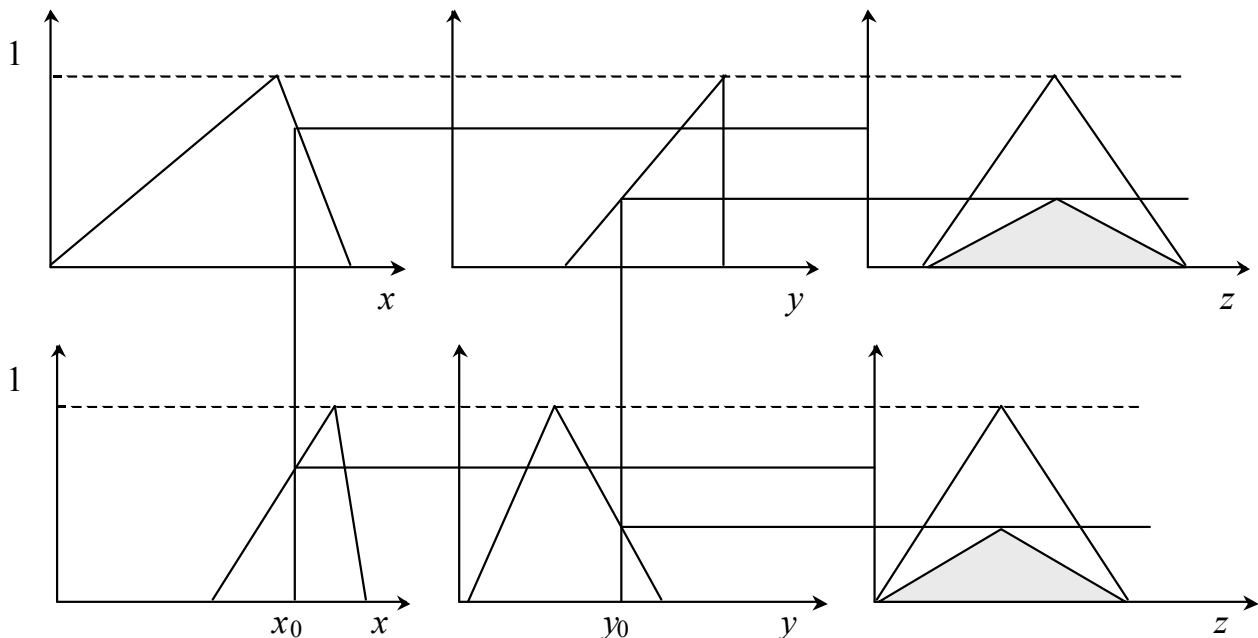


Рисунок 3.10 – Иллюстрация к алгоритму *Larsen*

Применение НЛ обеспечивает хорошие результаты, если:

- автоматизируемый процесс имеет очень сложную структуру с частично неизвестными или трудно измеряемыми параметрами, либо с неизвестными внутренними механизмами действия;
- отсутствует математическая модель процесса, либо имеется только качественная модель процесса, которым надо управлять;
- выходной величиной процесса или несколькими выходными величинами процесса управляет множество входных величин с разными видами управляющих сигналов (дискретные, непрерывные), нередко связанных друг с другом сложными способами;
- при управлении системой должны одновременно решаться различные задачи автоматизации (например – измерение, управление, регулирование, наблюдение, оптимизация, распознавание образов), связанные друг с другом.

В ряду основных промышленных применений теории нечетких множеств можно назвать управление, диагностику неисправностей, распознавание образов, обработку изображений, анализ надежности, проектирование систем, компьютеры, экспертные системы и т.д.

Для практической реализации аппарата нечеткой логики (НЛ) существуют три основных подхода:

- *программный* – реализация аппарата НЛ осуществляется программным способом на обычных контроллерах, не имеющих в своей программной модели команд, специально разработанных для реализации аппарата НЛ. Преимущества данного подхода заключаются в максимальной гибкости и экономичности разработанной системы. Недостаток подхода – требуются большие вычислительные мощности;
- *программно-аппаратный* – аппарат НЛ реализуется на специальных контроллерах, имеющих в своей программной модели основной набор команд для выполнения операций с НЛ;
- *аппаратный* – аппарат НЛ реализуется на специализированных элементах, которые обеспечивают максимальное быстродействие.

3.6. Интерпретация результатов хроматографического анализа растворенных в масле газов с использованием аппарата нечеткой логики

В качестве примера практического использования аппарата НЛ для решения задач диагностики рассмотрим реализацию метода обнаружения развивающихся дефектов трансформаторов на основе анализа динамики изменения концентраций газов, приведенного в разделе 6.6 первой части данного пособия.

Несмотря на то что алгоритм распознавания имеет практически детерминированный характер, отдельные его процедуры имеют элементы неопределенности (нечеткости). В частности, определение значений ключей (K_1 , K_2 , K_3) производится путем сравнения расчетных значений коэффициентов парной корреляции r с табличными значениями. Однако принимаемое решение будет зависеть не только от расчетного значения r , но и от выбранного уровня значимости α . В результате при одном и том же значении r принимаемое решение может измениться на противоположное, если изменится значение доверительной вероятности. Более того, наличие шумовой компоненты в результатах измерений приводит к тому, что они являются условно определенными. Поэтому практическую реализацию предложенного решающего правила целесообразней всего выполнить с использованием аппарата нечеткой логики.

Реализация данного правила состоит из трех этапов.

1. Определение состояния ключей. Входными переменными являются значения коэффициентов парной корреляции пяти газов со временем, между газами углеводородного ряда в диагностируемом трансформаторе и между газами углеводородного ряда в соседних трансформаторах. Выходными переменными являются значения ключей в диапазоне 0–1.

2. Диагностика трансформатора. Входными переменными являются значения переменных K_1 , K_2 , K_3 , которые получены в предыдущем блоке. Выходной переменной является один из возможных диагнозов из табл. 3.2.

3. Если поставленный на предыдущем этапе диагноз соответствует наличию дефекта, тогда производится уточнение вида дефекта согласно табл. 3.2. Входными переменными являются отношения пар газов, выходной переменной является вид дефекта.

Рассмотрим последовательно реализацию каждого из этапов [6] в

среде *Matlab Version 7.0*.

**Определение состояния ключей, характеризующих поведение
растворенных газов во времени**

Положение ключа K_1

Входными переменными являются значения коэффициентов парной корреляции между газами углеводородного ряда и временем эксплуатации.

Этап фазификации. На данном этапе выделяются входные лингвистические переменные:

$$ЛП_{вх\ i} \quad i = 1 \dots 5,$$

где i – номер коэффициента парной корреляции данного газа и времени.

Таблица 3.2 – Определение характера дефекта трансформаторов по содержанию газов в масле

Характер прогнозируемого дефекта	Отношение пар газов		
	$\frac{C_2H_2}{C_2H_4}$	$\frac{CH_4}{H_2}$	$\frac{C_2H_4}{C_2H_6}$
1. Нормальное состояние	–	0,1–1	<0,2
2. Частичные разряды	–	<0,1	<0,2
3. Разряды малой мощности	>1	0,1–0,5	>1
4. Разряды большой мощности	>1	0,1–1	>2
5. Поверхностные разряды	<1	0,3–0,5	>5
6. Термический дефект в диапазоне низких температур (150–300 °С)	–	>1	<1
7. Термический дефект в диапазоне средних температур (300–700 °С)	–	>1	1–4
8. Термический дефект высокой температуры (более 700 °С)	<0,2	>1	>4

Входные функции принадлежности будут иметь вид:

$ЛП_{вх\ 1}$: <корреляция метана со временем>

$ЛП_{вх\ 2}$: <корреляция ацетилена со временем>

$ЛП_{вх\ 3}$: <корреляция этилена со временем>

$ЛП_{вх\ 4}$: <корреляция этана со временем>

ЛП_{вх 5}: <корреляция водорода со временем>

Далее ЛП_{вх i} делится на несколько лингвистических термов ЛТ_{ij}, характеризующих особенности состояния данной переменной:

$$\text{ЛТ}_{ij}, j = 1 \dots n,$$

где j – номер термина ЛП_{вх i}; n – количество термов, на которое разбито ЛП_{вх i}.

Разбиение на термы проводилось следующим образом:

ЛП_{вх 1} разделена на два ЛТ

ЛТ₁₁: <корреляция значима – Y >

ЛТ₁₂: <корреляция не значима – N >

ЛП_{вх 2} разделена на два ЛТ

ЛТ₂₁: <корреляция значима – Y >

ЛТ₂₂: <корреляция не значима – N >

ЛП_{вх 3} разделена на два ЛТ

ЛТ₃₁: <корреляция значима – Y >

ЛТ₃₂: <корреляция не значима – N >

ЛП_{вх 4} разделена на два ЛТ

ЛТ₄₁: <корреляция значима – Y >

ЛТ₄₂: <корреляция не значима – N >

ЛП_{вх 5} разделена на два ЛТ

ЛТ₅₁: <корреляция значима – Y >

ЛТ₅₂: <корреляция не значима – N >

Затем проводится прием входного четкого значения для системы с нечеткой логикой, а также поочередное считывание из базы знаний всех функций принадлежности, которые имеют отношение к данному входному значению, и устанавливается значение достоверности измеренного значения каждому ЛТ_{ij} по функциям принадлежности.

В качестве функций принадлежности для каждого из термов была выбрана трапецеидальная функция, математическое выражение которой имеет вид:

$$f_T(x, a, b, c, d) = \left. \begin{array}{ll} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{array} \right\}.$$

Поскольку неопределенность при оценке значимости коэффициента парной корреляции обусловлена использованием различных уровней доверительной вероятности, то параметры функции принадлежности определялись по критическим значениям коэффициентов парной корреляции при различных уровнях доверительной вероятности. На рис. 3.11 приведена функция принадлежности для объема выборочных значений $N = 7$ (критическое значение коэффициента парной корреляции при $p = 0,95$ составляет 0,582, при $p = 99,95 - 0,898$). Терм Y имеет координаты (0,57; 0,66; 1; 1), а терм $N(-1; -1; 0,5; 0,59)$.

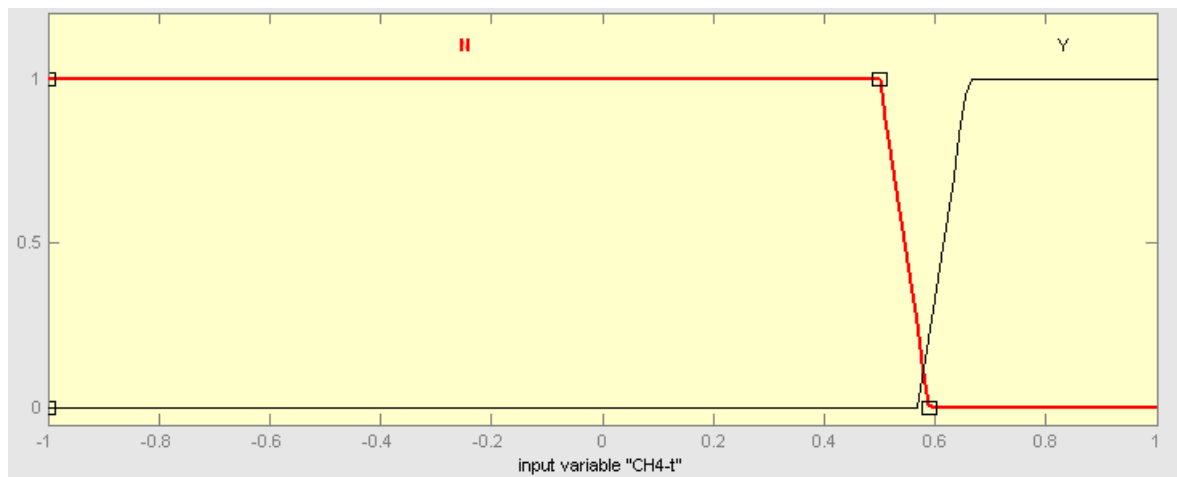


Рисунок 3.11 – Трапециевидальная функция принадлежности для объема выборочных значений $N = 7$

Выходные лингвистические переменные имеют вид:

ЛП_{вых 1}: <Наблюдается значимый рост газов углеводородного ряда во времени – *OPEN*>

ЛП_{вых 2}: <Значимый рост газов углеводородного ряда во времени отсутствует – *CLOSE*>

Разбиение на термы проводилось следующим образом:

ЛП_{ВЫХ 1} разделена на два ЛТ

ЛТ₁₁: <Рост есть – D>

ЛТ₁₂: <Рост отсутствует – N>

ЛП_{ВЫХ 2} разделена на два ЛТ

ЛТ₂₁: <Рост есть – D>

ЛТ₂₂: <Рост отсутствует – N>

В качестве функций принадлежности для каждого из термов была выбрана треугольная функция, математическое выражение которой имеет вид:

$$f_{\Delta}(x, a, b, c) = \left. \begin{array}{l} 0, \quad x < a \\ \frac{x-a}{b-a}, \quad a \leq x \leq b \\ \frac{c-x}{c-b}, \quad b \leq x \leq c \\ 0, \quad c \leq x \end{array} \right\}$$

Терм D имеет координаты (0; 1; 1), а терм N – (0; 0; 1). Графически функция принадлежности для выходных переменных приведена на рис. 3.12.

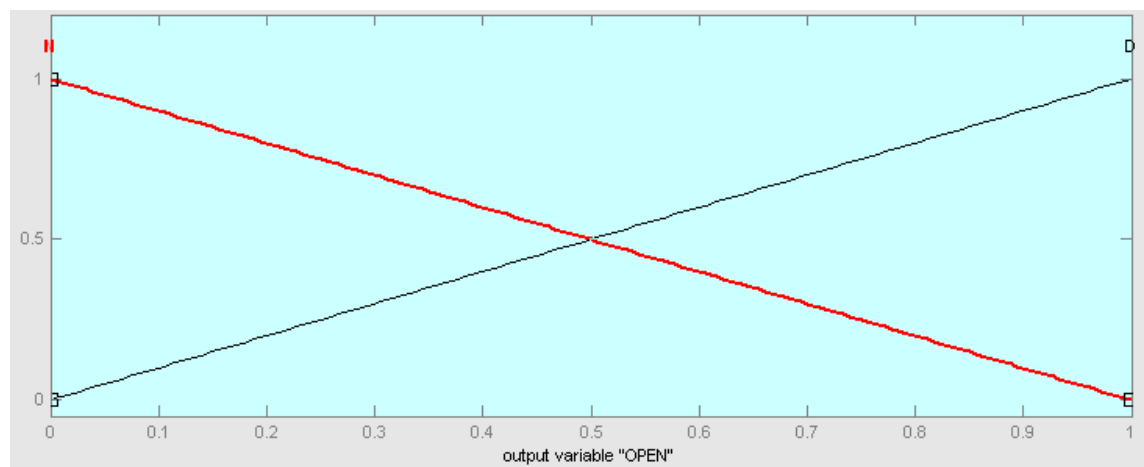


Рисунок 3.12 – Треугольная функция принадлежности выходной переменной

Результатом работы блока фаззификации является массив нечетких выходов, которые представляют собой набор пар ЛТ_{ij} и его значение достоверности

$$\text{ЛП}_{\text{ВХ } i} = \left\{ z_i \left| \mu_{\text{ЛТ}_{ij}}(z_i) \right. \right\}, \mu_{\text{ЛТ}_{ij}}(z_i) \in [0, 1],$$

где z_i – четкое входное значение коэффициента парной корреляции данного газа и времени.

Этап вычисления правила. Система с нечеткой логикой должна иметь базу правил, которая по сути представляет собой эмпирические знания эксперта о механизме управления. Для вычисления правил используются нечеткие входы, полученные от блока фаззификации, и правила, находящиеся в базе знаний. В левой части правил перебираются возможные ситуации на входе системы, а в правой части указывается, какая ЛПП описывает правильную реакцию системы.

База правил имеет следующий вид:

Правило 1: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 2: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 3: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 4: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 5: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 6: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 7: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 8: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 9: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 10: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 11: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 12: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 13: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 14: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 15: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$ И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 16: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 17: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 18: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 19: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 20: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 21: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 22: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 23: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 24: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 25: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 26: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 27: ЕСЛИ $[CH_4-t \in Y]$ И $[C_2H_4-t \in M]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in M]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$;

Правило 28: ЕСЛИ $[CH_4-t \in M]$ И $[C_2H_4-t \in Y]$, И $[C_2H_2-t \in M]$, И $[C_2H_6-t \in M]$, И $[H_2-t \in M]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$;

Правило 29: ЕСЛИ $[CH_4-t \in N]$ И $[C_2H_4-t \in N]$, И $[C_2H_2-t \in Y]$, И $[C_2H_6-t \in N]$, И $[H_2-t \in N]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$;

Правило 30: ЕСЛИ $[CH_4-t \in N]$ И $[C_2H_4-t \in N]$, И $[C_2H_2-t \in N]$, И $[C_2H_6-t \in Y]$, И $[H_2-t \in N]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$;

Правило 31: ЕСЛИ $[CH_4-t \in N]$ И $[C_2H_4-t \in N]$, И $[C_2H_2-t \in N]$, И $[C_2H_6-t \in N]$, И $[H_2-t \in Y]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$;

Правило 32: ЕСЛИ $[CH_4-t \in N]$ И $[C_2H_4-t \in N]$, И $[C_2H_2-t \in N]$, И $[C_2H_6-t \in N]$, И $[H_2-t \in N]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$.

Для предпосылок каждого из правил находим уровни отсечения с использованием операции минимум.

Этап дефаззификации. На данном этапе происходит преобразование нечеткой информации, содержащейся в виде достоверности лингвистического терма, в четко определенное значение. Дефаззификация производится по фигуре, полученной путем сложения всех функций принадлежности термов входной переменной. При выборе метода дефаззификации анализировались следующие методы.

1. *Метод центра тяжести.* Центр тяжести или центроид площади рассчитывается по формуле

$$y = \frac{\max \int x \cdot \mu(x) dx}{\max \int \mu(x) dx},$$

где y – результат дефаззификации; x – переменная, соответствующая выходной лингвистической переменной w ; $\mu(x)$ – функция принадлежности нечеткого множества, соответствующего выходной переменной w после этапа аккумуляции; \min и \max – левая и правая точки интервала носителя нечеткого множества рассматриваемой выходной переменной w .

2. *Метод центра площади.* Центр площади определяется как

$$\int_{\min}^u \mu(x) dx = \int_u^{\max} \mu(x) dx.$$

Другими словами, центр площади равен абсциссе, которая делит площадь, ограниченную графиком кривой функции принадлежности соответствующей выходной переменной, на две равные части.

3. *Метод левого модального значения.* Левое модальное значение рассчитывается по формуле:

$$y = \min\{x_m\},$$

где x_m – модальное значение (мода) нечеткого множества, соответствующего выходной переменной w после этапа аккумуляции.

В этом случае значение выходной переменной определяется как мода нечеткого множества для соответствующей выходной переменной или наименьшая из мод (самая левая), если нечеткое множество имеет несколько модальных значений.

4. *Метод правого модального значения.* Правое модальное значение рассчитывается по формуле:

$$y = \max\{x_m\}.$$

Другими словами, значение выходной переменной определяется как мода нечеткого множества для соответствующей выходной переменной или наибольшая из мод (самая правая), если нечеткое множество имеет несколько модальных значений.

Для выбора наиболее эффективного метода используем значения коэффициентов парной корреляции газов углеводородного ряда и времени эксплуатации для трансформатора «Постниково» Т-1 Донецкой области, которые составили: 0,771; 0,775; 0,747; 0,803 и 0. Поскольку данный трансформатор имеет дефект, то состояние ключа должно соответствовать $OPEN \in D$. Подставим данные значения коэффициентов парной корреляции на вход программы и, меняя методы дефаззификации, получим следующие значения выходной переменной OPEN:

- метод центра тяжести $OPEN \in D$ на 67 %;
- метод центра площади $OPEN \in D$ на 71 %;
- метод левого модального значения $OPEN \in D$ на 72 %;
- метод правого модального значения $OPEN \in D$ на 100 %.

Таким образом, в качестве метода дефаззификации был принят метод правого модального значения, который обеспечивает наибольшую достоверность при определении степени принадлежности выходных переменных.

На выходе данная система возвращает степень принадлежности ключа K_1 состояниям OPEN или CLOSE.

Положение ключа K_2

Входными переменными являются значения коэффициентов парной корреляции между газами углеводородного ряда в исследуемом трансформаторе.

Этап фаззификации. На данном этапе выделяются входные лингвистические переменные:

$$\text{ЛП}_{\text{вх } i} \quad i = 1 \dots 10,$$

где i – номер коэффициента парной корреляции данного газа и времени.

Входные функции принадлежности будут иметь вид:

$\text{ЛП}_{\text{вх } 1}$: <корреляция метана с ацетиленом>

$\text{ЛП}_{\text{вх } 2}$: <корреляция метана с этиленом>

$\text{ЛП}_{\text{вх } 3}$: <корреляция метана с этаном>

$\text{ЛП}_{\text{вх } 4}$: <корреляция метана с водородом>

$\text{ЛП}_{\text{вх } 5}$: <ацетилена с этиленом>

$\text{ЛП}_{\text{вх } 6}$: <ацетилена с этаном>

$\text{ЛП}_{\text{вх } 7}$: <ацетилена с водородом>

$\text{ЛП}_{\text{вх } 8}$: <этилена с этаном>

$\text{ЛП}_{\text{вх } 9}$: <этилена с водородом>

$\text{ЛП}_{\text{вх } 10}$: <этана с водородом>

Разбиение на термы проводилось следующим образом:

$\text{ЛП}_{\text{вх } 1}$ разделена на два ЛТ

ЛТ_{11} : <корреляция значима – Y >

ЛТ_{12} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 2}$ разделена на два ЛТ

ЛТ_{21} : <корреляция значима – Y >

ЛТ_{22} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 3}$ разделена на два ЛТ

ЛТ_{31} : <корреляция значима – Y >

ЛТ_{32} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 4}$ разделена на два ЛТ

ЛТ_{41} : <корреляция значима – Y >

ЛТ_{42} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 5}$ разделена на два ЛТ

ЛТ_{51} : <корреляция значима – Y >

ЛТ_{52} : <корреляция не значима – N >

ЛП_{вх 6} разделена на два ЛТ

ЛТ₆₁: <корреляция значима – Y >

ЛТ₆₂: <корреляция не значима – N >

ЛП_{вх 7} разделена на два ЛТ

ЛТ₇₁: <корреляция значима – Y >

ЛТ₇₂: <корреляция не значима – N >

ЛП_{вх 8} разделена на два ЛТ

ЛТ₈₁: <корреляция значима – Y >

ЛТ₈₂: <корреляция не значима – N >

ЛП_{вх 9} разделена на два ЛТ

ЛТ₉₁: <корреляция значима – Y >

ЛТ₉₂: <корреляция не значима – N >

ЛП_{вх 10} разделена на два ЛТ

ЛТ₁₀₁: <корреляция значима – Y >

ЛТ₁₀₂: <корреляция не значима – N >

В качестве функций принадлежности были выбраны функции приведенные на рис. 3.9 и 3.10.

Этап вычисления правила. База правил имеет следующий вид:

Правило 1: ЕСЛИ [$\text{CH}_4 - \text{C}_2\text{H}_2 \in Y$] И (ИЛИ) [$\text{CH}_4 - \text{C}_2\text{H}_4 \in Y$], И (ИЛИ) [$\text{CH}_4 - \text{C}_2\text{H}_6 \in Y$], И (ИЛИ) [$\text{CH}_4 - \text{H}_2 \in Y$], И (ИЛИ) [$\text{C}_2\text{H}_2 - \text{C}_2\text{H}_4 \in Y$], И (ИЛИ) [$\text{C}_2\text{H}_2 - \text{C}_2\text{H}_6 \in Y$], И (ИЛИ) [$\text{C}_2\text{H}_2 - \text{H}_2 \in Y$], И (ИЛИ) [$\text{C}_2\text{H}_4 - \text{C}_2\text{H}_6 \in Y$], И (ИЛИ) [$\text{C}_2\text{H}_4 - \text{H}_2 \in Y$], И (ИЛИ) [$\text{C}_2\text{H}_6 - \text{H}_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in N$];

Правило 2: ЕСЛИ [$\text{CH}_4 - \text{C}_2\text{H}_2 \in M$] И [$\text{CH}_4 - \text{C}_2\text{H}_4 \in M$], И [$\text{CH}_4 - \text{C}_2\text{H}_6 \in M$], И [$\text{CH}_4 - \text{H}_2 \in M$], И [$\text{C}_2\text{H}_2 - \text{C}_2\text{H}_4 \in M$], И [$\text{C}_2\text{H}_2 - \text{C}_2\text{H}_6 \in M$], И [$\text{C}_2\text{H}_2 - \text{H}_2 \in M$], И [$\text{C}_2\text{H}_4 - \text{C}_2\text{H}_6 \in M$], И [$\text{C}_2\text{H}_4 - \text{H}_2 \in M$], И [$\text{C}_2\text{H}_6 - \text{H}_2 \in M$], ТО [OPEN $\in N$ CLOSE $\in D$].

Для предпосылок каждого из правил находим уровни отсечения с использованием операции минимум.

Этап дефаззификации. Как и в предыдущем случае, в качестве метода дефаззификации был принят метод правого модального значения, который обеспечивает наибольшую достоверность при определении степени принадлежности выходных переменных.

На выходе данная система возвращает степень принадлежности ключа K_2 состояниям OPEN или CLOSE.

Положение ключа K_3

Входными переменными являются значения коэффициентов парной корреляции между газами углеводородного ряда, растворенными в масле соседних трансформаторов.

Этап фаззификации. На данном этапе выделяются входные лингвистические переменные:

$$\text{ЛП}_{\text{вх } i} \quad i = 1 \dots 5,$$

где i – номер коэффициента парной корреляции данного газа и времени.

Входные функции принадлежности будут иметь вид:

$\text{ЛП}_{\text{вх } 1}$: <корреляция метана с метаном>

$\text{ЛП}_{\text{вх } 2}$: <корреляция ацетилена с ацетиленом>

$\text{ЛП}_{\text{вх } 3}$: <корреляция этилена с этиленом>

$\text{ЛП}_{\text{вх } 4}$: <корреляция этана с этаном>

$\text{ЛП}_{\text{вх } 5}$: <корреляция водорода с водородом>

Разбиение на термы проводилось следующим образом:

$\text{ЛП}_{\text{вх } 1}$ разделена на два ЛТ

ЛТ_{11} : <корреляция значима – Y >

ЛТ_{12} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 2}$ разделена на два ЛТ

ЛТ_{21} : <корреляция значима – Y >

ЛТ_{22} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 3}$ разделена на два ЛТ

ЛТ_{31} : <корреляция значима – Y >

ЛТ_{32} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 4}$ разделена на два ЛТ

ЛТ_{41} : <корреляция значима – Y >

ЛТ_{42} : <корреляция не значима – N >

$\text{ЛП}_{\text{вх } 5}$ разделена на два ЛТ

ЛТ_{51} : <корреляция значима – Y >

ЛТ_{52} : <корреляция не значима – N >

В качестве функций принадлежности были выбраны функции, приведенные на рис. 3.9 и 3.10.

Этап вычисления правила. База правил имеет следующий вид:

Правило 1: ЕСЛИ [$\text{CH}_4 - \text{CH}_4 \in Y$] И [$\text{C}_2\text{H}_4 - \text{C}_2\text{H}_4 \in Y$], И [$\text{C}_2\text{H}_2 -$

$C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 2: ЕСЛИ [$CH_4 - CH_4 \in M$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 3: ЕСЛИ [$CH_4 - CH_4 \in Y$] И [$C_2H_4 - C_2H_4 \in M$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 4: ЕСЛИ [$CH_4 - CH_4 \in Y$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in M$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 5: ЕСЛИ [$CH_4 - CH_4 \in Y$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in M$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 6: ЕСЛИ [$CH_4 - CH_4 \in Y$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in M$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 7: ЕСЛИ [$CH_4 - CH_4 \in M$] И [$C_2H_4 - C_2H_4 \in M$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 8: ЕСЛИ [$CH_4 - CH_4 \in M$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in M$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 9: ЕСЛИ [$CH_4 - CH_4 \in M$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in M$] И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 10: ЕСЛИ [$CH_4 - CH_4 \in M$] И [$C_2H_4 - C_2H_4 \in Y$], И [$C_2H_2 - C_2H_2 \in Y$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in M$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 11: ЕСЛИ [$CH_4 - CH_4 \in Y$] И [$C_2H_4 - C_2H_4 \in M$], И [$C_2H_2 - C_2H_2 \in M$], И [$C_2H_6 - C_2H_6 \in Y$], И [$H_2 - H_2 \in Y$], ТО [OPEN $\in D$ CLOSE $\in M$];

Правило 12: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 13: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 14: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 15: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 16: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 17: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 18: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 19: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 20: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 21: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 22: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 23: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 24: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 25: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 26: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 27: ЕСЛИ $[CH_4 - CH_4 \in Y]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 28: ЕСЛИ $[CH_4 - CH_4 \in M]$ И $[C_2H_4 - C_2H_4 \in Y]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 29: ЕСЛИ $[CH_4 - CH_4 \in N]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in Y]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 30: ЕСЛИ $[CH_4 - CH_4 \in N]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in Y]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 31: ЕСЛИ $[CH_4 - CH_4 \in N]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in Y]$, ТО $[OPEN \in D \text{ CLOSE} \in M]$;

Правило 32: ЕСЛИ $[CH_4 - CH_4 \in N]$ И $[C_2H_4 - C_2H_4 \in M]$, И $[C_2H_2 - C_2H_2 \in M]$, И $[C_2H_6 - C_2H_6 \in M]$, И $[H_2 - H_2 \in M]$, ТО $[OPEN \in N \text{ CLOSE} \in D]$.

Для предпосылок каждого из правил находим уровни отсечения с использованием операции минимум.

Этап дефаззификации. Как и в предыдущих случаях, в качестве метода дефаззификации был принят метод правого модального значения, ко-

торый обеспечивает наибольшую достоверность при определении степени принадлежности выходных переменных.

На выходе данная система возвращает степень принадлежности ключа K_3 состояниям OPEN или CLOSE.

Определение состояния трансформатора

Входными переменными являются значения переменных K_1, K_2, K_3 , которые характеризуют поведение газов на диагностируемом промежутке. Численные значения этих переменных берутся с выходов предыдущих блоков.

Этап фаззификации. Выделяем входные лингвистические переменные:

$$\text{ЛП}_{\text{вх } i} \quad i = 1 \dots 3,$$

где i – номер ключа, характеризующего поведение кривых газов во времени.

Входные функции принадлежности будут иметь вид:

k_1 : <значимый рост концентрации газов во времени>

k_2 : <одновременный рост концентрации нескольких газов>

k_3 : <одновременный рост концентрации нескольких газов в соседних трансформаторах >

Далее $\text{ЛП}_{\text{вх } i}$ делим на несколько лингвистических термов ЛТ_{ij} , характеризующих особенности состояния данной переменной. Разбиение на термы проводилось следующим образом:

k_1 разделена на два ЛТ

k_{11} : <OPEN >

k_{12} : <CLOSE>

k_2 разделена на два ЛТ

k_{21} : <OPEN >

k_{22} : <CLOSE>

k_3 разделена на два ЛТ

k_{31} : <OPEN >

k_{32} : <CLOSE>

В качестве функций принадлежности для каждого из термов была выбрана трапецеидальная функция принадлежности с координатами для терма <OPEN > (0,45; 0,5; 1; 1), для терма <CLOSE> (0; 0; 0,48; 0,52). Дан-

ная функция принадлежности для входной переменной k_1 приведена на рис. 3.13.

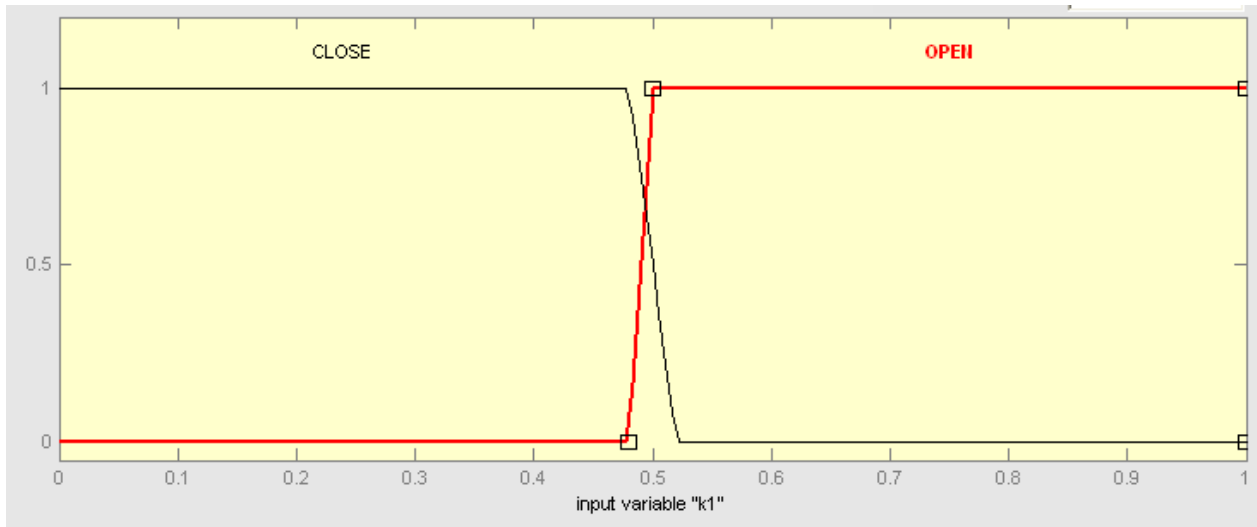


Рисунок 3.13 – Функция принадлежности для входной переменной k_1

В качестве выходных лингвистических переменных принимаем:

$$\text{ЛП}_{\text{вых } i} \quad i = 1 \dots 8,$$

где i – номер возможного состояния.

Используем следующий перечень состояний:

D_1 :<Вероятней всего дефект отсутствует>

D_2 :<Вероятней всего дефект отсутствует, картины газов идентичны для всех трансформаторов подстанции>

D_3 :<Рост концентрации газов без развития во времени, внешние воздействия>

D_4 :<Рост концентрации газов без развития во времени, картины газов идентичны для всех трансформаторов подстанции, внешние воздействия>

D_5 :<Рост концентрации отдельных газов, необходимы дополнительные анализы >

D_6 :<Рост концентрации отдельных газов, картины газов идентичны для всех трансформаторов подстанции, внешние воздействия >

D_7 :<Наличие дефекта>

D_8 :<Вероятней всего интенсивные внешние воздействия, необходимы дополнительные анализы>

В качестве функции принадлежности выходных переменных используем треугольную функцию, координатами которой являются для термина $\langle D \rangle$ (0; 1; 1), для термина $\langle N \rangle$ (0; 0; 1). Данная функция принадлежности для выходной переменной D_1 приведена на рис. 3.10.

Этап вычисления правила. База правил имеет следующий вид:

Правило 1: ЕСЛИ $[k_1 \in \text{CLOSE}]$ И $[k_2 \in \text{CLOSE}]$, И $[k_3 \in \text{CLOSE}]$, ТО $[D_1]$;

Правило 2: ЕСЛИ $[k_1 \in \text{CLOSE}]$ И $[k_2 \in \text{CLOSE}]$, И $[k_3 \in \text{OPEN}]$, ТО $[D_2]$;

Правило 3: ЕСЛИ $[k_1 \in \text{CLOSE}]$ И $[k_2 \in \text{OPEN}]$, И $[k_3 \in \text{CLOSE}]$, ТО $[D_3]$;

Правило 4: ЕСЛИ $[k_1 \in \text{CLOSE}]$ И $[k_2 \in \text{OPEN}]$, И $[k_3 \in \text{OPEN}]$, ТО $[D_4]$;

Правило 5: ЕСЛИ $[k_1 \in \text{OPEN}]$ И $[k_2 \in \text{CLOSE}]$, И $[k_3 \in \text{CLOSE}]$, ТО $[D_5]$;

Правило 6: ЕСЛИ $[k_1 \in \text{OPEN}]$ И $[k_2 \in \text{CLOSE}]$, И $[k_3 \in \text{OPEN}]$, ТО $[D_6]$;

Правило 7: ЕСЛИ $[k_1 \in \text{OPEN}]$ И $[k_2 \in \text{OPEN}]$, И $[k_3 \in \text{CLOSE}]$, ТО $[D_7]$;

Правило 8: ЕСЛИ $[k_1 \in \text{OPEN}]$ И $[k_2 \in \text{OPEN}]$, И $[k_3 \in \text{OPEN}]$, ТО $[D_8]$.

Объединение предпосылок внутри каждого правила выполняется с использованием операции пересечения (\min).

Этап дефаззификации. Выполненные исследования на верифицированных образцах показали, что максимальное значение достоверности обеспечивает метод правого модального значения.

На выходе данная система возвращает степень принадлежности диагностируемого трансформатора одному из возможных состояний (диагнозов).

Определение вида дефекта трансформатора

Входными переменными являются значения отношений следующих пар газов: C_2H_2/C_2H_4 , CH_4/H_2 , C_2H_4/C_2H_6 .

Этап фаззификации. Выделяем входные лингвистические переменные:

$$ЛП_{\text{вх } i} \quad i = 1 \dots 3,$$

где i – номер отношения пар газов.

Входные функции принадлежности будут иметь вид:

ЛП_{вх 1}: $\langle C_2H_2/C_2H_4 \rangle$

ЛП_{вх 2}: $\langle CH_4/H_2 \rangle$

ЛП_{вх 3}: $\langle C_2H_4/C_2H_6 \rangle$

Далее ЛП_{вх*i*} делим на несколько лингвистических термов ЛТ_{*ij*}, характеризующих особенности состояния данной переменной.

Разбиение на термы проводилось следующим образом:

ЛП_{вх 1} разделена на три ЛТ

ЛТ₁₁: $\langle \text{маленькое значение} - M \rangle$

ЛТ₁₂: $\langle \text{среднее значение} - S \rangle$

ЛТ₁₃: $\langle \text{большое значение} - H \rangle$

ЛП_{вх 2} разделена на четыре ЛТ

ЛТ₂₁: $\langle \text{маленькое значение} - M \rangle$

ЛТ₂₂: $\langle \text{небольшое значение} - L \rangle$

ЛТ₂₃: $\langle \text{среднее значение} - S \rangle$

ЛТ₂₄: $\langle \text{большое значение} - H \rangle$

ЛП_{вх 3} разделена на пять ЛТ

ЛТ₃₁: $\langle \text{маленькое значение} - M \rangle$

ЛТ₃₂: $\langle \text{небольшое значение} - L \rangle$

ЛТ₃₃: $\langle \text{среднее значение} - S \rangle$

ЛТ₃₄: $\langle \text{повышенное значение} - R \rangle$

ЛТ₃₅: $\langle \text{большое значение} - H \rangle$

Для выбора функций принадлежности были использованы результаты исследований законов распределений отношений пар газов, для бездефектного и дефектного состояний, которые опубликованы в работе [7]. В качестве примера на рис. 3.14 приведено взаимное расположение распределений отношения метана к водороду для бездефектного состояния (D_1) и термического дефекта с температурой $t > 700$ °С (D_8). Пунктирными вертикальными линиями обозначены границы областей с различными состояниями. Как видно из рисунка, функции плотностей распределений для дефектного и бездефектного состояний пересекаются, при этом значения отношений пар газов могут принимать значения, характерные для разных диагнозов. Данное обстоятельство подтверждает необходимость использования аппарата нечеткой логики для распознавания типа дефекта.

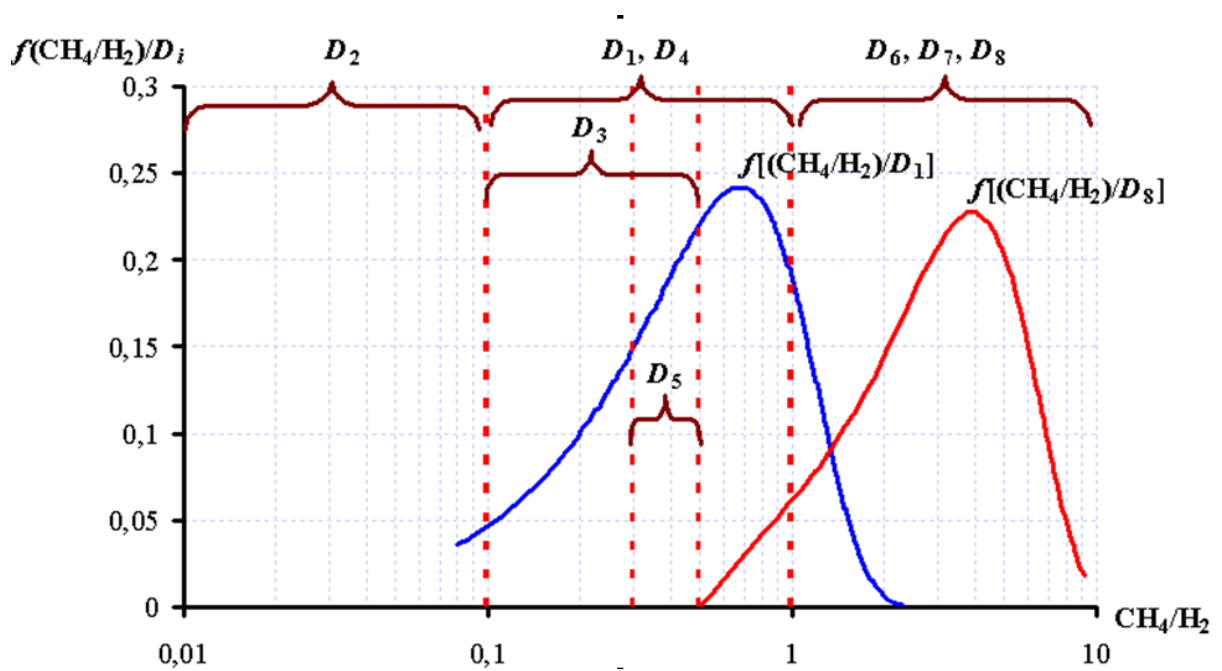


Рисунок 3.14 – Взаимное расположение распределений отношения метана к водороду для бездефектного состояния (D_1) и термического дефекта с температурой $t > 700\text{ }^\circ\text{C}$ (D_8)

С учетом результатов, полученных в [7], в качестве функций принадлежности для каждого из термов была выбрана трапецеидальная функция, а также, что является немаловажным, с точки зрения обеспечения максимальной достоверности поставленного диагноза, определены границы областей пересечения. Данные функции приведены на рис. 3.15–3.17.

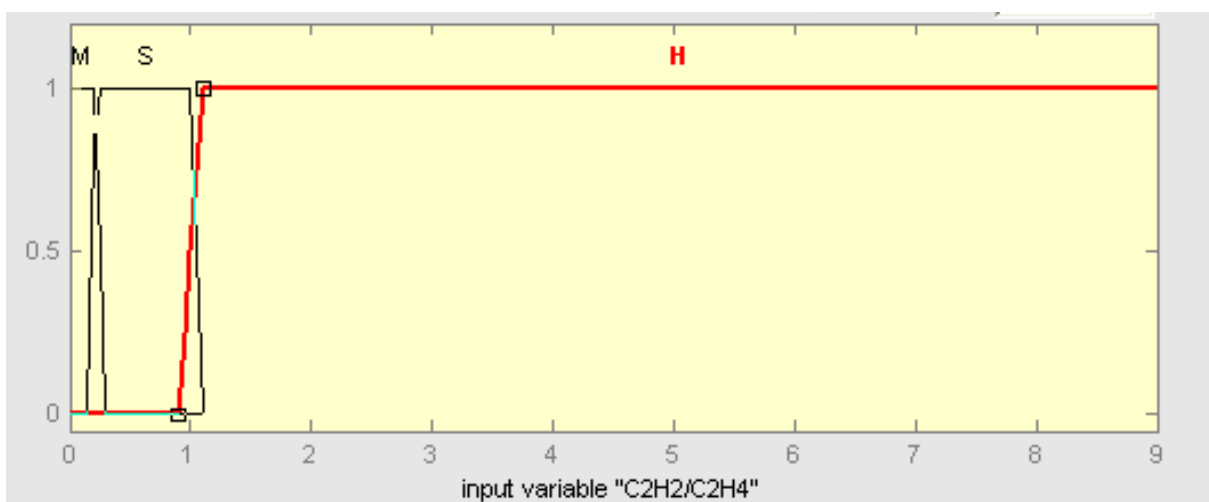


Рисунок 3.15 – Функция принадлежности отношения ацетилена к этилену

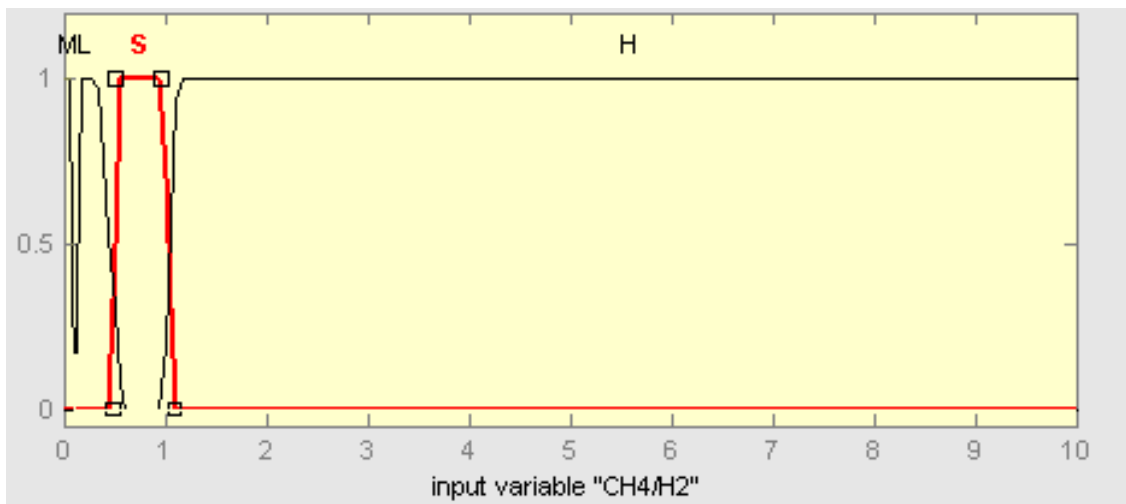


Рисунок 3.16 – Функция принадлежности отношения метана к водороду

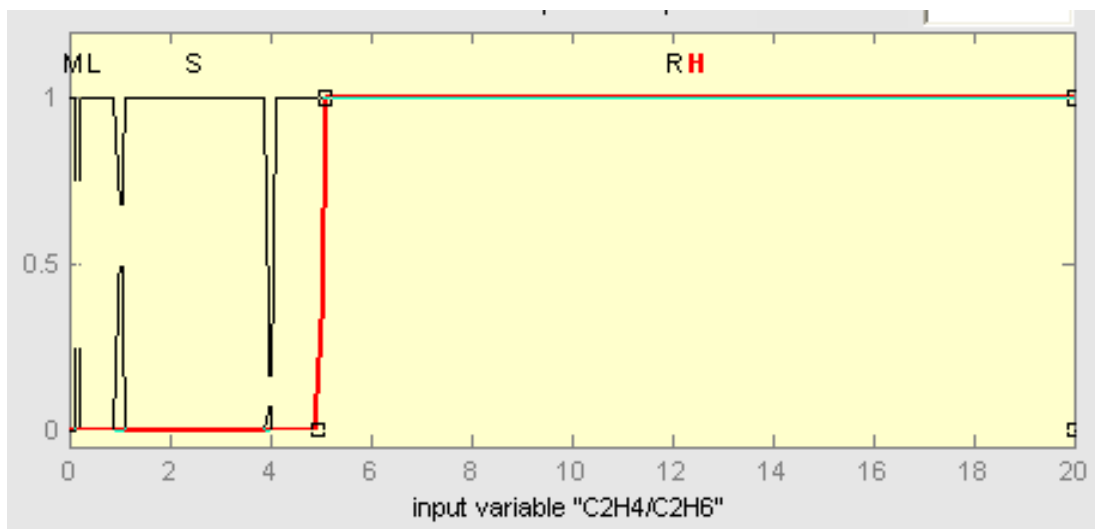


Рисунок 3.17 – Функция принадлежности отношения этилена к этану

Для обучения сети использовалась следующая кодировка (табл. 3.3).

Таблица 3.3 – Кодировка диагнозов трансформаторов

Код	Диагноз
1	Нормальное состояние
2	Частичные разряды
3	Разряды низкой энергии
4	Разряды высокой энергии
5	Разряды по поверхности и толщине изоляции
6	Термический дефект t от 300 до 700 °С
7	Термический дефект t от 300 до 700 °С
8	Термический дефект $t > 700$ °С

В качестве выходной лингвистической переменной принимаем:

$$\mathbf{Diagnoz}_i \quad i = 1 \dots 8,$$

где i – номер возможного состояния в соответствии с табл. 3.3. В качестве функции принадлежности выходных переменных используем треугольную функцию, координатами которой являются для термина $\langle D \rangle$ (0; 1; 1), для термина $\langle N \rangle$ (0; 0; 1). Данная функция принадлежности для выходной переменной приведена на рис. 3.10.

Этап вычисления правила. База правил имеет следующий вид:

Правило 1: ЕСЛИ $[C_2H_2/C_2H_4 \in M]$ И $[CH_4/H_2 \in S]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 1]$;

Правило 2: ЕСЛИ $[C_2H_2/C_2H_4 \in S]$ И $[CH_4/H_2 \in S]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 1]$;

Правило 3: ЕСЛИ $[C_2H_2/C_2H_4 \in H]$ И $[CH_4/H_2 \in S]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 1]$;

Правило 4: ЕСЛИ $[C_2H_2/C_2H_4 \in M]$ И $[CH_4/H_2 \in M]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 2]$;

Правило 5: ЕСЛИ $[C_2H_2/C_2H_4 \in S]$ И $[CH_4/H_2 \in M]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 2]$;

Правило 6: ЕСЛИ $[C_2H_2/C_2H_4 \in H]$ И $[CH_4/H_2 \in M]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 2]$;

Правило 7: ЕСЛИ $[C_2H_2/C_2H_4 \in H]$ И $[CH_4/H_2 \in L]$, И $[C_2H_4/C_2H_6 \in L]$, ТО $[Diagnoz \in 3]$;

Правило 8: ЕСЛИ $[C_2H_2/C_2H_4 \in H]$ И $[CH_4/H_2 \in S]$, И $[C_2H_4/C_2H_6 \in S]$, ТО $[Diagnoz \in 4]$;

Правило 9: ЕСЛИ $[C_2H_2/C_2H_4 \in M]$ И $[CH_4/H_2 \in L]$, И $[C_2H_4/C_2H_6 \in R]$, ТО $[Diagnoz \in 5]$;

Правило 10: ЕСЛИ $[C_2H_2/C_2H_4 \in M]$ И $[CH_4/H_2 \in H]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 6]$;

Правило 11: ЕСЛИ $[C_2H_2/C_2H_4 \in S]$ И $[CH_4/H_2 \in H]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 6]$;

Правило 12: ЕСЛИ $[C_2H_2/C_2H_4 \in H]$ И $[CH_4/H_2 \in H]$, И $[C_2H_4/C_2H_6 \in M]$, ТО $[Diagnoz \in 6]$;

Правило 13: ЕСЛИ $[C_2H_2/C_2H_4 \in M]$ И $[CH_4/H_2 \in H]$, И $[C_2H_4/C_2H_6 \in S]$, ТО $[Diagnoz \in 7]$;

Правило 14: ЕСЛИ [$C_2H_2/C_2H_4 \in S$] И [$CH_4/H_2 \in H$], И [$C_2H_4/C_2H_6 \in S$], ТО [**Diagnoz** $\in 7$];

Правило 15: ЕСЛИ [$C_2H_2/C_2H_4 \in H$] И [$CH_4/H_2 \in H$], И [$C_2H_4/C_2H_6 \in S$], ТО [**Diagnoz** $\in 7$];

Правило 16: ЕСЛИ [$C_2H_2/C_2H_4 \in M$] И [$CH_4/H_2 \in H$], И [$C_2H_4/C_2H_6 \in H$], ТО [**Diagnoz** $\in 8$].

Объединение предпосылок внутри каждого правила выполняется с использованием операции пересечения (**min**).

Этап дефазификации. Выполненные исследования на верифицированных образцах показали, что максимальное значение достоверности обеспечивает метод правого модального значения. На выходе данная система возвращает степень принадлежности диагностируемого трансформатора одному из возможных дефектов. Проиллюстрируем работу разработанной системы на примере распознавания состояния трансформатора ПС «Ленинская» Т-2 Луганскоблэнерго. По результатам ХАРГ в данном трансформаторе обнаружен термический дефект с температурой выше 700 °С. Данный трансформатор не использовался в процессе разработки экспертной системы и может рассматриваться как независимый объект. Значения коэффициентов парной корреляции приведены в табл. 3.4 и 3.5.

Таблица 3.4 – Значения коэффициентов парной корреляции между растворенными в масле газами для трансформаторов Т-1 и Т-2 ПС «Ленинская» Луганскоблэнерго

Трансформатор	Газ	Время	CO	CO ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆	C ₂ H ₂
Луганскоблэнерго ПС «Ленинская» Т - 1, N _i = 9, ρ _{крит.} , 7, 0,95 = 0,666	CO	-0,181	1,00	0,0683	0,660	-0,716	0,738	0,610	0,678
	CO ₂	-0,561	0,0683	1,00	0,184	0,395	-0,298	0,166	0,0256
	CH ₄	-0,365	0,660	0,184	1,00	-0,331	0,782	0,941	0,964
	C ₂ H ₂	0,09	-0,716	0,395	-0,331	1,00	-0,594	-0,391	-0,417
	C ₂ H ₄	0,101	0,738	-0,298	0,782	-0,594	1,00	0,729	0,952
	C ₂ H ₆	-0,334	0,610	0,166	0,941	-0,391	0,729	1,00	0,887
	C ₂ H ₂	-0,180	0,678	0,0256	0,964	-0,417	0,887	0,952	1,00
Луганскоблэнерго ПС «Ленинская» Т - 2, N _i = 9, ρ _{крит.} , 7, 0,95 = 0,666	CO	-0,033	1,00	-0,0906	0,639	0,207	0,612	0,532	0,650
	CO ₂	-0,422	-0,0906	1,00	-0,275	-0,096	-0,610	-0,492	-0,471
	CH ₄	0,197	0,639	-0,275	1,00	-0,144	0,828	0,853	0,888
	C ₂ H ₂	0,727	0,207	-0,096	-0,144	1,00	0,218	0,291	0,272
	C ₂ H ₄	0,551	0,612	-0,610	0,828	0,218	1,00	0,931	0,966
	C ₂ H ₆	0,699	0,532	-0,492	0,853	0,291	0,931	1,00	0,980
	C ₂ H ₂	0,767	0,650	-0,471	0,888	0,272	0,966	0,980	1,00

Значения отношения пар газов составили $C_2H_2/C_2H_4 = 0,11$; $C_2H_4/C_2H_6 = 6,85$; $CH_4/H_2 = 2,2$.

Таблица 3.5 – Значения коэффициентов корреляции между газами трансформаторов Т - 1 и Т - 2 ПС «Ленинская» Луганскоблэнерго

CO	CO ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆	C _Σ H _Σ
CO	CO ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆	C _Σ H _Σ
0,352	-0,175	-0,0876	0,375	-0,148	-0,327	-0,058

Результаты работы разработанной программы нечеткого вывода приведены на рис. 3.18–3.22. Как видно из рисунков, на всех этапах диагностики наблюдается полное совпадение между диагнозами, которые предлагает разработанная программа нечеткого вывода, и реальным состоянием трансформатора. Выполненное тестирование разработанной системы на трансформаторах с априорно известным состоянием показало ее высокую эффективность. Принципиально важным является тот факт, что разработанная программа нечеткого вывода позволяет выявлять и распознавать развивающиеся дефекты в маслонаполненном оборудовании на ранней стадии их развития, когда не произошло разрушение изоляции.

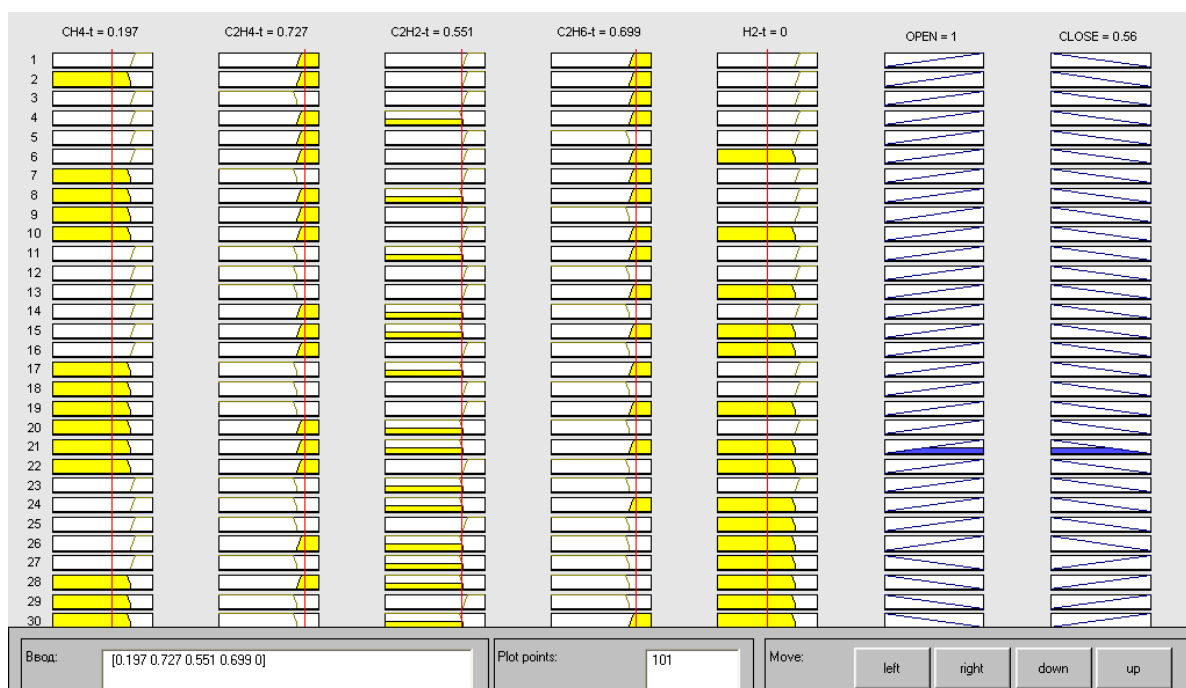


Рисунок 3.18 – Вид программы просмотра результатов нечеткого вывода при определении положения ключа K_1 для трансформатора ПС «Ленинская» Т - 2 Луганскоблэнерго



Рисунок 3.19 – Вид программы просмотра результатов нечеткого вывода при определении положения ключа K_2 для трансформатора ПС «Ленинская» Т - 2 Луганскоблэнерго

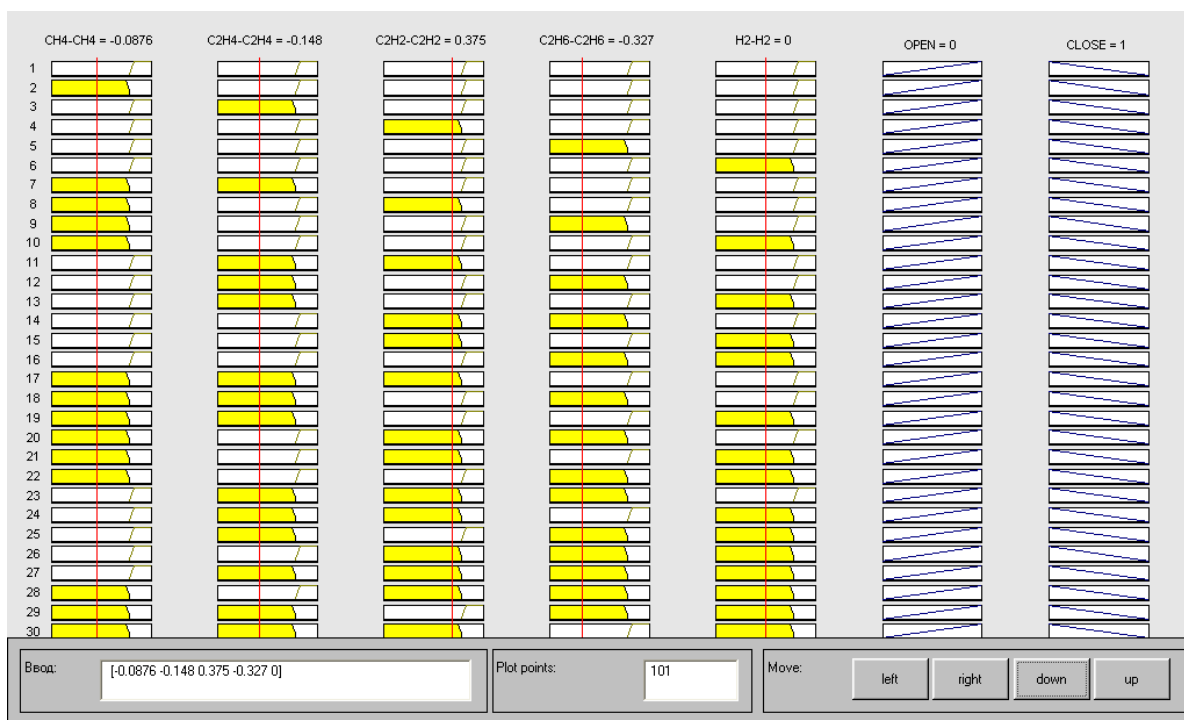


Рисунок 3.20 – Вид программы просмотра результатов нечеткого вывода при определении положения ключа K_3 для трансформатора ПС «Ленинская» Т - 2 Луганскоблэнерго



Рисунок 3.21 – Вид программы просмотра результатов нечеткого вывода при определении состояния трансформатора ПС «Ленинская» Т - 2 Луганскоблэнерго

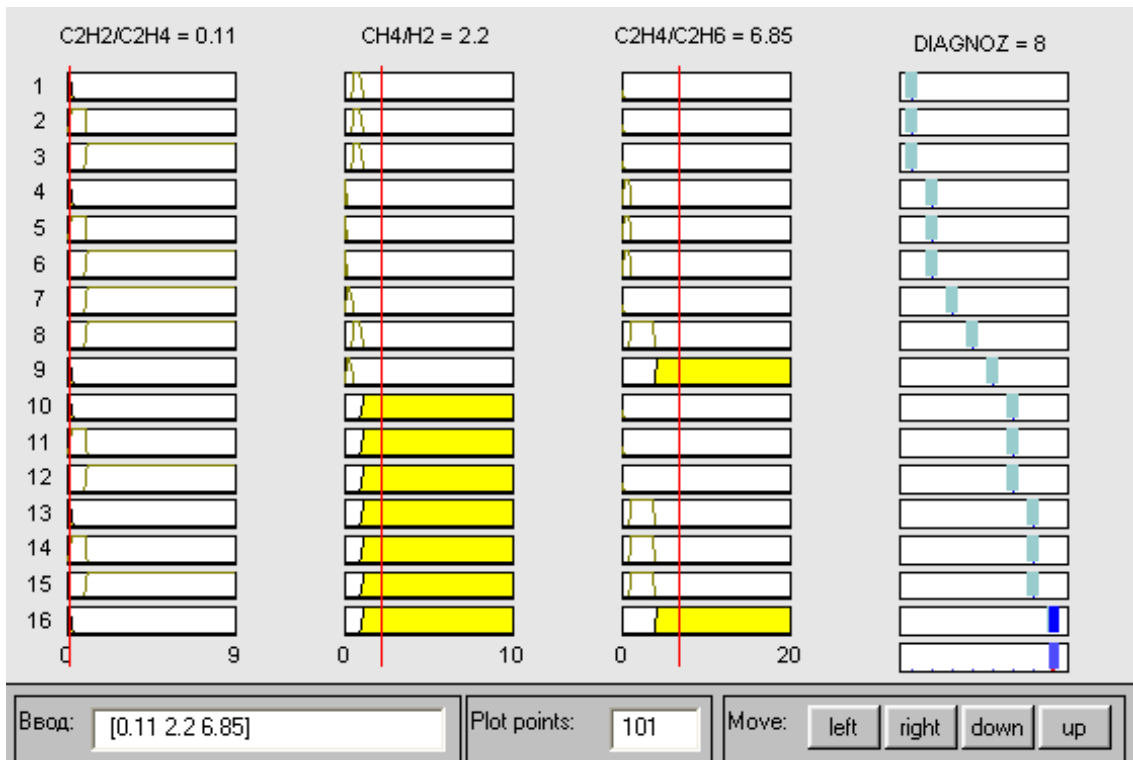


Рисунок 3.22 – Вид программы просмотра результатов нечеткого вывода при определении типа дефекта трансформатора ПС «Ленинская» Т - 2 Луганскоблэнерго

Вопросы для самоконтроля

1. Дать определение нечеткого множества.
2. Привести пример нечеткого множества.
3. Основные характеристики нечетких множеств.
4. Основные операции при работе с нечеткими множествами.
5. Нечеткие отношения и основные операции работы с ними.
6. Нечеткая импликация.
7. Построение функций принадлежности нечетких множеств.
8. Приведите наиболее часто встречающиеся функции принадлежности и дайте их описание.
9. Нечеткие и лингвистические переменные.
10. Алгоритм формирования нечеткого вывода.
11. Этап фаззификации.
12. Этап решающих правил.
13. Этап дефаззификации.
14. Алгоритм Mamdani.
15. Алгоритм Sugeno.
16. Алгоритм Tsukamoto.
17. Алгоритм Larsen.
18. Метод интерпретации результатов ХАРГ, разработанный на кафедре «Передача электрической энергии» НТУ «ХПИ».
19. Особенности реализации метода интерпретации результатов ХАРГ в среде нечеткой логики.

Литература к разделу 3

Основная

1. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации / С.А. Орловский. – М. : Наука, 1981. – 208 с.
2. Нечеткие множества в моделях управления и искусственного интеллекта / под ред. Д.А. Поспелова. – М. : Наука, 1986. – 312 с.
3. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде. – М. : Мир, 1976. – 165 с.
4. Леоненков А.В. Нечеткое моделирование в среде MATLAB Fuzzy TECH / А.В. Леоненков. – СПб. : БХВ – Петербург, 2003. – 736 с.

5. Діагностика маслонаповненого трансформаторного обладнання за результатами хроматографічного аналізу вільних газів, відібраних із газового реле, і газів, розчинених у ізоляційному маслі: СОУ – Н ЕЕ 46.501:2006. – Київ. – 2007. – 92 с.

6. Шутенко О.В. Формирование процедуры нечеткого вывода для обнаружения развивающихся дефектов высоковольтных трансформаторов / О.В. Шутенко // Вісник НТУ «ХПІ». – 2008. – № 44. – С. – 162–177.

7. Шутенко О.В. Анализ законов распределения отношений пар газов, рекомендуемых СОУ - Н ЕЕ 46.501 для распознавания типа дефекта в высоковольтных трансформаторах // О.В. Шутенко, Д.Н. Баклай // Энергетика та електрифікація. – Київ. – 2013. – № 8. – С. 36–47.

Дополнительная

8. Кофман А. Введение в теорию нечетких множеств / А. Кофман. – М. : Радио и связь, 1982. – 432 с.

9. Борисов А.Н. Обработка нечеткой информации в системах принятия решений / А.Н. Борисов. – М. : Радио и связь, 1989. – 304 с.

10. Прикладные нечеткие системы / Под ред. Т. Тэрано, К. Асаи, Сугэно. – М. : Мир, 1993. – 368 с.

11. Нечеткие множества и теория возможностей. Последние достижения / под ред. Р. Ягера. – М. : Радио и связь, 1986. – 409 с.

12. Борисов А.Н. Принятие решений на основе нечетких моделей. Примеры использования / А.Н. Борисов, О.А. Крумберг, И.П. Федоров. – Рига : Зинатне, 1990. – 184 с.

13. Малышев Н.Г. Нечеткие модели для экспертных систем в САПР / Н.Г. Малышев, Л.С. Берштейн, А.В.Боженюк. – М. : Энергоатомиздат, 1991. – 136 с.

14. Мелихов А.Н. Ситуационные советующие системы с нечеткой логикой / А.Н. Мелихов, Л.С. Бернштейн, С.Я. Коровин. – М. : Наука, 1990. – 736 с.

15. Беллман Р. Вопросы принятия решений в расплывчатых условиях / Р. Беллман, Л. Заде // Вопросы анализа и процедуры принятия решений. – М. : Мир, 1976. – 46 с.

РАЗДЕЛ 4

РАСПОЗНАВАНИЕ СОСТОЯНИЯ ЭЛЕКТРОЭНЕРГЕТИЧЕСКОГО ОБОРУДОВАНИЯ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

4.1. Биологический нейрон и его кибернетическая модель

Развитие искусственных нейронных сетей вдохновляется биологией. То есть, рассматривая сетевые конфигурации и алгоритмы, исследователи мыслят их в терминах организации мозговой деятельности [01]. Но на этом аналогия может и закончиться. Наши знания о работе мозга столь ограничены, что мало бы нашлось руководящих ориентиров для тех, кто стал бы ему подражать. Поэтому разработчикам сетей приходится выходить за пределы современных биологических знаний в поисках структур, способных выполнять полезные функции. Во многих случаях это приводит к необходимости отказа от биологического правдоподобия, мозг становится просто метафорой, и создаются сети, невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга.

Элементом клеточной структуры мозга является нервная клетка – *нейрон*. Нейрон в своем строении имеет много общих черт с другими клетками биоткани: тело нейрона окружено плазматической мембраной, внутри которой находится цитоплазма, ядро и другие составляющие клетки. Однако нервная клетка существенно отличается от иных по своему *функциональному назначению*. Нейрон выполняет прием, элементарное преобразование и дальнейшую передачу информации другим нейронам. Информация переносится в виде импульсов нервной активности, имеющих электрохимическую природу.

Нейроны крайне разнообразны по форме, которая зависит от их местонахождения в нервной системе и особенностей функционирования. На рис. 4.1 показана структура пары типичных биологических нейронов [1; 2]. Тело клетки содержит множество ветвящихся отростков двух типов. Отростки первого типа, называемые *дендритами* за их сходство с кроной раскидистого дерева, служат в качестве входных каналов для нервных им-

пульсов от других нейронов. Эти импульсы поступают в *сому* или *тело* клетки размером от 3 до 100 микрон, вызывая ее специфическое возбуждение, которое затем распространяется по выводному отростку второго типа – *аксону*. Длина аксонов обычно заметно превосходит размеры дендритов, в отдельных случаях достигая десятков сантиметров и даже метров. Гигантский аксон кальмара имеет толщину около миллиметра, и именно наблюдение за ним послужило выяснению механизма передачи нервных импульсов между нейронами.

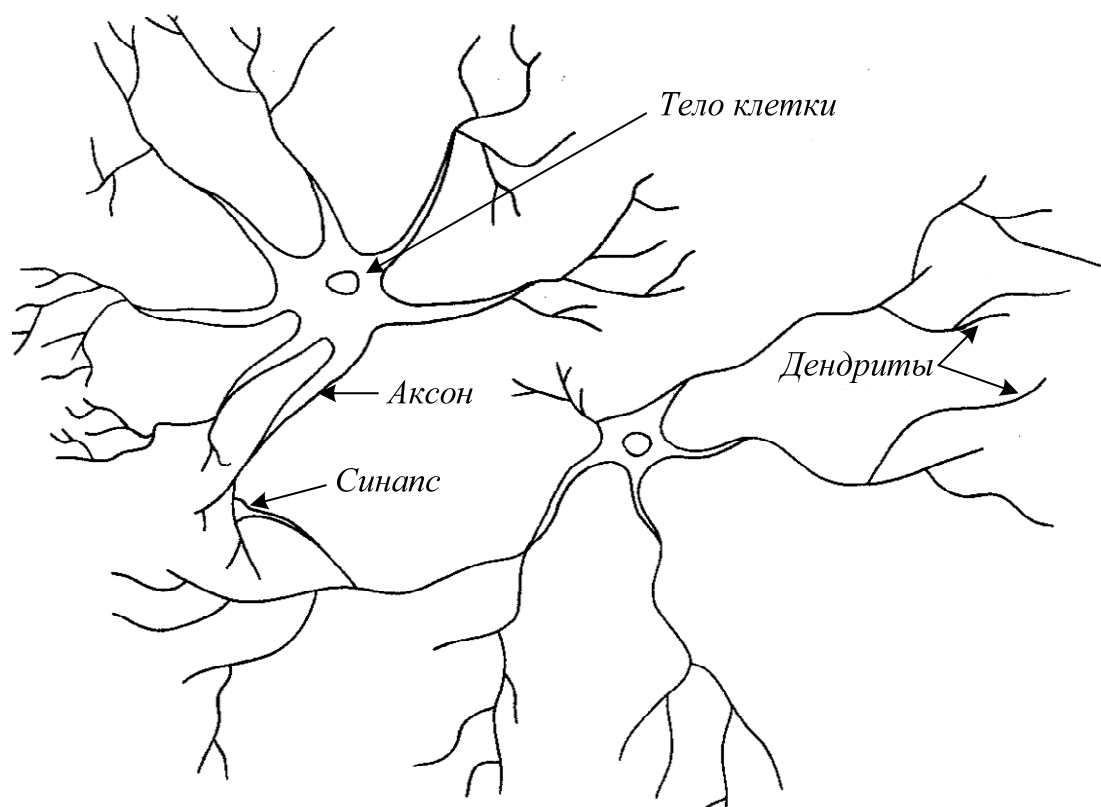


Рисунок 4.1 – Биологический нейрон

Тело нейрона, заполненное проводящим ионным раствором, окружено мембраной толщиной около 75 ангстрем, обладающей низкой проводимостью. Между внутренней поверхностью мембраны аксона и внешней средой поддерживается разность электрических потенциалов. Это осуществляется при помощи молекулярного механизма ионных насосов, создающих различную концентрацию положительных ионов K^+ и Na^+ внутри и вне клетки. Проницаемость мембраны нейрона селективна для этих ионов. Внутри аксона клетки, находящейся в состоянии покоя, активный

транспорт ионов стремится поддерживать концентрацию ионов калия более высокой, чем ионов натрия, тогда как в жидкости, окружающей аксон, выше оказывается концентрация ионов Na^+ .

Пассивная диффузия более подвижных ионов калия приводит к их интенсивному выходу из клетки, что обуславливает ее общий отрицательный относительно внешней среды *потенциал покоя*, составляющий около 65 мВ.

Под воздействием стимулирующих сигналов от других нейронов мембрана аксона динамически изменяет свою проводимость. Это происходит, когда суммарный внутренний потенциал превышает пороговое значение масштаба 50 мВ. Мембрана на короткое время, составляющее около 2 мс, изменяет свою полярность (деполяризуется) и достигает *потенциала действия* около +40 мВ. На микроуровне это объясняется кратковременным повышением проницаемости мембраны для ионов Na^+ и активным поступлением их в аксон. В дальнейшем, по мере выхода ионов калия, положительный заряд с внутренней стороны мембраны меняется на отрицательный, и наступает так называемый *период рефрактерности*, длящийся около 200 мс. В течение этого времени нейрон является полностью пассивным, практически неизменно сохраняя потенциал внутри аксона на уровне около 70 мВ. Импульс деполяризации клеточной мембраны, называемый *спайком*, распространяется вдоль аксона практически без затухания, поддерживаясь локальными ионными градиентами. Скорость перемещения спайка является относительно невысокой и составляет от 100 до 1 000 см/с.

Возбуждение нейрона в виде спайка передается другим нейронам, которые таким образом объединены в проводящую нервную импульсную сеть. Участки мембраны на аксоне, где размещаются области контакта аксона данного нейрона с дендритами других нейронов, называются *синапсами*. В области синапса, имеющего сложное строение, происходит обмен информацией о возбуждении между нейронами. Механизмы синаптической передачи достаточно сложны и разнообразны. Они могут иметь химическую и электрическую природу. В химическом синапсе в передаче импульсов участвуют специфические химические вещества – *нейромедиаторы*, вызывающие изменения проницаемости локального участка мембраны. В зависимости от типа вырабатываемого медиатора синапс может

обладать возбуждающим (эффективно проводящим возбуждение) или тормозящим действием. Обычно на всех отростках одного нейрона вырабатывается один и тот же медиатор, поэтому нейрон в целом функционально является тормозящим или возбуждающим. Это важное наблюдение о наличии нейронов различных типов в последующих главах будет существенно использоваться при проектировании искусственных систем.

4.2. Биологические нейронные сети

Взаимодействующие между собой посредством передачи через отростки возбуждений нейроны формируют *нейронные сети*. Переход от рассмотрения отдельного нейрона к изучению нейронных сетей является естественным шагом в нейробиологической иерархии [1–4].

Общее число нейронов в центральной нервной системе человека достигает 10^{10} – 10^{11} , при этом каждая нервная клетка связана в среднем с 10^3 – 10^4 других нейронов. Установлено, что в головном мозге совокупность нейронов в объеме масштаба 1 мм^3 формирует относительно независимую локальную сеть, несущую определенную функциональную нагрузку.

Выделяют несколько (обычно три) основных типов нейронных сетей, отличающихся структурой и назначением. Первый тип составляют *иерархические* сети, часто встречающиеся в сенсорных и двигательных путях. Информация в таких сетях передается в процессе последовательного перехода от одного уровня иерархии к другому. Нейроны образуют два характерных типа соединений – *конвергентные*, когда большое число нейронов одного уровня контактирует с меньшим числом нейронов следующего уровня, и *дивергентные*, в которых контакты устанавливаются со все большим числом клеток последующих слоев иерархии. Сочетание конвергентных и дивергентных соединений обеспечивает многократное дублирование информационных путей, что является решающим фактором надежности нейронной сети. При гибели части клеток сохранившиеся нейроны оказываются в состоянии поддерживать функционирование сети.

Ко второму типу нейронных сетей относятся *локальные* сети, формируемые нейронами с ограниченными сферами влияния. Нейроны локальных сетей производят переработку информации в пределах одного уровня

иерархии. При этом функционально локальная сеть представляет собой относительно изолированную тормозящую или возбуждающую структуру.

Важную функцию также выполняют так называемые дивергентные *сети с одним входом*. Командный нейрон, находящийся в основании такой сети, может оказывать влияние сразу на множество нейронов, и поэтому сети с одним входом выступают согласующим элементом в сложном сочетании нейросетевых систем всех типов.

Рассмотрим схематически нейронную сеть, формирующую простую рефлекторную цепь с передачей возбуждения от раздражителя к двигательной мышце (рис. 4.2).

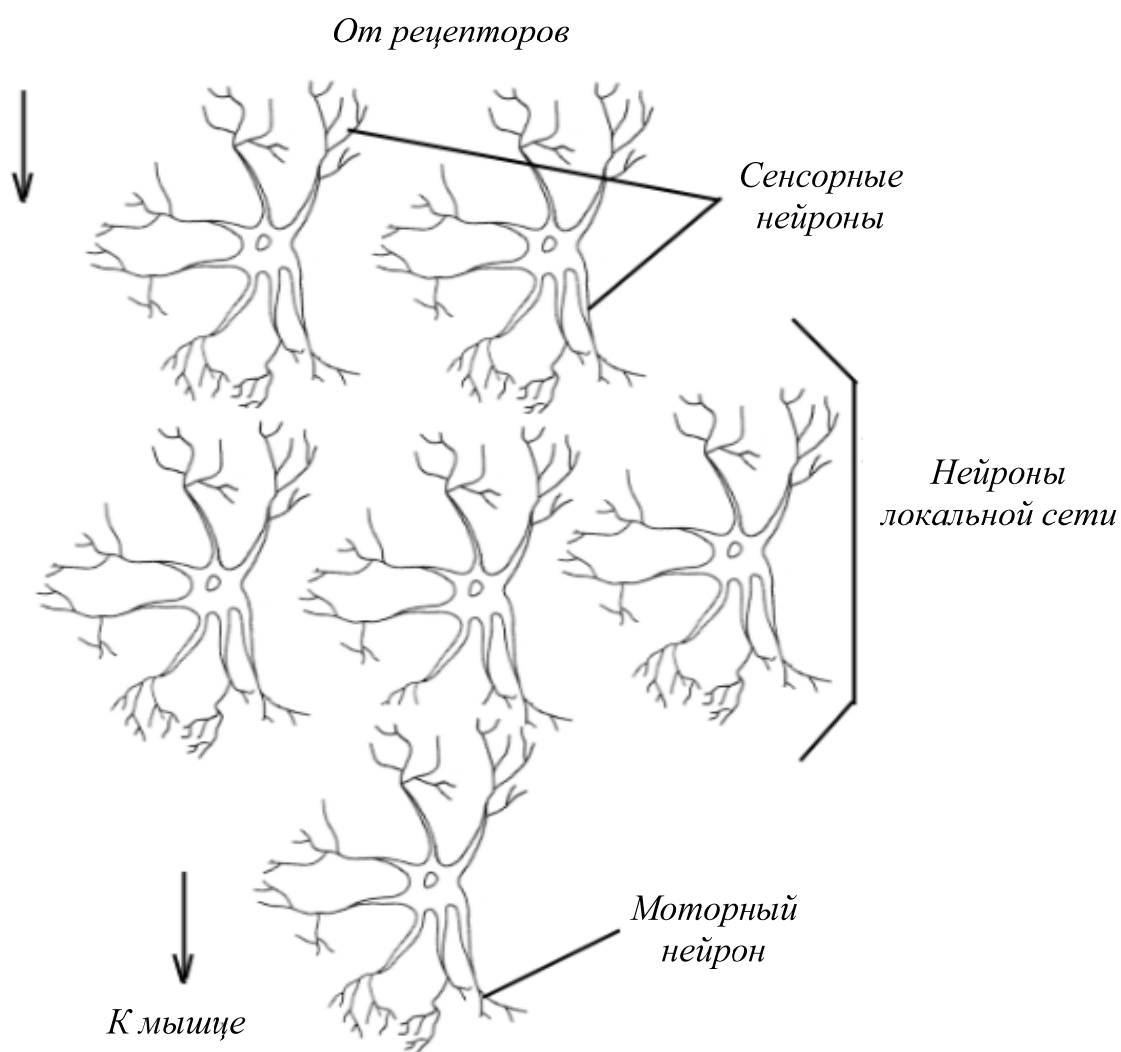


Рисунок 4.2 – Структура простой рефлекторной нейронной сети

Сигнал внешнего раздражителя воспринимается *сенсорными* нейронами, связанными с чувствительными клетками - рецепторами. Сенсорные нейроны формируют первый (нижний) уровень иерархии. Выработанные ими сигналы передаются нейронам локальной сети, содержащим множество прямых и обратных связей с сочетанием дивергентных и конвергентных соединений.

Характер преобразованного в локальных сетях сигнала определяет состояние возбуждения *моторных* нейронов. Эти нейроны, составляющие верхний в рассматриваемой сети уровень иерархии, образно говоря, «принимают решение», которое выражается в воздействии на клетки мышечной ткани посредством нервно - мышечных соединений.

4.3. Искусственный нейрон

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. Они представляют собой простые процессоры, вычислительные возможности которых обычно ограничиваются некоторым правилом комбинирования входных сигналов и правилом активации, позволяющим вычислить выходной сигнал по совокупности входных сигналов. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона. Хотя сетевые парадигмы весьма разнообразны, в основе почти всех их лежит эта конфигурация.

4.3.1. Нейрон Маккалоха – Питтса

В 1943 г. У. Маккалох и У. Питтс предложили [2; 5] в качестве искусственного нейрона использовать бинарный (релейный) элемент. Этот формальный нейрон вычислял взвешенную сумму n входных сигналов x_i и формировал на выходе единичный сигнал, в случае если вычисленная сумма превышала некоторый порог. В противном случае формальный нейрон возвращает нуль. Этими же авторами было показано, что при определенном выборе весов система параллельно функционирующих нейронов способна производить достаточно универсальные вычисления. В дальней-

шем эта модель совершенствовалась, и в настоящее время под нейроном Маккалоха – Питтса понимают многовходовой нелинейный преобразователь со взвешенными входными сигналами, показанный на рис. 4.3.

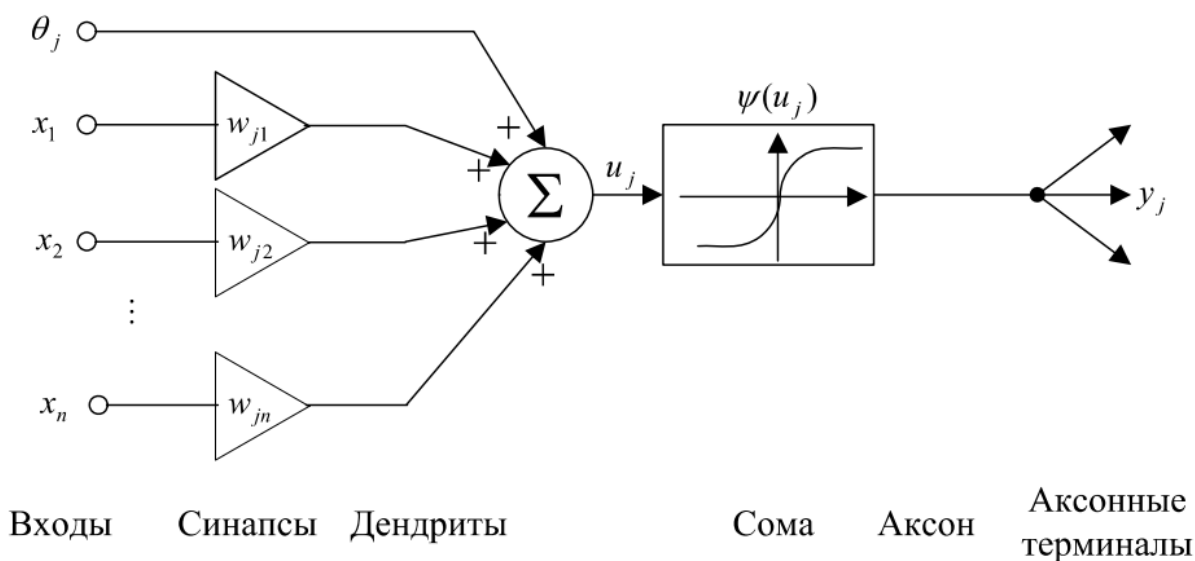


Рисунок 4.3 – Нейрон Маккалоха – Питтса

На входы j -го нейрона поступает n сигналов x_1, x_2, \dots, x_n , которые взвешиваются усилителями, реализующими синаптические веса, после чего взвешенные значения $w_{j1}x_1, w_{j2}x_2, \dots, w_{jn}x_n$ вместе с пороговым значением θ_j , именуемым также сигналом смещения, подаются на сумматор Σ , в результате чего формируется внутренний сигнал u_j . Сомы биологического нейрона моделируется с помощью некоторой нелинейной функции $\psi(u_j)$, называемой в теории искусственных нейронных сетей (ИНС) либо активационной, либо передаточной функцией формального нейрона. Таким образом, математическая модель Маккалоха – Питтса может быть записана в виде:

$$y_j = \psi \left(\sum_{i=1}^n w_{ji} x_i + \theta_j \right), \quad (4.1)$$

или

$$y_j = \psi \left(\sum_{i=0}^n w_{ji} x_i \right), \quad (4.2)$$

где $w_{j0} = \theta_j, x_0 \equiv 1$.

Вводя вектор синаптических весов j -го нейрона $w_j = (w_{j0}, w_{j1}, \dots, w_{jn})^T$ и вектор входов $x = (1, x_1, x_2, \dots, x_n)^T$, можно переписать (4.2) в векторной форме:

$$y_j = \Psi\left(w_j^T \cdot x\right), \quad (4.3)$$

более компактной и удобной для анализа поведения как нейронов, так и сформированных из них сетей.

Функциональные характеристики отдельных нейронов и сетей в целом определяются видом используемых активационных функций. Так, если первоначальная модель Маккалоха – Питтса использовала бинарный ограничитель (релейную функцию), то в настоящее время используется множество иных преобразований, некоторые из которых приведены на рис. 4.4. По причинам, которые будут изложены ниже, в нейронных сетях наибольшее распространение получила сигмоидальная функция

$$y_j = \frac{1}{1 + e^{-\gamma \cdot u_j}}, \quad (4.4)$$

характеристики которой в значительной мере зависят от параметра крутизны γ (с возрастанием значения параметра функция (4.4) приближается к релейной, не претерпевая при этом разрыва в точке 0).

Наряду с функциями, приведенными на рис. 4.4, широкое распространение получили биполярные активационные функции, некоторые из которых показаны на рис. 4.5.

$$\Psi(u) = \frac{1 - e^{-2\mu_j}}{1 + e^{-2\mu_j}}.$$

Среди биполярных активационных функций наибольшее распространение получили линейная

$$y_j = u_j \quad (4.5)$$

и функция гиперболического тангенса

$$y_j = \tanh(\gamma u_j) = \frac{1 - e^{-2\gamma \cdot u_j}}{1 + e^{-2\gamma \cdot u_j}}. \quad (4.6)$$

Первая из них используется в модели простейшего нейрона, называемого линейным ассоциатором, а вторая применяется в так называемых аппроксимирующих нейронных сетях.

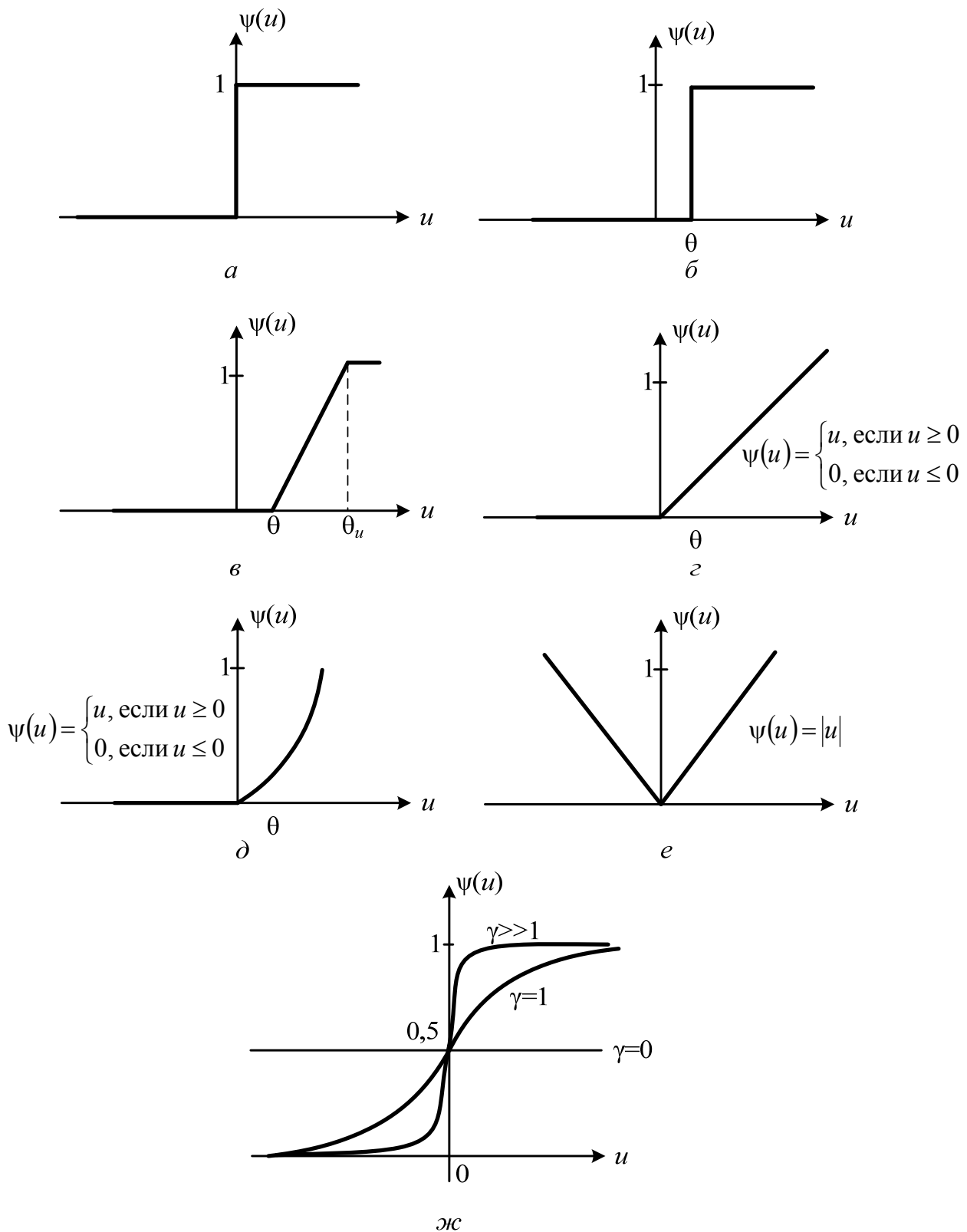


Рисунок 4.4 – Активационные функции нейронов;
 а – релейная функция; б – порогово - релейная функция;
 в – линейно пороговая функция; г – функция - выпрямитель;
 д – квадратичная функция; е – модульная функция; ж – сигмоидальная функция

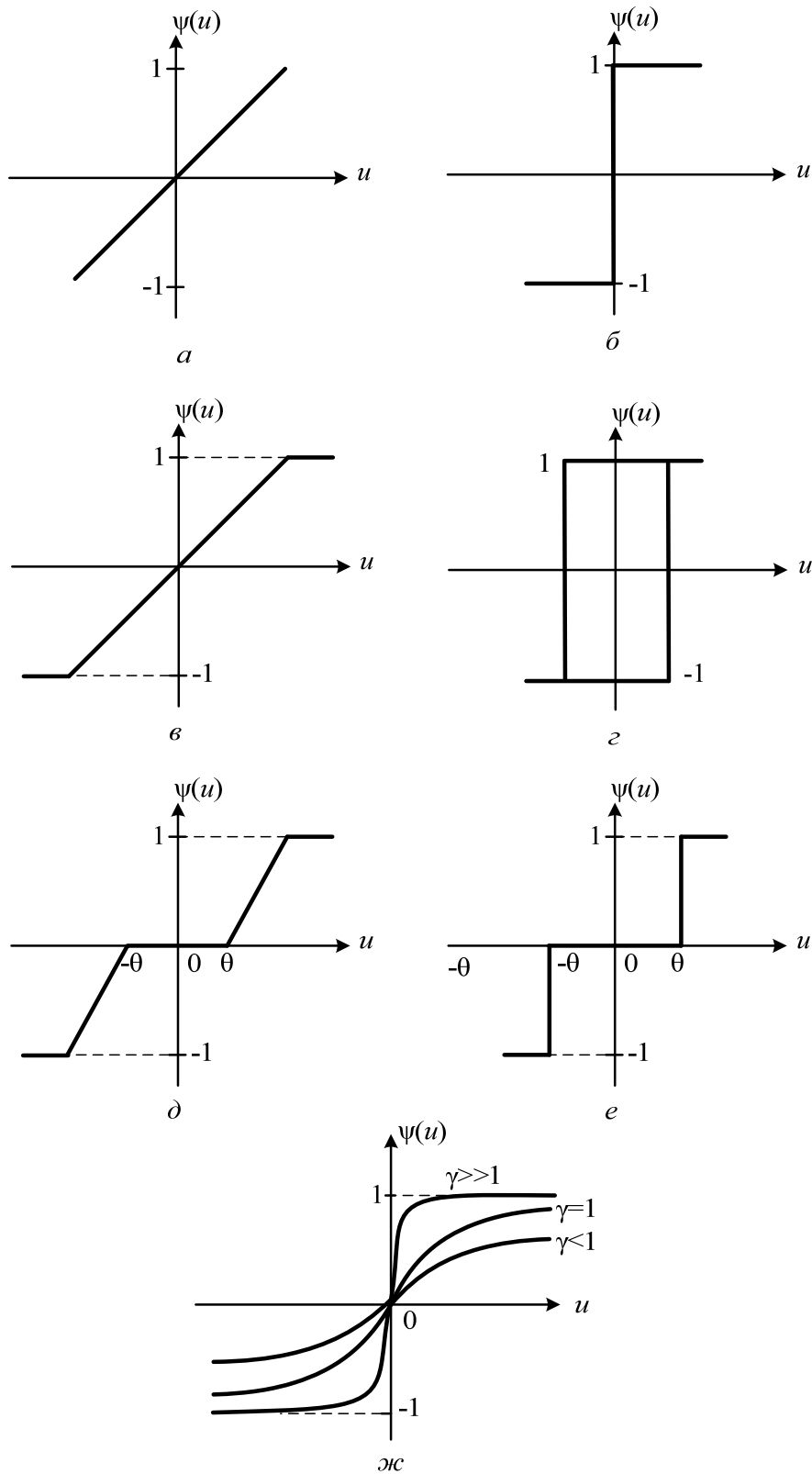


Рисунок 4.5 – Биполярные активационные функции:
a – линейный ассоциатор; *б* – сигнум – функция; *в* – ограничитель с насыщением;
г – ограничитель с гистерезисом; *д* – ограничитель с зоной нечувствительности;
е – сигнум - функция с зоной нечувствительности;
ж – функция гиперболического тангенса

4.3.2. Квадратичный нейрон

В ряде случаев нелинейное преобразование можно обеспечить и с помощью линейного ассоциатора путем специальной предварительной обработки входных сигналов. На рис. 4.6 показан квадратичный нейрон, вычисляющий функцию:

$$y_j = \theta_j + \sum_{i=1}^n w_{ji}x_i + \sum_{p=1}^n \sum_{l=1}^n w_{jpl}x_p x_l \quad (4.7)$$

с помощью набора элементарных блоков – умножителей и сумматоров сигналов θ_j , $w_{j1}x_1$, $w_{j11}x_1^2$, $w_{j12}x_1x_2$, ..., $w_{jpl}x_px_l$, ..., $w_{jnn}x_n^2$. Естественно, что количество синаптических весов в данном случае существенно увеличивается, однако, простота реализации зачастую обеспечивает преимущество именно таким моделям, с помощью которых можно обеспечить полиномиальное преобразование любой требуемой степени.

4.3.3. Обобщенный формальный нейрон

Одним из наиболее привлекательных свойств искусственных нейронных сетей являются их универсальные аппроксимирующие возможности, обеспечивающие широкое использование ИНС в разнообразных задачах обработки информации.

Исследования в этом направлении были предприняты Хехт - Нильсеном [6; 7], который на основе теоремы Колмогорова – Арнольда о представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одной переменной доказал, что функция многих переменных достаточно общего вида может быть представлена с помощью двухслойной нейронной сети, содержащей фиксированное число нейронов с заранее известными ограниченными функциями активации.

Позже было доказано, что конечная линейная комбинация фиксированных одномерных функций может однозначно аппроксимировать любую непрерывную функцию n действительных переменных на заданном гиперкубе.

В качестве таких функций были предложены так называемые сигмоидальные функции, являющиеся непрерывными, монотонными, возрастающими, ограниченными и имеющими отличные от нуля производные на всей области определения.

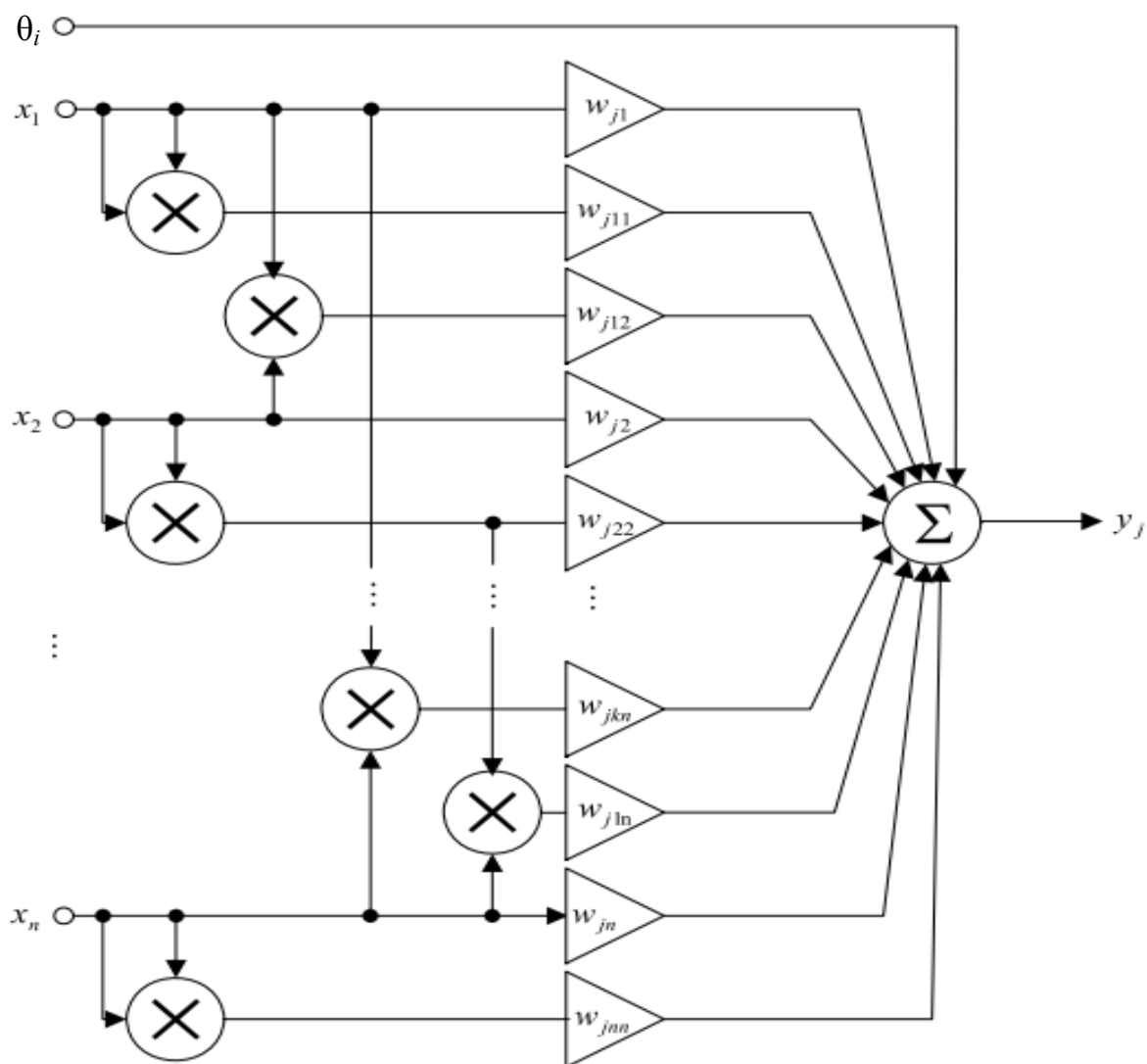


Рисунок 4.6 – Квадратичный нейрон

Рассмотрим формальный нейрон, приведенный на рис. 4.7 и являющийся обобщением нейрона Маккаллоха – Питтса, с тем отличием, что в него введены дополнительные настраиваемые блоки усиления γ_j и Γ_j . Здесь параметр γ_j управляет крутизной функции активации, а Γ_j – коэффициент усиления, определяющий максимальные и минимальные значения выходного сигнала y_j .

Несложно видеть, что выходной сигнал такого нейрона может быть записан в виде:

$$y_j = \Gamma_j \psi(\gamma_j w_j^T x). \quad (4.8)$$

Исторически первой в качестве сигмоидальной функции Цыбенко была предложена конструкция вида (1.4)

$$0 < \sigma(\gamma u) = \frac{1}{1 + e^{-\gamma \cdot u}} < 1, \quad (4.9)$$

определенная на множестве всех действительных чисел и принимающая только положительные значения. В практических приложениях более удобной оказалась кривая гиперболического тангенса (4.6)

$$-1 = \tanh(\gamma u_j) = \frac{1 - e^{-2\gamma \cdot u_j}}{1 + e^{-2\gamma \cdot u_j}} < 1, \quad (4.10)$$

связанная с сигмоидой соотношением

$$\sigma(\gamma u) = \frac{1}{2} \left(\tanh\left(\frac{\gamma u}{2}\right) + 1 \right). \quad (4.11)$$

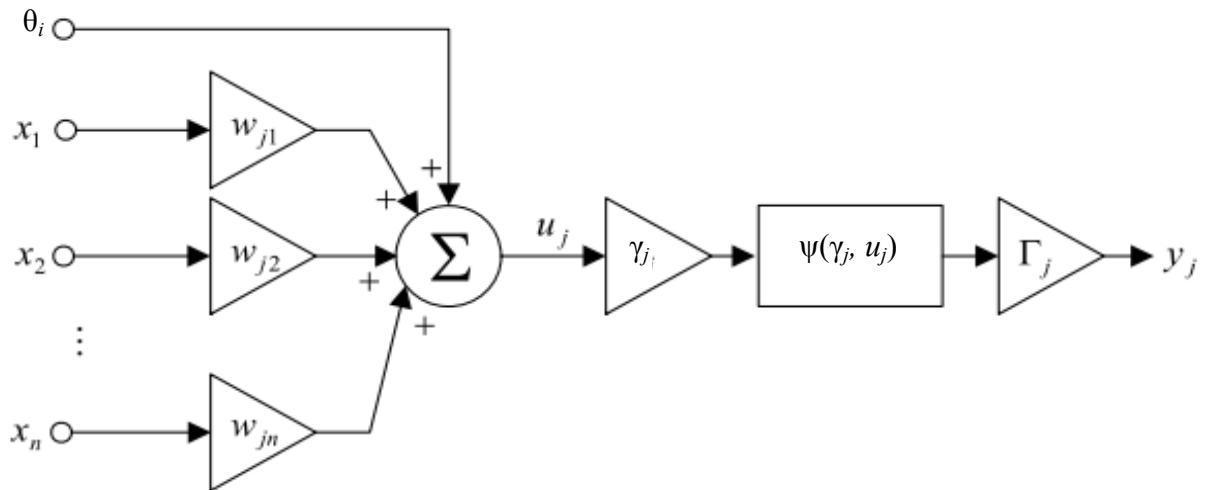


Рисунок 4.7 – Обобщенный формальный нейрон

Задав ограничения на квадрате $-1 < u_j < 1$, $-1 < y < 1$, в качестве возможных активационных функций нейрона (4.8) можно использовать следующие:

$$\psi_1(\gamma u) = \tanh(\gamma u) = \frac{1 - e^{-2\gamma \cdot u}}{1 + e^{-2\gamma \cdot u}}, \quad \Gamma_1 < \frac{1}{\tanh \gamma}, \quad (4.12)$$

$$\psi_2(\gamma u) = \frac{\gamma u}{\sqrt{1 + \gamma^2 u^2}}, \quad \Gamma_1 < \frac{\sqrt{1 + \gamma^2}}{\gamma}, \quad (4.13)$$

$$\psi_3(\gamma u) = \sin\left(\gamma u \frac{\pi}{2}\right), \quad \Gamma_3 < \frac{1}{\sin\left(\gamma \frac{\pi}{2}\right)}, \quad (4.14)$$

$$\psi_4(\gamma u) = \frac{2}{\pi} \operatorname{arctg}(\gamma u), \quad \Gamma_4 < \frac{\pi}{2 \operatorname{arctg} \gamma}, \quad (4.15)$$

$$\psi_5(\gamma u) = \gamma u - \frac{\gamma^3}{3} u^3, \quad \Gamma_5 < \frac{3}{3\gamma - \gamma^3}, \quad (4.16)$$

Графики данных зависимостей приведены на рис. 4.8.

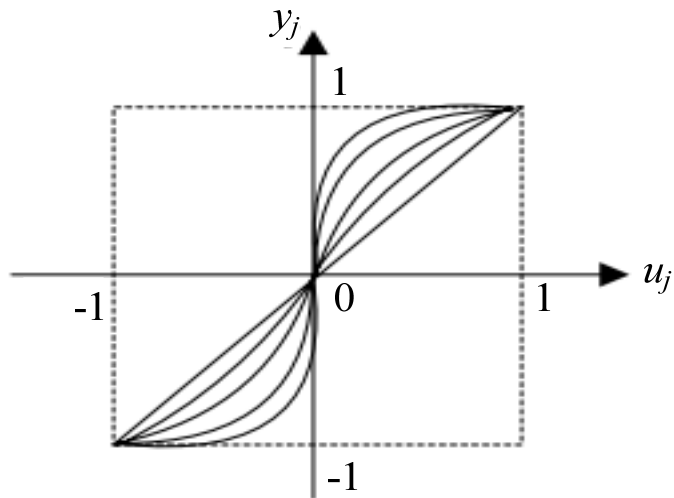


Рисунок 4.8 – Активационные функции обобщенного нейрона

Видно, что все эти функции отвечают предъявляемым требованиям. Так как при обучении нейронных сетей приходится иметь дело как с самими активационными функциями, так и с их производными, целесообразным является такое представление функций $\psi_j^{(\bullet)}$, которое упрощало бы вычисление производных, обеспечивая тем самым и упрощение процесса настройки нейронов сети.

Одним из возможных подходов к получению такого удобного описания является представление функций $\psi_j^{(\bullet)}$ в виде степенного ряда. Так, разложив (4.12)–(4.15) в ряд Маклорена, можно получить

$$\psi_1(\gamma u) = \gamma u - \frac{1}{3}(\gamma u)^3 + \frac{2}{15}(\gamma u)^5 - \frac{17}{315}(\gamma u)^7 + \dots, \quad (4.17)$$

$$\psi_2(\gamma u) = \gamma u - \frac{1}{2}(\gamma u)^3 + \frac{3}{8}(\gamma u)^5 - \frac{39}{336}(\gamma u)^7 + \dots, \quad (4.18)$$

$$\psi_3(\gamma u) = \frac{\pi}{2} \left(\gamma u - \frac{\pi^2}{24}(\gamma u)^3 + \frac{\pi^4}{1920}(\gamma u)^5 - \frac{\pi^6}{322560}(\gamma u)^7 + \dots \right), \quad (4.19)$$

$$\psi_4(\gamma u) = \frac{2}{\pi} \left(\gamma u - \frac{1}{3}(\gamma u)^3 + \frac{1}{5}(\gamma u)^5 - \frac{1}{7}(\gamma u)^7 + \dots \right), \quad (4.20)$$

$$\psi_5(\gamma u) = \gamma u - \frac{1}{3}(\gamma u)^3, \quad (4.21)$$

при этом первые четыре члена разложения (4.17)–(4.20) обеспечивают точность аппроксимации не хуже 10^{-2} [2].

Характер изменения коэффициентов рядов позволяет предположить, что выражения (4.17)–(4.20) можно представить в виде ряда:

$$\psi^*(\gamma u) = \varphi_0 \gamma u + \varphi_1 (\gamma u)^3 + \varphi_2 (\gamma u)^5 + \varphi_3 (\gamma u)^7 + \dots = \sum_{l=0}^L \varphi_l (\gamma u)^{2l+1}, \quad (4.22)$$

коэффициенты φ_l которого удовлетворяют уравнению авторегрессии первого порядка [2]:

$$\varphi_{l+1} = \alpha \varphi_l + \xi_{l+1}, \quad -1 < \alpha < 0, \quad \varphi_0 = 1, \quad (4.23)$$

в котором параметр α и дисперсия случайной компоненты ξ могут быть оценены с помощью известных соотношений метода наименьших квадратов:

$$\alpha = \frac{\sum_{l=0}^{L-1} \varphi_{l+1} \cdot \varphi_l}{\sum_{l=0}^{L-1} \varphi_l^2}, \quad \sigma_{\xi}^2 = \frac{\sum_{l=0}^{L-1} (\varphi_{l+1} \cdot \alpha \varphi_l)^2}{L-1}. \quad (4.24)$$

Результаты расчетов с использованием тех же четырех членов разложения показали [2], что параметр α , обеспечивающий требуемые свойства активационной функции, для всех рассмотренных кривых (4.12)–(4.16) лежит в интервале $-0,5 < \alpha < -0,3$, а $\sigma_{\xi}^2 < 10^{-2}$. На рис. 4.9 в качестве примера показана активационная функция, представленная рядом (4.22), с $\alpha = -0,3$ (сплошная линия) и $\alpha = -0,5$ (пунктирная линия), и коэффициентом усиления:

$$\Gamma^* = \left(\sum_{l=0}^{L-1} \varphi_l \right)^{-1}. \quad (4.25)$$

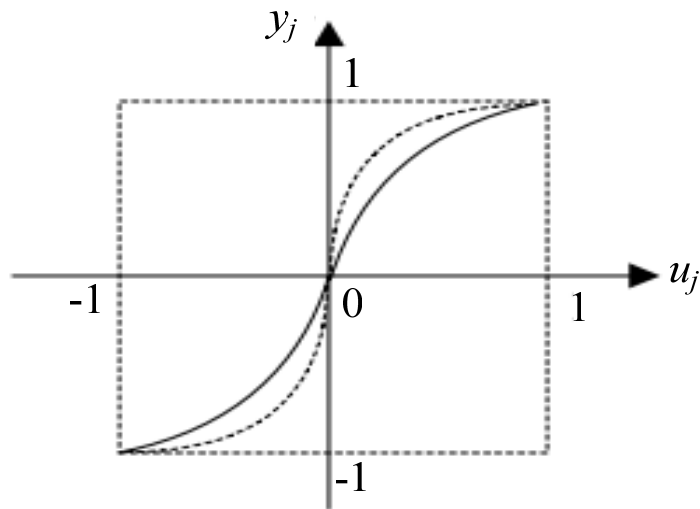


Рисунок 4.9 – Активационная функция в виде степенного ряда

Очевидно, что данная функция обладает всеми указанными выше свойствами сигмоидальных функций, при этом дифференцирование полинома в вычислительном смысле проще дифференцирования, например, функций (4.12)–(4.15).

4.3.4. Нейрон Фукушимы

В середине 70-х годов К. Фукушима разработал искусственную нейронную сеть, получившую название «когнитрон» и являющуюся формальной моделью биологической системы восприятия и распознавания, инвариантную к поворотам, сдвигам, изменениям масштабов образов [8; 9].

В основе когнитрона лежит нейрон Фукушимы, отличающийся от нейрона Маккаллоха – Питтса тем, что если у последнего синаптические веса могут быть положительными, отрицательными или нулевыми, то в модели Фукушимы все веса полагаются неотрицательными, а входы разбиты на две группы: возбуждающие и тормозящие. На рис. 4.10 приведена схема нейрона Фукушимы.

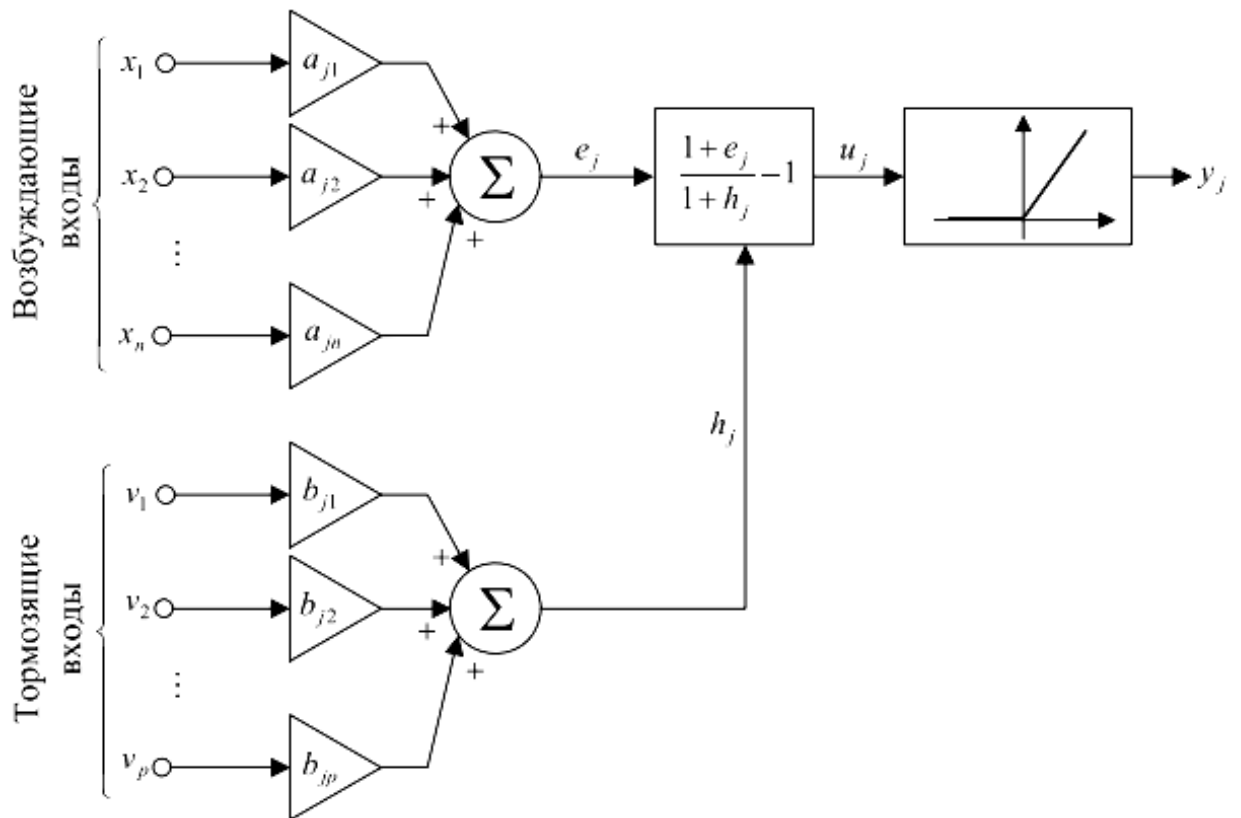


Рисунок 4.10 – Нейрон Фукушимы

Сигнал e_j , являющийся взвешенной суммой возбуждающих входов, компенсируется сигналом h_j , являющимся взвешенной комбинацией тормозящих сигналов. Выходной сигнал нейрона описывается соотношением:

$$y_j = \psi \left(\frac{1 + \sum_{i=1}^n a_{ji} \cdot x_i}{1 + \sum_{i=1}^p b_{ji} \cdot v_i} \right), \quad (4.26)$$

где

$$\psi(u_j) = \begin{cases} u_j, & \text{если } u_j \geq 0, \\ 0 & \text{в противном случае} \end{cases} \quad (4.27)$$

активационная функция типа «выпрямитель» (см. рис. 4.4, з).

При выполнении условия:

$$\frac{1 + \sum_{i=1}^n a_{ji} \cdot x_i}{1 + \sum_{i=1}^p b_{ji} \cdot v_i} > 1 \quad (4.28)$$

на выходе нейрона появляется сигнал

$$y_j = \frac{\sum_{i=1}^n a_{ji} \cdot x_i - \sum_{i=1}^p b_{ji} \cdot v_i}{1 + \sum_{i=1}^p b_{ji} \cdot v_i}. \quad (4.29)$$

Если $\sum_{i=1}^p b_{ji} \cdot v_i \ll 1$, нейрон Фукушимы работает как линейный ассоциатор (4.5), то есть:

$$y_j \approx \sum_{i=1}^n a_{ji} \cdot x_i - \sum_{i=1}^p b_{ji} \cdot v_i. \quad (4.30)$$

4.3.5. Неокогнитрон и инвариантное распознавание образов

Новая работа Фукушимы «НЕОКОГНИТРОН» была опубликована в 1980 г. [10; 11], хотя и имеет много общих черт с его прародителем «КОГНИТРОНОМ», но одновременно он претерпел значительные изменения и усложнения, в соответствии с появлением новых нейробиологических данных.

НЕОКОГНИТРОН состоит из иерархии нейронных слоев, каждый из которых состоит из массива плоскостей. Каждый элемент массива состоит из пары плоскостей нейронов (рис. 4.11). Первая плоскость состоит из так называемых *простых* нейронов, которые получают сигналы от предыдущего слоя и выделяют определенные образы. Эти образы далее обрабатываются *сложными* нейронами второй плоскости, задача которых состоит в том, чтобы сделать выделенные образы менее зависимыми от их положения. Области связей показаны большими белыми кружками, а области конкуренции – маленькими темными на рис. 4.11.

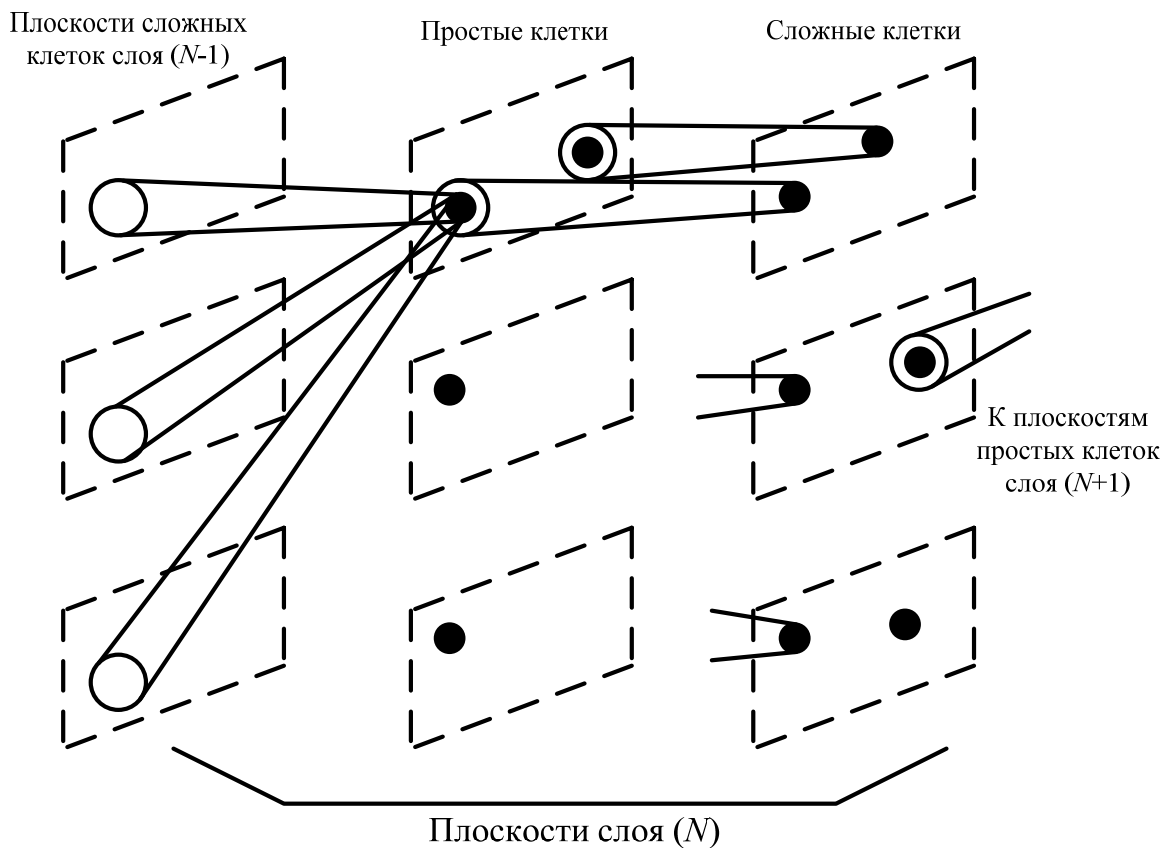


Рисунок 4.11 – Общая схема НЕОКОГНИТРОНА

Нейроны каждой пары плоскостей обучаются реагировать на определенный образ, представленный в определенной ориентации. Для другого образа или для нового угла поворота образа требуется новая пара плоскостей. Таким образом, при больших объемах информации НЕОКОГНИТРОН представляет собой огромную структуру с большим числом плоскостей и слоев нейронов.

Простые нейроны чувствительны к небольшой области входного образа, называемой *рецептивной* областью (или что то же самое, областью связей). Простой нейрон приходит в возбужденное состояние, если в его рецептивной области возникает определенный образ. Рецептивные области простых клеток перекрываются и покрывают все изображение. Сложные нейроны получают сигналы от простых клеток, при этом для возбуждения сложного нейрона достаточно одного сигнала от любого простого нейрона. Тем самым, сложная клетка регистрирует определенный образ независимо от того, какой из простых нейронов выполнил детектирование, и, значит, независимо от его расположения.

По мере распространения информации от слоя к слою картинка нейронной активности становится все менее чувствительной к ориентации, расположению образа и, в определенных пределах, к его размеру. Нейроны выходного слоя выполняют окончательное инвариантное распознавание.

Обучение НЕОКОГНИТРОНА аналогично уже рассмотренному обучению КОГНИТРОНА. При этом изменяются только синаптические веса простых клеток. Тормозящие нейроны вместо средней активности нейронов в области связей используют квадратный корень из взвешенной суммы квадратов входов:

$$v_i = \sqrt{\sum_j (b_j \cdot u_j)^2}. \quad (4.31)$$

Такая формула для активности тормозящей клетки менее чувствительна к размеру образа. После выбора простого нейрона, веса которого будут обучаться, он рассматривается в качестве представителя слоя, и веса всех остальных нейронов будут обучаться по тем же правилам. Таким образом, все простые клетки обучаются одинаково, выдавая при распознавании одинаковую реакцию на одинаковые образы.

Для уменьшения объема обрабатываемой информации рецептивные поля нейронов при переходе со слоя на слой расширяются, а число нейронов уменьшается. В выходном слое на каждой плоскости остается только один нейрон, рецептивное поле которого покрывает все поле образа предыдущего слоя. В целом функционирование НЕОКОГНИТРОНА происходит следующим образом. Копии входного изображения поступают на все плоскости простых клеток первого слоя. Далее все плоскости функционируют параллельно, передавая информацию следующему слою. По достижении выходного слоя, в котором каждая плоскость содержит один нейрон, возникает некоторое окончательное распределение активности. На результат распознавания указывает тот нейрон, активность которого оказалась максимальной. При этом существенно разным входным изображениям будут соответствовать разные результаты распознавания.

НЕОКОГНИТРОН успешно проявил себя при распознавании символов. Нужно отметить, что структура этой сети необычайно сложна и объем вычислений очень велик, поэтому компьютерные модели НЕОКОГНИТРОНА будут слишком дорогими для промышленных приложений. Возможной альтернативой является, конечно, переход на аппаратные или оп-

тические реализации, однако их рассмотрение находится за рамками данного пособия.

4.3.6. Стохастическая модель нейрона

Рассмотренные выше модели нейронов являются детерминированными, поскольку реализуют однозначные отображения входных сигналов в выходной. В ряде приложений более эффективным является использование так называемых стохастических нейронов [2], в качестве которых могут быть использованы нейроны Маккаллоха – Питтса с активационной сигнум - функцией (рис. 4.4, б). Отличие состоит в том, что выходной сигнал такого нейрона определяется не только значениями входных сигналов и синаптических весов, но и некоторой функцией распределения вероятностей состояний $P(v)$. При этом:

$$y_j = \begin{cases} +1, & \text{с вероятностью } P(v), \\ -1, & \text{с вероятностью } 1 - P(v). \end{cases} \quad (4.32)$$

В качестве такой функции чаще всего используется сигмоид:

$$P(v) = \frac{1}{1 + e^{\frac{v}{T}}}, \quad (4.33)$$

где T – переменная, называемая в теории ИНС псевдотемпературой и определяющая уровень неопределенности состояний; v – параметр, описывающий эффект синаптического шума в биологических системах. При уменьшении псевдотемпературы ($T \rightarrow 0$) стохастический нейрон превращается в детерминированный.

4.3.7. Динамические нейроны

Все введенные ранее искусственные нейроны описываются алгебраическими уравнениями и являются статическими моделями. Это означает, что фактор времени явно не влияет на поведение нейрона: выходная реакция в текущий k -й момент времени $y_j(k)$ определяется только входными сигналами в этот же момент времени $x_1(k), x_2(k), \dots, x_n(k)$ и никак не зависит от прошлых состояний.

В задачах оптимизации и цифровой обработки сигналов широкое распространение получили динамические нейроны, описываемые разностными или дифференциальными уравнениями, поведение которых существ-

венным образом определяется их предысторией. К ним относятся нейроны Хопфилда, Гроссберга, нейроны - осцилляторы и другие модели [1–4].

Простейшим динамическим нейроном является нейрон Хопфилда, который представлен на рис. 4.12.

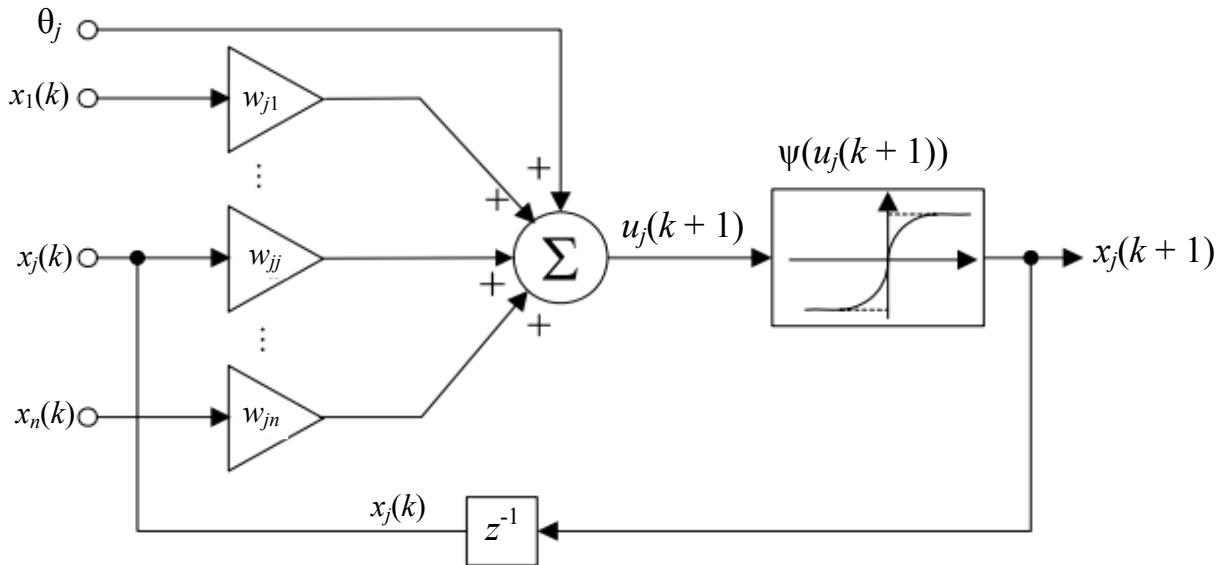


Рисунок 4.12 – Нейрон Хопфилда

Приведенный нейрон описывается рекуррентным уравнением:

$$x_j(k+1) = \psi \left(\sum_{i=1}^n w_{ji} x_i(k) + \theta_j \right), \quad k = 0, 1, 2, \dots; \quad j = 1, 2, \dots, n \quad (4.34)$$

В цепь обратной связи включен элемент чистой задержки, реализующий операцию сдвига назад ($z^{-1}x(k+1) = x(k)$) и обеспечивающий обычному нейрону требуемые динамические свойства. Заметим, что с точки зрения современной теории управления этот нейрон есть не что иное, как модель нелинейной динамической системы в пространстве состояний. С другой стороны, уравнение (4.34) описывает нелинейную марковскую последовательность первого порядка [12], поведение которой определяется только предыдущим состоянием. Несложно ввести модификацию стандартного нейрона Хопфилда, состояние которой будет определяться более отдаленной предысторией. На рис. 4.13 приведен динамический нейрон.

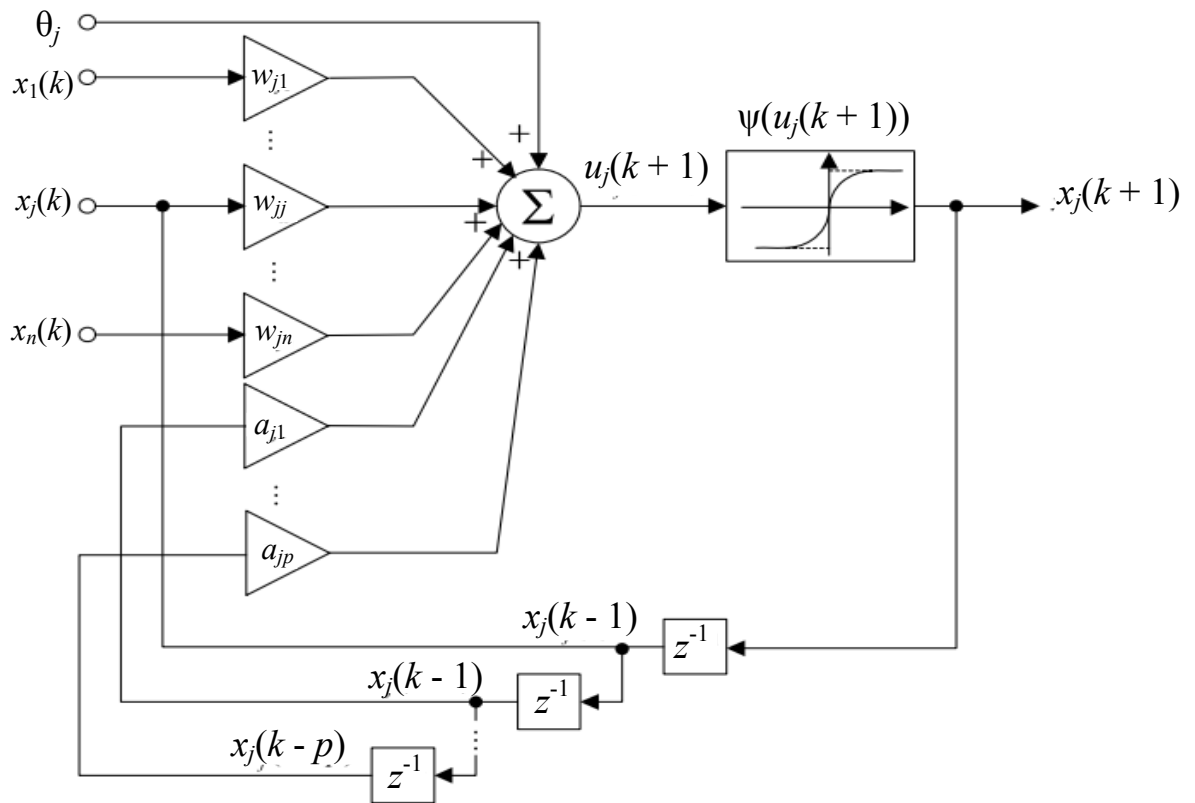


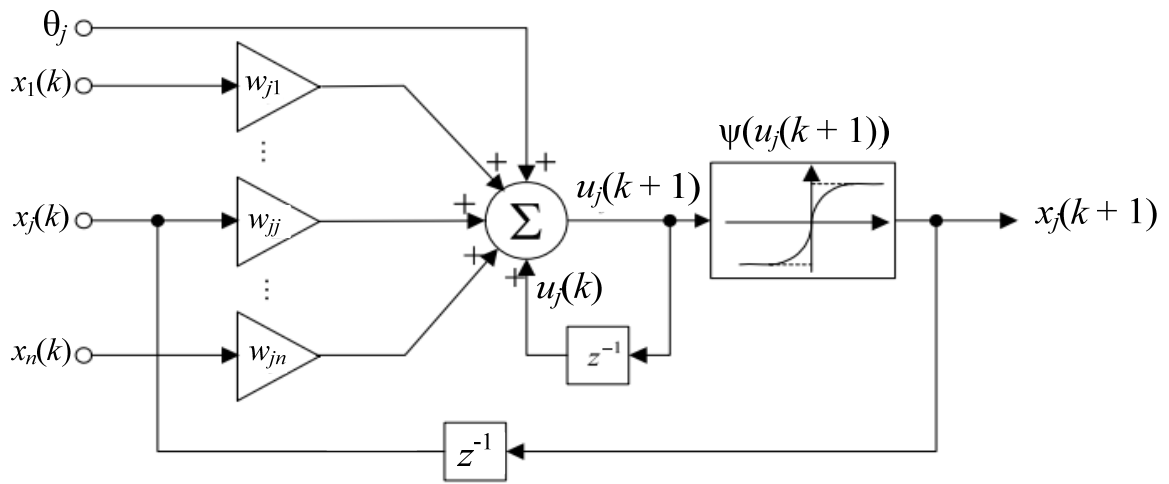
Рисунок 4.13 – Модифицированный нейрон Хопфилда

Данный нейрон описывается нелинейным уравнением авторегрессии вида:

$$x_j(k+1) = \psi \left(\sum_{l=1}^p a_{jl} x_l(k-1) + \sum_{i=1}^n w_{ji} x_i(k) + \theta_j \right), \quad (4.35)$$

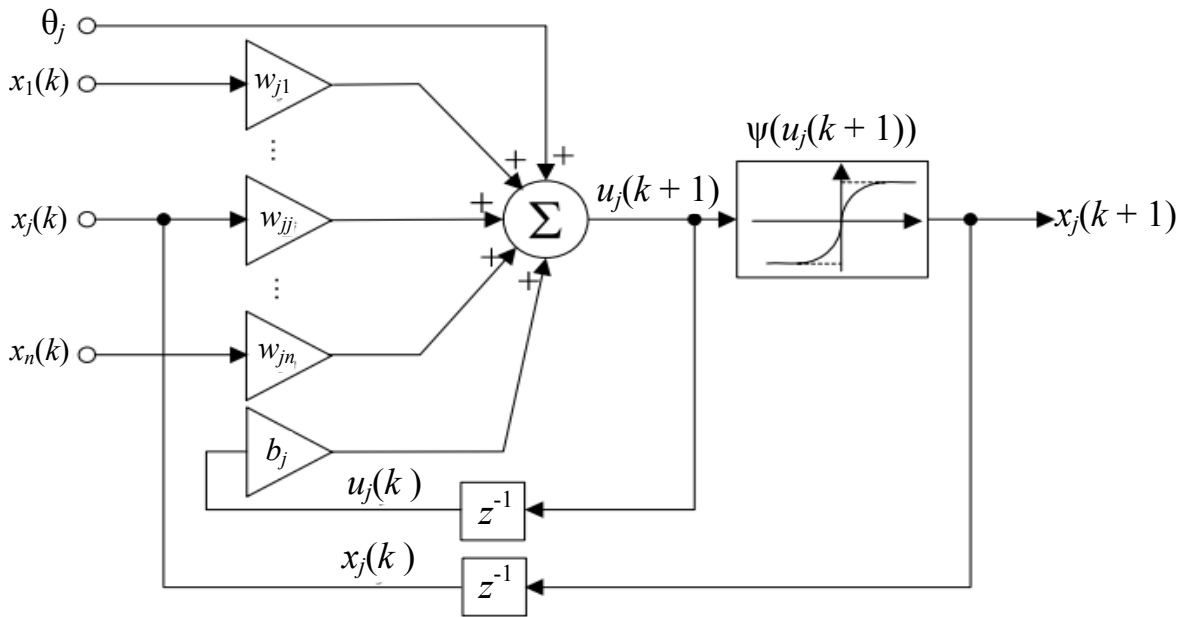
$$k = 0, 1, 2, \dots; \quad j = 1, 2, \dots, n$$

Более сложной конструкцией является нейрон Гроссберга, в котором в обратную связь поступают не только выходное состояние x_j , но и внутренний сигнал u_j . На рис. 4.14 приведены возможные реализации этой динамической модели.



a

$$\begin{cases} u_j(k+1) = u_j(k) + \sum_{i=1}^n w_{ji}x_i(k) + \theta_j, \\ x_j(k+1) = \psi(u_j(k+1)), \quad k = 0, 1, 2, \dots; \quad j = 1, 2, \dots, n. \end{cases} \quad (4.36)$$



б

$$\begin{cases} u_j(k+1) = b_j u_j(k) + \sum_{i=1}^n w_{ji}x_i(k) + \theta_j, \\ x_j(k+1) = \psi(u_j(k+1)), \quad k = 0, 1, 2, \dots; \quad j = 1, 2, \dots, n. \end{cases} \quad (4.37)$$

Рисунок 4.14 – Возможные реализации динамического нейрона Гроссберга

4.4. Персептроны

Основной особенностью искусственных нейронных сетей и, естественно, образующих их нейронов является способность к обучению, в процессе которого синаптические веса настраиваются с помощью того или иного адаптивного алгоритма с целью наиболее эффективного решения поставленной проблемы. Хотя один нейрон и способен выполнять простейшие процедуры распознавания, сила нейронных вычислений проистекает от соединений нейронов в сетях.

4.4.1. Адалина

Одним из простейших обучающихся нейронов является адаптивный линейный элемент (AD ALINE), предложенный Б. Уидроу [13; 14] и приведенный на рис. 4.15.

Структурно адалина весьма напоминает нейрон Маккаллоха – Питтса с активационной сигнум-функцией и состоит из двух основных частей: адаптивного линейного ассоциатора и нелинейной активационной функции. Адалина имеет $n + 1$ входов – x_0, x_1, \dots, x_n и два выхода – аналоговый u_j и бинарный y_j .

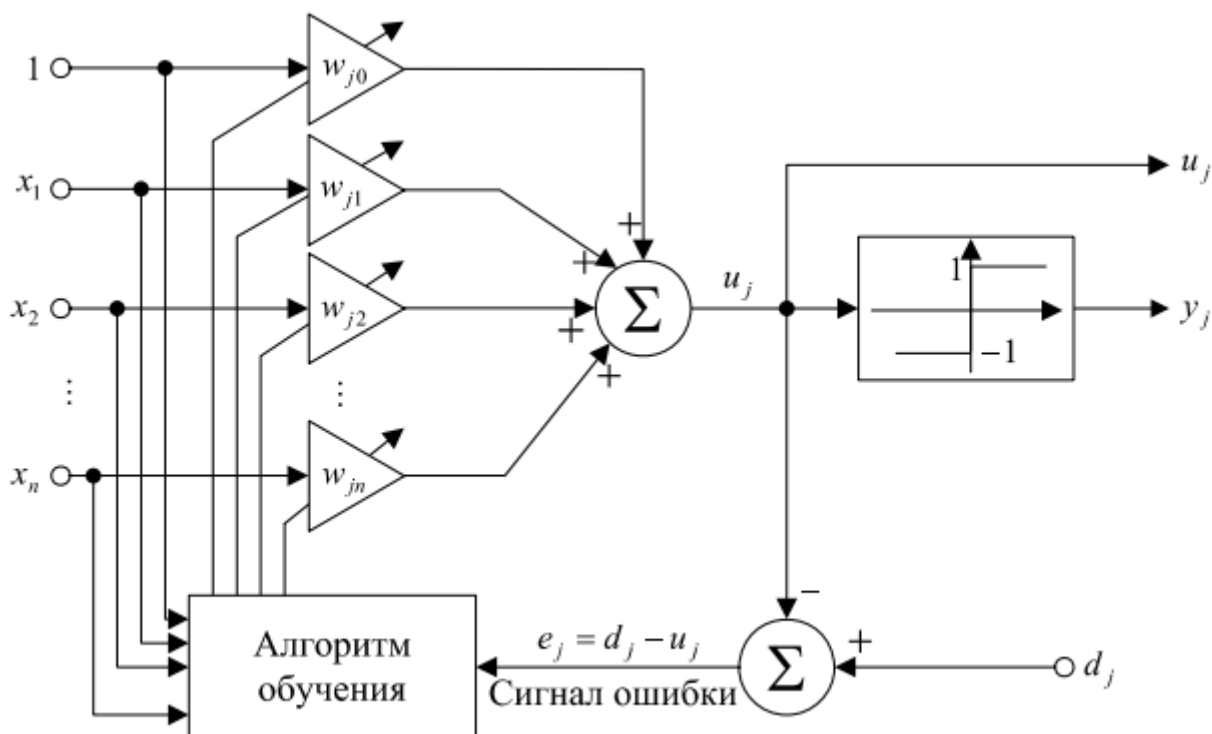


Рисунок 4.15 – Структура адалины

Кроме того, имеется дополнительный вход, на который подается обучающий сигнал d_j , показывающий, какой должна быть желаемая реакция нейрона на каждый конкретный набор входных сигналов. Аналоговый выход u_j представляет собой взвешенную сумму входов x_j :

$$u_j = \sum_{i=0}^n w_{ji} x_i = w_j^T x, \quad (4.38)$$

а бинарный выход u_j может принимать значения $+1$ или -1 , в зависимости от полярности аналогового сигнала u_j . Выходной сигнал u_j сравнивается с внешним обучающим сигналом d_j , и возникающий при этом сигнал ошибки $e_j = d_j - u_j$ поступает в алгоритм обучения, который перестраивает синаптические веса так, чтобы минимизировать некоторую функцию ошибки e_j , называемую критерием обучения. В качестве такой функции чаще всего применяют квадратичную, что позволяет использовать для обучения не только «родной» алгоритм, синтезированный Б. Уидроу и М. Хоффом специально для адальны, но и множество рекуррентных процедур адаптивной идентификации [2].

Адалина может использоваться в качестве элементарного нейрона в составе ИНС, а также самостоятельно в задачах распознавания образов, обработки сигналов, реализации логических функций.

4.4.2. Персептрон Розенблатта

Одной из первых искусственных сетей, способных к перцепции (восприятию) и формированию реакции на воспринятый стимул, явился PERCEPTRON Розенблатта [15]. Персептрон рассматривался его автором не как конкретное техническое вычислительное устройство, а как модель работы мозга. Нужно заметить, что после нескольких десятилетий исследований современные работы по искусственным нейронным сетям редко преследуют такую цель.

Элементарный персептрон Ф. Розенблатта структурно подобен адальне (рис. 4.16). Основное отличие состоит в алгоритме обучения, поскольку в адальне ошибка $e_j = d_j - u_j$ является линейной функцией от векторов входов x , в то время как ошибка обучения персептрона $e_j = d_j - u_j$ от входов зависит нелинейно, причем характер этой нелинейности определяется активационной функцией $\psi^{(\bullet)}$, которая в общем случае может быть недифференцируемой (например, сигнум-функция). Данное обстоятельство

во существенно затрудняет процесс обучения персептрона и требует использования специальных процедур.

В работах Розенблатта было сделано заключение о том, что нейронная сеть рассмотренной архитектуры будет способна к воспроизведению *любой* логической функции, однако, как было показано позднее М. Минским и С. Пейпертом [16], этот вывод оказался неточным. Были выявлены принципиальные неустраняемые ограничения однослойных персептронов, и впоследствии стал в основном рассматриваться многослойный вариант персептрона, в котором имеются несколько слоев процессорных элементов.

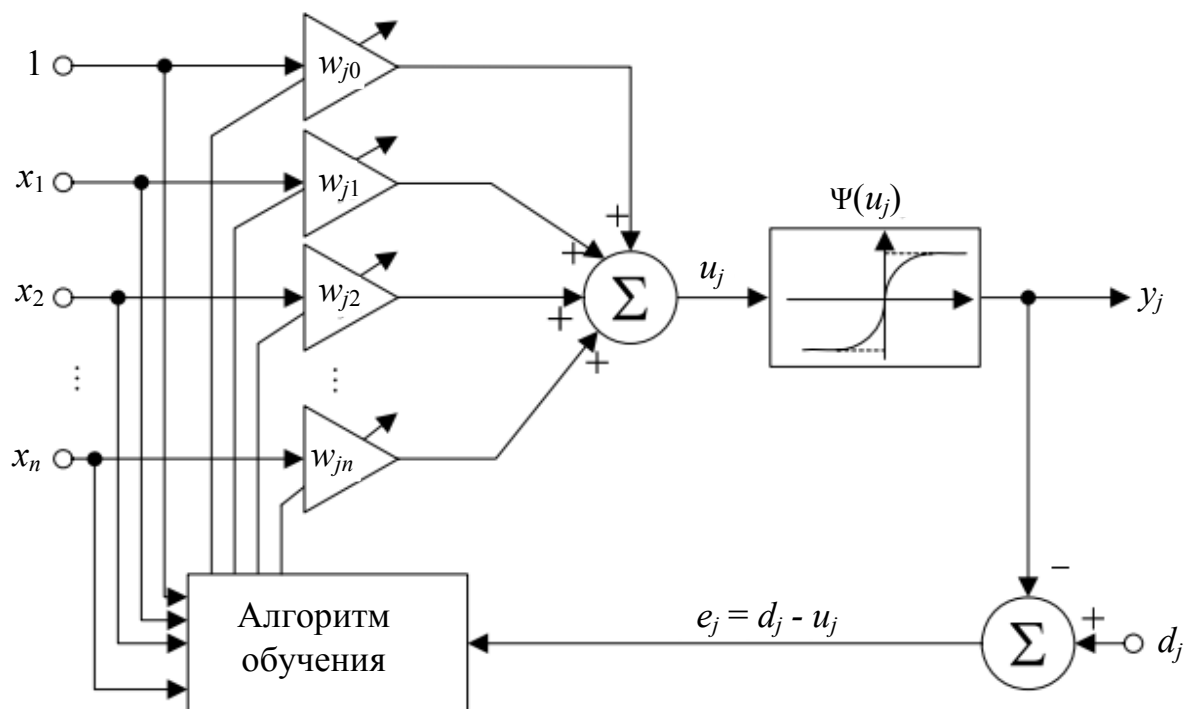


Рисунок 4.16 – Элементарный персептрон Розенблатта

С сегодняшних позиций однослойный персептрон представляет скорее исторический интерес, однако, на его примере могут быть изучены основные понятия и простые алгоритмы обучения нейронных сетей.

4.5. Многослойный персептрон

Новый всплеск интереса к персептронам приходится на конец 80 - х годов. Именно к этому периоду относится появление многослойных искусственных нейронных сетей и алгоритмов их обучения. Как уже отмеча-

лось, свойства искусственных нейронных сетей (ИНС) в значительной мере определяются их топологией (архитектурой). На сегодня большинство существующих нейронных сетей в зависимости от архитектуры может быть разделено на три большие категории [2]:

– *многослойные* с прямой передачей информации, в которых отдельные нейроны объединены в слои, между ними информация передается в одном направлении от входа к выходу (рис. 4.17, а);

– *рекуррентные* (с обратной связью), в которых сигналы с выхода могут подаваться на вход или внутренние слои сети (рис. 4.17, б);

– *клеточные*, в которых каждый нейрон связан только со своими соседями (рис. 4.17, в).

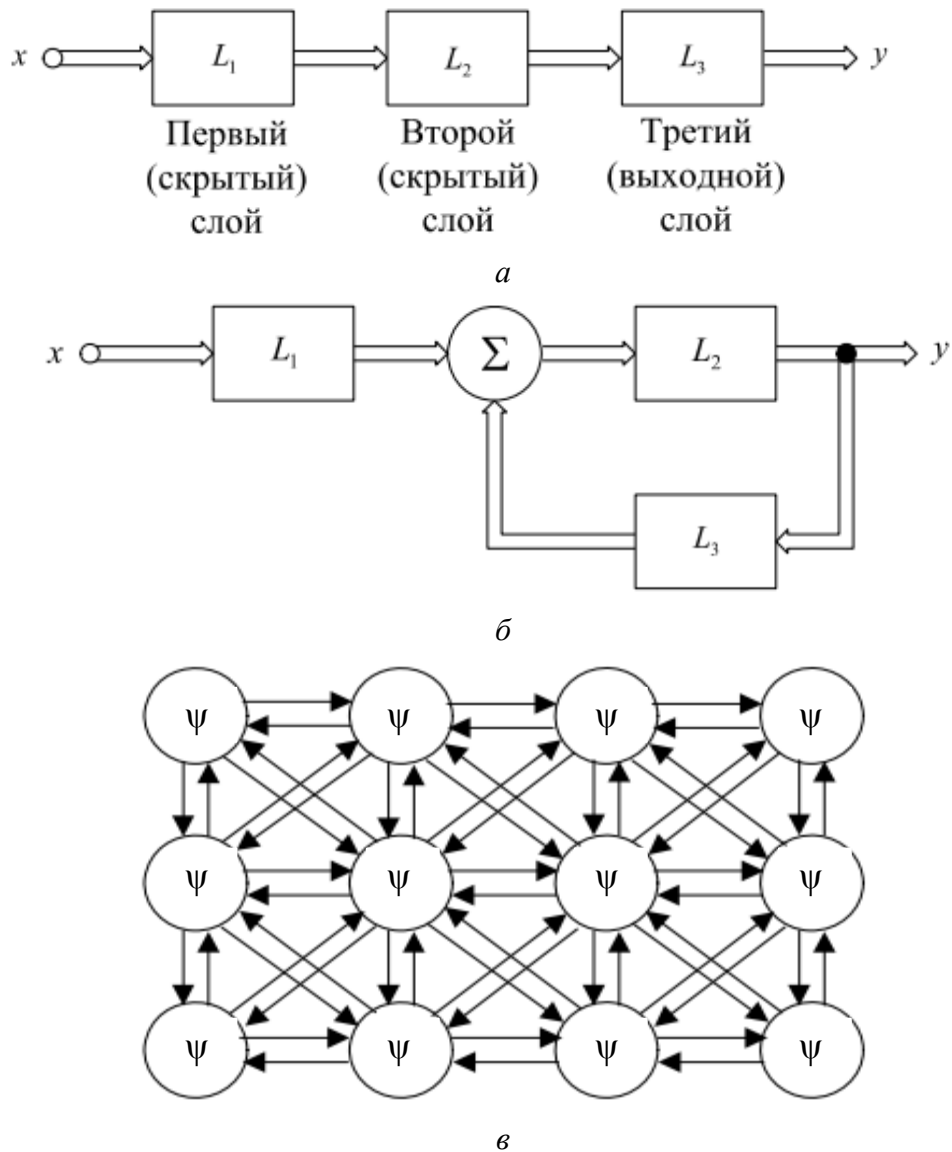


Рисунок 4.17 – Архитектуры нейронных сетей

Наиболее ярким представителем ИНС с прямой передачей информации являются многослойные персептроны. В общем случае каждый нейрон данного слоя однонаправленно связан со всеми нейронами последующего слоя. Эти связи организованы через синаптические веса, которые действуют как усилители в соответствующих каналах. Например, в трехслойном персептроне, схема которого приведена на рис. 4.18, *a* (сигналы смещения θ_j для простоты изображения опущены), нейроны сгруппированы в последовательные слои: нулевой, первый, второй и третий. Нейроны нулевого слоя, иногда называемого входным, не производят никаких вычислений, а служат лишь для передачи вектора входных сигналов $x = (x_1, x_2, \dots, x_n)^T$ на нейроны слоя, называемого первым скрытым слоем. Сигналы с первого слоя поступают на второй скрытый слой и далее – на третий. Последний, третий слой является выходным, и в нем формируется вектор выходных сигналов в ИНС в целом. Теоретически количество слоев может быть произвольно большим, однако рядом авторов было доказано [2], что для решения сколь угодно сложных задач аппроксимации и распознавания образов достаточно лишь трех слоев, то есть трехслойный персептрон является универсальным аппроксиматором.

Заметим, что в многослойных персептронах связи между нейронами внутри слоя или от высшего к низшему слою запрещены, при этом количество нейронов, а соответственно и синаптических весов, в каждом слое различно. Каждый нейрон (а это, как правило, элементарный персептрон Розенблатта) имеет один выход и множество входов, которые, в свою очередь, являются выходами всех нейронов предыдущего слоя.

Обозначив внутренний сигнал j -го нейрона s -го слоя, содержащего n_s нейронов, как $u_j^{[s]}$ ($s = 1, 2, 3; j = 1, 2, \dots, n_s$), несложно представить его в виде взвешенной суммы выходов предыдущего слоя

$$u_j^{[s]} = \sum_{i=0}^{n_s-1} w_{ji}^{[s]} o_i^{[s-1]}, \quad (4.39)$$

где $w_{ji}^{[s]}$ – синаптические веса, связывающие каждый нейрон s -го слоя со всеми нейронами предыдущего слоя так, что $o_i^{[s-1]} = x_i^{[s]}$, $o_i^{[0]} = x_i$, $o_i^{[3]} = y_i$.

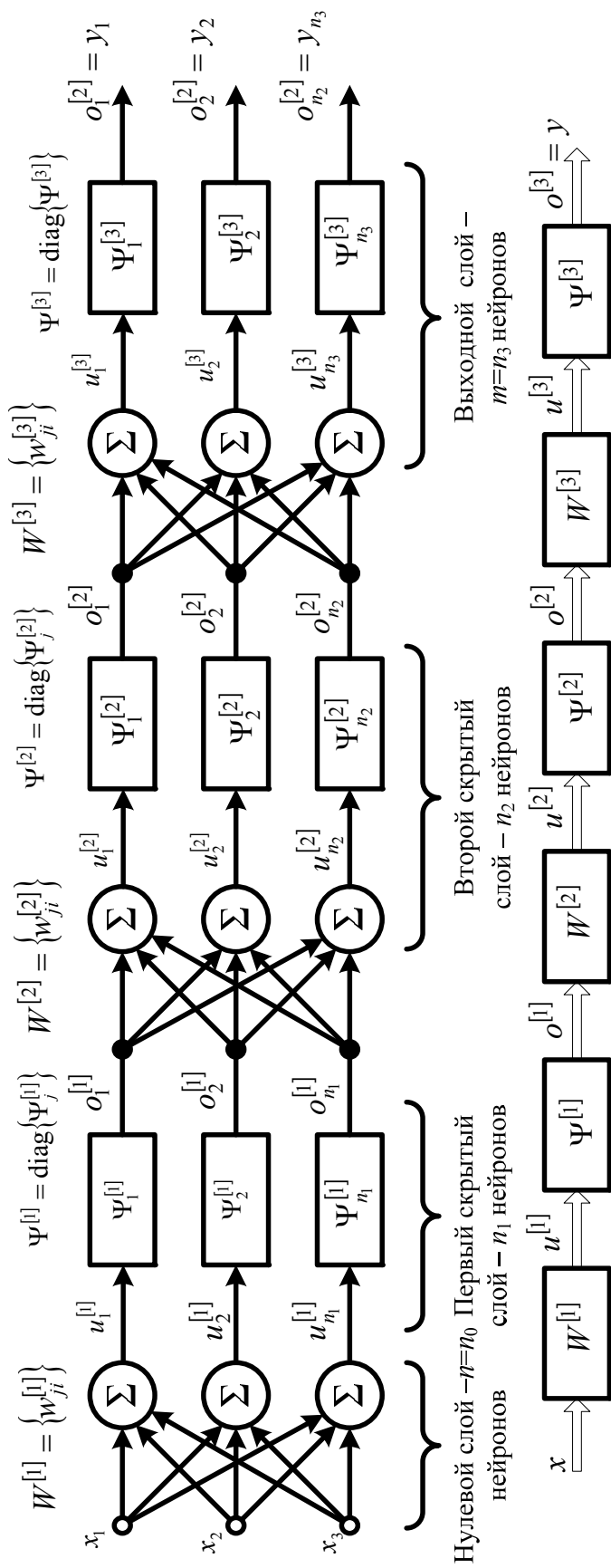


Рисунок 4.18 – Трехслойный перцептрон

Тогда выход каждого нейрона есть нелинейное преобразование вида:

$$o_j^{[s]} = \psi_j^{[s]}(u_j^{[s]}) = \psi_j^{[s]} \left(\sum_{i=0}^{n_s-1} w_{ji}^{[s]} o_i^{[s-1]} \right), \quad (4.40)$$

где для сокращения записи величина смещения $\theta_j^{[s]}$ обозначается как синаптический вес $w_{j0}^{[s]} = \theta_j^{[s]}$, а $o_0^{[s-1]} = 1$.

Используя векторно - матричную форму, можно записать преобразование, осуществляемое каждым слоем, в компактном виде:

$$o^{[s]} = \Psi^{[s]}(W^{[s]}x^{[s]}) = \Psi^{[s]}(W^{[s]}o^{[s-1]}), \quad (4.41)$$

где $o^{[s]} = (o_0^{[s]}, o_1^{[s]}, o_2^{[s]}, \dots, o_{n_s}^{[s]})^T - ((n_s + 1) \cdot 1)$ – выходной вектор сигналов, $x^{[s]} = (x_0^{[s]}, x_1^{[s]}, \dots, x_{n_{s-1}}^{[s]})^T = o^{[s-1]} - ((n_s + 1) \cdot 1)$ – входной вектор; $W^{[s]} - ((n_s + 1) \cdot (n_{s-1} + 1))$ – матрица синаптических весов; $\Psi^{[s]}(\bullet)$ – диагональный нелинейный оператор, образованный активационными функциями всех нейронов $\psi_j^{[s]}$.

С учетом введенных обозначений уравнение трехслойного персептрона принимает вид:

$$y = \psi(x) = \Psi^{[3]}(W^{[3]}\Psi^{[2]}(W^{[2]}\Psi^{[1]}(W^{[1]}x))), \quad (4.42)$$

а сама ИНС может быть представлена в компактной форме, приведенной на рис. 4.18, б.

На сегодня многослойный персептрон, пожалуй, наиболее распространенная нейросеть, нашедшая применение во многих областях. Однако для того чтобы обеспечить требуемое поведение ИНС в процессе преобразования входных сигналов в выходные, необходимо соответствующим образом организовать процедуру обучения. Процессы обучения искусственных нейронных сетей подробно будут рассмотрены в последующих разделах.

4.5.1. Радиально-базисные нейронные сети

В нервных системах биологических организмов существуют нейроны, чей выходной сигнал «локален», или «настроен» на некоторую узкую ограниченную область входного пространства [2]. Сеть, построенная на искусственных нейронах, обладающих выраженными локальными харак-

теристиками, была предложена в 1988 г. в качестве альтернативы многослойным перцептронам и получила название радиально - базисной нейронной сети (*Radial Basis Function Neural Network – RBFN*).

На рис. 4.19 приведена стандартная схема радиально-базисной сети с n -входами и m -выходами, осуществляющая нелинейное преобразование вида:

$$y_j = F_j(x) = w_{j0} + \sum_{i=1}^h w_{ji} \phi_i(x), \quad j = 1, 2, \dots, m, \quad (4.43)$$

где $\phi_i(x)$ – радиально - базисные функции, определяющие характер отображения из n - мерного пространства входов в m - мерное пространство выходов $R^n \rightarrow R^m$.

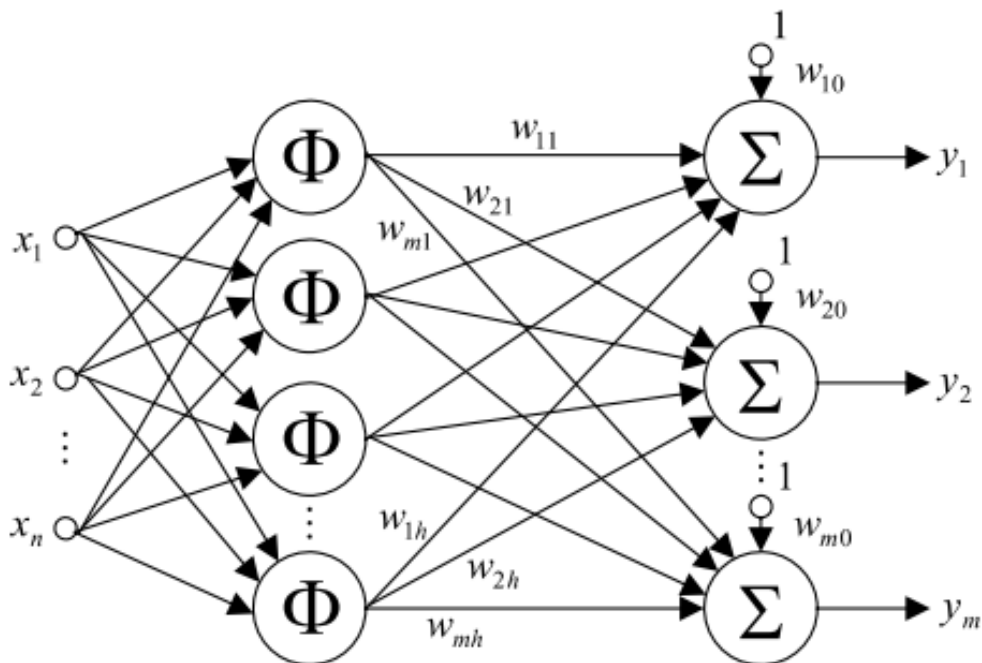


Рисунок 4.19 – Радиально-базисная нейронная сеть

Входной слой такой сети – это сенсоры, которые связывают ИНС с окружающей средой. Единственный скрытый слой, образованный нейронами Φ , осуществляет нелинейное преобразование входного пространства R^n в скрытое пространство R^h , как правило, высокой размерности ($h \gg n$). И, наконец, выходной слой, образованный адаптивными линейными ассоциаторами, формирует отклик сети $y = (y_1, y_2, \dots, y_m)^T$ на входной сигнал сети $x = (x_1, x_2, \dots, x_m)^T$.

Радиально-базисные сети реализуют идею, состоящую в том, что ли-

нейно неразделимая задача распознавания образов в пространстве R^n может стать линейно делимой в пространстве более высокой размерности R^h .

Свойства такой сети полностью определяются радиально-базисными функциями Φ , используемыми в нейронах скрытого слоя и формирующими некоторый базис для входных векторов - образов x . Радиально - базисная функция

$$\varphi(x) = \Phi(\|x - c\|, \sigma) = \Phi(r, \sigma) \quad (4.44)$$

– это многомерная функция, зависящая от расстояния $r = \|x - c\|$ между входным вектором x и собственным центром c и параметра ширины (масштаба) σ , определяющая локальную область входного пространства, на которую «реагирует» данная функция. Таким образом, каждый нейрон скрытого слоя вычисляет расстояние между входным вектором и своим центром и осуществляет над ним некоторое нелинейное преобразование $\Phi(r, \sigma)$.

Важно заметить, что в отличие от монотонных активационных функций многослойных сетей, радиально-базисные функции, как правило, симметричны и «накрывают» узкую область входного пространства. Достаточно часто радиально-базисные функции имеют колоколообразную форму и могут быть представлены в виде производных функций активации стандартных нейронов.

Наибольшее распространение получили гауссовские функции, имеющие пик в центре и монотонно убывающие по мере удаления от центра. В связи с этим в теории и практике ИНС кроме схемы 4.19 достаточно широко распространено представление, приведенное на рис. 4.20, при этом без потери общности рассматривается структура с одним выходом, осуществляющая отображение $R^n \rightarrow R^1$.

В большинстве случаев, связанных с практическими приложениями, центры узлов c_i и параметры ширины σ_i фиксированы, а настраиваются только синаптические веса w_i . При решении более сложных задач (распознавание образов, классификация и кластеризация и т.п.) во внимание принимаются все три множества параметров $c_i \in R^n$, $\sigma_i, w_i \in R^1$, $i = 0, 1, 2, \dots, h$. Однако при этом следует учитывать, что число базисных функций экспоненциально растет с размерностью входного пространства n . Таким образом, в радиально-базисных сетях на первый план выступает так назы-

ваемое «проклятие размерности», ограничивающее эффективность этих ИНС в задачах с большим числом входных признаков.

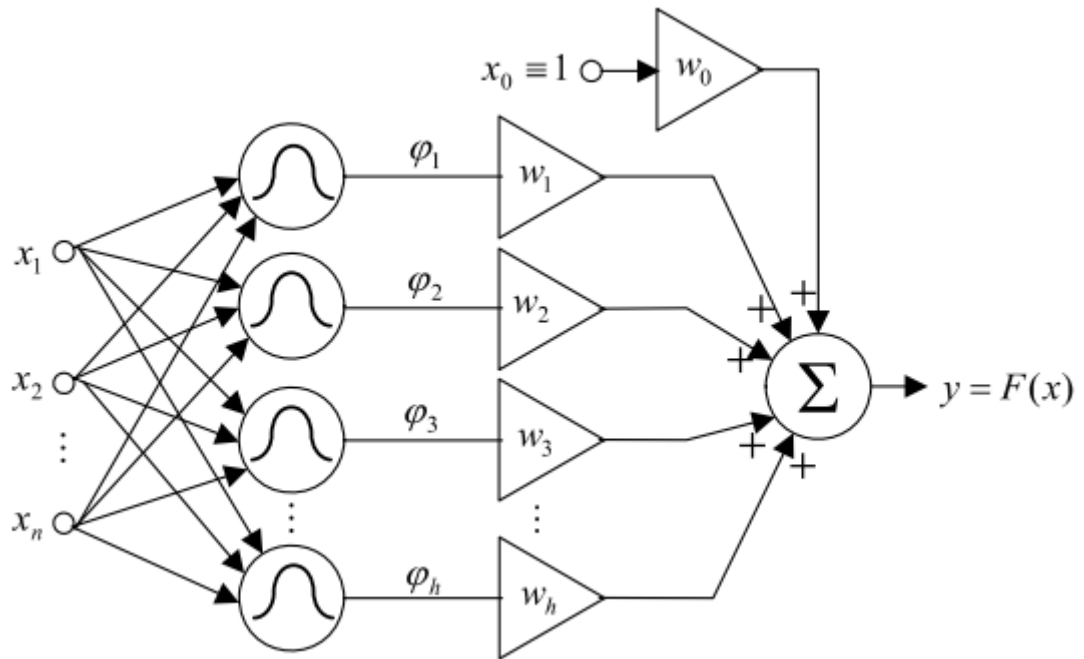


Рисунок 4.20 – Радиально-базисная сеть, реализующая отображение

$$y = F(x) = w_0 + \sum_{i=1}^h w_i \varphi_i(x)$$

Кроме наиболее популярного гауссиана:

$$\varphi(x) = \Phi_1(\|x - c\|, \sigma) = \Phi_1(r, \sigma) = e^{-\left(\frac{r^2}{\sigma^2}\right)}, \quad (4.45)$$

в радиально-базисных сетях используются и иные функции, например,

> мультикватричная $\Phi_2(r, \sigma) = \left(r^2 + \sigma^2\right)^{\frac{1}{2}}, \quad (4.46)$

> обратная мультикватричная $\Phi_3(r, \sigma) = \left(r^2 + \sigma^2\right)^{-\frac{1}{2}}, \quad (4.47)$

> сплайновая $\Phi_4(r, \sigma) = \left(\frac{r}{\sigma}\right)^2 \log\left(\frac{r}{\sigma}\right), \quad (4.48)$

а также

$$\Phi_5(r, \sigma) = \left(\frac{r}{\sigma}\right)^a \log\left(\frac{r}{\sigma}\right), \quad a > 0, \quad (4.49)$$

$$\Phi_6(r, \sigma) = \left(\frac{r}{\sigma}\right)^a e^{\left(-\frac{r}{\sigma}\right)}, \quad (4.50)$$

$$\Phi_7(r, \sigma) = \left(1 + \frac{r}{\sigma}\right)^{-2}, \quad (4.51)$$

$$\Phi_8(r, \sigma) = \left(1 + \frac{r}{\sigma}\right)^{-1}, \quad (4.52)$$

$$\Phi_9(r, \sigma) = \left(\frac{r}{\sigma}\right)^2 + a_1 \left(\frac{r}{\sigma}\right) + a_2, \quad (4.53)$$

$$\Phi_{10}(r, \sigma) = \cos\left(\frac{r}{\sigma}\right), \quad (4.54)$$

$$\Phi_{11}(r, \sigma) = 1 - \left(\frac{r}{\sigma}\right), \quad (4.55)$$

и хотя не все они имеют колоколообразную форму, например (4.46), (4.48), (4.51), их применение в задачах идентификации, моделирования, прогнозирования, распознавания образов, кластеризации и нейроуправления [2] оказалось вполне оправданным.

Как отмечалось выше, наиболее широкое распространение получила функция активации (4.45), расширить возможности которой можно, используя многомерный гауссиан:

$$\varphi(x) = \Phi(\|x - c\|, \Sigma) = e^{-(x-c)^T \Sigma^{-1} (x-c)} = e^{\left(-\|x-c\|_{\Sigma^{-1}}^2\right)}, \quad (4.56)$$

где ковариационная матрица Σ определяет форму, размер и ориентацию так называемого рецепторного поля радиально-базисной функции. При $\Sigma = \sigma^2 I$ (здесь I – $(n \cdot n)$ единичная матрица) рецепторное поле представляет гиперсферу с центром c и радиусом σ ; при $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2)$ – это гиперэллипсоид, чьи оси совпадают с осями входного пространства и имеют длину $2\sigma_i$ по i -й оси, и, наконец, при Σ – недиагональной положительно определенной матрице:

$$\Sigma = Q^T \Lambda Q, \quad (4.57)$$

матрица собственных значений Λ определяет форму и размер рецепторного поля, а ортогональная матрица вращения Q – его ориентацию.

В качестве простейшего, но характерного примера, демонстрирующего возможности радиально-базисных сетей, можно рассмотреть задачу интерполяции многомерной функции n переменных, заданной на N точках $x(1), x(2), \dots, x(k), \dots, x(N)$. При этом известны соответствующие значения этой функции $d(1), \dots, d(k), \dots, d(N)$. Задача состоит в нахождении отображения $F : R^n \rightarrow R^1$, удовлетворяющего условиям интерполяции:

$$F(x(k)) = d(k), \quad k = 1, 2, \dots, N. \quad (4.58)$$

Радиально - базисная техника в данном случае состоит в выборе такой функции F :

$$F(x) = \sum_{k=1}^N w_k \Phi_k(\|x - c\|, \sigma_k) = \sum_{k=1}^N w_k \Phi_k(\|x - x(k)\|, \sigma_k), \quad (4.59)$$

где $\Phi(\bullet)$ – радиально-базисная функция, центрами которой являются точки $x(k)$. С учетом (4.58), (4.59) можно записать систему линейных уравнений для нахождения коэффициентов (синаптических весов) w_k :

$$\begin{pmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1N} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{N1} & \Phi_{N2} & \cdots & \Phi_{NN} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{pmatrix}, \quad (4.60)$$

где $\Phi_{kj} = \Phi(\|x(j) - x(k)\|, \sigma_k)$, $k = 1, \dots, N$; $j = 1, \dots, N$.

Обозначив $d = (d(1), d(2), \dots, d(N))^T$, $w = (w_1, w_2, \dots, w_N)^T$, $\Phi = \{\Phi_{kj}\}$, несложно переписать (4.60) в виде:

$$\Phi w = d, \quad (4.61)$$

откуда

$$w = \Phi^+ d, \quad (4.62)$$

где Φ^+ – матрица, псевдообратная к Φ .

Заметим также, что при $N > h$, с использованием стандартной техники метода наименьших квадратов несложно решать задачи нелинейного регрессионного анализа.

Как уже отмечалось, радиально-базисные сети подобно многослойным ИНС являются универсальными аппроксиматорами, однако в силу того, что в них присутствует только один нелинейный скрытый слой, а

настраиваются параметры линейного выходного слоя, для их обучения могут быть использованы стандартные процедуры, обладающие высоким быстродействием и фильтрующими свойствами, что крайне важно в задачах обработки «зашумленных» наблюдений.

Вместе с тем объединение преимуществ многослойных и радиально-базисных сетей может привести к весьма интересным результатам. Так, в [17, 18] предложены архитектуры многослойных радиально-базисных сетей, подобные многослойным персептронам (см. рис. 4.18), где в качестве отдельных нейронов используются радиально - базисные сети с n входами и одним выходом (см. рис. 4.20). По сравнению с многослойными персептронами они обладают более высокой скоростью обучения, не страдая при этом от «проклятия размерности», ограничивающего возможности обычных радиально - базисных нейронных сетей.

4.6. Обучение нейронных сетей

Основным свойством биологического мозга является его способность к обучению, а поскольку искусственная нейронная сеть является моделью мозга, понятие «обучение» является также ключевым в теории ИНС.

Математическими проблемами, связанными с процессами обучения, занимается направление в общей теории искусственных нейронных сетей, получившее название «нейроматематика» [1–4].

С позиции нейроматематики процесс обучения рассматривается как адаптация параметров, а возможно, и архитектуры сети для решения поставленной задачи путем оптимизации принятого критерия качества. Такая формулировка является общепринятой и неявно предполагает, что в основе нейроматематики лежат методы оптимизации и идентификации.

Обычно полагается, что процесс обучения имеет перманентный характер и с течением времени сеть улучшает свои характеристики, постепенно «приближаясь» к оптимальному решению поставленной задачи.

Тип и характер обучения определяются прежде всего объемом априорной и текущей информации о среде, в которую «погружена» сеть, а также критерием качества (целевой функцией), характеризующим степень соответствия нейросети решаемой ею задачи. Информация о внешней среде задана, как правило, в виде обучающей выборки образов или примеров,

обрабатывая которую, сеть извлекает сведения, необходимые для получения искомого решения. Именно характер и объем этой информации определяют как тип обучения, так и конкретный алгоритм.

4.6.1. Основные парадигмы и правила обучения

Наиболее популярной и очевидной к настоящему времени является парадигма обучения «с учителем», схематически представленная на рис. 4.21.

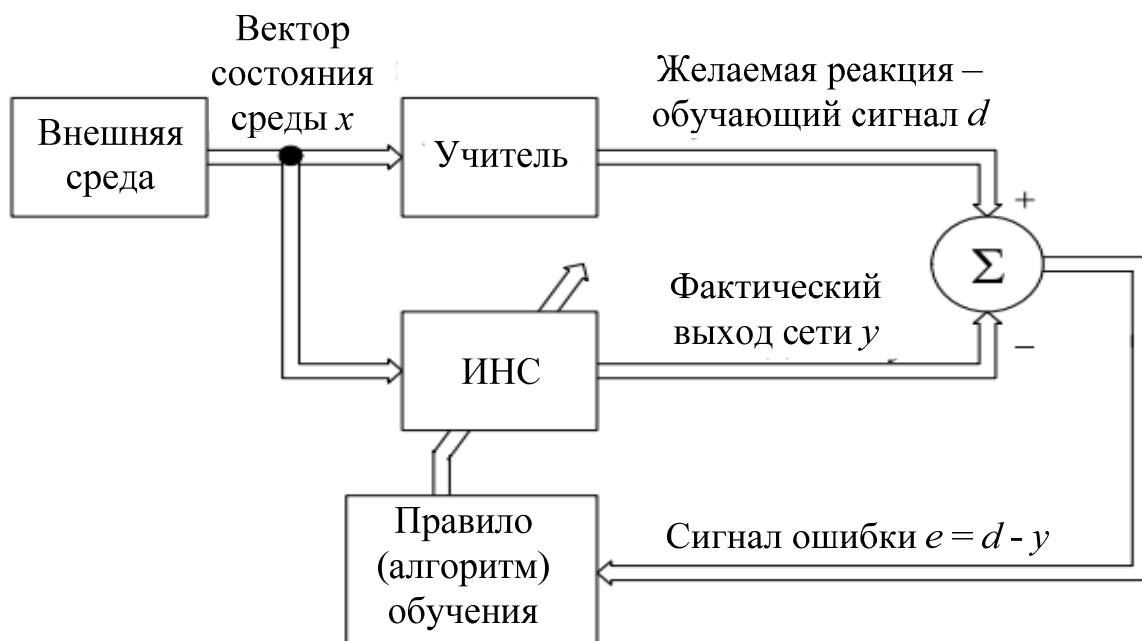


Рисунок 4.21 – Схема обучения с учителем

В данной схеме «учителю» известна информация о внешней среде, заданная в виде последовательности или пакета входных векторов x , а также «правильная реакция» на эти сигналы, представленная в виде обучающего сигнала d . Естественно, что реакция необученной сети y отличается от «правильной» реакции учителя, в результате чего возникает ошибка $e = d - y$. В процессе обучения необходимо так настроить параметры ИНС, чтобы некоторая скалярная функция от ошибки $E(e)$ (критерий качества) достигла своего минимального значения. Обученной считается сеть, которая в некотором, как правило, статистическом смысле повторяет реакцию учителя. Поскольку информация о внешней среде обычно имеет нестационарный характер, процесс обучения идет непрерывно, для чего используются те или иные рекуррентные процедуры.

Альтернативой этой парадигме является обучение «без учителя», или самообучение, когда правильная реакция на сигналы внешней среды неизвестна. Процесс самообучения схематически представлен на рис. 4.22.

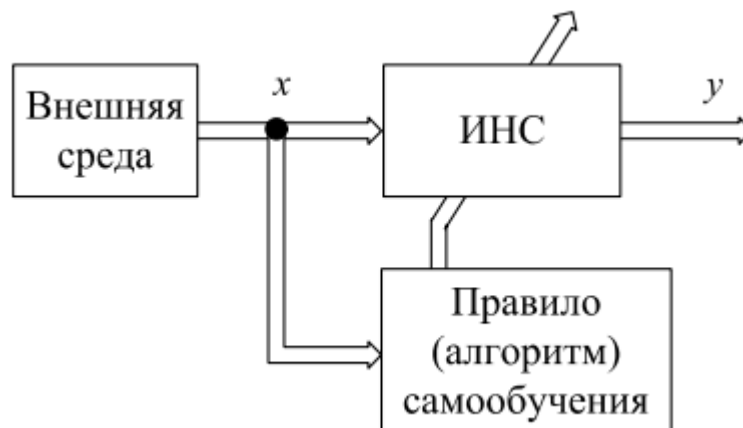


Рисунок 4.22 – Схема самообучения

Сети, реализующие парадигму самообучения, предназначены, как правило, для анализа внутренней латентной структуры входной информации и решают задачи автоматической классификации, кластеризации, факторного анализа, компрессии данных.

Своеобразным компромиссом между двумя этими парадигмами является обучение с *подкреплением* (не путать с обучением с поощрением), при котором доступна лишь косвенная информация о правильной реакции на входной сигнал x . На рис. 4.23 приведена схема процесса обучения с подкреплением. Нейронная сеть производит отображение входной информации x в выходной вектор y в виде $y = F(x)$, однако, поскольку обучающий сигнал d в явном виде не задан, невозможно получить ошибку $e = d - y$, на основании которой происходит обучение. Предполагается, что имеются некоторые априорные знания, позволяющие связать эвристический сигнал подкрепления \tilde{d} с ненаблюдаемым желаемым выходом d с помощью некоторой функции \tilde{F} , отображающей d в \tilde{d} . Обычно эта функция учитывает связь выходных сигналов сети y с наблюдаемыми событиями во внешней среде, для чего в схему обучения вводится дополнительный блок – «критик», отображающий поведение сети в сигнал $\tilde{y} = \tilde{F}(F(x))$. Далее вычисляется эвристическая ошибка $\tilde{e} = \tilde{d} - \tilde{y}$, на основе которой и реализуется процесс обучения.



Рисунок 4.23 – Схема обучения с подкреплением

Процесс обучения с подкреплением разбивается на два относительно независимых этапа: обучение тому, как выходной сигнал сети y влияет на наблюдаемые переменные среды x , то есть восстановление отображения \tilde{F} и, собственно, обучение сети на основе минимизации принятого критерия $E(\tilde{e})$.

Эта парадигма тесно связана с идеями динамического программирования и в теории искусственных нейронных сетей известна как нейродинамическое программирование [19].

Достаточно широкое распространение получила парадигма смешанного обучения, когда часть параметров сети настраивается с помощью обучения с учителем, а другая часть или архитектура в целом – с помощью самообучения. Этот подход получил наибольшее распространение при обучении радиально - базисных ИНС.

С введенными парадигмами тесно связаны правила обучения, лежащие в основе конкретных алгоритмов. С. Хайкин [19] определяет пять основных правил: обучение на основе коррекции по ошибке, обучение по Больцману, обучение по Хэббу, обучение памяти и конкурентное обучение.

Правило коррекции по ошибке – типичный случай обучения с учителем, при этом с помощью тех или иных процедур оптимизации и адаптивной идентификации минимизируется априори заданная скалярная целевая функция $E(e)$. С этим правилом связано наибольшее число известных алго-

ритмов обучения, которое к настоящему времени перевалило за сотню.

В основе обучения по Больцману лежат принципы теоретической термодинамики, при этом настройка синаптических весов стохастической сети обеспечивает требуемое (желаемое) распределение вероятностей состояний отдельных нейронов. В какой-то мере обучение, по Больцману, может рассматриваться как распространение идей обучения с учителем на стохастический случай.

С самообучением тесно связаны правило Хэбба и обучение памяти, в основе которых лежит нейрофизиологический постулат, гласящий о том, что если нейроны с обеих сторон синапса находятся в возбужденном состоянии, то сила связи между ними возрастает (увеличивается синаптический вес), и, наоборот, если соседние нейроны находятся в разных состояниях, то связь между ними ослабевает.

В конкурентном обучении могут быть реализованы все описанные парадигмы, при этом его отличительной особенностью является процесс «соревнования» нейронов выходного слоя по принципу *«winner takes all»*, то есть возбуждается только один выходной нейрон – «победитель». Наиболее яркими примерами сетей, использующих это правило, являются сети адаптивного резонанса (*ART*) и самоорганизующиеся карты (*SOM*).

Более подробно правила обучения, порождаемые ими алгоритмы и соответствующие им архитектуры нейросетей будут рассмотрены в последующих разделах.

4.6.2. Задачи обучения

В настоящем подразделе рассмотрены некоторые прикладные инженерные задачи, решаемые нейросетями, обученными соответствующим образом. Естественно, что этот список далеко не полон и включает только проблемы, которые традиционно интересуют специалистов в области компьютерных наук, инженерии и управления.

4.6.2.1. Распознавание образов

Наряду с обучением, распознавание является одной из основных функций биологического мозга. Получая данные из окружающего мира с помощью биологических сенсоров, мозг достаточно просто распознает источник данных и выделяет из него требуемую информацию. Так, человек

без особых проблем узнает знакомое лицо, хотя видел его давно, и оно успело измениться, голос, искаженный телефонными помехами, город, в котором не был много лет и т.д. Это узнавание и есть результат обучения, причем в идеальном случае нейросеть должна узнавать предъявляемые ей образы не хуже, чем это делает живой организм.

Формально распознавание образов определяется как процесс, в результате которого получаемый образ (сигнал) относится к одному из априори назначенных классов (категорий). В процессе обучения нейросети предъявляются различные образы с известной классификацией (обучающая выборка), а в результате сеть должна распознать объект, который ранее не предъявлялся, но который принадлежит той же совокупности, что и обучающая выборка. Задача распознавания статистична по своей природе, при этом образы представляются случайными векторами в многомерном пространстве признаков, а результат обучения состоит в построении решающих гиперповерхностей, разделяющих «в среднем» пространство признаков на соответствующие классы.

Как правило, нейросетевые распознающие системы состоят из двух частей. Первая – это самообучающаяся сеть, решающая задачу селекции и выделения признаков, а вторая – это сеть, настраиваемая с помощью внешнего обучающего сигнала, содержащего информацию о принадлежности образов обучающей выборки определенным классам. В принципе такая последовательность решения задачи характерна для большинства распознающих систем: сначала понижение размерности вектора признаков с помощью, например, традиционного преобразования Карунена – Лоэва, а затем, собственно, построение разделяющих гиперповерхностей. Преимущество нейросетевого подхода перед другими методами распознавания образов состоит в том, что нейросети способны восстанавливать разделяющие гиперповерхности сколь угодно сложной формы, не опираясь на гипотезы о компактности или линейной делимости классов.

4.6.2.2. Ассоциация и кластеризация

Биологическим системам, наряду с обучением и распознаванием, присуща также способность к ассоциациям, то есть восстановлению (воспоминанию) ранее предъявленных образов по некоторым косвенным стимулам. Любому человеку знакомы случаи, когда какой-то случайный звук

или запах вызывал в воображении сложные зрительные образы.

В нейросетях ассоциации реализуются в двух формах: автоассоциация и гетероассоциация. В случае автоассоциации сеть обрабатывает множество последовательно предъявляемых ей образов, причем эти образы могут быть зашумлены либо искажены. Выделяя и запоминая основные признаки предъявляемых образов, сеть приобретает способность восстанавливать (вспоминать) ранее показанные ей примеры. Гетероассоциация отличается тем, что произвольное множество входных образов связывается (ассоциируется) с произвольным множеством выходных примеров. Основное отличие между этими формами состоит в том, что автоассоциация реализуется на основе парадигмы самообучения, а гетероассоциация – обучения с учителем.

Пусть $x(k)$ – входной образ-вектор (стимул), в общем случае произвольно взятый из обучающей выборки и предъявленный сети ассоциативной памяти, а $y(k)$ – запомненный (выходной) образ-вектор. Ассоциация образов, выполняемая сетью, описывается отношением $x(k) \rightarrow y(k)$, $k = 1, 2, \dots, N$, где N – число образов, запомненное ИНС. Входной образ $x(k)$ действует как стимул, вызывающий отклик $y(k)$, а в последствии служит ключом к восстановлению.

В автоассоциативной памяти $y(k) = x(k)$, то есть входное и выходное пространства сети совпадают. В гетероассоциативной памяти $y(k) \neq x(k)$, при этом размерности пространств, как правило, также не совпадают.

В работе ассоциативных нейросетей выделяют две фазы: накопления, которая соответствует периоду обучения, и восстановления, которая предполагает воспоминание запомненного образа после предъявления зашумленного или искаженного стимула.

Число N образов, накопленных в ассоциативной памяти, является мерой емкости сети. При проектировании таких сетей основной проблемой является выбор и обеспечение максимальной емкости, выраженной как отношение числа запоминаемых примеров N к общему числу нейронов сети при минимальном числе некорректно восстанавливаемых образов.

К проблеме автоассоциации тесно примыкает задача кластеризации (автоматической классификации), когда сеть, анализируя обучающую выборку $x(k)$, размещает «похожие» образы по группам – кластерам. Предъявляемый зашумленный образ, ранее не показанный сети, по ассоциации с

уже запомненными, должен быть отнесен к «родному кластеру». Сети, реализующие кластеризацию образов, используются обычно для сжатия данных и извлечения из них знаний.

4.6.2.3. Аппроксимация функций

С проблемой обучения тесно связана достаточно часто возникающая на практике задача аппроксимации функций, заданных на некотором множестве точек.

Рассмотрим нелинейное отображение «вход-выход», описываемое функциональным соотношением:

$$d = f(x), \quad (4.63)$$

где d и x – $(m \times 1)$ и $(n \times 1)$ – векторы выходов и входов соответственно;

$f(\bullet)$ – неизвестная вектор-функция, которую необходимо оценить с помощью заданной обучающей выборки $\{x(k), d(k)\}, k = 1, 2, \dots, N$.

Задача обучения аппроксимирующей нейросети состоит в нахождении функции $F(x)$ в некотором смысле достаточно близкой к $f(x)$, такой что

$$\|F(x) - f(x)\| \leq \varepsilon \text{ для всех } x(k), k = 1, 2, \dots, N, \quad (4.64)$$

где $F(x)$ – отображение, реализуемое сетью; ε – малое положительное число.

Если объем выборки N достаточно велик, а сеть имеет достаточное число настраиваемых синаптических весов, ошибка аппроксимации e может быть сделана сколь угодно малой, хотя здесь есть опасность превращения сети из аппроксимирующей в интерполирующую, рассмотренную в третьем разделе.

Несложно видеть, что проблема аппроксимации в данном контексте полностью совпадает с задачей обучения с учителем, где последовательность $x(k)$ выполняет функцию входного сигнала ИНС, а $d(k)$ – обучающего сигнала.

Способность нейросетей аппроксимировать неизвестные отображения «вход - выход» находит два важнейших приложения в задачах интеллектуального управления [2]. Первое из них – идентификация объектов управления, или эмуляция – в терминах нейроуправления. Схема системы идентификации (эмуляции) приведена на рис. 4.24, при этом предполагается, что многомерный статический объект описывается соотношением

(4.63), а нейронная сеть, включенная параллельно объекту, обучается в реальном времени, «подгоняя» свои выходные сигналы к выходам реального объекта.

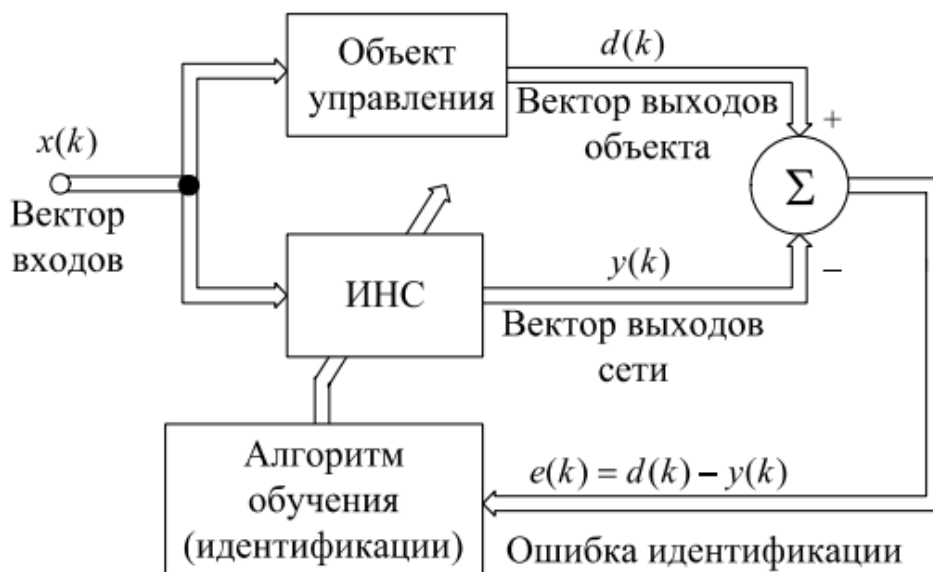


Рисунок 4.24 – Схема системы идентификации

Второе приложение – это обратное моделирование, используемое в некоторых адаптивных системах управления и состоящее в том, что для объекта управления (4.63) требуется построить «обратную систему», которая генерирует вектор $x(k)$ как отклик на входной сигнал $d(k)$. В общем виде обратная система имеет форму:

$$x = f^{-1}(d), \quad (4.65)$$

однако, поскольку функция f либо не известна, либо слишком сложна, разумным выходом представляется использование ИНС в качестве обратной модели так, как это показано на рис. 4.25.

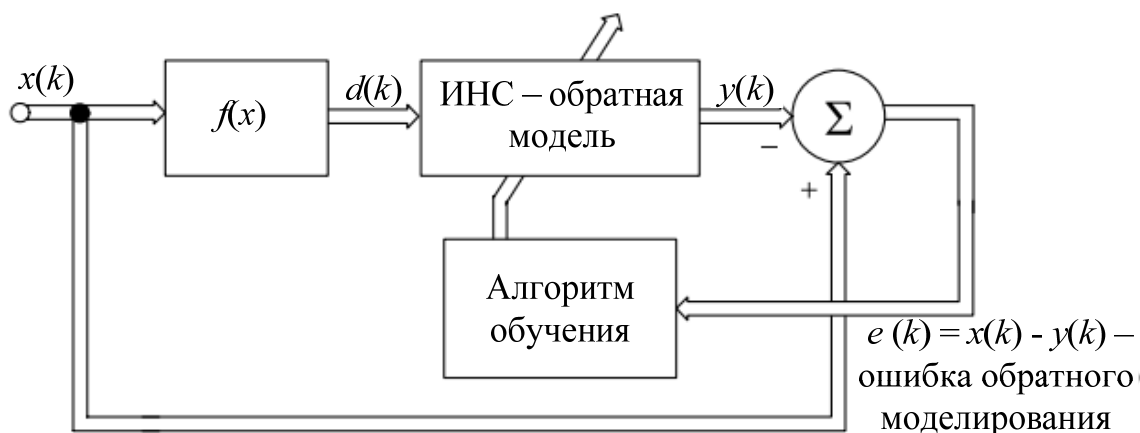


Рисунок 4.25 – Схема обратного моделирования

В этой схеме роли сигналов $x(k)$ и $d(k)$ поменялись: вектор $d(k)$ используется как вход сети, а $x(k)$ – как желаемый отклик (обучающий сигнал). Подобно системе идентификации сигнал ошибки $e(k) = x(k) - y(k)$ используется для обучения ИНС.

4.6.2.4. Управление и оптимизация

Управление объектами в условиях структурной и параметрической неопределенности – еще одна задача, связанная с обучением нейросетей. На рис. 4.26 приведена схема управления с обратной связью, при этом предполагается, что в распоряжении проектировщика системы управления нет информации ни о структуре нелинейного объекта, ни, тем более, о его параметрах.

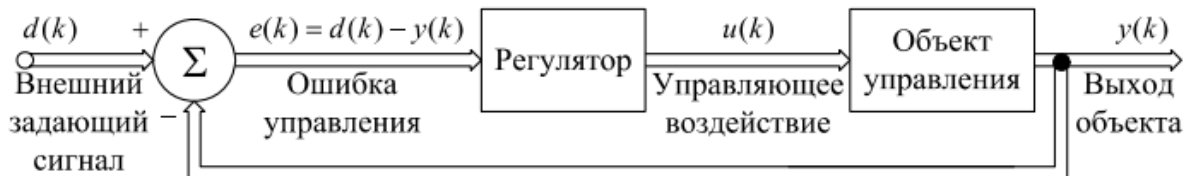


Рисунок 4.26 – Система управления с обратной связью

Целью управления является выработка управляющих сигналов $u(k)$, обеспечивающих устойчивое слежение выхода объекта $y(k)$ за внешним задающим сигналом (желаемой траекторией движения) $d(k)$.

Поскольку об объекте управления ничего не известно, в качестве регулятора можно использовать нейросеть, входом которой является вектор ошибок управления $e(k) = d(k) - y(k)$, а выходом – сигнал управления $u(k)$, подаваемый на объект.

Синтез оптимального управления связан с оценкой якобиана объекта

$J = \left\{ \frac{\partial y_j}{\partial u_i} \right\}$, для определения которого опять-таки могут быть использованы

аппроксимирующие свойства обучаемой ИНС.

В теории адаптивного управления сформировалось два основных направления. Первое – непрямой или идентификационный подход, при котором в схему вводится настраиваемая модель, обучаемая в темпе с процессом управления и параметры которой в линейном случае являются оценками элементов матрицы-якобиана. В нашем случае в систему управ-

ления дополнительно вводится нейросеть - эмулятор, оценивающая частные производные $\frac{\partial y_j}{\partial u_i}$, которые далее используются нейросетью - регулятором.

Альтернативой идентификационному является прямой подход к синтезу регулятора, при котором предполагается, что проектировщику доступна информация о знаках частных производных $\frac{\partial y_j}{\partial u_i}$. В прямой системе управления присутствует одна нейросеть-регулятор, обучаемая с помощью алгоритмов, использующих только знаки обрабатываемых сигналов.

Вплотную к задаче управления примыкает задача оптимизации, когда требуется определить экстремум многомерной неявно заданной функции при наличии ограничений. И хотя для решения проблемы оптимизации спроектированы специальные архитектуры нейросетей, достаточно большой класс задач может быть решен в рамках систем нейроуправления, когда в качестве целевой функции используется лагранжиан, учитывающий всевозможные ограничения, накладываемые на переменные объекта.

4.6.2.5. Фильтрация, сглаживание, прогнозирование

Системы обработки «зашумленных» сигналов в условиях неопределенности в настоящее время находят широкое применение в самых разнообразных приложениях. Собственно, понятие «обработка сигналов» традиционно включает в себя три задачи: фильтрацию, сглаживание и прогнозирование. Если в распоряжении исследователя есть выборка «загрязненных» наблюдений: $x(1), x(2), \dots, x(k), \dots, x(N)$, то задача фильтрации сводится к нахождению наилучшей оценки процесса $\hat{x}(N|N)$ в момент времени N по информации о N наблюдениях, сглаживания – оценки $\hat{x}(k|N)$ при $k < N$ и прогнозирования – $\hat{x}(N+l|N)$ при $N+l > N$, где l – горизонт упреждения.

Последние годы внимание исследователей привлечено еще к одной нетрадиционной задаче обработки – «слепой» сепарации и идентификации сигналов. Предполагается, что имеется множество неизвестных источников сигналов $\{u_i(k)\}_{i=1}^n$, которые не зависят друг от друга. Сенсоры воспринимают эти сигналы не покомпонентно, а в смеси, представляющей со-

бой неизвестную линейную комбинацию $x(k) = Au(k)$ так, как это показано на рис. 4.27.

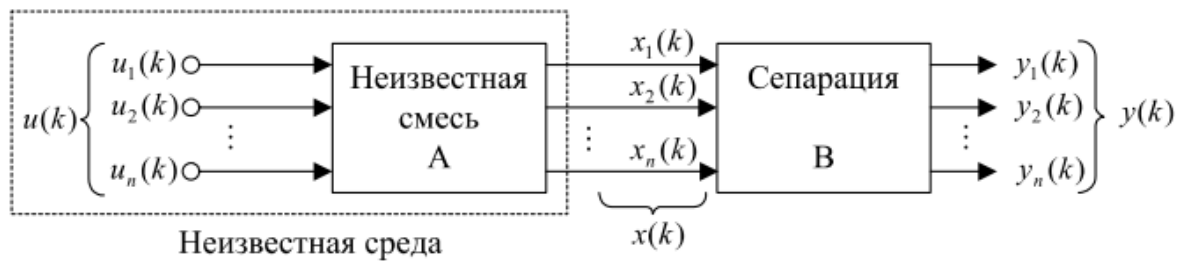


Рисунок 4.27 – Схема слепой сепарации

Задача сводится к восстановлению вектора $y(k) \approx u(k)$ по данным наблюдений вектора $x(k)$ при неизвестной $(n \times n)$ – матрице A . Несложно видеть, что первые три задачи весьма близки к проблеме идентификации, а задача слепой сепарации практически совпадает с задачей обратного моделирования и сводится к нахождению оператора сепарации $B = A^{-1}$. Естественно, что применение ИНС для решения этих задач принципиальных затруднений не вызывает.

Остановимся кратко на задаче текущего прогнозирования стохастической последовательности $x(k)$ по данным о ее предыстории $x(k-1), x(k-2), \dots$. Проблема сводится к нахождению оценки $x(k) = F(x(k-1), (x(k-2), \dots, x(k-p)))$ в реальном времени в темпе с поступлением данных. В линейном случае эта задача хорошо исследована и успешно может быть решена с помощью адаптивных прогнозирующих авторегрессионных моделей [2]. Для построения же нелинейных прогнозов наиболее целесообразным представляется применение ИНС, например, так, как это показано на рис. 4.28.

В настоящее время нейросетевые прогнозирующие модели успешно используются для решения широкого круга задач науки, техники, экономики.

4.6.3. Теорема об обучении перцептрона

Обучение сети состоит в подстройке весовых коэффициентов каждого нейрона. Пусть имеется набор пар векторов (x^α, y^α) , $\alpha = 1 \dots p$, называемый *обучающей выборкой*. Будем называть нейронную сеть обученной на данной обучающей выборке, если при подаче на входы сети каждого вектора x^α на выходах всякий раз получается соответствующий вектор y^α .

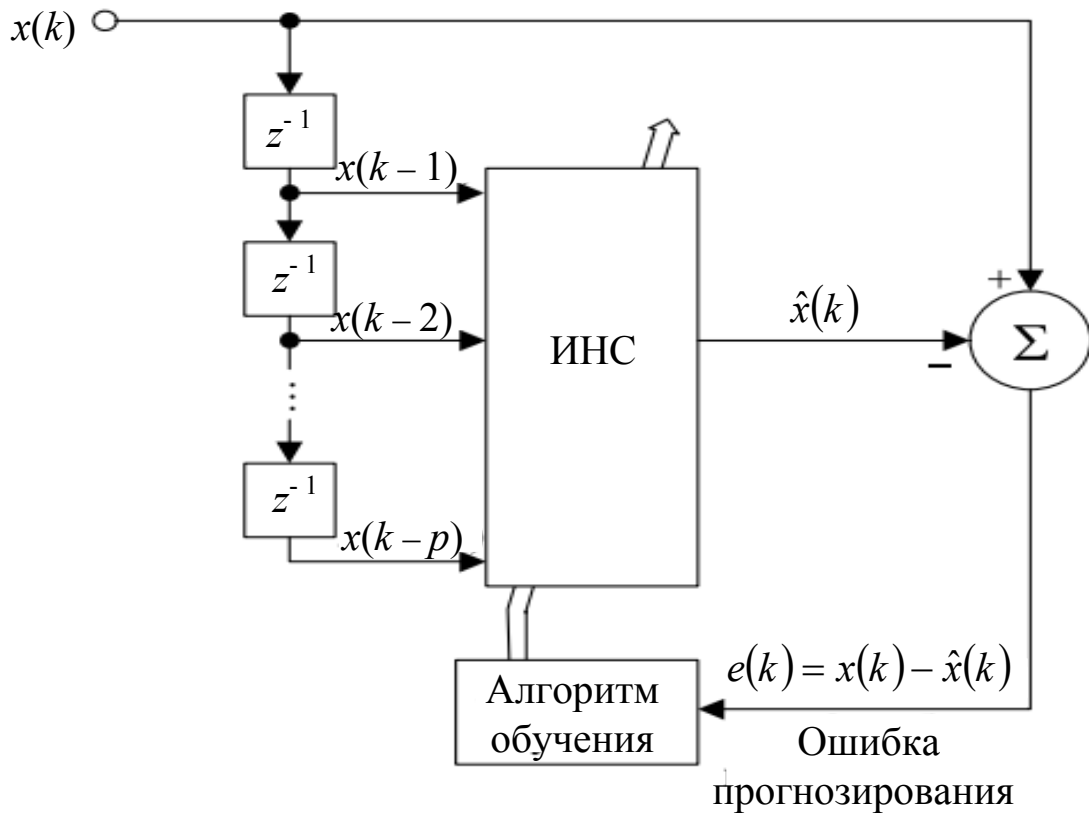


Рисунок 4.28 – Схема нейросетевого прогнозирования

Предложенный Ф. Розенблаттом метод обучения состоит в итерационной подстройке матрицы весов, последовательно уменьшающей ошибку в выходных векторах. Алгоритм включает несколько шагов.

Шаг 0. Начальные значения весов всех нейронов $w(t=0)$ полагаются случайными.

Шаг 1. Сети предъявляется входной образ \underline{x}^α , в результате формируется выходной образ $\tilde{y}^\alpha \neq y^\alpha$.

Шаг 2. Вычисляется вектор ошибки $\delta^\alpha = (y^\alpha - \tilde{y}^\alpha)$, выполняемый сетью на выходе. Дальнейшая идея состоит в том, что изменение вектора весовых коэффициентов в области малых ошибок должно быть пропорционально ошибке на выходе и равно нулю, если ошибка равна нулю.

Шаг 3. Вектор весов модифицируется по следующей формуле:
 $w(t + \Delta t) = w(t) + \eta \cdot x^\alpha \cdot (\delta^\alpha)^T$. Здесь $0 < \eta < 1$ – темп обучения.

Шаг 4. Шаги 1–3 повторяются для всех обучающих векторов. Один цикл последовательного предъявления всей выборки называется *эпохой*.

Обучение завершается по истечении нескольких эпох, когда итерации сойдутся, то есть вектор весов перестает изменяться, или когда полная просуммированная по всем векторам абсолютная ошибка станет меньше некоторого малого значения.

Используемая на шаге 3 формула учитывает следующие обстоятельства:

а) модифицируются только компоненты матрицы весов, отвечающие ненулевым значениям входов;

б) знак приращения веса соответствует знаку ошибки, то есть положительная ошибка ($\delta > 0$, значение выхода меньше требуемого) приводит к усилению связи;

в) обучение каждого нейрона происходит независимо от обучения остальных нейронов, что соответствует важному с биологической точки зрения принципу *локальности* обучения.

Данный метод обучения был назван Ф. Розенблаттом «методом коррекции с обратной передачей сигнала ошибки». Позднее более широко стало известно название « δ - правило». Представленный алгоритм относится к широкому классу алгоритмов обучения *с учителем*, поскольку известны как входные векторы, так и требуемые значения выходных векторов (имеется учитель, способный оценить правильность ответа ученика).

Доказанная Розенблаттом теорема о сходимости обучения по дельта - правилу свидетельствует о том, что персептрон способен обучиться любому обучающему набору, который *он способен представить*.

4.6.4. Линейная делимость

Каждый нейрон персептрона является формальным пороговым элементом, принимающим единичные значения в случае, если суммарный взвешенный вход больше некоторого порогового значения:

$$y_i = \begin{cases} 1, & \sum_i w_{ij}x_i > \theta_j, \\ 0, & \sum_i w_{ij}x_i \leq \theta_j. \end{cases} \quad (4.66)$$

Таким образом, при заданных значениях весов и порогов, нейрон

имеет определенное значение выходной активности для каждого возможного вектора входов. Множество входных векторов, при которых нейрон активен ($y = 1$), отделено от множества векторов, на которых нейрон пассивен ($y = 0$) *гиперплоскостью*, уравнение которой есть суть:

$$\sum_i w_{ij}x_i - \theta_j = 0. \quad (4.67)$$

Следовательно, нейрон способен *отделить* (иметь различный выход) только такие два множества векторов входов, для которых имеется гиперплоскость, отсекающая одно множество от другого. Такие множества называют *линейно разделимыми*. Линейная неразделимость множества аргументов, отвечающих различным значениям функции, означает, что функция «исключающее или», столь широко используемая в логических устройствах, не может быть представлена формальным нейроном.

Столь скромные возможности нейрона и послужили основой для критики перцептронного направления Ф. Розенблатта со стороны М. Минского и С. Пейперта [15, 16].

При возрастании числа аргументов ситуация еще более катастрофична: *относительное* число функций, которые обладают свойством линейной разделимости, резко уменьшается. А значит, и резко сужается класс функций, который может быть реализован перцептроном (так называемый класс функций, обладающий свойством перцептронной представляемости). Следует отметить, что в начале 70-х годов прошлого века, это ограничение было преодолено путем введения нескольких слоев нейронов, однако критическое отношение к классическому перцептрону сильно заморозило общий круг интереса научных исследований в области искусственных нейронных сетей.

4.6.5. Обучение методом обратного распространения ошибок

Для обучения многослойной сети в 1986 г. Руммельхартом и Хинтоном [20] был предложен алгоритм обратного распространения ошибок (error back propagation). Многочисленные публикации о промышленных применениях многослойных сетей с этим алгоритмом обучения подтвердили его принципиальную работоспособность на практике.

В начале возникает резонный вопрос: а почему для обучения многослойного перцептрона нельзя применить уже известное δ - правило Розен -

блатта? Ответ состоит в том, что для применения метода Розенблатта необходимо знать не только текущие выходы нейронов y , но и требуемые *правильные* значения Y . В случае многослойной сети эти правильные значения имеются только для нейронов *выходного* слоя. Требуемые значения выходов для нейронов скрытых слоев неизвестны, что и ограничивает применение δ - правила.

Основная идея обратного распространения состоит в том, как получить оценку ошибки для нейронов скрытых слоев. Заметим, что *известные* ошибки, имеющие место в нейронах выходного слоя, возникают вследствие *неизвестных* пока ошибок нейронов скрытых слоев. Чем больше значение синаптической связи между нейроном скрытого слоя и выходным нейроном, тем сильнее ошибка первого влияет на ошибку второго. Следовательно, оценку ошибки элементов скрытых слоев можно получить как взвешенную сумму ошибок последующих слоев. При обучении информация распространяется от низших слоев иерархии к высшим, а оценки ошибок, делаемые сетью, – в обратном направлении, что и отражено в названии метода.

Перейдем к подробному рассмотрению этого алгоритма. Для упрощения обозначений ограничимся ситуацией, когда сеть имеет только один скрытый слой. Матрицу весовых коэффициентов от входов к скрытому слою обозначим W , а матрицу весов, соединяющих скрытый и выходной слой, – как V . Для индексов примем следующие обозначения: входы будем нумеровать только индексом i , элементы скрытого слоя – индексом j , а выходы, соответственно, индексом k .

Пусть сеть обучается на выборке (X^α, Y^α) , $\alpha = 1 \dots p$. Активности нейронов будем обозначать малыми буквами y с соответствующим индексом, а суммарные взвешенные входы нейронов – малыми буквами x .

Общая структура алгоритма аналогична рассмотренной выше, с исключением формул подстройки весов.

Алгоритм обратного распространения ошибки

Шаг 0. Начальные значения весов всех нейронов всех слоев $V(t = 0)$ и $W(t = 0)$ полагаются случайными числами;

Шаг 1. Сети предъявляется входной образ X^α , в результате формируется выходной образ $y \neq Y^\alpha$. При этом нейроны последовательно от слоя к слою функционируют по следующим формулам:

– скрытый слой

$$x_j = \sum_i W_{ij} X_i^\alpha; \quad y_i = f(x_j); \quad (4.68)$$

– выходной слой

$$x_k = \sum_j W_{jk} Y_j; \quad y_k = f(x_k), \quad (4.69)$$

где $f(x)$ – сигмоидальная функция, определяемая по формуле (4.4).

Шаг 2. Функционал квадратичной ошибки сети для данного входного образа имеет вид:

$$\varepsilon = \frac{1}{2 \cdot \sum_k (y_k - Y_k^\alpha)^2}. \quad (4.70)$$

Данный функционал подлежит минимизации. Классический градиентный метод оптимизации состоит в итерационном уточнении аргумента согласно формуле:

$$V_{jk}(t+1) = v_{jk}(t) - h \cdot \frac{\partial \varepsilon}{\partial V_{jk}}. \quad (4.71)$$

Функция ошибки в явном виде не содержит зависимости от веса V_{jk} , поэтому воспользуемся формулами неявного дифференцирования сложной функции:

$$\frac{\partial \varepsilon}{\partial y_k} = \delta_k = (y_k - Y_k^\alpha); \quad (4.72)$$

$$\frac{\partial \varepsilon}{\partial x_k} = \frac{\partial \varepsilon}{\partial y_k} \cdot \frac{\partial y_k}{\partial x_k} = \delta_k \cdot y_k \cdot (1 - y_k); \quad (4.73)$$

$$\frac{\partial \varepsilon}{\partial V_{jk}} = \frac{\partial \varepsilon}{\partial y_k} \cdot \frac{\partial y_k}{\partial x_k} \cdot \frac{\partial x_k}{\partial V_{jk}} = \delta_k \cdot y_k \cdot (1 - y_k) \cdot y_j. \quad (4.74)$$

Здесь учтено полезное свойство сигмоидальной функции $f(x)$: ее производная выражается только через само значение функции, $f'(x) = f(1 - f)$. Таким образом, все необходимые величины для подстройки весов выходного слоя V получены.

Шаг 3. На этом шаге выполняется подстройка весов скрытого слоя. Градиентный метод по-прежнему дает:

$$W_{ij}(t+1) = W_{ij}(t) - h \cdot \frac{\partial \varepsilon}{\partial W_{ij}}. \quad (4.75)$$

Вычисления производных выполняются по тем же формулам, за исключением некоторого усложнения формулы для ошибки δ_j .

$$\frac{\partial \varepsilon}{\partial x_k} = \frac{\partial \varepsilon}{\partial y_k} \cdot \frac{\partial y_k}{\partial x_k} = \delta_k \cdot y_k \cdot (1 - y_k); \quad (4.76)$$

$$\frac{\partial \varepsilon}{\partial y_j} = \delta_j = \sum_k \frac{\partial \varepsilon}{\partial x_k} \cdot \frac{\partial x_k}{\partial y_j} = \sum_k \delta_k \cdot y_k (1 - y_k) \cdot V_{jk}; \quad (4.77)$$

$$\begin{aligned} \frac{\partial \varepsilon}{\partial W_{ij}} &= \frac{\partial \varepsilon}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_j} \cdot \frac{\partial x_j}{\partial W_{ij}} = \delta_j \cdot y_j \cdot (1 - y_j) \cdot X_i^\alpha = \\ &= \left[\sum_k \delta_k \cdot y_k (1 - y_k) \cdot V_{jk} \right] \cdot \left[y_j \cdot (1 - y_j) \cdot X_i^\alpha \right] \end{aligned} \quad (4.78)$$

При вычислении δ_j здесь и был применен принцип обратного распространения ошибки: частные производные берутся только по переменным *последующего* слоя. По полученным формулам модифицируются веса нейронов скрытого слоя. Если в нейронной сети имеется несколько скрытых слоев, процедура обратного распространения применяется последовательно для каждого из них, начиная со слоя, предшествующего выходному, и далее до слоя, следующего за входным. При этом формулы сохраняют свой вид с заменой элементов выходного слоя на элементы соответствующего скрытого слоя.

Шаг 4. Шаги 1–3 повторяются для всех обучающих векторов. Обучение завершается по достижении малой полной ошибки или максимально допустимого числа итераций, как и в методе обучения Розенблатта.

Как видно из описания шагов 2–3, обучение сводится к решению задачи оптимизации функционала ошибки градиентным методом. Весь смысл обратного распространения ошибки состоит в том, что для ее оценки для нейронов скрытых слоев можно принять взвешенную сумму ошибок последующего слоя.

Параметр h имеет смысл темпа обучения и выбирается достаточно малым для сходимости метода. О сходимости необходимо сделать несколько дополнительных замечаний. Во-первых, практика показывает, что сходимость метода обратного распространения весьма медленная. Невысокий темп сходимости является «генетической болезнью» всех градиент-

ных методов, так как локальное направление градиента отнюдь не совпадает с направлением к минимуму. Во - вторых, подстройка весов выполняется независимо для каждой пары образов обучающей выборки. При этом улучшение функционирования на некоторой заданной паре может, вообще говоря, приводить к ухудшению работы на предыдущих образах. В этом смысле *нет* достоверных (кроме весьма обширной практики применения метода) гарантий сходимости.

Исследования показывают, что для представления произвольного функционального отображения, задаваемого обучающей выборкой, достаточно всего *два слоя* нейронов. Однако на практике, в случае сложных функций, использование более чем одного скрытого слоя может давать экономию полного числа нейронов.

В завершение сделаем замечание относительно настройки порогов нейронов. Легко заметить, что порог нейрона может быть сделан эквивалентным дополнительному весу, соединенному с фиктивным входом, равным -1 . Действительно, выбирая $W_0 = \theta$, $x_0 = -1$ и начиная суммирование с нуля, можно рассматривать нейрон с нулевым порогом и одним дополнительным входом:

$$\begin{aligned} y &= f\left(\sum_{i=1}^n W_i x_i - \theta\right) = f\left(\sum_{i=1}^n W_i x_i + (-1) \cdot \theta\right) = \\ &= f\left(\sum_{i=1}^n W_i x_i + W_0 x_0\right) = f\left(\sum_{i=0}^n W_i x_i\right). \end{aligned} \tag{4.79}$$

Все изложенные в алгоритме обратного распространения формулы суммирования по входам начинаются с нулевого индекса.

4.6.6. Алгоритмы самообучения. Правило обучения Хэбба

Настоящий подраздел, в отличие от предыдущих, посвящен процедурам обучения без учителя (самообучения), представляющим собой, по определению Б. Уидроу [14], алгоритмы адаптации синаптических весов в разомкнутом контуре. Наиболее широкое распространение самообучение получило в задачах кластеризации, квантования непрерывного пространства входов, понижения размерности пространства сигналов (нелинейного

факторного анализа), выделения информативных признаков при распознавании образов и т.п.

Самообучение лежит в основе таких получивших широкое распространение ИНС, как самоорганизующиеся карты Кохонена, сети Гроссберга, сети, основанные на теории адаптивного резонанса, конкурентные сети и др. Здесь мы рассмотрим правила обучения отдельных нейронов, а также области их целесообразного использования.

Обучение по Д. Хэббу является наиболее известным, а его история насчитывает более пятидесяти лет [2]. Суть его состоит в том, что если два соседних нейрона, связанных через синаптическую связь, активируются одновременно, то сила этой связи увеличивается; если же эти нейроны активируются асинхронно, то эта сила или уменьшается, либо исключается вообще. Простой алгоритм, реализующий такой механизм обучения, получил название *правила Хэбба*. Рассмотрим его подробно.

Пусть задана обучающая выборка образов ξ^α , $\alpha = 1 \dots p$. Требуется построить процесс получения матрицы связей W , такой, что соответствующая нейронная сеть будет иметь в качестве стационарных состояний образы обучающей выборки (значения порогов нейронов T обычно полагаются равными нулю).

В случае одного обучающего образа правило Хэбба приводит к требуемой матрице:

$$W_{ij} = \xi_i \cdot \xi_j. \quad (4.80)$$

Покажем, что состояние $S = \xi$ является стационарным для сети Хопфилда с указанной матрицей. Действительно, для любой пары нейронов i и j энергия их взаимодействия в состоянии ξ достигает своего минимально возможного значения

$$E_{ij} = -\left(\frac{1}{2}\right) \cdot \xi_i \cdot \xi_j \cdot \xi_i \cdot \xi_j = -\frac{1}{2}. \quad (4.81)$$

При этом E – полная энергия – равна: $E = -(1/2) N^2$, что отвечает глобальному минимуму.

Для запоминания других образов может быть использован итерационный процесс:

$$W_{ij}^{(\alpha)} = W_{ij}^{(\alpha-1)} + \xi_i^{(\alpha)} \cdot \xi_j^{(\alpha)}, \quad W^{(0)} = 0, \quad \alpha = 1 \dots p,$$

который приводит к полной матрице связей в формуле Хэбба:

$$W_{ij} = \sum_{\alpha=1}^p \xi_i^{(\alpha)} \cdot \xi_j^{(\alpha)}. \quad (4.82)$$

Устойчивость совокупности образов не столь очевидна, как в случае одного образа. Ряд исследований показывает, что нейронная сеть, обученная по правилу Хэбба, может в среднем, при больших размерах сети N , хранить не более чем $p \approx 0,14 N$ различных образов. Устойчивость может быть показана для совокупности ортогональных образов, когда

$$\left(\frac{1}{N}\right) \sum_{j=1}^N \xi_i^{(\alpha)} \cdot \xi_j^{(\beta)} = \delta_{\alpha\beta} = \begin{cases} 1, & \alpha = \beta, \\ 0, & \alpha \neq \beta. \end{cases} \quad (4.83)$$

В этом случае для каждого состояния ξ^α произведение суммарного входа i -го нейрона h_i на величину его активности $S_i = \xi_i^\alpha$ оказывается положительным, следовательно, само состояние ξ^α является состоянием притяжения (устойчивым *аттрактором*):

$$h_i \cdot \xi_i^\alpha = \sum_j \left[\left(\sum_{\beta} \xi_i^\beta \cdot \xi_j^\beta \right) \cdot \xi_i^\alpha \right] \cdot \xi_i^\alpha = N > 0. \quad (4.84)$$

Таким образом, правило Хэбба обеспечивает устойчивость сети Хопфилда на заданном наборе относительно небольшого числа ортогональных образов.

4.7. Нейронные сети с алгоритмом самообучения

4.7.1. Модель Кохонена

Модель Кохонена [21] выполняет обобщение предьявляемой информации. В результате работы нейронной сети Кохонена получается образ, представляющий собой карту распределения векторов из обучающей выборки. Таким образом, в модели Кохонена выполняется решение задачи нахождения кластеров в пространстве входных образов.

Данная сеть обучается без учителя на основе самоорганизации. По мере обучения векторы весов нейронов стремятся к центрам кластеров – групп векторов обучающей выборки. На этапе решения информационных

задач сеть относит новый предъявленный образ к одному из сформированных кластеров, указывая тем самым категорию, к которой он принадлежит.

Рассмотрим архитектуру нейронной сети Кохонена и правила обучения подробнее. Сеть Кохонена состоит из одного слоя нейронов. Число входов каждого нейрона равно размерности входного образа. Количество же нейронов определяется той степенью подробности, с которой требуется выполнить кластеризацию набора библиотечных образов. При достаточном количестве нейронов и удачных параметрах обучения нейронной сети Кохонена может не только выделить основные группы образов, но и установить «тонкую структуру» полученных кластеров. При этом близким входным образам будет соответствовать близкие карты нейронной активности.

Обучение начинается с задания случайных значений матрице связей W_n^m . В дальнейшем происходит процесс самоорганизации, состоящий в модификации весов при предъявлении на вход векторов обучающей выборки. Для каждого нейрона можно определить его расстояние до вектора входа:

$$d_m = \sum_{i=1}^N (x_i(t) - W_i^m(t))^2. \quad (4.85)$$

Далее выбирается нейрон $m = m^*$, для которого это расстояние минимально. На текущем шаге обучения t будут модифицироваться только веса нейронов из окрестности нейрона m^* :

$$W_n^m(t+1) = W_n^m(t) + \eta(x_n(t) - W_n^m(t)). \quad (4.86)$$

Первоначально в окрестности любого из нейронов находятся все нейроны сети, впоследствии эта окрестность сужается. В конце этапа обучения подстраиваются только веса самого ближайшего нейрона. Темп обучения $\eta(t) < 1$ с течением времени также уменьшается. Образы обучающей выборки предъявляются последовательно, и каждый раз происходит подстройка весов. Нейронная сеть Кохонена может обучаться и на искаженных версиях входных векторов, в процессе обучения искажения, если они не носят систематический характер, сглаживаются.

4.7.2. Звезды Гроссберга

Входная звезда (*Instar*) С. Гроссберга [2; 22] представляет собой нейрон, аналогичный по структуре адальне, предназначенный для решения простейших задач распознавания образов и осуществляющий преобразование:

$$y_j = \psi \left(\sum_{i=1}^n w_{ji}^T x_i + \theta_j \right), \quad (4.87)$$

где $\psi(u_j) = \begin{cases} 1, & \text{если } u_j \geq 0, \\ 0, & \text{в противном случае.} \end{cases} \quad (4.88)$

Схема входной звезды приведена на рис. 4.29.

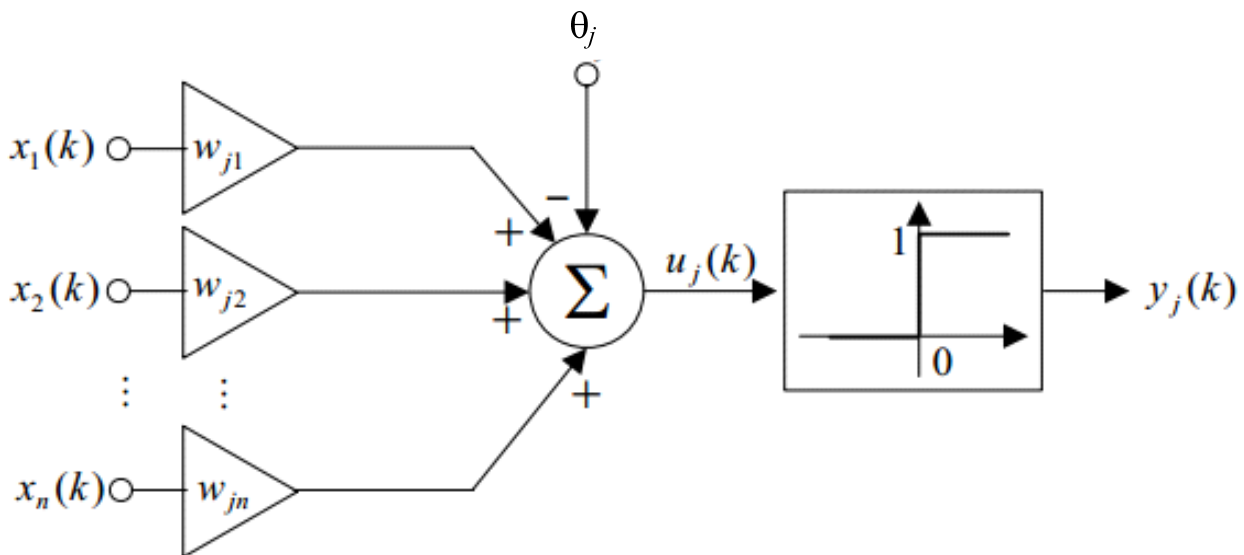


Рисунок 4.29 – Входная звезда Гроссберга

Несложно видеть, что этот нейрон активизируется (на выходе появляется 1) в случае, если вектор входных сигналов $x(k)$ в некотором смысле близок к текущему вектору синаптических весов $w_j(k)$, то есть при выполнении условия:

$$w_j^T(k)x(k) = \|w_j(k)\| \|x(k)\| \cos\theta \geq \theta_j, \quad (4.89)$$

где θ – угол между векторами $w_j(k)$ и $x(k)$; θ_j – сигнал смещения, задающий порог «близости» векторов, который определяет срабатывание входной звезды.

Если принять $\theta_j = \|w_j\| \|x\|$, то звезда активизируется только в случае, если входной сигнал совпадает с вектором синаптических весов, то есть распознается только один образ. Чем меньше значение θ_j , тем больше возможных образов может активизировать нейрон, который становится при этом все менее «разборчивым».

Обучение входной звезды производится с помощью алгоритма, имеющего в данном случае вид:

$$w_j(k+1) = w_j(k) + \eta y_j(k)x(k) - \alpha y_j(k)w_j(k). \quad (4.90)$$

Отличительной особенностью алгоритма (4.90) является то, что самообучение происходит только в активизированном состоянии, когда $y_j(k) = 1$.

Положив для простоты $\alpha = \eta$, получаем так называемое стандартное правило самообучения входной звезды:

$$w_j(k+1) = w_j(k) + \eta y_j(k)(x(k) - w_j(k)). \quad (4.91)$$

При $y_j(k) = 0$, согласно (4.91), обучение не происходит, то есть $w_j(k+1) = w_j(k)$. При $y_j(k) = 1$ алгоритм приобретает вид:

$$w_j(k+1) = w_j(k) + \eta(x(k) - w_j(k)) = (1 - \eta)w_j(k) + \eta x(k), \quad (4.92)$$

то есть вектор синаптических весов «подтягивается» ко входному образу на расстояние, пропорциональное параметру шага η . Чем больше η , тем ближе $w_j(k+1)$ к $x(k)$ и при $\eta = 1$ совпадает с ним. Обычно в реальных задачах используется переменное значение $\eta(k)$. Можно также отметить, что в целях вычислительных удобств вместо вектора $x(k)$ чаще используют его нормированный аналог:

$$x(k) = \frac{x(k)}{\|x(k)\|}. \quad (4.93)$$

Своеобразным антиподом входной звезды является *выходная звезда (Outstar)*, предназначенная для решения задач восстановления образов, схема которой приведена на рис. 4.30.

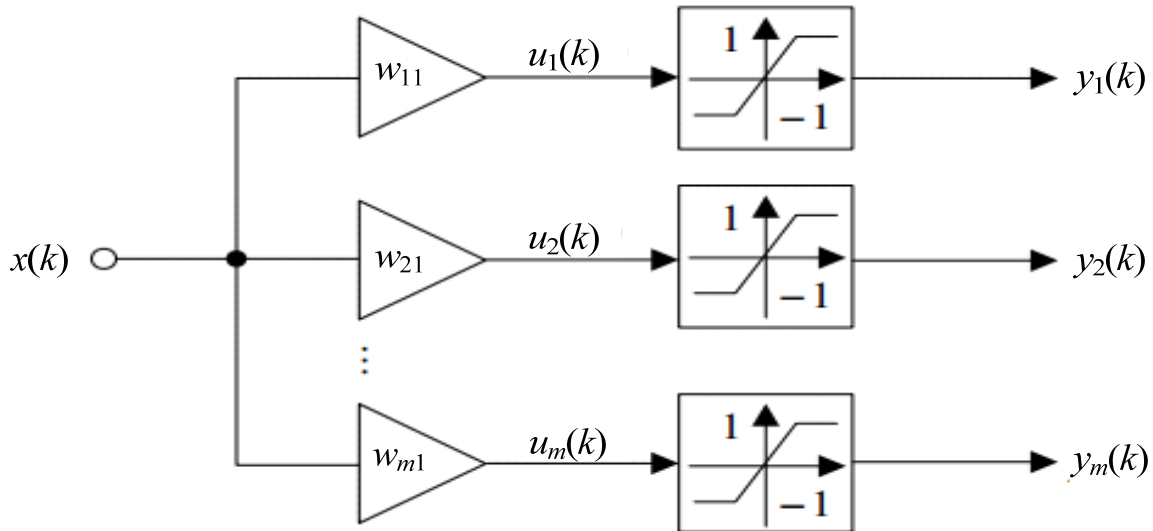


Рисунок 4.30 – Выходная звезда Гроссберга

Этот нейрон имеет скалярный вход и векторный выход и осуществляет преобразование:

$$y_j = \psi(w_{j1}x), \quad j = 1, 2, \dots, m \quad (4.94)$$

с функцией активации:

$$\begin{cases} u_j, & \text{если } -1 \leq u_j \leq 1, \\ 1, & \text{если } 1 \leq u_j, \\ -1, & \text{если } u_j < -1. \end{cases} \quad (4.95)$$

Правило самообучения выходной звезды имеет вид:

$$w_{ji}(k+1) = w_{ji}(k) + \eta y_j(k)x_i(k) - \alpha x_i(k)w_{ji}(k), \quad (4.96)$$

а при $\eta = \alpha$:

$$w_{ji}(k+1) = w_{ji}(k) + \eta x_i(k)(y_j(k) - w_{ji}(k)), \quad (4.97)$$

то есть настройка синаптических весов происходит только в случае $x_i(k) \neq 0$. Здесь, как видно, в процессе самообучения синаптические веса «подтягиваются» к выходному вектору $y(k)$.

В векторной форме правило самообучения имеет вид:

$$w_i(k+1) = w_i(k) + \eta x_i(k)(y(k) - w_i(k)), \quad (4.98)$$

где $w_i(k)$ – i -й столбец матрицы коэффициентов $W(k+1)$.

4.8. Сети с обратными связями

В предыдущих разделах были рассмотрены нейронные сети с прямой передачей информации, в которых сигнал распространяется и обрабатывается в одном направлении – от входа к выходу ИНС, а последовательность, в которой предъявляются обучающие образы, в общем случае не имеет значения. Здесь мы рассмотрим второй важный класс ИНС – рекуррентные нейронные сети, имеющие замкнутые петли обратной связи в своей топологии. В этих сетях на первый план выступает фактор времени: входные сигналы в ИНС должны быть заданы в форме временной последовательности, автокорреляционные свойства которой выявляются и анализируются в процессе обработки.

В рекуррентных сетях в основном используются два способа организации обратной связи: локальная обратная связь на уровне отдельных нейронов и глобальная, охватывающая сеть в целом, хотя возможны и промежуточные варианты. Так, если в качестве базового строительного блока рекуррентной сети принять многослойный персептрон, то локальная обратная связь организуется на уровне отдельного слоя, глобальная связывает нейроны выходного слоя со входами сети, однако при этом возможны варианты связи от скрытого слоя ко входному или от скрытого к предыдущему скрытому слою.

В настоящее время сформировалось два больших класса рекуррентных сетей: сети, реализующие отображение «вход - выход» с учетом временного фактора, и сети ассоциативной памяти. Последние будут рассмотрены в последующем разделе, а здесь мы подробно остановимся на сетях первого класса, широко применяемых для решения задач оптимизации, идентификации, эмуляции, прогнозирования, управления, диагностики и т.п., словом, там, где фактор времени имеет существенное значение.

Синтез и анализ рекуррентных нейронных сетей является предметом рассмотрения достаточно нового направления в теории ИНС – нейродинамики [19], вобравшей в себя аппарат нейроматематики, теории автоматического управления и адаптивных систем. Основная проблема, возникающая при синтезе рекуррентных сетей (как впрочем и всех систем с обратной связью), связана с обеспечением их устойчивости, от которой, собственно, и зависит возможность решения поставленной задачи.

С точки зрения нейродинамики рекуррентные сети рассматриваются как многосвязные динамические нелинейные стохастические диссипативные системы с большим числом степеней свободы, для анализа устойчивости которых неприменимы традиционные инженерные критерии устойчивости. В связи с этим основные подходы к разработке и анализу рекуррентных сетей связаны с аппаратом функций Ляпунова [2], идентификацией устойчивых состояний (аттракторов) и минимизацией тех или иных форм энергетических функций.

К данному классу сетей относятся следующие сети: Хопфилда, Хэмминга, Элмена, Джордана, сети с временными задержками, а также скрытые марковские модели и др. Ниже будут рассмотрены некоторые из этих сетей.

4.8.1. Модель Хопфилда

Исторически первой рекуррентной нейронной сетью явилась ИНС Хопфилда [23; 24], получившая в дальнейшем широкое распространение для решения задач комбинаторной оптимизации, а также в качестве ассоциативной памяти. Схема данной сети, основными строительными блоками которой являются нейроны Хопфилда (см. рис. 4.12), приведена на рис. 4.31.

Функционирование сети описывается системой нелинейных разностных уравнений:

$$x_j(k+1) = \psi_j \left(\sum_{i=1}^n w_{ji} x_i(k) + \theta_j \right), \quad k = 1, 2, \dots; \quad j, i = 1, 2, \dots, n, \quad (4.99)$$

или

$$x_j(k+1) = \Psi(Wx(k) + \theta), \quad x(0) = x_0. \quad (4.100)$$

Вид уравнений (4.99), (4.100) свидетельствует о том, что они являются ни чем иным, как стандартным описанием нелинейной динамической системы в пространстве состояний. Для заданных начальных условий $x(0)$ при соответствующем выборе матрицы синаптических весов W такая система в процессе своего функционирования сходится к точке равновесия x^* , минимизируя при этом некоторую энергетическую функцию (функцию Ляпунова, гамильтониан и т.п.) сети $E(k)$.

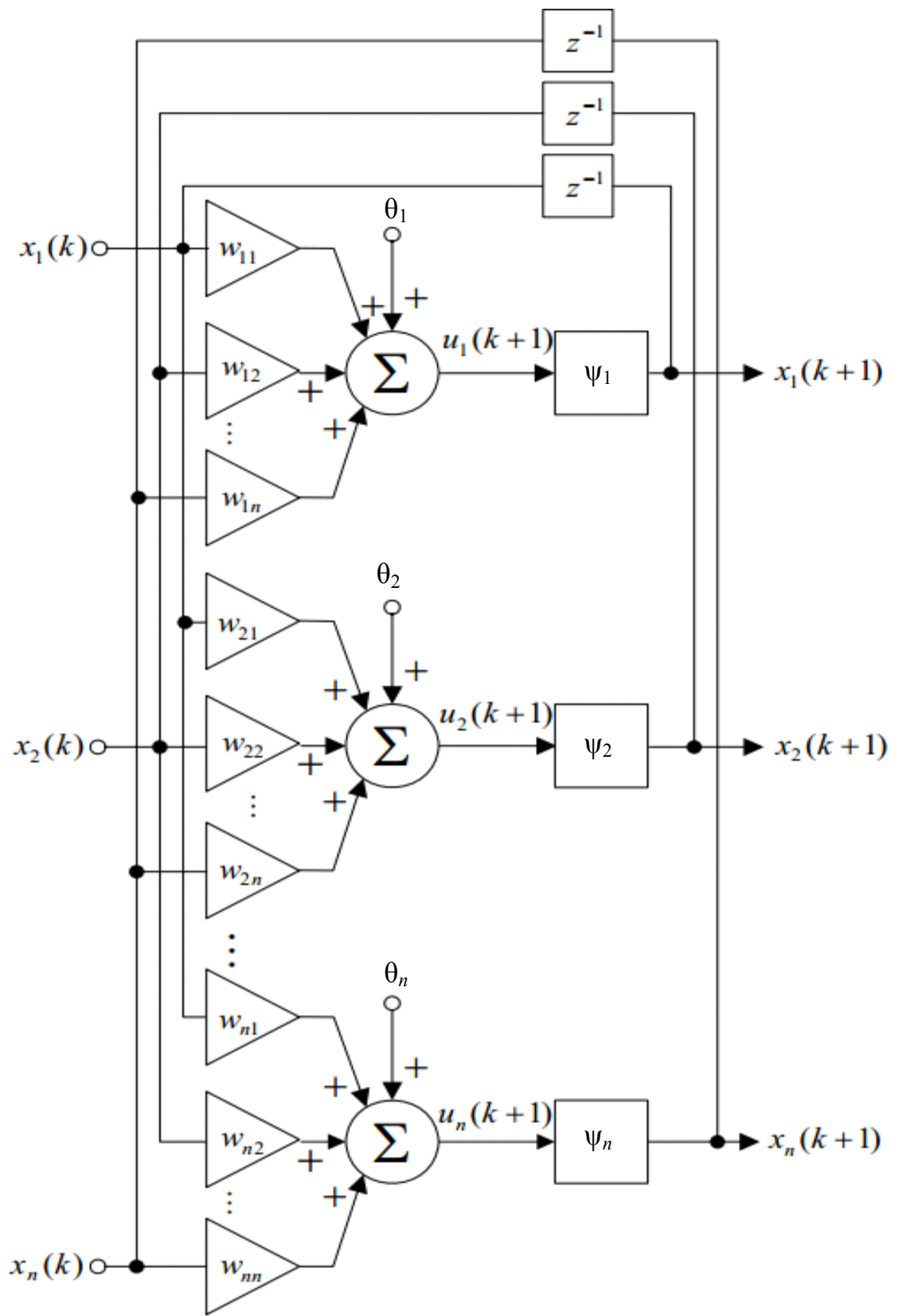


Рисунок 4.31 – Нейронная сеть Хопфилда

Точка равновесия $x^* = x(k + 1) = x(k)$ в общем случае может быть устойчивой, неустойчивой или нейтральной.

Естественно, что для решения практических задач наибольший интерес представляют устойчивые состояния, с формальной точки зрения характеризующиеся тем, что для любого $\varepsilon_1 > 0$ существует $\varepsilon_2 > 0$ такое, что:

$$\begin{cases} \|\psi(x(0), k) - x^*\| < \varepsilon_1, \\ \|x(0) - x^*\| < \varepsilon_2, \end{cases} \quad (4.101)$$

для всех $k = 0, 1, 2, \dots$

В качестве активационных функций в сетях Хопфилда наибольшее распространение получил не гиперболический тангенс, а более простая сигмоидальная функция.

В этом случае уравнение (4.99) естественно приобретает вид:

$$x_j(k + 1) = \text{sign} \left(\sum_{i=1}^n w_{ji} x_i(k) + \theta_j \right) = \begin{cases} 1 & \text{если } u_j(k + 1) \geq 0, \\ -1 & \text{в противном случае.} \end{cases} \quad (4.102)$$

В процессе своего функционирования такая сеть, минимизируя энергетическую функцию:

$$E(k) = -\frac{1}{2} x^T(k) W x(k) - x^T(k) \theta, \quad (4.103)$$

сходится к устойчивому состоянию, что в общем случае гарантируется при симметрической матрице синаптических весов $W (w_{ji} = w_{ij})$ с нулевыми элементами на главной диагонали ($w_{jj} = 0$).

Для обучения этой сети обычно используется автоассоциативное правило обучения Хэбба в пакетном варианте:

$$W = \frac{1}{n} \sum_{k=1}^N x(k) x^T(k) - \frac{N}{n} I, \quad (4.104)$$

где N – объем обучающей выборки; I – $(n \times n)$ – единичная матрица.

Согласно уравнению (4.102), каждый нейрон может находиться в двух состояниях: +1 или -1, а сама сеть также может функционировать в двух режимах. Если в каждый момент времени k все нейроны могут изменять свое состояние, то это – параллельный режим, если же в каждый момент свое состояние может менять только один нейрон, то говорят о последовательном режиме.

Рассмотрим ситуацию, когда сеть работает в последовательном режиме и в момент времени $k + 1$ j -й нейрон принимает состояние:

$$x_j(k+1) = x_j(k) + \Delta x_j. \quad (4.105)$$

Изменение энергетической функции (4.103) при этом имеет вид:

$$\Delta E(k) = E(k+1) - E(k) = -\frac{1}{2} \Delta x_j \left(\sum_{i=1}^n w_{ji} x_i(k) + \sum_{p=1}^n w_{pj} x_p(k) \right) - w_{jj} (\Delta x_j)^2 - \Delta x_j \theta_j, \quad (4.106)$$

а с учетом симметричности матрицы W :

$$\Delta E(k) = -\Delta x_j \left(\sum_{i=1}^n w_{ji} x_i(k) + \sum_{p=1}^n w_{pj} x_p(k) \right) - \Delta x_j \theta_j. \quad (4.107)$$

Заметив, что знаки Δx_j и $\left(\sum_{i=1}^n w_{ji} x_i(k) + \theta_j \right)$ совпадают, можно сде-

лать вывод, что уменьшение энергетической функции $\Delta E(k) < 0$ может быть обеспечено при $w_{jj} \geq 0$, а признаком достижения устойчивого состояния является выполнение условия:

$$x^* = x(k) = \text{sign}(Wx(k) + \theta), \quad (4.108)$$

которое обеспечивается при $k > n^3$ тактов работы сети.

4.8.2. Сеть Хэмминга

Нейронная сеть Хэмминга была предложена для решения задачи распознавания путем сравнения предъявляемого образа с одним из эталонов, формирующих обучающую выборку $x(1), x(2), \dots, x(N)$. Особенностью данной сети является то, что и анализируемый образ, и эталоны задаются в форме бинарных векторов, а расстояние между ними определяется хэмминговой метрикой:

$$H \|x - x_j\| = H \left\{ \left(\overline{x_j} \wedge x_i(j) \right) \vee \left(x_i \wedge \overline{x_i(j)} \right) \mid i = 1, 2, \dots, n \right\}, \quad (4.109)$$

то есть количеством отличающихся элементов векторов x и $x(j)$.

На рис. 4.32 представлена схема нейросети Хэмминга, представляющая собой двухслойную архитектуру и являющаяся своеобразным гибридом вероятностной нейронной сети с прямой передачей информации и рекуррентной сети Хопфилда.

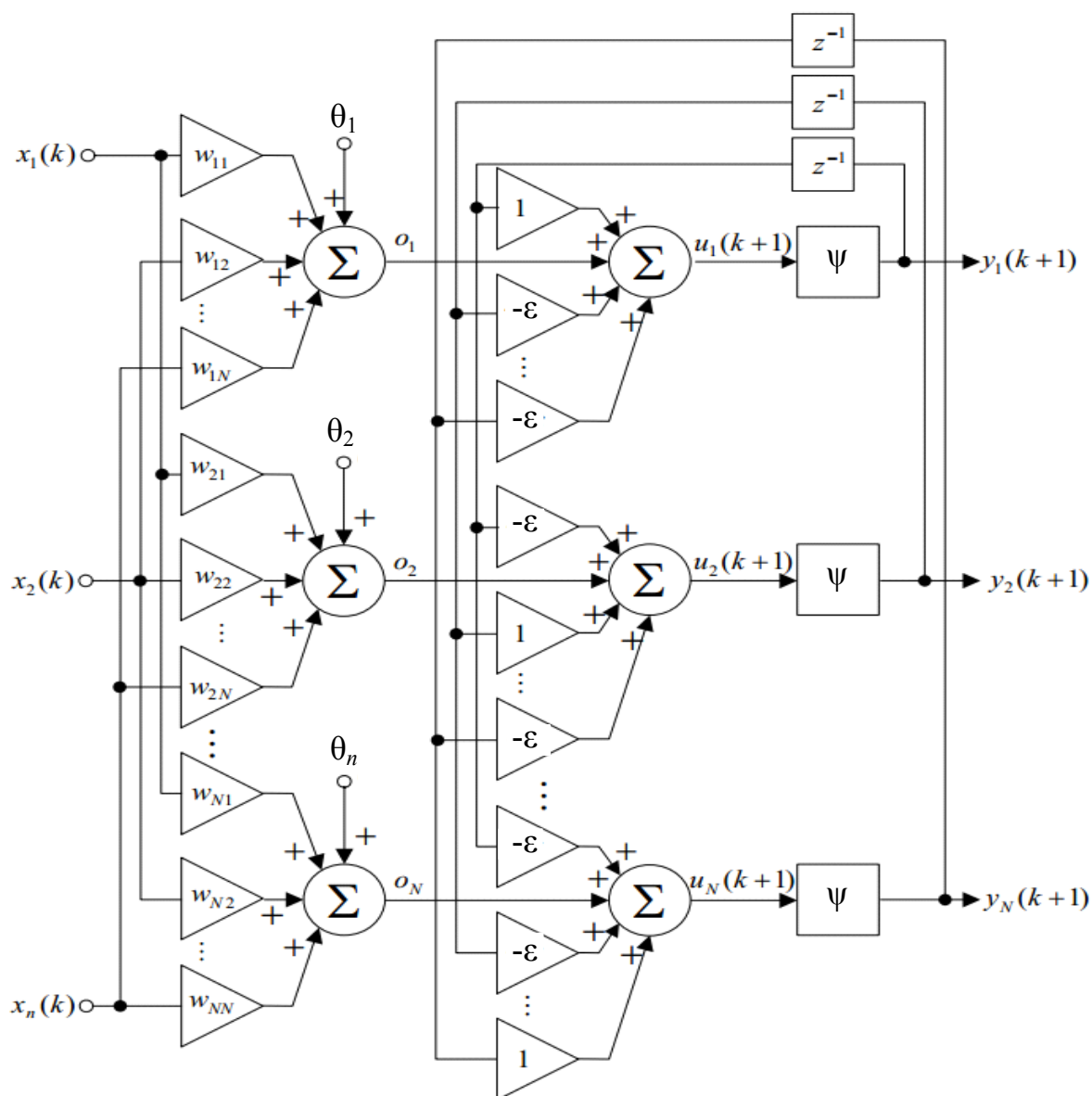


Рисунок 4.32 – Нейронная сеть Хэмминга

Сеть имеет n входов, N нейронов по числу имеющихся эталонов в скрытом слое и N рекуррентных нейронов в выходном слое. Скрытый слой образован линейными ассоциаторами, синаптические веса которых подобно вероятностной нейронной сети устанавливаются заранее и определяются значениями компонент векторов-эталонов $x(j)$, $j = 1, 2, \dots, N$:

$$w_{ji} = \frac{x_i(j)}{2}, \theta_j = \frac{n}{2}. \quad (4.110)$$

При подаче на вход сети произвольного вектора x на выходах нейронов скрытого слоя появляются сигналы [2]:

$$o_j = \left(\sum_{i=1}^n w_{ji} x_i(k) + \theta_j \right) = n - H \|x - x(j)\|. \quad (4.111)$$

Нейрон с максимальным выходным сигналом соответствует эталону $x(j)$, ближайшему в смысле Хэмминга к вектору x . Рекуррентный выходной слой сети образован нейронами, выходы которых через элементы задержки z^{-1} связаны с входами остальных нейронов этого слоя отрицательными обратными связями с коэффициентом усиления $0 < \varepsilon < N^{-1}$. Кроме того, каждый нейрон охвачен положительной обратной связью, которая связывает его выход с собственным входом. Выходной сигнал рекуррентных нейронов y_j определяется активационной функцией-выпрямителем, вид которой приведен на рис. 4.4, *г*.

Выходные сигналы скрытого слоя задают начальные состояния рекуррентных нейронов:

$$y_j(0) = o_j, \quad (4.112)$$

реализующих последовательность вычислений:

$$\begin{cases} u_j(k+1) = y_j(k) - \varepsilon \sum_{l=1, l \neq j}^N y_l(k), \\ y_j(k+1) = \psi(u_j(k+1)). \end{cases} \quad (4.113)$$

В результате итераций выходные сигналы всех нейронов постепенно уменьшаются, устремляясь к нулю, кроме одного, имевшего наибольшее начальное значение (4.112). Этот нейрон и определяет тот эталон, к которому будет отнесен предъявленный вектор-образ x .

К преимуществам сети Хэмминга следует отнести ее быстрое действие и простоту, хотя возможность работы только с бинарными векторами несколько ограничивает ее применимость.

4.9. Нейронные сети ассоциативной памяти

Одним из важных свойств биологического мозга является способность к запоминанию информации и ее последующему восстановлению с помощью системы ассоциаций, при этом однажды увиденный когда-то образ может быть опознан даже спустя длительное время, несмотря на все произошедшие с ним изменения. Эту функцию мозга моделирует доста-

точно широкий класс ИНС, называемых нейронными сетями ассоциативной памяти [2].

Такая искусственная память представляет собой, как правило, нейронную сеть либо с прямой передачей информации, либо рекуррентную, которая в режиме обучения накапливает предъявляемые ей образы (чаще всего в векторной форме), а в режиме восстановления вызывает их из памяти, ассоциируя вновь предъявляемые данные с ранее увиденными.

В отличие от стандартных блоков памяти ЭВМ, в ассоциативной памяти информация «размазывается» по всему множеству нейронов, входящих в сеть, что естественно повышает надежность ее хранения; доступ к информации производится по содержанию, а не по адресу; предусмотрена возможность устойчивой работы с искаженной или неполной информацией, что позволяет правильно опознавать по «зашумленному» фрагменту образ в целом.

На рис. 4.33 приведена простейшая архитектура ИНС ассоциативной памяти, представляющая собой по сути элементарную однослойную сеть с прямой передачей информации.

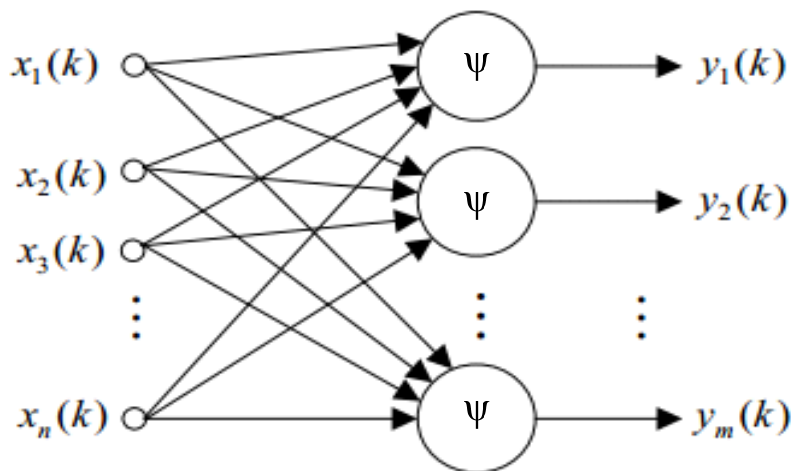


Рисунок 4.33 – Элементарная ИНС ассоциативной памяти

Основное отличие этой сети, состоит в том, что если обычная ИНС реализует однозначное отображение:

$$R^m \ni y(k) = F(x(k) \in R^n), \quad (4.114)$$

то ассоциативная память (АП) отображает в $y(k)$ все возможные векторы входов x , принадлежащие некоторой окрестности $x(k)$ такой, что $\|x - x(k)\| < \varepsilon$.

К настоящему времени распространение получили три основных типа АП:

- *гетероассоциативные сети*, реализующие отображение:

$$x \in R^n \rightarrow y(k) \in R^m, \quad (4.115)$$

для всех x , принадлежащих окрестности $\|x - x(k)\| < \varepsilon$;

- *автоассоциативные сети*, являющиеся специальным подклассом гетероассоциативных и реализующих отображение,

$$x \in R^n \rightarrow x(k) \in R^n, \quad (4.116)$$

для всех x , принадлежащих окрестности $\|x - x(k)\| < \varepsilon$, то есть ассоциирующие каждый образ с самим собой. Основная функция таких сетей – восстановление неполной и искаженной информации, например, узнавание образа по обрывку старой фотографии;

- *сети для распознавания образов*, также являющиеся частным случаем гетероассоциативных, в которых каждый входной вектор x ассоциируется со скаляром $y(k)$, то есть

$$x \in R^n \rightarrow y(k) \in R^1, \quad (4.117)$$

для всех x , принадлежащих окрестности $\|x - x(k)\| < \varepsilon$, то есть ассоциирующие каждый образ с самим собой. Основная задача, решаемая такой сетью, идентификация «имени» входного образа.

На рис. 4.34 приведены три основных типа АП.



Рисунок 4.34 – Типы ИНС ассоциативной памяти

К сетям с ассоциативной памятью относятся ассоциативная память Хопфилда, двунаправленная ассоциативная память, BSB-модели и др. Ниже рассмотрена двунаправленная ассоциативная память.

4.9.1. Двухнаправленная ассоциативная память

Двухнаправленная ассоциативная память (ДАП), разработанная в 1988 г. Бертом Коско, является гетероассоциативной. Она сохраняет пары образов и выдает второй образец пары, когда ассоциированный с ним первый образец подается на вход сети. Архитектура сети представлена на рис. 4.35.

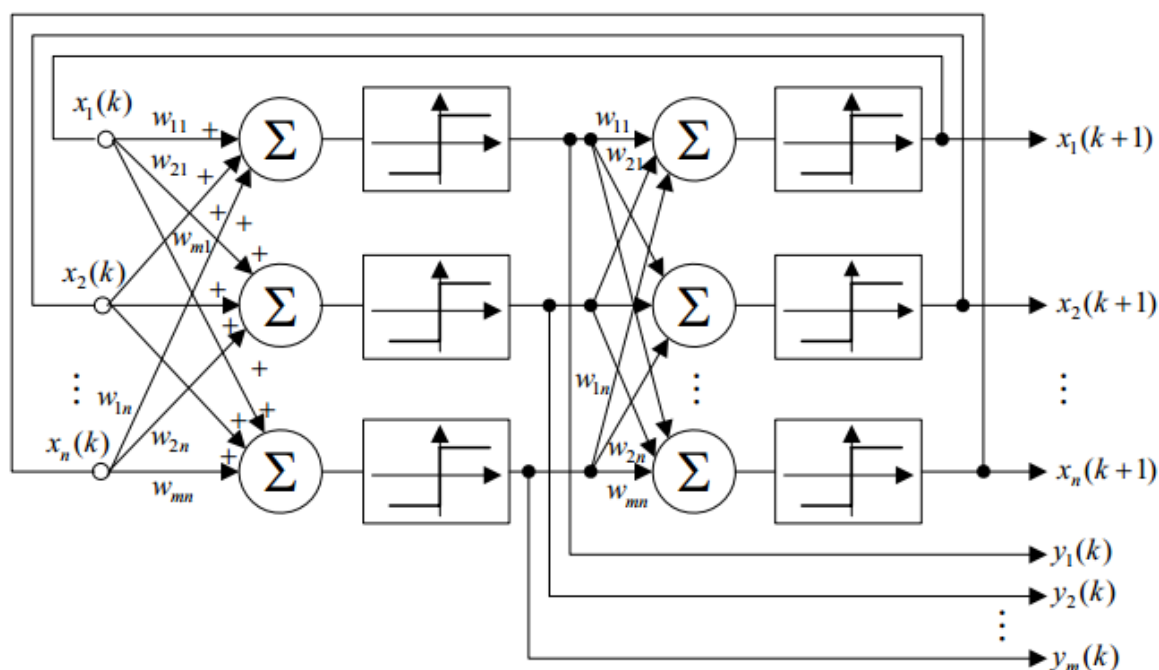


Рисунок 4.35 – Двухнаправленная ассоциативная память

Данная сеть, кроме входного и выходного, содержит скрытый слой, при этом организация архитектуры позволяет осуществлять встречный обмен информацией между скрытым и выходным слоями за счет того, что матрицы синаптических весов этих слоев являются взаимно транспонированными.

При подаче на вход сети n -мерного бинарного вектора $x(1)$ на выходе скрытого слоя появляется m -мерный векторный сигнал

$$y(1) = \text{sign}(Wx(1) + \theta_x), \quad (4.118)$$

который поступает на выходной слой, генерирующий вектор

$$x(2) = \text{sign}(W^T y(1) + \theta_y). \quad (4.119)$$

По каналу обратной связи $x(2)$ передается на вход, в результате чего в скрытом слое вычисляется:

$$y(2) = \text{sign}(Wx(2) + \theta_x), \quad (4.120)$$

а в выходном

$$x(3) = \text{sign}(W^T y(2) + \theta_y) \quad (4.121)$$

После k итераций сеть порождает пару:

$$\begin{cases} y(k) = \text{sign}(Wx(k) + \theta_x), \\ x(k+1) = \text{sign}(W^T y(k) + \theta_y), \end{cases} \quad (4.122)$$

минимизируя при этом энергетическую функцию:

$$E(x, y) = -\frac{1}{2} y^T Wx + \frac{1}{2} \theta_x^T y + \frac{1}{2} \theta_y^T x. \quad (4.123)$$

В результате устойчивые состояния сети (аттракторы) могут быть описаны системой уравнений:

$$\begin{cases} y^* = \text{sign}(Wx^* + \theta_x), \\ x^* = \text{sign}(W^T y^* + \theta_y) \end{cases} \quad (4.124)$$

Двунаправленная память обучается с помощью хэббовского правила. Несложно видеть, что при этом после предъявления первого стимула сеть формирует первый аттрактор:

$$y = \text{sign}(y(1)x^T(1)x) = \text{sign } y(1) = y(1), \quad (4.125)$$

если $0 < \cos(x(1), x) \leq 1$, и

$$x = \text{sign}(x(1)y^T(1)y) = \text{sign } x(1) = x(1), \quad (4.126)$$

если $0 < \cos(y(1), y) \leq 1$.

В случае, если необходимо запомнить $(N \times n)$ – матрицу ключевых образов, синаптические веса сети вычисляются согласно формуле:

$$W(N) = \sum_{k=1}^N y(k) x^T(k) = YX^T, \quad (4.127)$$

где N – максимальное количество запоминаемых образов, определяемое оценкой

$$N \leq \frac{n}{2 \log_2 n} \quad (4.128)$$

в случае нулевых смещений θ_x и θ_y и оценкой

$$N \leq 2^n, \quad (4.129)$$

если каждый нейрон сети имеет свое ненулевое смещение.

Двунаправленная ассоциативная память может использоваться и в качестве автоассоциативной, обладая при этом более высокой емкостью и скоростью сходимости по сравнению с памятью Хопфилда.

4.10. Адаптивные резонансные нейронные сети

С конца 60-х годов прошлого столетия С. Гроссберг с группой коллег исследовал нейроархитектуры, предназначенные для автоматической классификации образов (кластеризации), при этом основная проблема, возникшая в ходе работ, была связана с устранением противоречий между устойчивостью и пластичностью сети, то есть способностью ИНС запоминать новую информацию без разрушения ранее накопленной. В результате предпринятых исследований была создана теория адаптивного резонанса и на ее основе разработана целая группа архитектур [2]: ART 1, предназначенная для работы с бинарными векторами; ART 2, обрабатывающая сигналы с произвольными действительными значениями; ART 3, имеющая иерархическую структуру и описываемая системой дифференциальных уравнений, соответствующих динамике некоторых химических реакций; FUZZY ART, осуществляющая нечеткую кластеризацию; ARTMAP, представляющая собой комбинацию из двух резонансных сетей и др. В общем случае все ART-сети имеют аналогичную архитектуру и реализуют однотипную стратегию кластеризации на основе самообучения, основным преимуществом которой является возможность работы в реальном времени при априорно неизвестном количестве классов.

Адаптивные резонансные сети являются «ближайшими родственниками» двунаправленной ассоциативной памяти (см. подраздел 4.9.1) и подобно ей содержат две взаимодействующие нейроподсистемы, которые вступают в резонанс в процессе накопления и восстановления обрабатываемых образов.

На рис. 4.36 приведена базовая структура ART-сети, состоящая из нейроподсистемы сравнения F_C , нейроподсистемы распознавания F_R и управляющей подсистемы, координирующей работу F_C и F_R в процессе их резонансного взаимодействия.

Данная сеть содержит n нейронов в подсистеме сравнения F_C , число которых совпадает с размерностью входного вектора x , и m нейронов в подсистеме распознавания по количеству запоминаемых образов, количество которых может увеличиваться в процессе работы ИНС, при этом все нейроны каждой из нейроподсистем связаны друг с другом восходящими и нисходящими связями, определяемыми $(m \times n)$ и $(n \times m)$ – матрицами настраиваемых синаптических весов $W = \{w_{ji}\}$ и $V = \{v_{ij}\}$ соответст-

венно. Заметим также, что синаптический вес w_{ji} связывает i -й нейрон в F_C с j -м нейроном в F_R , а v_{ij} – j -й нейрон в F_R с i -м нейроном в F_C . Кроме того, узлы нейросистемы распознавания охвачены обратными связями, обеспечивающими работу F_R в режиме «победитель получает все», в результате чего в каждый момент времени только один нейрон в F_R может находиться в активном состоянии.

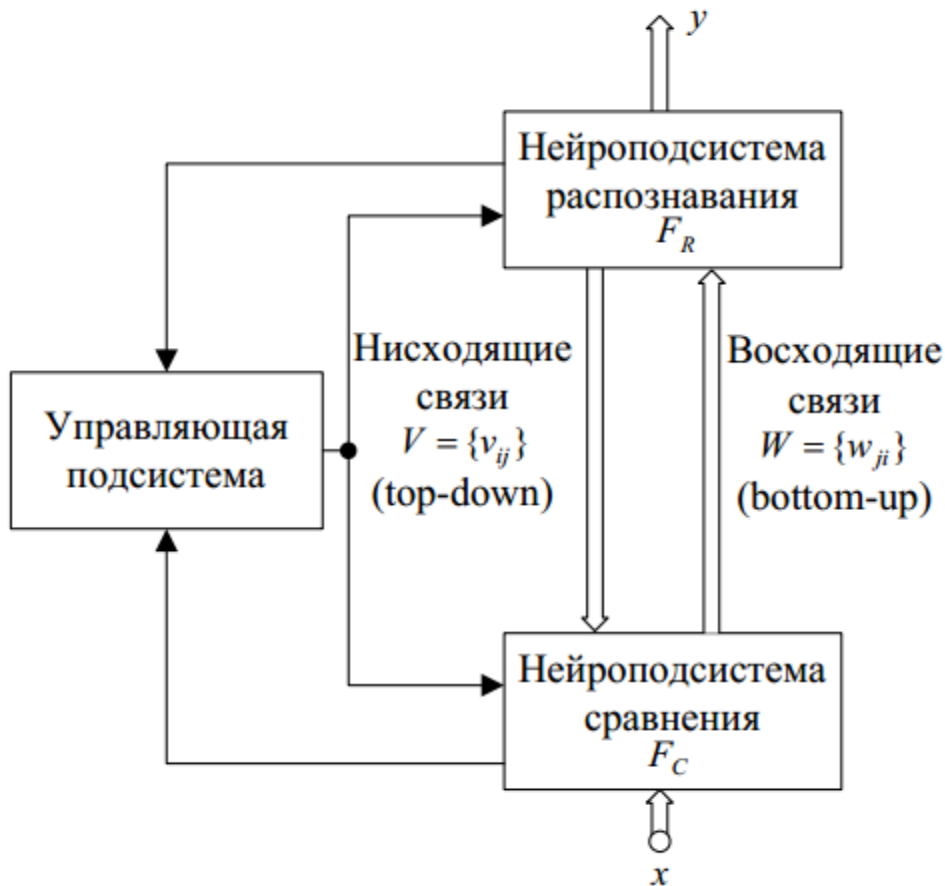


Рисунок 4.36 – Блок-схема адаптивной резонансной нейронной сети

В процессе своей работы адаптивные резонансные сети реализуют так называемую комбинированную парадигму распознавания-обучения. Первый предъявляемый образ автоматически считается центром (эталонном, прототипом) первого кластера, и именно он определяет весовые параметры связи F_C с единственным нейроном подсистемы F_R . Следующий образ сравнивается с прототипом первого кластера, и если они в некотором смысле близки, то происходит уточнение весов w_{1i} и v_{i1} . В противном же

случае в F_R формируется второй нейрон - эталон второго кластера. Таким образом, в результате циклического обмена информацией по восходящим-нисходящим связям реализуется одновременное самообучение-восстановление, состоящее в том, что в подсистеме F_C определяется «близость» предъявляемого образа к ранее накопленным эталонам, а в подсистеме F_R выясняется, к какому из кластеров конкретно принадлежит показанный вектор. Если при этом окажется, что данный образ расположен «далеко» от всех центров, управляющая подсистема сформирует в подсистеме распознавания новый нейрон.

Работу адаптивной резонансной сети рассмотрим для случая, когда априорно полагается, что данные могут быть разбиты на m кластеров, при этом, если фактическое число кластеров окажется меньше чем m , то некоторые из нейронов подсистемы F_R никогда не будут принимать активное состояние, в противном же случае, если m окажется меньшим фактического количества кластеров, сеть должна будет сформировать новый нейрон.

Начальные значения синаптических весов сети обычно полагаются равными:

$$w_{ji}(0) = \frac{1}{n+1}, v_{ij}(0) = 1, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m, \quad (4.130)$$

и поскольку они определяют эталон, соответствующий j -му нейрону в F_R , все нейроны подсистемы распознавания в исходном состоянии находятся в равных условиях.

После подачи на вход сети предварительно пронормированного n -мерного образа-вектора x нейроподсистема сравнения F_C вычисляет скалярное произведение:

$$y_j = \sum_{i=1}^n w_{ji}(0)x_i = w_j^T(0)x, \quad j = 1, 2, \dots, m, \quad (4.131)$$

если сеть только начала работу или если уже было отработано k итераций:

$$y_j = \sum_{i=1}^n w_{ji}(k)x_i = w_j^T(k)x, \quad j = 1, 2, \dots, m. \quad (4.132)$$

С учетом нормированности входных сигналов и прототипов кластеров соотношения (4.131), (4.132) есть не что иное, как косинусы углов между предъявляемым образом и имеющимися эталонами, определяющие «близость» образа к имеющимся центрам.

Вычисленные скалярные произведения далее поступают в подсистему распознавания F_R , где определяется максимальное значение:

$$y_j^* = \max_j \{y_j\} \quad (4.133)$$

то есть нейрон - победитель, чьи параметры наиболее близки к предъявленному образу.

После того как в подсистеме F_R найден нейрон-победитель, начинается фаза нисходящей верификации, когда сигнал y_j^* через синаптические связи v_{ij} возвращается в подсистему F_C в виде:

$$\hat{x}_i = v_{ij} y_j^*, \quad (4.134)$$

после чего происходит сравнение с предъявленным образом x . Так, для бинарных векторов, чьи компоненты принимают значения ± 1 , осуществляется проверка неравенства:

$$\sum_{i=1}^n \hat{x}_i = \rho \sum_{i=1}^n |x_i|, \quad (4.135)$$

или, что то же самое

$$\sum_{i=1}^n v_{ij} y_j^* = \rho \sum_{i=1}^n |x_i|, \quad (4.136)$$

где $0 < \rho < 1$ – так называемый параметр сходства, определяющий меру «близости» вектора x к центру j -го кластера. Чем больше значение этого параметра, тем более «жестко» осуществляется классификация, а при $\rho = 1$ каждый образ, отличный от предыдущих, порождает новый кластер.

Если неравенства (4.135), (4.136) выполняются, происходит уточнение синаптических весов в виде:

$$\begin{cases} \frac{dw_{ji}(t)}{dt} = \eta \delta \cdot (-\alpha w_{ji}(t) + w_{ji}(t)x_i), \\ \frac{dv_{ij}(t)}{dt} = \eta \delta \cdot (-\alpha v_{ij}(t) + v_{ij}(t)x_i) \end{cases} \quad (4.137)$$

в непрерывном времени или

$$\begin{cases} w_{ji}(k+1) = \delta^* \frac{w_{ji}(k)x_i}{0,5 + \sum_{i=1}^n w_{ji}(k)x_i}, \\ v_{ij}(k+1) = \delta^* v_{ij}(k)x_i \end{cases} \quad (4.138)$$

в дискретном времени.

Здесь, как и ранее, $\eta > 0$ – скалярный параметр, определяющий скорость обучения; $0 \leq \alpha \leq 1$ – фактор забывания, а

$$\delta^* = \begin{cases} 1, & \text{если } j\text{-й нейрон в } F_R \text{ является победителем,} \\ 0, & \text{для всех остальных нейронов.} \end{cases} \quad (4.139)$$

Если же контрольные неравенства не выполняются, то обычно делается попытка найти новый нейрон-победитель, исключив из рассмотрения предыдущий. Если в результате не будет найден подходящий к данному образу кластер, можно либо смягчить условия классификации, уменьшив значение параметра сходства ρ , либо сформировать в нейроподсистеме распознавания новый нейрон-прототип, чьи параметры определяются последним неотклассифицированным образом.

Несмотря на свои достоинства, связанные с переменным числом кластеров и возможностью работы в реальном времени, адаптивные резонансные нейронные сети получили достаточно ограниченное распространение из-за своей громоздкости и низкой помехоустойчивости, провоцирующей неограниченный рост числа нейронов в подсистеме распознавания.

4.11. Нейронная сеть встречного распространения

Еще одним примером гибридных ИНС являются нейронные сети встречного распространения (*Counterpropagation neural networks*), предложенные Р. Хехт–Нильсеном [25; 26] в качестве альтернативы многослойным сетям с прямой передачей информации, обучаемым на основе обратного распространения ошибок. Эти сети сокращают время обучения, как минимум, на порядок по сравнению с многослойными персептронами, хотя несколько проигрывают им по точности. Исходно эти сети были разработаны для аппроксимации по экспериментальным данным некоторого отображения $y = F(x)$ и нахождения обратного оператора $x = F^{-1}(y)$, хотя впоследствии применялись в основном в задачах распознавания образов, компрессии данных, а также в качестве ассоциативной памяти.

В простейшем случае сеть встречного распространения представляет собой гибрид самоорганизующейся карты Т. Кохонена и звезд С. Гроссберга (подразделы 4.7.1, 4.7.2) и соответственно сочетает в себе конку-

рентное самообучение с контролируемым обучением с учителем. К настоящему времени известен ряд вариантов сетей встречного распространения, простейшим из которых является однонаправленная ИНС, чья архитектура приведена на рис. 4.37.

Данная сеть предназначена для аппроксимации некоторого априори неизвестного оператора $F : X \rightarrow Y$, отображающего n -мерное пространство входов в m -мерное пространство выходов, по данным обучающей выборки $x(1), y(1); x(2), y(2); \dots; x(N), y(N)$.

Как видно из рисунка, нулевой слой сети содержит $n + m$ рецепторов и условно может быть разбит на две секции, одна из которых передает на скрытый слой образ $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$, а другая – на выходной слой $y(k) = (y_1(k), y_2(k), \dots, y_m(k))^T$.

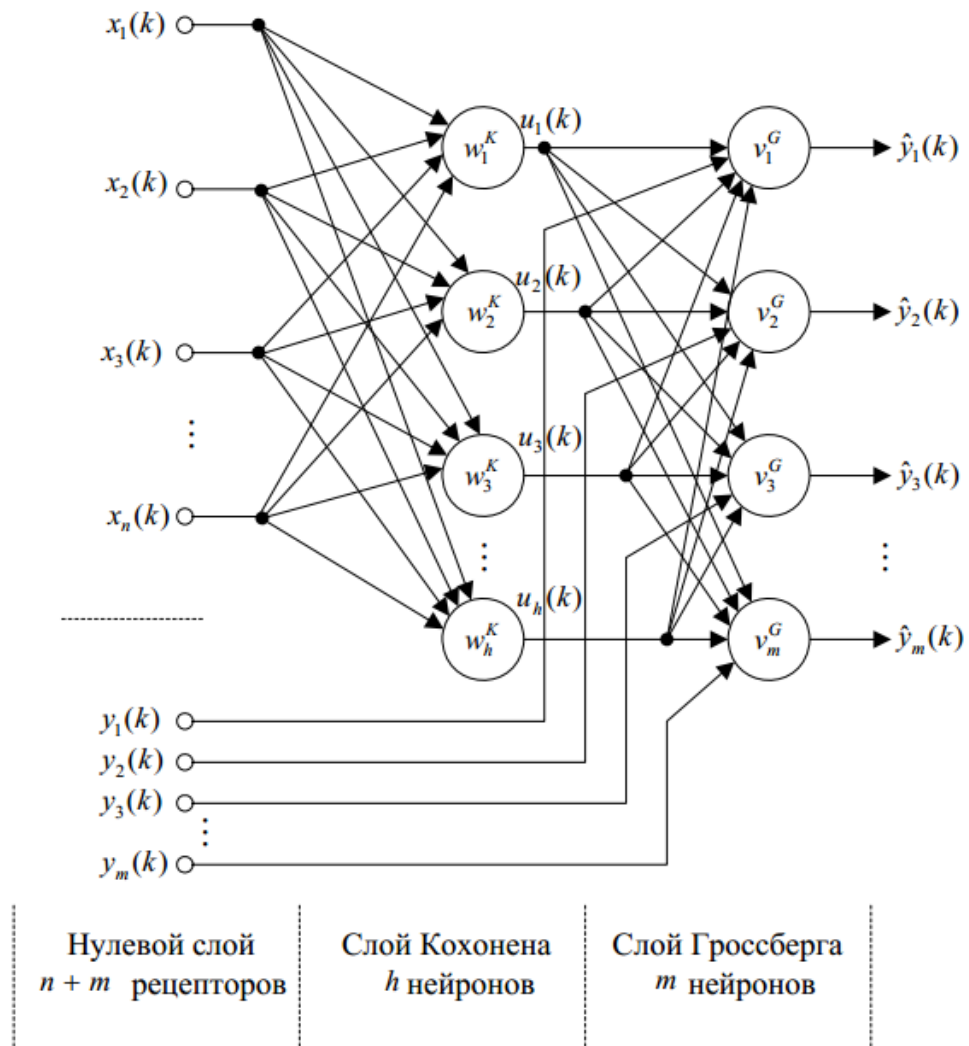


Рисунок 4.37 – Однонаправленная нейронная сеть встречного распространения

Скрытый слой, именуемый также слоем Кохонена, представляет собой по сути самоорганизующуюся карту, содержит h нейронов и реализует отображение:

$$u(k) = W^k x(k), \quad (4.140)$$

где $u(k) = (u_1(k), u_2(k), \dots, u_h(k))^T$; $W^k = \{w_{li}^k\}$; $l = 1, 2, \dots, h$; $i = 1, 2, \dots, n$ – $(h \times n)$ – матрица настраиваемых синаптических весов.

Выходной слой, именуемый также слоем Гроссберга, образован m выходными звездами и реализует отображение:

$$\hat{y}(k) = V^G u(k), \quad (4.141)$$

где $V^G = \{v_{jl}^G\}$; $j = 1, 2, \dots, m$; $l = 1, 2, \dots, h$; – $(m \times h)$ – матрица настраиваемых синаптических весов.

Поскольку в качестве нейронов сети обычно используются адаптивные линейные ассоциаторы, несложно записать преобразования, осуществляемые отдельными нейронами

$$\begin{cases} u_l(k) = w_l^{KT} x(k), & l = 1, 2, \dots, h; \\ \hat{y}_j(k) = v_j^{GT} u(k), & j = 1, 2, \dots, m \end{cases} \quad (4.142)$$

и сеть в целом

$$\hat{y}(k) = V^G W^K x(k). \quad (4.143)$$

Обучение сети встречного распространения происходит в два этапа. На первом этапе по данным выборки $x(1), x(2), \dots, x(N)$ в слое Кохонена происходит процесс самоорганизации, в результате которого входное пространство разбивается на множество кластеров, при этом, как правило, реализуется стратегия самообучения типа «победитель получает все»:

$$w_l^K(k+1) = \begin{cases} w_l^K(k) + \eta_w(k)(x(k) - w_l^K(k)), & \text{если } l\text{-й нейрон победил,} \\ w_l^K, & \text{в противном случае.} \end{cases} \quad (4.144)$$

На втором этапе по выборке $x(1), y(1); x(2), y(2); \dots; x(N), y(N)$ настраивается выходной слой в соответствии с правилом обучения выходной звезды:

$$v_{jl}^G(k+1) = v_{jl}^G(k) + \eta_v(k) u_l(k) (y_j(k) - v_{jl}^G(k)), \quad (4.145)$$

где

$$u_l(k) = \begin{cases} 1, & \text{если } l\text{-й нейрон победил,} \\ 0, & \text{в противном случае,} \end{cases} \quad (4.146)$$

то есть фактически происходит уточнение только тех весов, которые связывают нейроны слоя Гроссберга с нейроном-победителем слоя Кохонена. В результате, если на обученную сеть подать образ $p(x)$, не принадлежащий обучающей выборке, то сначала в слое Кохонена будет установлена его принадлежность определенному кластеру, после чего нейрон-победитель «возбудит» выходные звезды так, что на их выходах появятся сигналы, соответствующие координатам прототипа (центроида) этого кластера.

4.12. Нечеткие нейронные сети

В настоящее время предложены различные схемы классификации нейронных сетей и соответствующие алгоритмы их обучения. Однако нейронные сети все же обладают недостатками [27; 28]. Несмотря на то что они являются очень удобными для задач распознавания, классификации образов, прогнозирования и идентификации, лицо, принимающее решение (ЛПР), не может получить ответа на вопрос: как осуществляются эти процессы? Для пользователя обученная сеть подобна «черному ящику». Многие из недостатков могут быть разрешены с помощью систем с нечеткой логикой, которые используют основные понятия теории нечетких множеств (см. раздел 3).

Можно сказать, что системы с нечеткой логикой являются удобными и полезными для объяснения получаемых с их помощью результатов, они обеспечивают более высокую устойчивость к воздействию мешающих факторов. Однако такие системы не могут автоматически обучаться и приобретать новые знания. В итоге, искусственные нейронные сети и системы с нечеткой логикой эквивалентны друг другу, но, тем не менее, у них имеются собственные достоинства и недостатки.

Основная идея [3; 27; 28], положенная в основу нечетких нейронных сетей, заключается в том, что используется существующая выборка данных для определения параметров функций принадлежности, которые лучше всего соответствуют некоторой системе логического вывода, то есть выводы делаются на основе аппарата нечеткой логики. А для нахождения

параметров функций принадлежности используются алгоритмы обучения нейронных сетей. Такие системы могут использовать заранее известную информацию, обучаться, приобретать новые знания, прогнозировать временные ряды, выполнять классификацию образов, и, кроме этого, они являются вполне наглядными для пользователя.

4.12.1. Основные понятия и определения нечетких нейронных сетей

Для пояснения сущности нечетких нейронных (гибридных) сетей рассмотрим еще раз простую нейронную сеть, имеющую два входа и только один нейрон (рис. 4.38).

Здесь входные сигналы x_i «взаимодействуют» с весами w_i , образуя произведения:

$$p_i = x_i w_i, i = 1, 2. \quad (4.147)$$

Такая частная информация (произведения) объединяется с использованием операции суммирования, образуя вход *net* нейрона:

$$net = p_1 + p_2 = w_1 x_1 + w_2 x_2. \quad (4.148)$$

Выход нейрона образуется в результате преобразования входа *net* некоторой активационной функцией:

$$y = f(net) = f(w_1 x_1 + w_2 x_2), \quad (4.149)$$

например, сигмоидного типа.

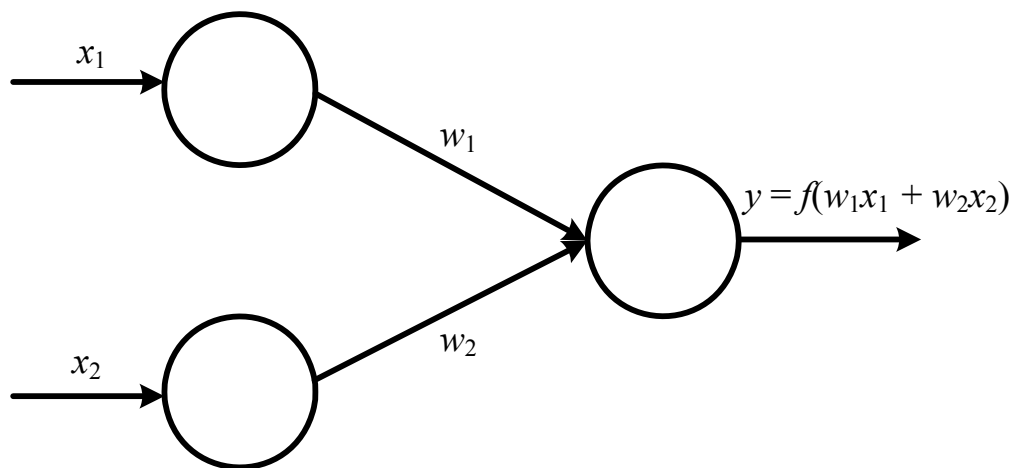


Рисунок 4.38 – Элементарная НС

Приведенную однейронную сеть, в которой используются операции умножения, суммирования и сигмоидная функция активации, будем называть *стандартной нейронной сетью*.

В случае применения других операций, таких как *t*-норма или *t*-конорма приходим к нейронной сети, которая будет называться *гибридной*.

Гибридная нейронная сеть – это нейронная сеть с четкими сигналами, весами и активационной функцией, но с объединением x_i и w_i , p_1 и p_2 с использованием *t*-нормы, *t*-конормы или некоторых других непрерывных операций (см. раздел 3.2.2).

Входы, выходы и веса гибридной нейронной сети – вещественные числа, принадлежащие отрезку $[0,1]$. Рассмотрим следующие примеры элементарных гибридных нейронных сетей.

Нечеткий нейрон «И»

Сигналы x_i и веса w_i в данном случае объединяются с помощью треугольной конормы:

$$p_i = S(w_i, x_i), \quad i = 1, 2, \dots \quad (4.150)$$

а выход образуется с применением треугольной нормы:

$$y = AND(p_1, p_2) = T(p_1, p_2) = T(S(w_1, x_1), S(w_2, x_2)). \quad (4.151)$$

Если принять $T = \min$, $S = \max$, тогда нечеткий нейрон «И» реализует композицию \min - \max :

$$y = \min\{w_1 \vee x_1, w_2 \vee x_2\}. \quad (4.152)$$

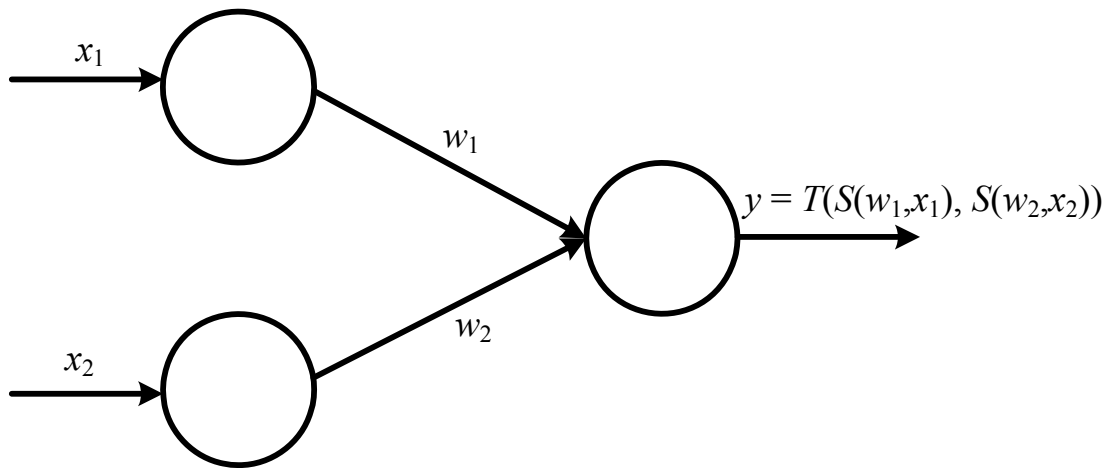


Рисунок 4.39 – Структура гибридного нейрона «И»

Нечеткий нейрон «ИЛИ»

Сигналы x_i и веса w_i здесь объединяются с помощью треугольной нормы:

$$p_i = T(w_i, x_i), \quad i = 1, 2, \quad (4.153)$$

а выход образуется с применением треугольной конормы:

$$y = OR(p_1, p_2) = S(p_1, p_2) = S(T(w_1, x_1), T(w_2, x_2)). \quad (4.154)$$

Если принять $T = \min$, $S = \max$, тогда нечеткий нейрон «ИЛИ» реализует композицию $\max - \min$:

$$y = \max \{w_1 \wedge x_1, w_2 \wedge x_2\}. \quad (4.155)$$

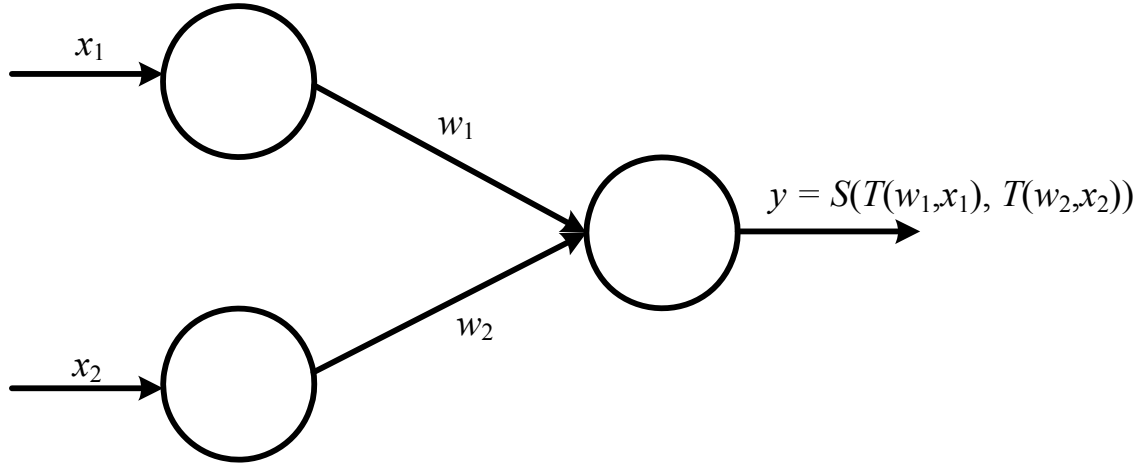


Рисунок 4.40 – Нечеткий нейрон «ИЛИ»

Алгоритм Сугено является наиболее часто используемым при построении адаптивных систем нейро-нечеткого вывода.

На основе алгоритма нечеткого логического вывода строится система рассуждений (рис. 4.41).

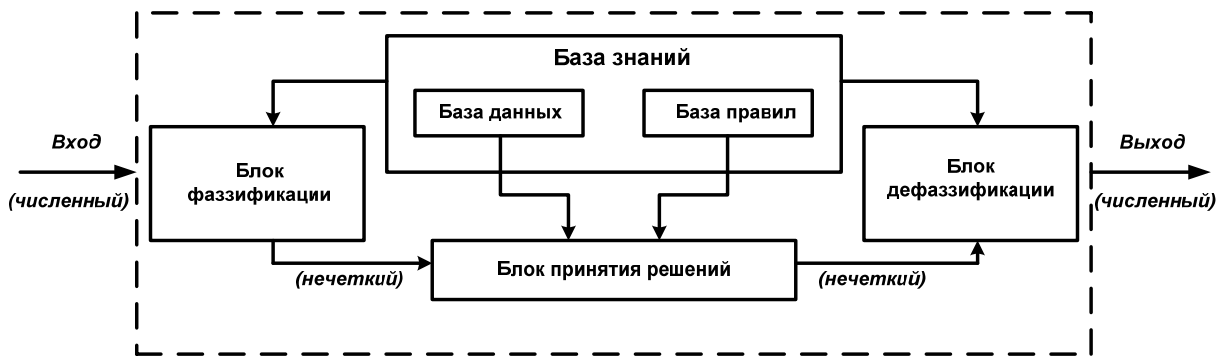


Рисунок 4.41 – Система нечетких рассуждений

Система нечетких рассуждений состоит из пяти функциональных блоков:

- блока фаззификации, преобразующего численные входные значения в степени соответствия лингвистическим переменным;

- базы правил, содержащей набор нечетких правил типа «если - то»;
- базы данных, в которой определены функции принадлежности нечетких множеств, используемых в нечетких правилах;
- блока принятия решений, совершающего операции вывода на основании имеющихся правил;
- блока дефаззификации, преобразующего результаты вывода в численные значения.

Традиционно база правил и база данных объединяются в общий блок – базу знаний. Наибольшее распространение в настоящее время [3] получили архитектуры нечеткой НС вида *ANFIS* и *TSK* (Сеть Такаги – Сугено – Канга). Доказано, что такие сети являются универсальными аппроксиматорами.

4.12.2. Сеть Такаги – Сугено – Канга

Наибольшую популярность среди нечётких систем адаптивного типа приобрела модель вывода Такаги – Сугено – Канга (*TSK*) [29]. В этой модели функция заключения определяется функциональной зависимостью. Благодаря этому, дефуззификатор на выходе системы не требуется, а модель вывода значительно упрощается.

В сети Такаги – Сугено – Канга (сокращенно, *TSK*) выходной сигнал рассчитывается с помощью выражения:

$$y(X) = \frac{\sum_{k=1}^M w_k y_i(X)}{\sum_{k=1}^M w_k}, \quad (4.156)$$

где $y_k(x) = p_{k0} + \sum_{j=1}^N p_{kj} x_j$ – k -й полиномиальный компонент аппроксимации;

$w_k = \mu_A^{(k)}(x)$ – веса компонентов рассчитываются по следующей формуле (с использованием рациональной формы функции Гаусса).

Приведенным выражениям соответствует пятислойная нейронная сеть, структурная схема которой представлена на рис. 4.42.

В такой сети выделяют 5 слоев.

1. Первый слой выполняет отдельную фаззификацию каждой пере-

менной x_i , $i = \overline{1, N}$, определяя для каждого k -го правила вывода значение ФП $\mu_A^{(k)}(x_i)$ в соответствии с функцией фаззификации. Это параметрический слой с параметрами $c_j^{(k)}$, $\sigma_j^{(k)}$, $b_j^{(k)}$, которые подлежат адаптации в процессе обучения.

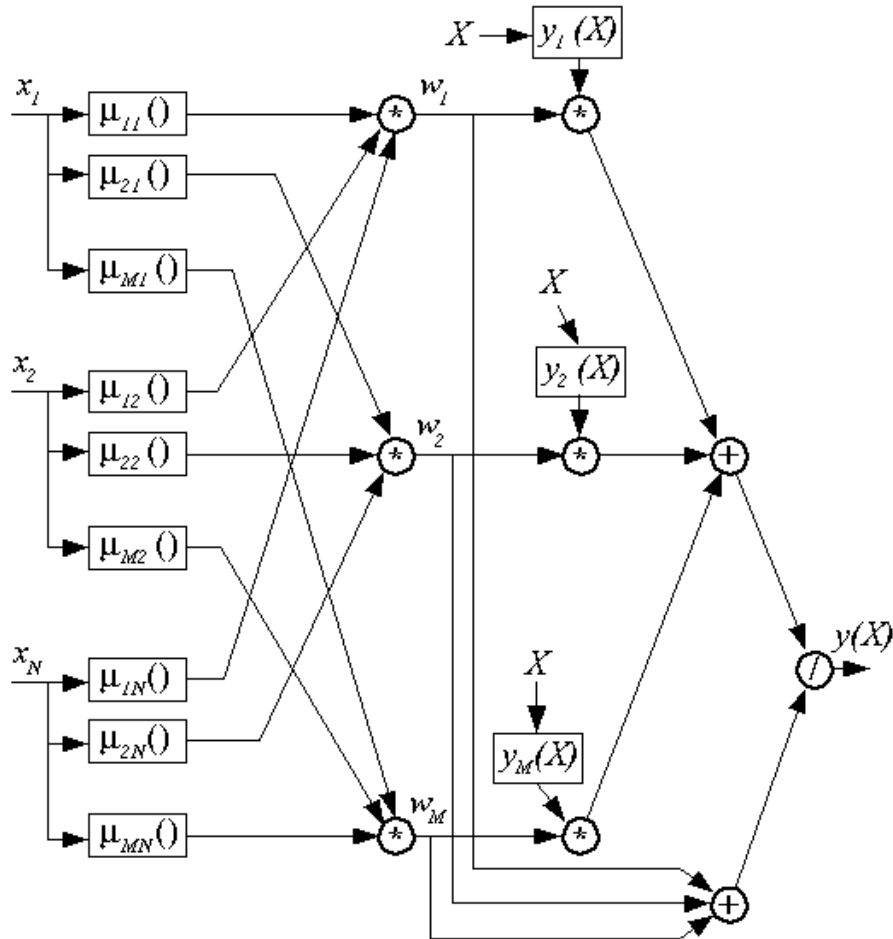


Рисунок. 4.42 – Структурная схема сети Такаги – Сугено – Канга

2. Второй слой выполняет агрегирование отдельных переменных x_i , определяя результирующую степень принадлежности $\omega_k = \mu_A^{(k)}(x)$ для вектора x условиям k -го правила. Это не параметрический слой.

3. Третий слой представляет собой генератор функции TSK, в котором рассчитывается значение $y_k(x) = p_{k0} + \sum_{j=1}^N p_{kj}x_j$. В этом слое также

происходит умножение функции $y_k(x)$ на w_k , сформированные на предыдущем слое. Это параметрический слой, в котором адаптации подлежат линейные параметры (веса), p_{k0} , p_{kj} для $j = \overline{1, N}$, $k = \overline{1, M}$, определяющие функции последствий правил.

4. Четвертый слой составляют 2 нейрона - сумматора, один из которых рассчитывает взвешенную сумму сигналов $y_k(x)$, а второй определяет

сумму весов
$$\sum_{k=1}^M \omega_k .$$

5. Пятый слой состоит из одного-единственного нейрона. В нем веса подлежат нормализации, и вычисляется выходной сигнал $y(x)$ в соответствии с выражением:

$$y(x) = \frac{f_1}{f_2} = \frac{\sum_{k=1}^M \omega_k y_k(x)}{\sum_{k=1}^M \omega_k} . \quad (4.157)$$

Это так же не параметрический слой.

Из приведенного описания следует, что нечеткая сеть *TSK* содержит только 2 параметрических слоя (первый и третий), параметры которых уточняются в процессе обучения. Параметры первого слоя $(c_j^{(k)}, \sigma_j^{(k)}, b_j^{(k)})$ будем называть нелинейными, а параметры третьего слоя $\{p_{kj}\}$ – линейными весами.

Общее выражение для функциональной зависимости для сети *TSK* задается так:

$$y(x) = \frac{1}{\sum_{k=1}^M \prod_{j=1}^M \mu_A^k(x_j)} \sum_{k=1}^M (p_{k0} + \sum_{j=1}^N p_{kj} x_j) \prod_{j=1}^N \mu_A^{(k)}(x_j) . \quad (4.158)$$

Рассмотрим гибридный алгоритм обучения. В гибридном алгоритме параметры, подлежащие адаптации, делятся на 2 группы. Первая из них состоит из линейных параметров p_{kj} третьего слоя, а вторая группа – из

параметров нелинейной ФП первого слоя. Уточнение параметров происходит в 2 этапа.

На первом этапе при фиксации отдельных значений параметров функции принадлежности, решая систему линейных уравнений, рассчитываются линейные параметры p_{kj} полинома TSK . При известных значениях ФП зависимость для выхода можно представить в виде линейной формы относительно параметра p_{kj} :

$$y_k(x) = \sum_{k=1}^M \omega_k' (p_{k0} + \sum_{j=1}^N p_{kj} x_j), \quad (4.159)$$

где $\omega_k' = \frac{\prod_{j=1}^N \mu_A^{(k)}(x_j)}{\sum_{r=1}^M \prod_{j=1}^N \mu_A^{(r)}(x_j)}$, $k = \overline{1, M}$.

При размерности обучающей выборки $L(x^{(l)}, d^{(l)})$, ($l = 1, 2, \dots, L$) и замене выходного сигнала сети ожидаемым значением $d^{(l)}$ получим систему из L линейных уравнений вида:

$$\begin{bmatrix} \omega_{11}' & \omega_{11}'x_1^{(1)} & \dots & \omega_{11}'x_N^{(1)} & \dots & \omega_{1M}' & \omega_{1M}'x_1^{(1)} & \dots & \omega_{1M}'x_N^{(1)} \\ \omega_{21}' & \omega_{21}'x_1^{(2)} & \dots & \omega_{21}'x_N^{(2)} & \dots & \omega_{2M}' & \omega_{2M}'x_1^{(2)} & \dots & \omega_{2M}'x_N^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \omega_{L1}' & \omega_{L1}'x_1^{(L)} & \dots & \omega_{L1}'x_N^{(L)} & \dots & \omega_{LM}' & \omega_{LM}'x_1^{(L)} & \dots & \omega_{LM}'x_N^{(L)} \end{bmatrix} \times \begin{bmatrix} p_{10} \\ p_{11} \\ \dots \\ p_{1N} \\ \dots \\ p_{M0} \\ p_{M1} \\ \dots \\ p_{MN} \end{bmatrix} = \begin{bmatrix} d^{(1)} \\ d^{(2)} \\ \dots \\ d^{(L)} \end{bmatrix}, \quad (4.160)$$

где ω_{il}' – означает уровень активации (вес) условия i -го правила при предъявлении l -го входного вектора x^l .

Это выражение можно записать в матричном виде:

$$Ap = d. \quad (4.161)$$

Размерность матрицы A равняется $L(N + 1)M$. При этом количество строк L обычно бывает значительно больше количества столбцов

$(N+1)M$. Решение этой системы уравнений можно получить как обычными методами, так и за один шаг, используя псевдоинверсию матрицы A :

$$p = A^+ d, \quad (4.162)$$

где A^+ – псевдоинверсная матрица.

На втором этапе после фиксации значения линейных параметров p_{kj} рассчитываются фактические выходные сигналы $y^{(l)}$, $l=1,2,\dots,L$, для этого используется линейная зависимость:

$$y^{(L)} = Ap. \quad (4.163)$$

После этого рассчитывается вектор ошибки $\varepsilon = y - d$ и критерий

$$E = \frac{1}{2} \sum_{l=1}^L (y(x^{(l)}) - d^{(l)})^2. \quad (4.164)$$

Сигналы ошибок направляются через сеть в обратном порядке согласно методу *Back Propagation* вплоть до первого слоя, где могут быть рассчитаны компоненты $c_j^{(k)}$, $\sigma_j^{(k)}$, $b_j^{(k)}$. После вычисления вектора градиента делается шаг спуска градиентным методом. Соответствующие формулы обучения (для самого простого метода быстрого спуска) принимают вид:

$$c_j^{(k)}(n+1) = c_j^{(k)}(n) - \eta_c \frac{\partial E(n)}{\partial c_j^{(k)}}, \quad (4.165)$$

$$\sigma_j^{(k)}(n+1) = \sigma_j^{(k)}(n) - \eta_\sigma \frac{\partial E(n)}{\partial \sigma_j^{(k)}}, \quad (4.166)$$

$$b_j^{(k)}(n+1) = b_j^{(k)}(n) - \eta_b \frac{\partial E(n)}{\partial b_j^{(k)}}, \quad (4.167)$$

где n – номер итерации.

После уточнения нелинейных параметров снова запускается процесс адаптации линейных параметров функции *TSK* (первый этап) и нелинейных параметров (второй этап). Этот цикл продолжается до тех пор, пока не стабилизируются все параметры процесса.

В качестве примера рассмотрим обобщенную колоколообразную функцию принадлежности:

$$\mu_A(x) = \frac{1}{1 + \left(\frac{x-c}{\sigma}\right)^{2b}}, \quad (4.168)$$

соответствующие формулы градиентного метода целевой функции для одной пары данных (x, d) принимают вид:

$$\frac{\partial E}{\partial c_j^{(k)}} = (y(x) - d) \sum_{r=1}^M (p_{r0} + \sum_{j=1}^N p_{rj} x_j) \cdot \frac{\partial w_r'}{\partial c_j^{(k)}}, \quad (4.169)$$

$$\frac{\partial E}{\partial \sigma_j^{(k)}} = (y(x) - d) \sum_{r=1}^M (p_{r0} + \sum_{j=1}^N p_{rj} x_j) \cdot \frac{\partial w_r'}{\partial \sigma_j^{(k)}}, \quad (4.170)$$

$$\frac{\partial E}{\partial b_j^{(k)}} = (y(x) - d) \sum_{r=1}^M (p_{r0} + \sum_{j=1}^N p_{rj} x_j) \cdot \frac{\partial w_r'}{\partial b_j^{(k)}}. \quad (4.171)$$

Соответствующие производные

$$\frac{\partial w_r'}{\partial c_j^{(k)}}, \quad \frac{\partial w_r'}{\partial \sigma_j^{(k)}}, \quad \frac{\partial w_r'}{\partial b_j^{(k)}}$$

принимают следующий вид:

$$\frac{\partial w_r'}{\partial c_j^{(k)}} = \frac{\delta_{rk} m(x_j) - l(x_j)}{[m(x_j)]^2} \prod_{i=1, i \neq j}^M \frac{\left[\frac{2b_j^{(k)}}{\sigma_j^{(k)}} \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)} - 1} \right]}{\left[1 + \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)}} \right]^2} \mu_A^{(k)}(x_i), \quad (4.172)$$

$$\frac{\partial w_r'}{\partial \sigma_j^{(k)}} = \frac{\delta_{rk} m(x_j) - l(x_j)}{[m(x_j)]^2} \prod_{i=1, i \neq j}^M \frac{\left[\frac{2b_j^{(k)}}{\sigma_j^{(k)}} \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)}} \right]}{\left[1 + \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)}} \right]^2} \mu_A^{(k)}(x_i), \quad (4.173)$$

$$\frac{\partial w_r'}{\partial b_j^{(k)}} = \frac{\delta_{rk} m(x_j) - l(x_j)}{[m(x_j)]^2} \prod_{i=1, i \neq j}^M \frac{\left[-2 \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)}} \ln \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right) \right]}{\left[1 + \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)}} \right]^2} \mu_A^{(k)}(x_j) \quad (4.174)$$

для $r = 1, 2, \dots, M$,

где δ_{rk} – дельта Кронекера, $l(x_j) = \prod_{j=1}^N \mu_A^{(k)}(x_j)$; $m(x_j) = \sum_{k=1}^M \prod_{j=1}^N \mu_A^{(k)}(x_j)$.

При практической реализации гибридного метода обучения нечетких сетей доминирующим фактором их адаптации считается первый этап, на котором веса p_{kj} подбираются с использованием псевдоинверсии за один шаг. Для уравнивания его влияния второй этап много раз повторяется в каждом цикле.

Представленный гибридный алгоритм – один из наиболее эффективных способов обучения нечетких нейронных сетей. Его характерной чертой является разделение процесса на два отделенных во времени этапа. Если учитывать, что расчетная сложность каждого алгоритма оптимизации нелинейно зависит от количества параметров, которые подлежат оптимизации, то уменьшение размерности задач оптимизации значительно сокращает объем расчетных операций и повышает скорость сходимости алгоритма. Благодаря этому гибридный алгоритм является более эффективным в сравнении с обычным градиентным методом.

4.12.3. ANFIS-сети

Рассмотрим класс адаптивных сетей функционально эквивалентных системам нечетких рассуждений [3; 27; 28]. Подобная архитектура носит название *ANFIS* (это аббревиатура *Adaptive - Network - Based Fuzzy Inference System* – адаптивная сеть нечеткого вывода). Она была предложена Янгом (*Jang*) в начале девяностых. *ANFIS* является одним из первых вариантов гибридных нейро-нечетких сетей – нейронной сети прямого пространства сигнала особого типа. Архитектура нейронечеткой сети изоморфна (подобна) нечеткой базе знаний. В нейро-нечетких сетях используются дифференцируемые реализации треугольных норм (умножение и

вероятностное ИЛИ), а также гладкие функции принадлежности. Это позволяет применять для настройки нейро-нечетких сетей быстрые алгоритмы обучения нейронных сетей, основанные на методе обратного распространения ошибки.

Ниже описываются архитектура и правила функционирования каждого слоя *ANFIS*-сети. *ANFIS* реализует систему нечеткого вывода Сугено в виде пятислойной нейронной сети прямого распространения сигнала. Назначение слоев следующее:

- первый слой – термы входных переменных;
- второй слой – антецеденты (посылки) нечетких правил;
- третий слой – нормализация степеней выполнения правил;
- четвертый слой – заключения правил;
- пятый слой – агрегирование результата, полученного по различным правилам.

Входы сети в отдельный слой не выделяются. На рис. 4.43 изображена *ANFIS* - сеть с двумя входными переменными (x_1 и x_2) и четырьмя нечеткими правилами.

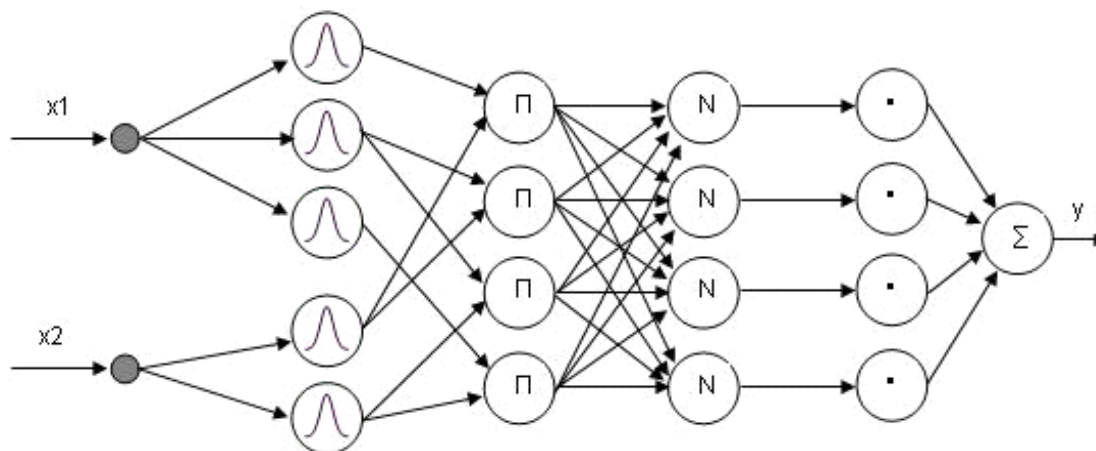


Рисунок 4.43 – Пример *ANFIS* - сети

Для лингвистической оценки входной переменной x_1 используется 3 терма, для переменной x_2 – 2 терма.

ANFIS - сеть функционирует следующим образом.

1-й слой. Термы входных переменных. Каждый узел первого слоя представляет один терм с функцией принадлежности. Количество узлов первого слоя равно сумме мощностей терм-множеств входных переменных. Выходом узла является степень принадлежности значения входной

переменной соответствующему нечеткому терму. Другими словами, $\mu_{A_j}(x_i)$ – функция принадлежности переменной A_j , определяющей степень, с которой x_i удовлетворяет A_j . Параметры этого слоя относятся к так называемым параметрам предпосылок.

2-й слой. Антецеденты (посылки) нечетких правил. Каждый узел данного слоя является фиксированным узлом, перемножающим входные сигналы, причем выходное значение узла τ_i представляет собой вес некоторого правила:

$$\tau_i = \prod \mu_{ij}. \quad (4.175)$$

Количество узлов второго слоя равно m . Каждый узел этого слоя соответствует одному нечеткому правилу. Узел второго слоя соединен с теми узлами первого слоя, которые формируют антецеденты соответствующего правила. Следовательно, каждый узел второго слоя может принимать от 1 до n входных сигналов. Выходом узла является степень выполнения правила, которая рассчитывается как произведение входных сигналов.

3-й слой. Нормализация степеней выполнения правил. Каждый i -й узел данного слоя определяет отношение веса i -го правила к сумме весов всех правил:

$$\hat{\tau}_i = \frac{\tau_i}{\sum_j \tau_j}. \quad (4.176)$$

Выходные сигналы 3-го слоя называются нормализованными весами. Количество узлов третьего слоя также равно m . Каждый узел этого слоя рассчитывает относительную степень выполнения нечеткого правила.

4-й слой. Заключение правил. Узлы данного слоя определяются линейными функциями принадлежности выходных переменных:

$$y_i = c_0 + \sum c_j x_j, \quad (4.177)$$

где y_i – линейная функция принадлежности; x_j – значение входа j ; c_j – параметр линейного уравнения, рассчитанный следующим образом:

$$c_i = w_i \cdot p_i, \text{ где } w_i = \frac{1}{\max(x_i)}, \quad (4.178)$$

где x_j – параметр функции принадлежности входного слоя; p_i – параметр функции принадлежности выходного слоя.

Количество узлов четвертого слоя также равно m . Каждый узел соединен с одним узлом третьего слоя, а также со всеми входами.

5-й слой. Агрегирование результата, полученного по различным правилам. Единственный узел данного слоя является фиксированным узлом, в котором вычисляется полное выходное значение адаптивной сети Y как сумма всех входных сигналов:

$$Y = \sum \hat{\tau}_j \cdot y_i. \quad (4.179)$$

Способ реализации ANFIS-сети

База правил. База правил представляет собой множество нечетких правил $R^k, k = 1, \dots, N$ вида:

$$\begin{aligned} R^k : & \mathbf{IF}(x_1 \text{ – это } A_1^k \text{ AND } x_2 \text{ – это } A_2^k \dots \text{ AND } x_n \text{ – это } A_n^k) \\ & \mathbf{THEN} (y_1 \text{ – это } B_1^k \text{ AND } y_2 \text{ – это } B_2^k \dots \text{ AND } y_m \text{ – это } B_m^k), \end{aligned} \quad (4.180)$$

где N – количество нечетких правил; A_j^k, B_j^k – нечеткие множества.

При скалярном выводе данная формула примет вид:

$$\begin{aligned} R^k : & \mathbf{IF}(x_1 \text{ – это } A_1^k \text{ AND } x_2 \text{ – это } A_2^k \dots \text{ AND } x_n \text{ – это } A_n^k) \\ & \mathbf{THEN} (y \text{ – это } B^k). \end{aligned} \quad (4.181)$$

Таким образом, база правил системы нечеткого вывода представляется в виде совокупности нечетких предикатных правил вида:

Π_1 : если x есть A_1 , тогда y есть B_1 ,

Π_2 : если x есть A_2 , тогда y есть B_2 ,

...

Π_n : если x есть A_n , тогда y есть B_n .

Если ввести обозначения $x = x_1 \times x_2 \times \dots \times x_n$, $A = A_1 \times A_2 \times \dots \times A_n$, то правило можно представить в виде нечеткой импликации:

$$R^k : A^k \rightarrow B^k, k = 1, \dots, N. \quad (4.182)$$

Правило R^k можно интерпретировать как нечеткое отношение, определенное на множестве $x \times y$, то есть $R^k \subseteq x \times y$ – нечеткое множество с функцией принадлежности:

$$\mu_{R^k}(x, y) = \mu_{A^k \rightarrow B^k}(x, y). \quad (4.183)$$

Если в качестве нечеткой импликации будет использоваться операция умножения, то получим формулу:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \mu_B(y). \quad (4.184)$$

Блок вывода (блок принятия решения). Для определения функции принадлежности нечеткого множества B^k можно воспользоваться формулой:

$$\mu_{B^k}(y) = \sup \left\{ \mu_{B^k}(y) \prod_{i=1}^n \mu_{A_i'}(x_i) \mu_{A^k}(x_i) \right\}. \quad (4.185)$$

Блок фаззификации. Применим операцию типа синглетон (*Singleton* применяется, если должен существовать только один экземпляр некоторого класса, и этому экземпляру нужна глобальная точка доступа):

$$A'(x) = \begin{cases} 1, & x = \bar{x}, \\ 0, & x \neq \bar{x} \end{cases}, \quad (4.186)$$

супремум в (4.185) достигается при $\mu_{A_i'}(\bar{x}) = 1$, при этом выражение (4.185) принимает вид:

$$\mu_{B^k}(y) = \left\{ \mu_{B^k}(y) \prod_{i=1}^n \mu_{A^k}(\bar{x}_i) \right\}. \quad (4.187)$$

Блок дефаззификации. Применим метод *Центра тяжести (CoGS, Centre of Gravity for Singletons)*:

$$\bar{y} = \frac{\sum_{i=1}^N \bar{y}^k \cdot \mu_{\bar{B}^k}(\bar{y}^k)}{\sum_{i=1}^N \mu_{\bar{B}^k}(\bar{y}^k)}, \quad (4.188)$$

где \bar{y}^k – центр нечеткого множества B^k , то есть точка, в которой функция принадлежности нечеткого множества B^k достигает максимума.

При подстановке выражения (4.187) в формулу (4.188) и учитывая, что максимальное значение в точке $\bar{y}^k=1$, можно записать:

$$\bar{y} = \frac{\sum_{i=1}^N \bar{y}^k \left(\prod_{i=1}^n \mu_{A_i}^k(\bar{x}^k) \right)}{\sum_{i=1}^N \left(\prod_{i=1}^n \mu_{A_i}^k(\bar{x}^k) \right)}. \quad (4.189)$$

Завершающий этап проектирования – это определение формы и параметров функций принадлежности нечетких множеств A .

Например, это может быть функция Гаусса

$$\mu_{A_i}^k(x_1) = e^{-\frac{(x_1 - \bar{x}_i^k)^2}{\sigma_i^k}}, \quad (4.190)$$

где \bar{x}_i^k – это центр; σ_i^k – ширина гауссовской кривой.

Объединим все представленные элементы. Воспользуемся выражениями (4.188), (4.185), (4.186) и (4.190) и тогда окончательно имеем:

$$\bar{y} = \frac{\sum_{i=1}^N \bar{y}^k \left(\prod_{i=1}^n e^{-\frac{(x_i - \bar{x}_i^k)^2}{\sigma_i^k}} \right)}{\sum_{i=1}^N \left(\prod_{i=1}^n e^{-\frac{(x_i - \bar{x}_i^k)^2}{\sigma_i^k}} \right)}. \quad (4.191)$$

Приведенное выражение представляет собой один из способов реализации *ANFIS* - сети.

4.12.4. Применение адаптивной системы нейро-нечеткого вывода на примере решения задачи прогнозирования временных рядов

С помощью адаптивной сети нейро - нечеткого вывода задача прогнозирования формализуется через задачу распознавания образов [3].

Данные о прогнозируемой переменной за некоторый промежуток времени образуют образ, класс которого определяется значением прогно-

зируемой переменной в некоторый момент времени за пределами данного промежутка, то есть значением переменной через интервал прогнозирования.

Прогнозирование осуществляется по тому же принципу, что и формирование обучающей выборки. При этом выделяются две возможности: одношаговое и многошаговое прогнозирование. Многошаговое прогнозирование используется для осуществления долгосрочного прогноза и предназначено для определения основного тренда и главных точек изменения тренда для некоторого промежутка времени в будущем. Одношаговое прогнозирование используется для краткосрочных прогнозов, обычно – абсолютных значений последовательности. Осуществляется прогноз только на один шаг вперед, но используется реальное, а не прогнозируемое значение для осуществления прогноза на следующем шаге.

Прогнозирующая нейронная сеть должна иметь всего один выход и столько входов, сколько предыдущих значений мы хотим использовать для прогноза, например, 4 последних значения.

Рассмотрим основные этапы проектирования и обучения нечеткой нейронной сети для решения задачи прогнозирования. Проектирование адаптивной системы нейро-нечеткого вывода, как и любой нейронной сети, состоит из построения системы и ее обучения. Построение состоит из проектирования базы знаний, которая состоит из базы данных, хранящей параметры функции принадлежности в виде системы нечеткого вывода и базы правил генерируемых на основе базы данных. Таким образом, алгоритм решения можно представить в виде диаграммы активности (рис. 4.44).

Проектирование нейро-нечеткой сети реализуется в несколько шагов:

- 1) генерация *FIS* - структуры – в рамках данного этапа определяются параметры функций принадлежности;
- 2) генерация базы правил – на данном этапе определяется число правил, предпосылки и заключения для каждого правила;
- 3) задаются параметры обучения (количество эпох, минимальная ошибка обучения и скорость обучения сети) и начальные параметры (обнуляется счетчик эпох и другие переменные, используемые при обучении);

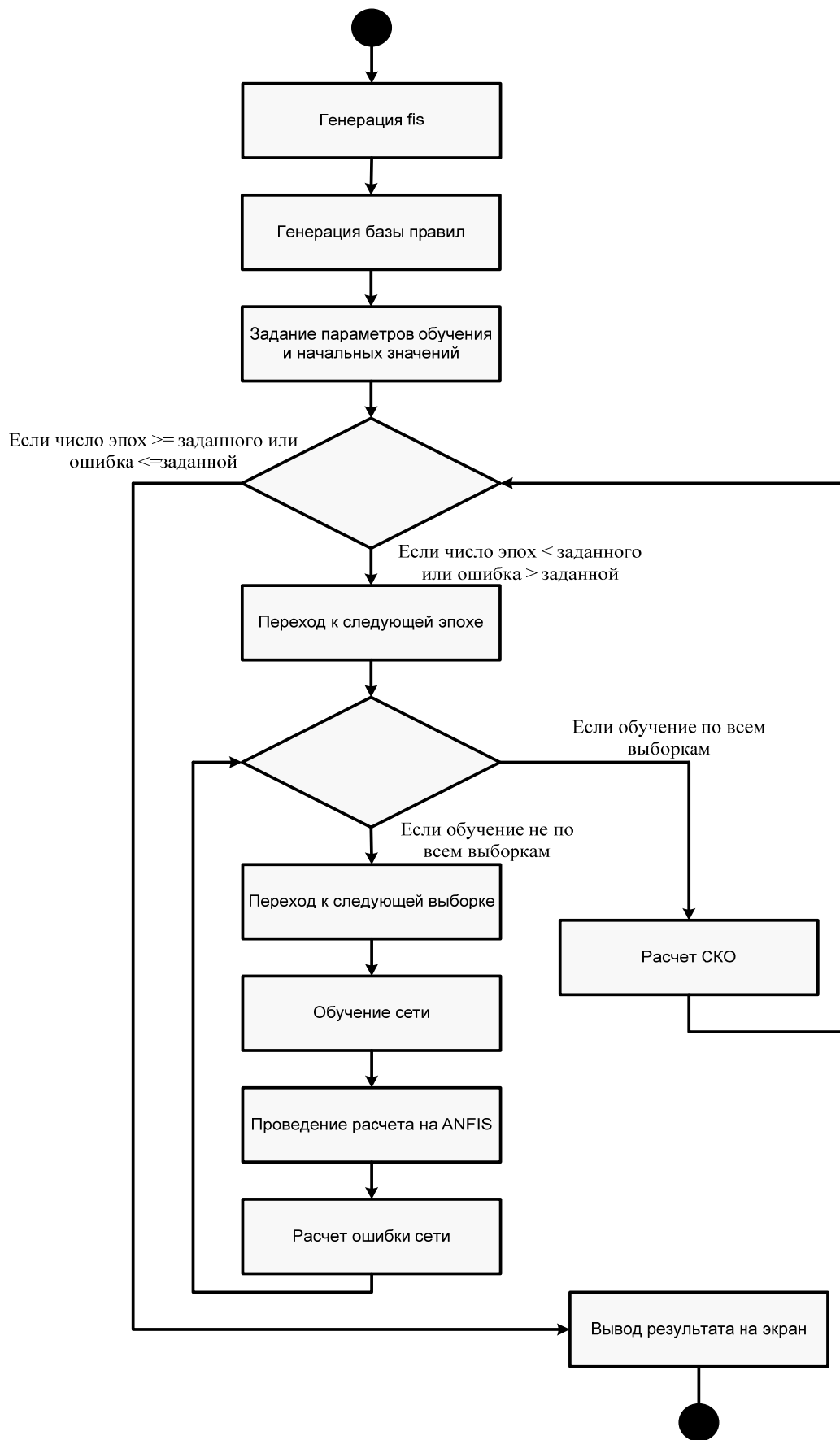


Рисунок 4.44 – Алгоритм функционирования сети

4) если количество эпох меньше заданного, или требуемая ошибка больше среднеквадратичной, начинаем новую эпоху обучения и переходим на шаг 5, иначе – на шаг 9;

5) если есть необученные выборки, то считываем данные очередной выборки и переходим на этап 6, иначе – на этап 8;

6) обучение сети по текущей выборке методом обратного распространения ошибки;

7) определяем ошибку после обучения и переходим на шаг алгоритма 5;

8) рассчитываем ошибку обучения как среднеквадратичное отклонение ошибки каждой выборки и переходим на шаг 4;

9) вывод результатов на экран.

Первый и второй этап характеризуют создание базы знаний, на этапах 3–8 происходит обучение созданной сети.

Этап обучения сети. Когда в сети только один слой, алгоритм ее обучения с учителем довольно очевиден, так как правильные выходные состояния нейронов единственного слоя заведомо известны, и подстройка синаптических связей идет в направлении, минимизирующем ошибку на выходе сети.

По этому принципу строится, например, алгоритм обучения однослойного персептрона. В многослойных же сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, не известны, и двух- или многослойный персептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах нейронной сети.

Один из вариантов решения этой проблемы – разработка наборов выходных сигналов, соответствующих входным, для каждого слоя нейронной сети, что, конечно, является очень трудоемкой операцией и не всегда осуществимо. Вторым вариантом – динамическая подстройка весовых коэффициентов синапсов, в ходе которой выбирают, как правило, наиболее слабые связи и изменяются на малую величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети.

И, наконец, третий, более приемлемый вариант – распространение сигналов ошибки от выходов нейронной сети к ее входам в направлении, обратном прямому распространению сигналов в обычном режиме работы.

На рис. 4.45, представлен процесс обучения нейросети. Этот алгоритм обучения нейронной сети получил название процедуры обратного распространения (см. раздел 4.6.5).

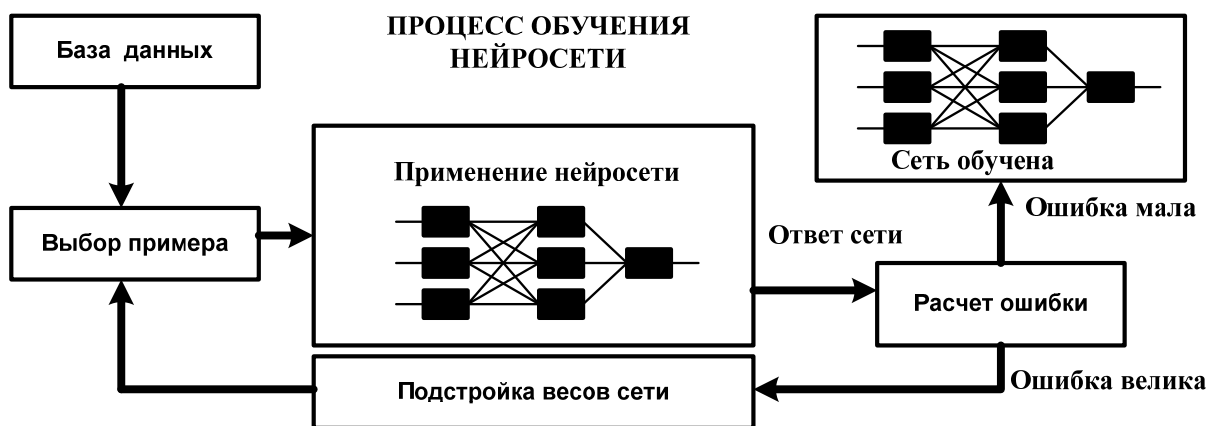


Рисунок 4.45 – Процесс обучения сети

Типовые процедуры обучения нейронных сетей могут быть применены для настройки *ANFIS*-сети, так как в ней используются только дифференцируемые функции. Обычно применяется комбинация градиентного спуска в виде алгоритма обратного распространения ошибки и метода наименьших квадратов.

Алгоритм обратного распространения ошибки настраивает параметры antecedent rules, то есть функций принадлежности. Методом наименьших квадратов оцениваются коэффициенты заключений правил, так как они линейно связаны с выходом сети. Каждая итерация процедуры настройки выполняется в два этапа.

На первом этапе на входы подается обучающая выборка, и по невязке (разности) между желаемым и действительным поведением сети итерационным методом наименьших квадратов находятся оптимальные параметры узлов четвертого слоя. На втором этапе остаточная невязка передается с выхода сети на входы, и методом обратного распространения ошибки модифицируются параметры узлов первого слоя. При этом найденные на первом этапе коэффициенты заключений правил не изменяются. Во время прямого прохода определяются параметры вывода методом наименьших квадратов. Во время обратного прохода обновляются параметры предпосылок с помощью метода снижения градиента. Итерационная про-

цедура настройки продолжается, пока невязка превышает заранее установленное значение.

Таким образом, при прямом проходе модифицируются параметры заключений правил по следующей формуле:

$$y_i(t+1) = y_i(t) + \gamma \cdot \frac{\partial E}{\partial y_i(t)} = y_i(t) + \frac{\gamma(y^* - y) \cdot \prod_j \mu_{ik}(x_j)}{\sum_j \prod_n \mu_{nm}(x_n)}, \quad (4.192)$$

где γ – константа обучения; t – эпоха обучения; i – номер правила; y^* – требуемое значение выхода сети; y – полученное значение выхода сети; k – j -я предпосылка правила i ; m - n -я предпосылка правила j .

Во время обратного прохода обновляются параметры предпосылок с помощью метода снижения градиента. Например, при функции принадлежности Гаусса формулы модификации параметров a и b определены следующим образом:

$$a_{di}(t+1) = a_{di}(t) + \frac{\lambda(y^* - y) \cdot \sum_j (y_k - y) \prod_{l < j} \mu_{lj}(x_l)}{\sum_j \prod_n \mu_{nm}(x_n)} \times \frac{(x_d - a_{di}) \cdot \mu_{di}(x_d)}{b_{di}^2}, \quad (4.193)$$

$$b_{di}(t+1) = b_{di}(t) + \frac{\beta(y^* - y) \cdot \sum_j (y_k - y) \prod_{l < j} \mu_{lj}(x_l)}{\sum_j \prod_n \mu_{nm}(x_n)} \times \frac{(x_d - a_{di})^2 \cdot \mu_{di}(x_d)}{b_{di}^3}. \quad (4.194)$$

В качестве ошибки обучения сети используется среднеквадратичная ошибка, рассчитанная по следующей формуле:

$$E = \sqrt{\frac{\sum_{i=1}^n (y_i^* - y_i)^2}{n-1}}, \quad (4.195)$$

где n – число выборок для обучения.

Алгоритм обучения состоит из определения правил, влияющих на результат при данных значениях входов сети, определения необходимого

приращения заключения правил, настройки заключений правил, определения значения выхода сети при настроенных параметрах, определения остаточной ошибки путем нахождения разницы между выходом сети и требуемым значением сети, определения приращения предпосылок правил и настройки предпосылок правил.

Схема алгоритма обучения нейро-нечеткой сети в виде диаграммы активности представлена на рис. 4.46.

Алгоритм обучения адаптивной системы нейро-нечеткого вывода можно разбить на следующие шаги.

1. Определяем для выборки правила, которые влияют на результат, это те правила $\hat{\tau}_j$ [см. (4.176)], которые не равны нулю.

2. Если есть нерассмотренные правила, влияющие на результат, считываем очередное правило и переходим на шаг 3, иначе – на шаг 5.

3. Рассчитываем приращение для параметров функций принадлежности, определенных в заключении рассматриваемого правила.

4. Изменяем значение параметров функций принадлежности на рассчитанную величину по формуле (4.192) и переходим на шаг 2.

5. Рассчитываем выходное значение сети после изменения параметров выходного слоя.

6. Определяем остаточную ошибку.

7. Если есть нерассмотренные правила, влияющие на результат, считываем очередное правило и переходим на шаг 8, иначе – на шаг 12.

8. По предпосылкам правил определяем, какие узлы первого слоя влияют на результат.

9. Рассчитываем производную функции принадлежности для определенных на предыдущем шаге узлов.

10. Рассчитываем приращение для параметров функций принадлежности по формуле.

11. Изменяем значение параметров функций принадлежности на рассчитанную величину по формулам (4.193), (4.194) и переходим на шаг 7.

12. Завершаем обучение и выходим из процедуры.

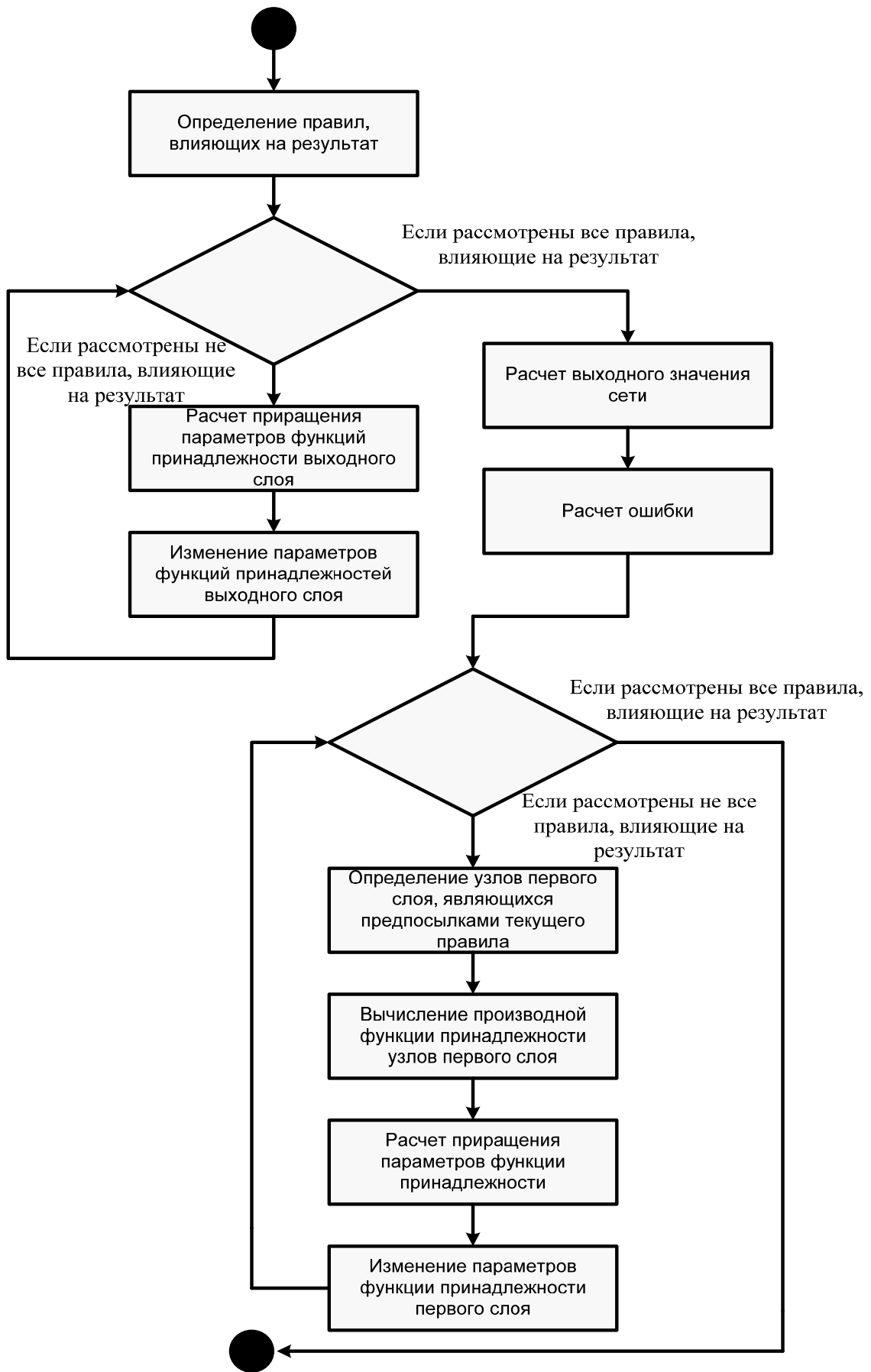


Рисунок 4.46 – Алгоритм обучения сети

Третий и четвертый шаги алгоритма реализуют обучение узлов выходного слоя, пятый и шестой шаги вычисляют остаточную ошибку после обучения выходного слоя и реализуют обучение узлов первого слоя методом градиентного спуска.

4.13. Прогнозирование значений показателей качества трансформаторных масел с помощью нейро-нечеткой системы *ANFIS*

Прогнозирование остаточного ресурса высоковольтных маслонаполненных трансформаторов является одним из приоритетных направлений при решении задач повышения эксплуатационной надежности и продления срока службы высоковольтного оборудования. Одним из элементов электрической изоляции, состояние которого во многом определяет срок службы всего трансформатора, является трансформаторное масло. Основная сложность при решении задач прогноза значений трансформаторного масла заключается в том, что для построения адекватной модели прогноза необходимо учитывать не только влияние условий эксплуатации и наличие корреляционных связей между показателями качества изоляции, но и сложный, нелинейный характер дрейфа показателей качества масла во времени. Задачу значительно усложняет то обстоятельство, что исходная информация об уровне эксплуатационных воздействий является неполной и в ряде случаев условно определенной. Поэтому для решения задачи прогноза целесообразно использовать математический аппарат, который бы учитывал как нелинейный характер зависимостей показателей от времени, так и нечеткость исходной информации, например, нечеткие нейронные сети.

В качестве исходных данных использовались результаты периодических испытаний трансформаторного масла по Донецкой, Луганской, Полтавской, Сумской, Харьковской областям Украины, а также АР Крым, за период с 1959 по 2010г. Всего были проанализированы результаты наблюдений по 249 трансформаторам напряжением 110 кВ и номинальной мощностью от 6,3 до 63 МВА. Общий объем выборки составил свыше 44 тысяч (44 728 значений по двадцати показателям масла).

Физическими основами для прогнозирования являлись результаты следующих исследований:

1) в работе [30] установлено, что зависимости показателей качества трансформаторного масла от времени эксплуатации имеют сложный нелинейный характер, что обосновывает необходимость использования для прогнозирования аппарата нечетких нейронных сетей;

2) в работе [31] показано наличие практически функциональной зависимости между характеристиками нагрузки трансформаторов и интенсивностью дрейфа показателей качества масла;

3) исследования, выполненные в [32], позволили выявить показатели качества трансформаторного масла, между которыми имеется значимая статистическая связь.

В процессе исследований анализировалась возможность:

1) прогнозирования значений кислотного числа масла по известным значениям загрузки трансформатора и длительности эксплуатации;

2) прогнозирования значений цвета масла по значению кислотного числа и длительности эксплуатации;

3) прогнозирования значений содержания в масле водорастворимых кислот по известным значениям загрузки трансформатора, значению кислотного числа и длительности эксплуатации.

Для численной оценки прогнозирования значений показателей качества масла разрабатывался алгоритм, реализованный на ЭВМ в среде *Matlab Version 7.0 (free version)* со встроенным пакетом *Fuzzy Logic Toolbox*.

При разработке нечеткой нейронной сети для прогнозирования рассматривалось применение следующих алгоритмов нечеткого вывода: алгоритм Мамдани и алгоритм Сугено [27]. В последствии на основе тестирования сетей, созданных с применением этих алгоритмов, была выбрана нечеткая нейронная сеть, в которой реализуется алгоритм Сугено. Это связано с тем, что нечеткая нейронная сеть с алгоритмом Сугено имеет меньшую ошибку обучения и ошибку прогнозирования. Далее для прогнозирования значений кислотного числа и цвета масла разрабатывались нечеткие нейронные сети с алгоритмом Сугено.

4.13.1. Прогнозирование кислотного числа трансформаторного масла

Учитывая наличие функциональной связи между временем эксплуатации, загрузкой трансформаторов и кислотным числом масел, прогнозирование значений кислотного числа проводилось следующим образом.

Входными переменными являлись:

- длительность эксплуатации в годах;
- средняя нагрузка трансформаторов за анализируемый период в процентах.

В качестве исходных данных использовались результаты измерений кислотного числа масла, значения времени эксплуатации и средней загрузки трансформаторов по данным АК «Харьковоблэнерго» [33]. Используя пакет *Fuzzy Logic Toolbox*, встроенный в среду *Matlab Version 7.0 (free version)*, была разработана нейро-нечеткая сеть. Общий вид созданной нечеткой нейронной сети представлен на рис. 4.47.

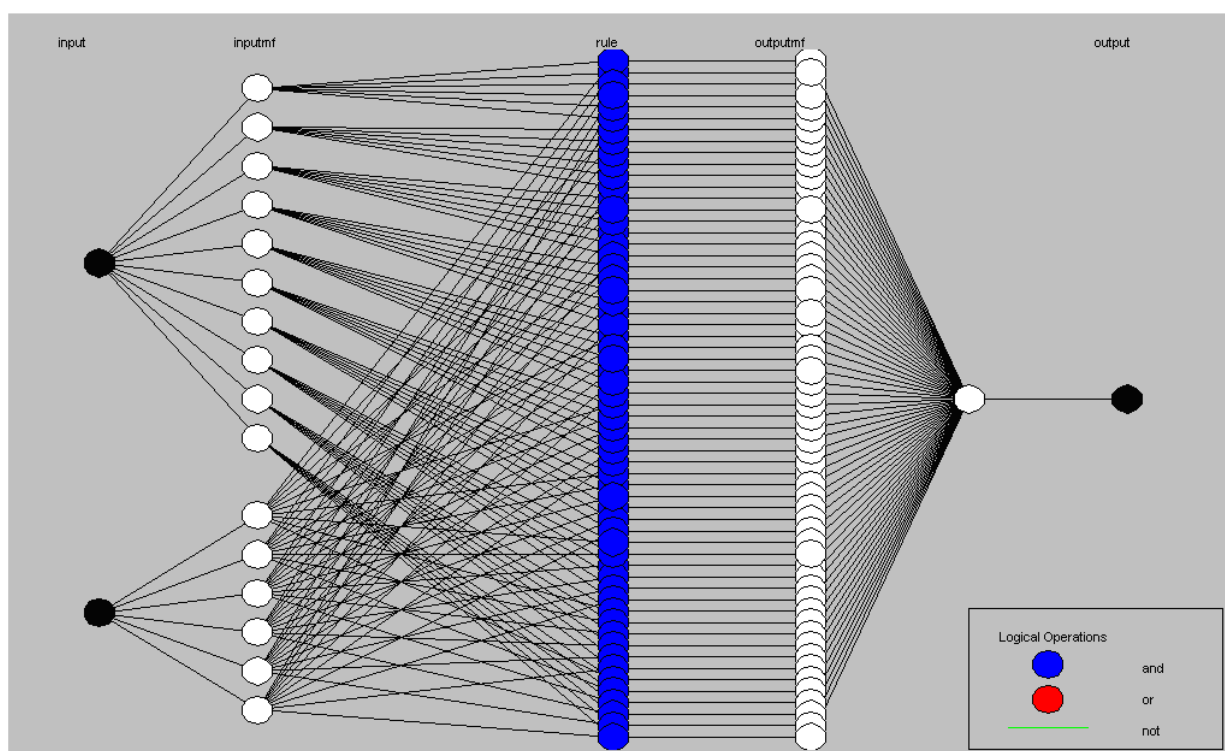


Рисунок 4.47 – Топология нечеткой нейронной сети для прогнозирования кислотного числа трансформаторных масел

У созданной сети 16 входов, 10 входов по времени и 6 входов по мощности. Было выбрано две функции принадлежности для каждой входной переменной. В качестве функций принадлежности для каждого из термов была выбрана трапецеидальная функция, математическое выражение которой имеет вид:

$$f_T(x, a, b, c, d) = \left. \begin{array}{l} 0, \quad x \leq a \\ \frac{x-a}{b-a}, \quad a \leq x \leq b \\ 1, \quad b \leq x \leq c \\ \frac{d-x}{d-c}, \quad c \leq x \leq d \\ 0, \quad d \leq x \end{array} \right\}. \quad (4.196)$$

Для выходного параметра тип функции принадлежности был задан как *Linear*.

Количество циклов обучения созданной нечеткой нейронной сети составило 600 эпох. На рис. 4.48 показана зависимость ошибки обучения от количества циклов обучения.

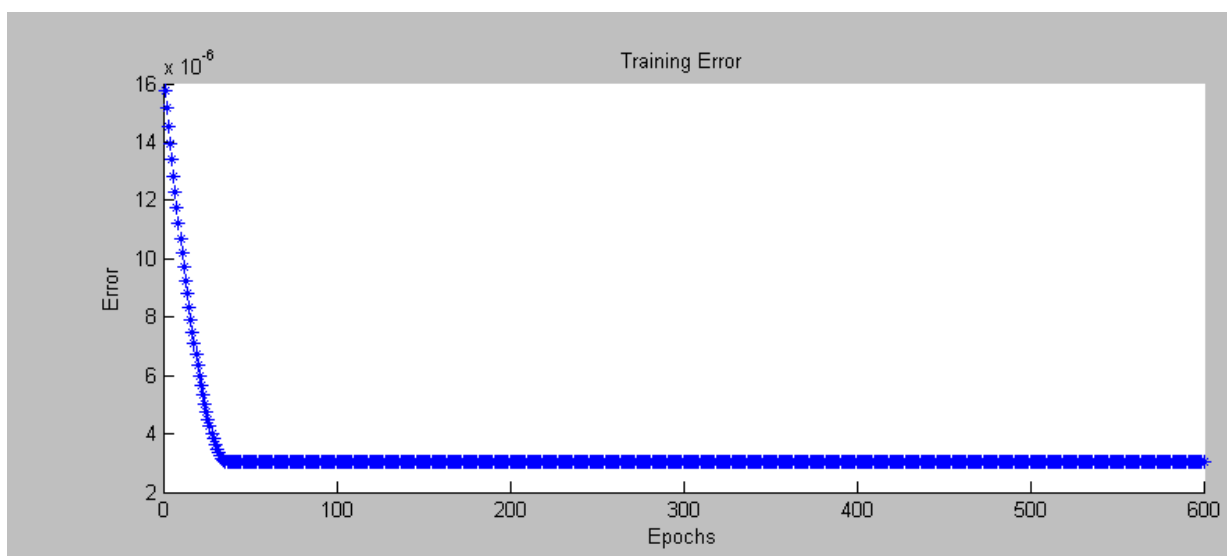


Рисунок 4.48 – Зависимость ошибки обучения нечеткой нейронной сети от количества циклов обучения

Ошибка обучения составила $3,01 \cdot 10^{-6}$ мг КОН на 1 г масла, после чего была протестирована на выборке из данных, которые не использовались

в обучающей выборке. На рис. 4.49 приведены фактические (проверочная выборка) и спрогнозированные значения кислотного числа.

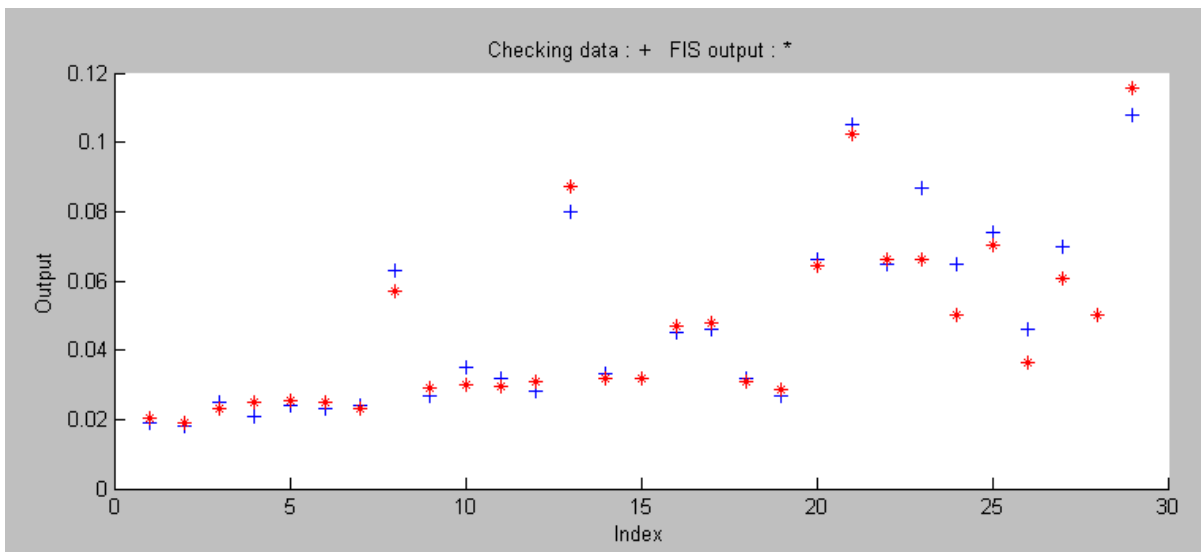


Рисунок 4.49 – Фактические (+) и спрогнозированные (*) значения кислотного числа

Рис. 4.50 отображает ошибку прогнозирования созданной нечеткой нейронной сети.

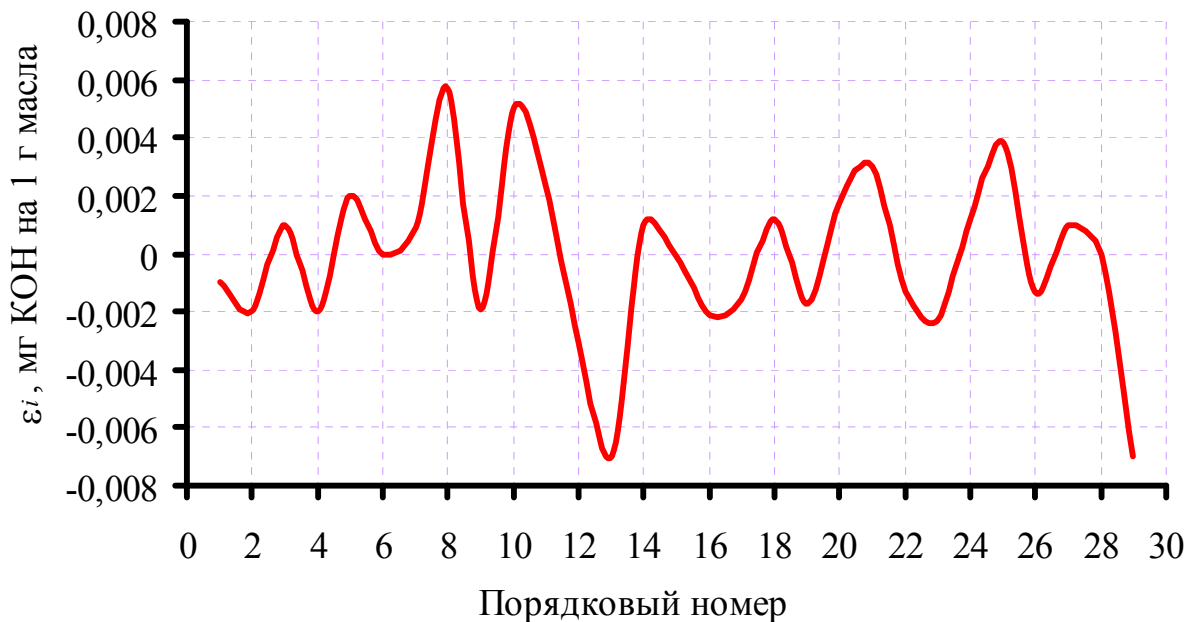


Рисунок 4.50 – Ошибка прогнозирования кислотного числа трансформаторного масла по значениям средней мощности и времени эксплуатации

В результате поведение сети можно считать вполне адекватным, среднеквадратическая погрешность прогноза составила 0,00 292 мг КОН на 1 г масла, что, учитывая ограниченный объем обучающей выборки, является приемлемым.

4.13.2. Прогнозирование цвета трансформаторного масла

Прогнозирование цвета трансформаторного масла осуществлялось на основе выделенной корреляционной связи между цветом масла и кислотным числом.

Входными переменными являлись:

- длительность эксплуатации в годах;
- кислотное число трансформаторного масла в миллиграммах КОН на 1 г масла.

В качестве исходных данных использовались результаты измерений кислотного числа масла, значения времени эксплуатации и значения цвета масла по данным АК «Донбассоблэнерго». Топология нечеткой нейронной сети для прогнозирования цвета трансформаторного масла аналогична приведенной на рис. 4.51.

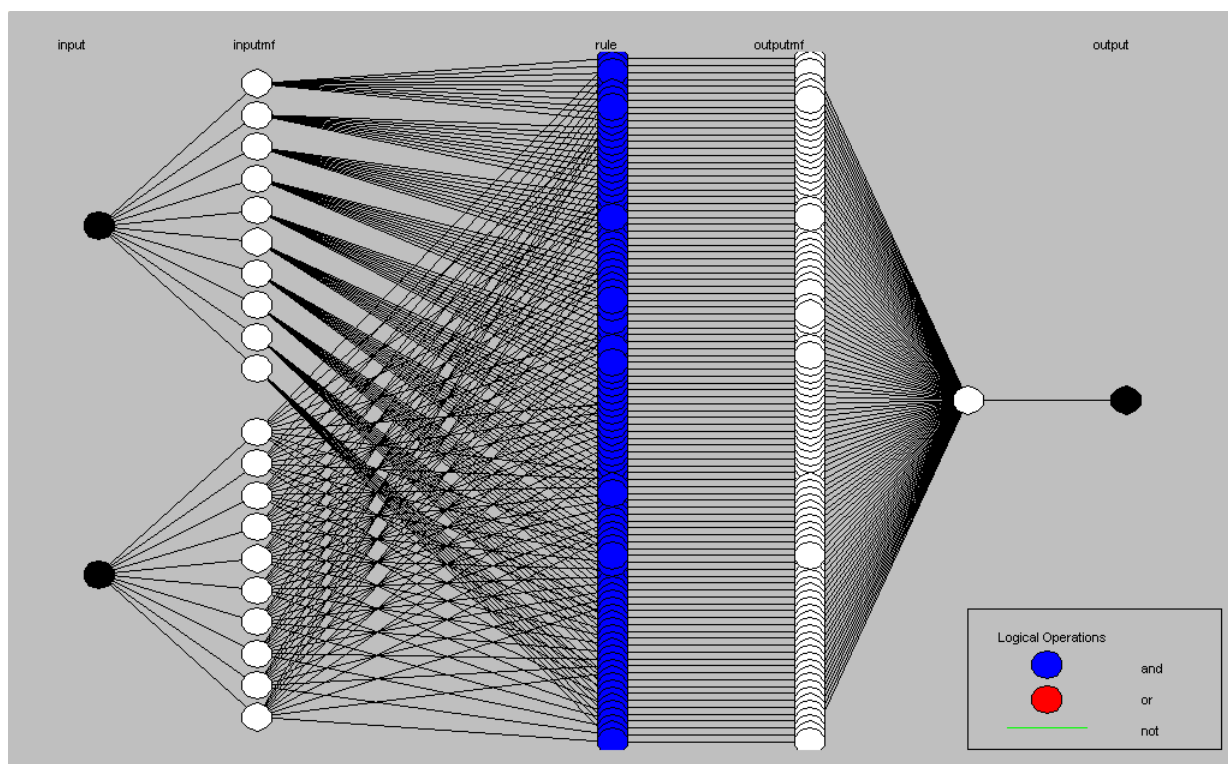


Рисунок 4.51 – Топология нечеткой нейронной сети для прогнозирования цвета трансформаторных масел

Отличия заключаются лишь в том, что у созданной сети 20 входов: 10 входов по времени и 10 входов по кислотному числу. Так же как и для предыдущей сети выбрано две функции принадлежности для каждой входной переменной, эти функции являются трапецеидальными. Для выходного параметра тип функции принадлежности был задан как const. Количество циклов обучения созданной нечеткой нейронной сети составило 500 эпох, ошибка обучения составила 0,157 ЦЧ. Зависимость ошибки обучения от количества циклов обучения приведена на рис. 4.52.

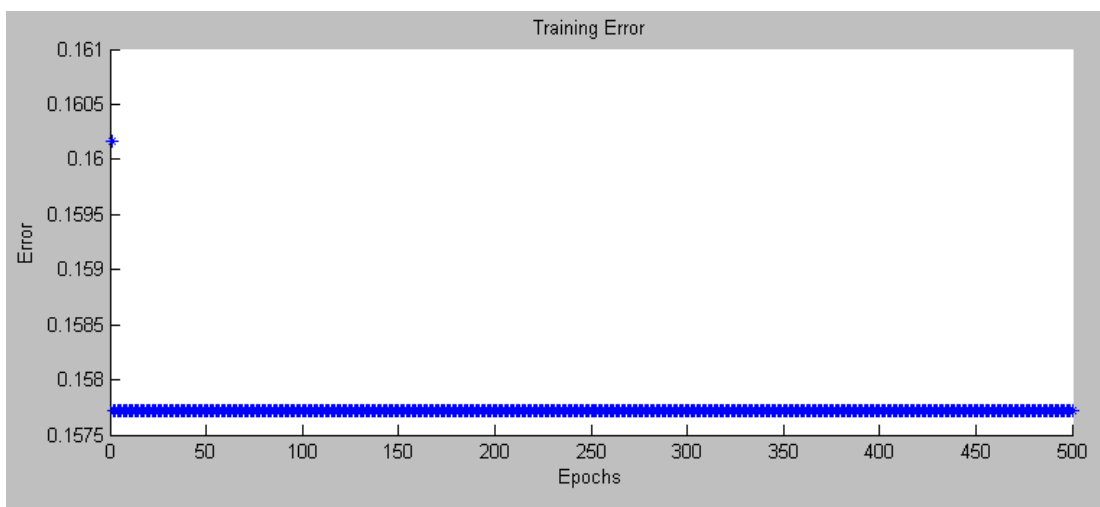


Рисунок 4.52 – Зависимость ошибки обучения нечеткой нейронной сети от количества циклов обучения

На рис. 4.53 приведены результаты проверки обученной сети на независимой выборке.

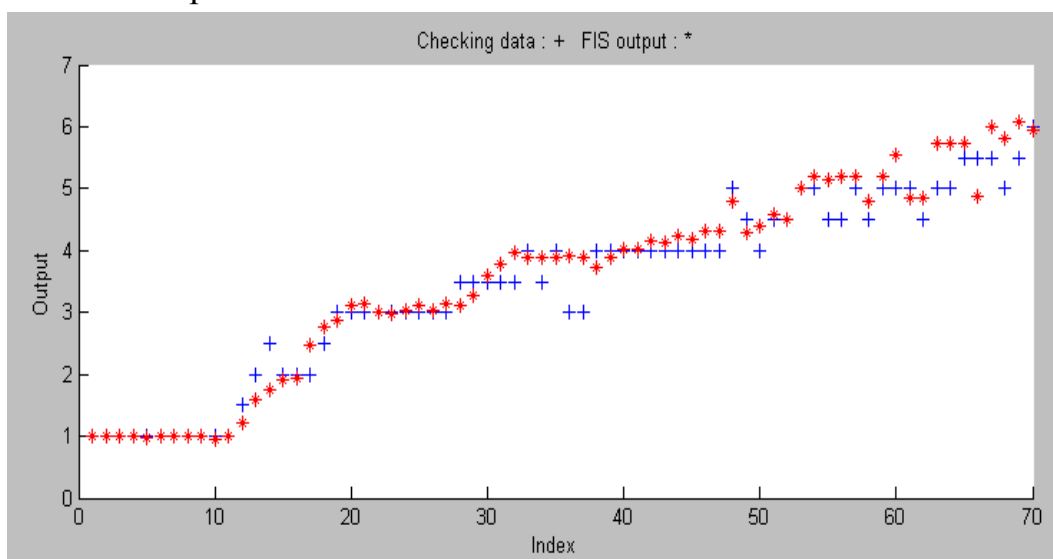


Рисунок 4.53 – Фактические (+) и спрогнозированные (*) значения цвета

трансформаторного масла

Как видно из рисунка, экспериментальные и прогнозируемые значения практически совпадают. Рис. 4.54 отображает ошибку прогнозирования созданной нечеткой нейронной сети. Среднеквадратическая погрешность прогноза составила 0,352 ЦЧ, что позволяет признать обученную нейронную сеть адекватной.

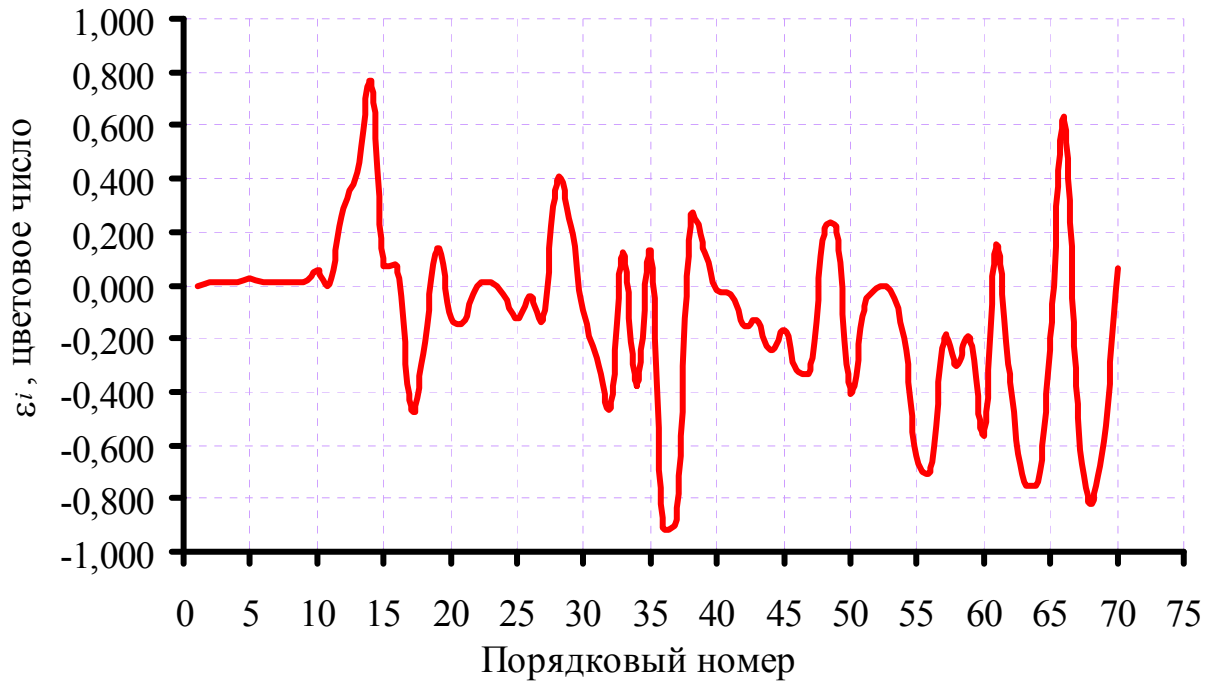


Рисунок 4.54 – Ошибка прогнозирования цвета трансформаторного масла по значениям времени эксплуатации и кислотного числа масла

4.13.3. Прогнозирование концентраций водорастворимых кислот в трансформаторном масле

Прогнозирование концентраций водорастворимых кислот в трансформаторном масле осуществлялось как на основе полученной связи между загрузкой трансформаторов и содержанием водорастворимых кислот, так и на основе выделенной корреляционной связи между кислотным числом масел и содержанием водорастворимых кислот.

Входными переменными являлись:

- длительность эксплуатации в годах;
- средняя загрузка трансформаторов за анализируемый период в процентах;
- кислотное число трансформаторного масла в миллиграммах КОН на 1 г масла.

В качестве исходных данных использовались результаты измерений кислотного числа масла, значения времени эксплуатации и значения цвета масла по данным АК «Харьковоблэнерго».

Далее была разработана топология нечеткой нейронной сети, которая приведена на рис. 4.55.

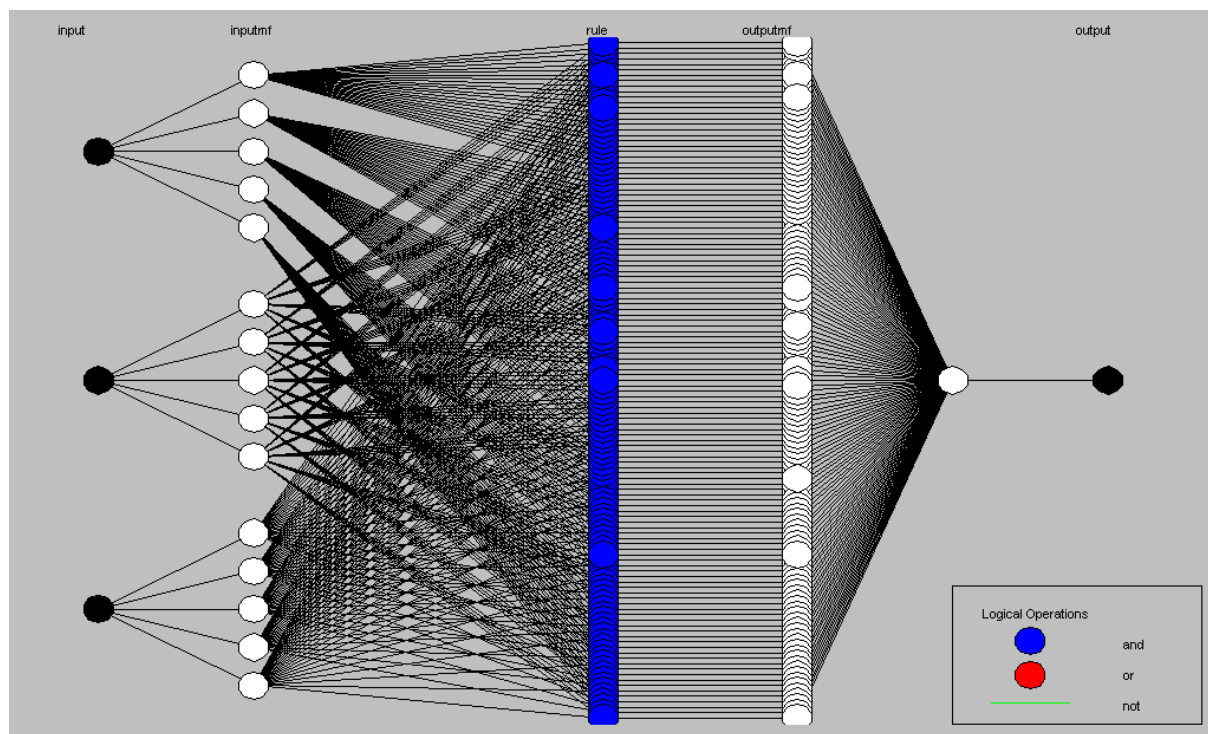


Рисунок 4.55 – Топология нечеткой нейронной сети для прогнозирования содержания водорастворимых кислот в трансформаторных маслах

Как видно из рисунка, нечеткая нейронная сеть имеет 15 входов: по 5 входов по каждой входной переменной. Так же как и для предыдущих сетей выбрано две функции принадлежности для каждой входной переменной, эти функции являются трапецеидальными. Для выходного параметра тип функции принадлежности был задан как *Linear*. Количество циклов обучения созданной нечеткой нейронной сети составило 500 эпох, ошибка обучения составила $7,83 \cdot 10^{-9}$ мг КОН на 1 г масла. Зависимость ошибки обучения нечеткой нейронной сети от количества циклов обучения приведена на рис. 4.56.

На рис. 4.57 приведены результаты проверки обученной сети на независимой выборке. Как видно из рисунка, экспериментальные и прогнозируемые точки находятся очень близко, о чем свидетельствует и значение погрешности, которое составило 0,00 074 мг КОН на 1 г масла.

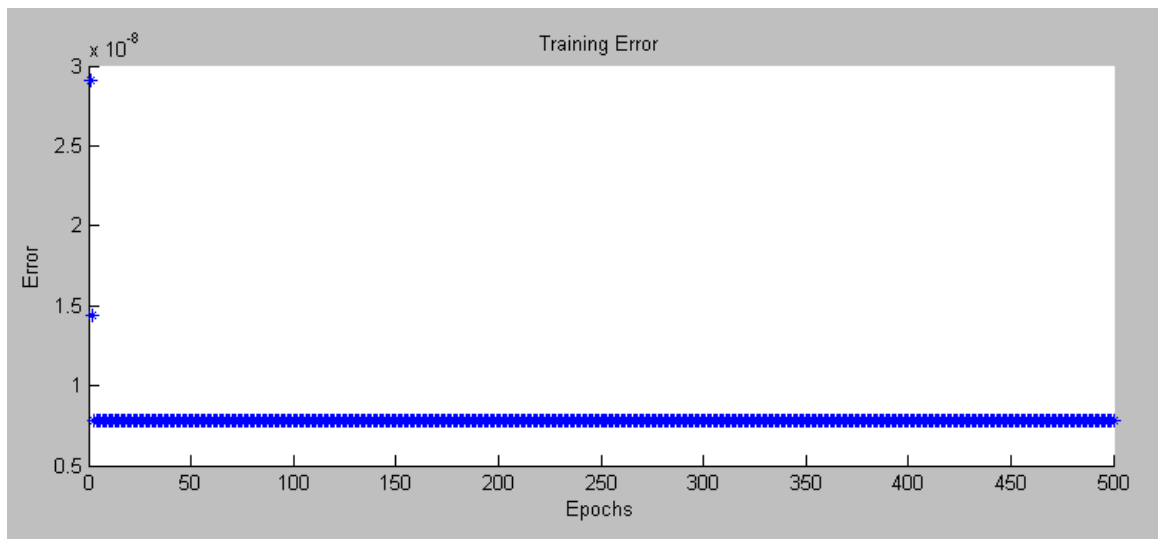


Рисунок 4.56 – Зависимость ошибки обучения нечеткой нейронной сети от количества циклов обучения

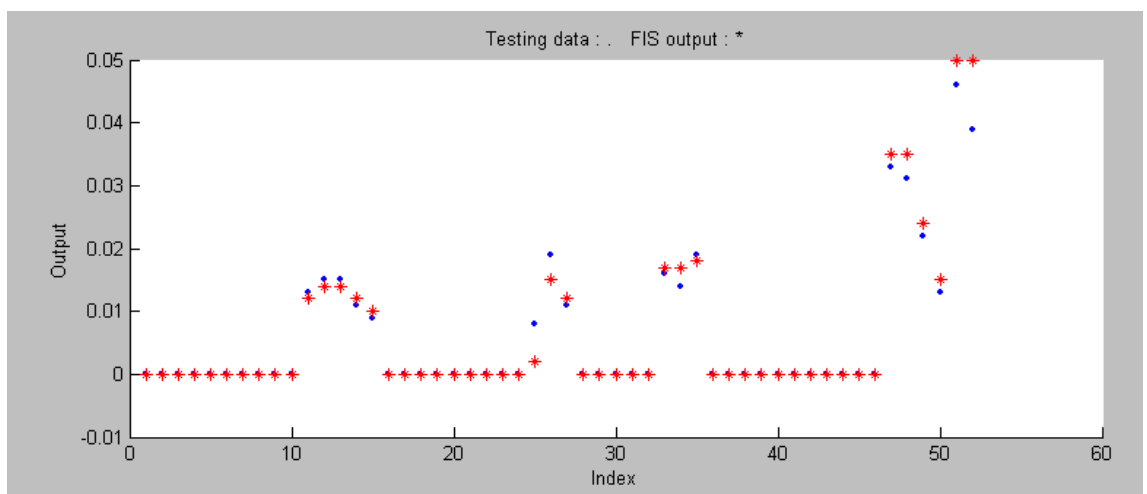


Рисунок 4.57 – Фактические (+) и спрогнозированные (*) значения содержания в масле водорастворимых кислот

Рис. 4.58 отображает ошибку прогнозирования созданной нечеткой нейронной сети. Как видно из рисунка, погрешности не коррелированы между собой, что позволяет признать полученную нейро-нечеткую сеть адекватной.

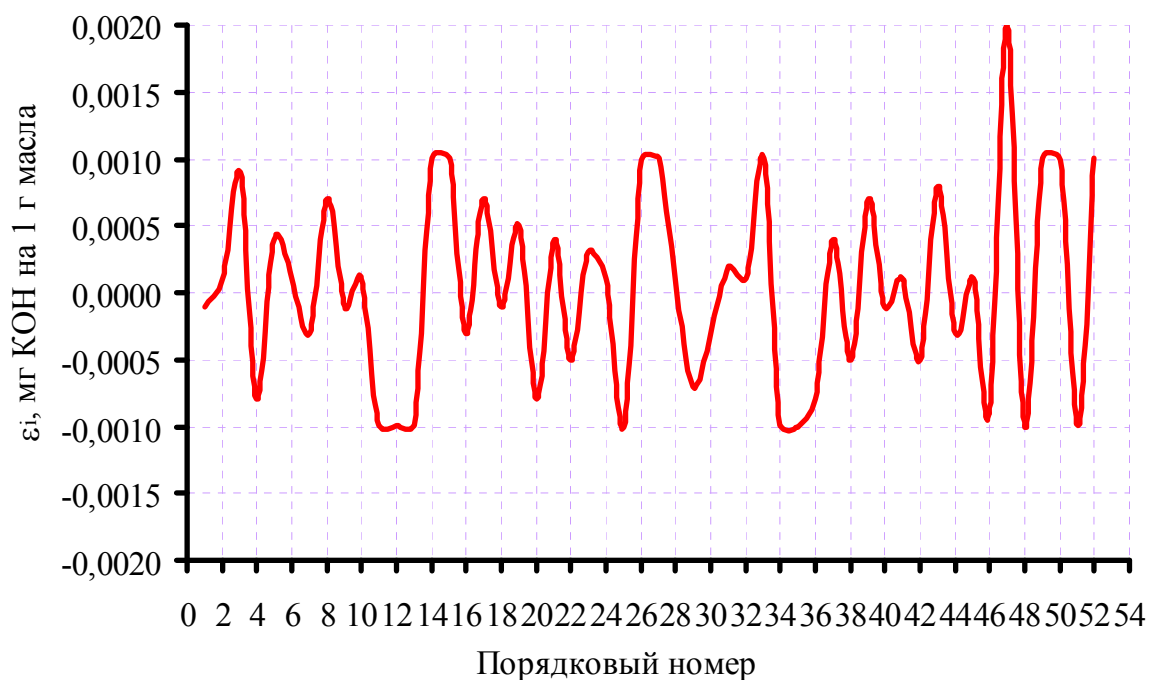


Рисунок 4.58 – Ошибка прогнозирования содержания в трансформаторном масле водорастворимых кислот по значениям времени эксплуатации, нагрузке трансформатора и кислотному числу масла

Таким образом, исследуемые нечеткие нейронные сети позволяют достаточно точно прогнозировать значения показателей качества трансформаторного масла, даже при достаточно ограниченном объеме обучающих выборок, учитывая при этом как корреляционные связи между показателями качества масла, так и влияния условий эксплуатации. Повышение точности и расширения пределов прогноза возможно за счет увеличения объема обучающих выборок.

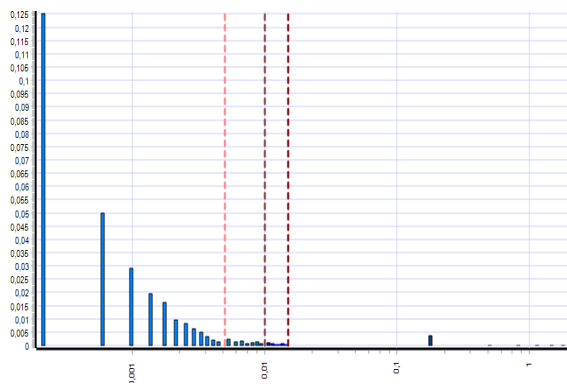
4.14. Оценка возможностей нечетких нейронных сетей для интерпретации результатов хроматографического анализа растворенных в масле газов

В настоящее время для обнаружения и распознавания типа дефектов высоковольтного маслонаполненного оборудования как отечественными [34–38], так и зарубежными [39–49] исследователями широко используются математический аппарат нечеткой логики и нейронных сетей. В большинстве из опубликованных работ в качестве решающего правила используются международные, национальные или ведомственные стандарты или

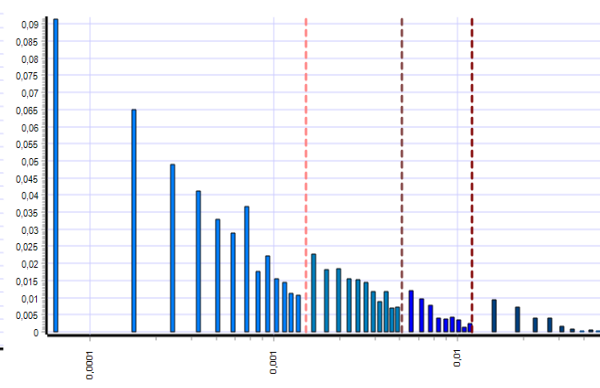
методики по интерпретации результатов хроматографического анализа растворенных в масле газов (ХАРГ). Как правило, при обучении и тестировании разработанных систем нечеткого вывода или нейронных сетей использовались результаты ХАРГ, соответствующие бездефектному либо дефектному состоянию оборудования. При этом обученные сети, как правило, показывают достаточно высокие значения достоверности распознавания. Однако, несмотря на достаточно большое количество публикаций по использованию нейронных сетей для интерпретации результатов ХАРГ, ряд вопросов остаются неосвещенными. В частности не рассмотрены вопросы, связанные с возможностью нейронных сетей распознавать развивающиеся дефекты на ранней стадии их развития, не исследована реакция нейронных сетей на значениях концентраций газов, полученных после аварийных воздействий со стороны электрических сетей на силовые трансформаторы. В связи с этим на кафедре «Передача электрической энергии» НТУ «ХПИ» выполнен анализ возможности обучения нейронных сетей для распознавания состояния трансформатора по результатам ХАРГ.

Для обучения сети использовались результаты ХАРГ по Донецкой, Луганской, Сумской и Харьковской областям Украины. Всего проанализированы результаты наблюдений по 426 трансформаторам напряжением 110 и 330 кВ негерметичного исполнения. Поскольку данные трансформаторы отличаются и по классу напряжения, и по номинальной мощности, и по конструкции, а самое главное, по условиям эксплуатации, то есть по значениям загрузки, по частоте и уровню воздействия токов короткого замыкания по кратности воздействий перенапряжений и т.д. Все это приводит к тому, что значения концентраций газов изменяются в достаточно широком диапазоне. Наглядно значимые различия в распределении значений концентраций для отдельных газов в бездефектных трансформаторах негерметичного исполнения иллюстрирует рис. 4.59.

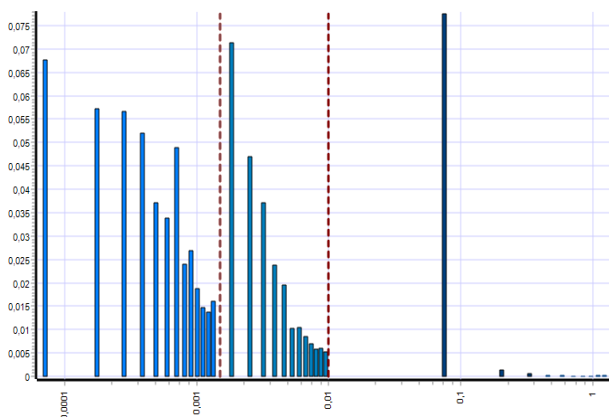
Для удобства анализа значений концентраций газов были использованы уровни концентраций, которые рекомендованы в [50], для диагностики состояния высоковольтного оборудования напряжением до 330 кВ. Значения концентраций газов, соответствующие разным уровням, приведены в табл. 4.1.



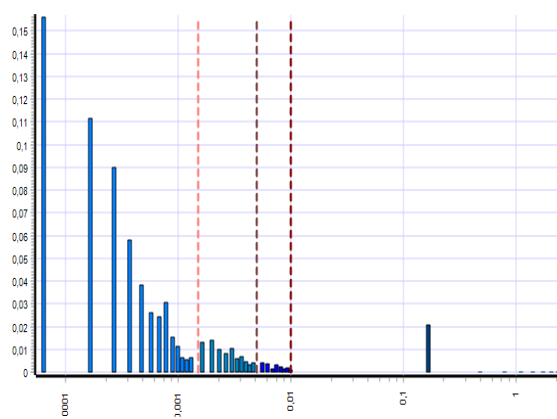
a



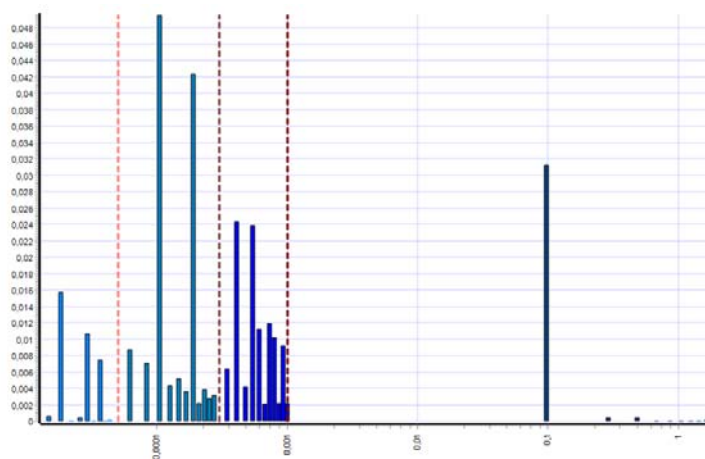
б



в



г



д

Рисунок 4.59 – Диаграммы распределения концентраций газов в бездефектных трансформаторах негерметичного исполнения:
a – водород; *б* – метан; *в* – этилен; *г* – этан; *д* – ацетилен

Таблица 4.1 – Уровни состояния маслonaполненного оборудования по значениям концентраций газов

Уровень концентраций	Растворенные в масле газы				
	H ₂	CH ₄	C ₂ H ₆	C ₂ H ₄	C ₂ H ₂
1	<0,01	<0,005	<0,005	<0,0015	<0,00005
2	0,01–0,015	0,005–0,012	0,005–0,01	0,0015–0,01	0,00005–0,001
3	>0,015	>0,012	>0,005	>0,0015	>0,001

Согласно [50], если значения концентраций газов соответствуют уровню 1, то это свидетельствует о нормальном, бездефектном состоянии оборудования. Если концентрация хотя бы одного из газов соответствует уровню 2, то решение о состоянии оборудования производится на основании анализа значений скоростей нарастания суммы газов углеводородного ряда (дефект считается «присутствующим», если эта скорость превышает 30 мл/сутки). Если значения концентраций хотя бы одного из газов соответствуют уровню 3, то прогнозируется наличие дефекта без учета скорости нарастания концентраций газов.

Выполненный анализ показал, что из 7 393 результатов измерения концентраций водорода 5 161 значение или 69,81 % не превышало предела обнаружения хроматографа. Еще 2 106 значений (28,49 %) концентраций не превышали значения аналитического порога распознавания (0,005 % об), регламентированного в [50]. Только 71 значение (0,96 %) соответствовало уровню 1 (менее 0,01 % об). Уровню 2 (0,01–0,015 % об), соответствовали 26 значений (0,35 %), а уровню 3 (более 0,015 % об) – 29 значений (0,39 %).

Концентрации метана ниже предела обнаружения хроматографом выявлены в 2 304 пробах масла, что составляет 31,6 % всех наблюдений по данному газу, еще 3 342 (45,2 %) концентраций метана имели значения ниже аналитического порога распознавания (0,0015 % об). Уровню 1 (до 0,005 % об) соответствовали 1160 значений (15,69 %), уровню 2 (0,005–0,012 % об) – 367 значений (4,96 %), а уровню 3 (более 0,012 % об) – 220 значений (2,98 %).

Концентрации этана не превышают предела обнаружения хроматографа в 1 957 пробах (26,47 %), еще в 4 485 пробах масла (60,67 %), концентрации этана не превышали аналитического порога распознавания (0,0015 % об). Уровню 1 (до 0,005 % об) соответствовали 619 значений

концентраций (8,37 %), уровню 2 (0,005–0,01 % об) – 153 значения (4,96 %), а уровню 3 (более 0,01 % об) – 179 значений (2,42 %).

Из 7 393 значений концентраций этилена ниже предела обнаружения, хроматографом выявлено 1 090 значений (14,74 %), 3 763 (50,90 %), не превышающих аналитического порога распознавания (который составляет 0,0015 % об). Здесь следует учесть, что для этилена значение аналитического уровня распознавания совпадает со значением верхней границы уровня 1. В связи с этим в дальнейшем для этилена все значения, соответствующие уровню 1, отнесены к значениям, не превышающим аналитического порога распознавания. В 1 914 пробах значения концентраций этилена (25,89 %) соответствуют уровню 2 (0,0015–0,01 % об), а 626 значений этилена (8,47 %) соответствуют уровню 3 (более 0,01 % об).

Концентрации ацетилена, не превышающие предела обнаружения хроматографом, выявлены в 4 551 пробе масла, что составляет 61,56 % всех наблюдений по данному газу, еще 1 602 значения (21,67 %) концентраций ацетилена имели значения ниже аналитического порога распознавания. В связи с тем, что для ацетилена уровень распознавания (0,0003 % об) превышает верхнюю границу уровня 1 (0,00005 % об), значения ацетилена с концентрациями выше аналитического порога распознавания, но ниже уровня 3, отнесены к уровню 2 (от 0,00005 до 0,001 % об). Уровню 2 соответствовали 982 значения (13,28 %), а уровню 3 (более 0,001 % об) – 258 значений (3,49 %).

Таким образом, на основании анализа установлено, что в бездефектных трансформаторах негерметичного исполнения возможно превышение концентрациями газов верхних граничных значений уровня 2, что может быть интерпретировано как наличие дефекта. Наибольшую вероятность превышения граничного значения уровня 2 выявлено у этилена, потом ацетилена, метана, этана и меньше всего – у водорода.

В то же время наибольшую вероятность реализации концентраций газов ниже аналитического порога распознавания имеет водород, далее – ацетилен, этан и метан. Для этилена данная вероятность является самой низкой. Максимальное количество значений с концентрациями выше аналитического порога распознавания имеет C_2H_4 (2 540 значений, или 34,36 %). Далее следует CH_4 (1 747 значений, или 23,63 %), потом C_2H_2 (1 240 значений, или 16,77 %). Меньше всего значений с концентрациями

выше аналитического порога распознавания выявлено у C_2H_6 (951 значение, или 12,86 %) и H_2 (126 значений, или 1,70 %).

Полученные результаты позволяют сделать вывод о значительных количественных и качественных различиях газосодержания масел в бездефектном герметичном и негерметичном маслonaполненном оборудовании. Кроме того, в бездефектных трансформаторах значения концентраций могут соответствовать значениям, характерным для дефектного состояния (уровень 3). Как показал анализ, выполненный в работах [51–53], одной из основных причин превышения концентраций газов граничных значений в бездефектных трансформаторах являются аварийные воздействия со стороны электрических сетей (короткие замыкания, перенапряжения, перегрузки и т.д.).

Также для обучения нейронной сети были использованы результаты ХАРГ для дефектного оборудования, которые получены как в результате сотрудничества авторов с энергетическими компаниями Украины, так и из открытых отечественных и зарубежных литературных источников. Общий объем выборочных значений составил 1 103 измерения. Распределение выборочных значений по типам дефектов приведено в табл. 4.2.

Как видно из таблицы, в представленной к анализу выборке разные типы дефекта имеют разный объем выборочных значений, то есть разную вероятность их появления. Наибольшее число дефектов связано с перегревами в диапазоне температур выше $700\text{ }^{\circ}\text{C}$ (в таблице обозначен как перегрев высоких температур) и перегревами в области высоких температур, которые сопровождаются электрическими разрядами. Меньше всего результатов испытаний получено для оборудования с электрическими разрядами низкой энергии и перегревами, переходящими в дуговой разряд. Выполненный анализ показал, что в дефектном оборудовании значения концентраций газов существенно зависят от стадии развития дефекта, при этом на начальных стадиях значения концентраций могут и не превышать граничных значений, соответствующих уровню 3 [50], но по мере развития дефекта значения концентраций возрастают. Еще одним немаловажным фактором, влияющим на значения концентраций отдельных газов, является тип дефекта. Как правило, максимальные значения концентраций наблюдаются у газов, характерных для данного типа дефектов. Для сопутствующих газов значения концентраций несколько меньше. В то же время значе-

ния концентраций газов, не характерных для данного типа дефекта, имеют значения, соответствующие либо уровням 1 или 2, либо же вообще не превышают предела обнаружения хроматографа.

Таблица 4.2 – Распределение выборочных значений по типам дефектов

№ п/п	Тип дефекта	Объем вы-борки
1	Частичные разряды	115
2	Частичные разряды высокой интенсивности	15
3	Искровые и ползучие разряды	81
4	Разряды низкой энергии	17
5	Дуговой разряд (газ с максимальной концентрацией H_2)	67
6	Дуговой разряд (C_2H_2)	43
7	Разряды большой энергии (C_2H_2)	53
8	Перегревы низких температур (CH_4)	48
9	Перегревы низких температур (C_2H_6)	57
10	Перегревы средних температур (CH_4)	68
11	Перегревы средних температур (C_2H_4)	81
12	Перегревы высоких температур (C_2H_4)	260
13	Перегревы низких температур и разряды	35
14	Перегревы, переходящие в дугу (CH_4)	16
15	Перегревы, переходящие в разряды (C_2H_6)	27
16	Перегревы высоких температур и разряды	120

Примечание. В скобках приведен газ с максимальной концентрацией

В качестве критерия наличия дефекта в высоковольтных маслонаполненных трансформаторах использовался критерий граничных концентраций газов. Кодировка диагнозов трансформаторов приведена в табл. 4.3.

Таблица 4.3 – Кодировка диагнозов трансформаторов

Код	Диагноз
1	Нормальное состояние
2	Подозрительные концентрации
3	Дефектное состояние

В табл. 4.4 приведен фрагмент обучающей выборки, которая использовалась на первом этапе синтеза и обучения сети.

Таблица 4.4 – Фрагмент выборки для обучения нечеткой сети

H ₂	CH ₄	C ₂ H ₆	C ₂ H ₄	C ₂ H ₂	D
0,000	0,000	0,0003	0,0004	0,0004	1
0,000	0,000	0,0003	0,001	0,0005	1
0,000	0,000	0,0001	0,0003	0,0003	1
0,000	0,000	0,0001	0,0003	0,0002	1
0,000	0,000	0,0001	0,0002	0,000	1
0,000	0,000	0,0001	0,0001	0,00013	1
0,000	0,000	0,0001	0,0002	0,000	1
0,0140	0,0044	0,0033	0,0078	0,00003	2
0,00027	0,00064	0,00044	0,00216	0,00004	2
0,00129	0,00084	0,00037	0,00439	0,00001	2
0,00063	0,00047	0,00032	0,00179	0,00004	2
0,00124	0,00042	0,00006	0,00198	0,00002	2
0,00575	0,0004	0,00052	0,00308	0,00002	2
0,00339	0,00116	0,00100	0,00286	0,00000	2
0,00143	0,00066	0,00028	0,00219	0,00001	2
0,00049	0,0009	0,00021	0,00164	0,00000	2
0,00077	0,00141	0,00015	0,00201	0,00000	2
0,00052	0,00080	0,00016	0,00168	0,00000	2
0,00000	0,0008	0,0003	0,0044	0,00000	2
0,135	0,022	0,001	0,0003	0,000	3
0,053	0,0345	0,0085	0,0266	0,0250	3
0,0772	0,0155	0,0179	0,0061	0,0171	3
0,788	0,755	0,473	0,03	0,297	3
0,1879	0,0029	0,0005	0,0036	0,0421	3
0,19	0,0285	0,0031	0,0957	0,773	3
0,0024	0,0013	0,0005	0,0043	0,0319	3
0,003	0,05	0,009	0,0006	0,000	3
0,0063	0,0117	0,0146	0,0073	0,000	3
0,0073	0,0177	0,0037	0,0052	0,000	3
0,0136	0,016	0,0038	0,0187	0,0003	3
0,0525	0,0109	0,0113	0,0527	0,042	3

Как видно из таблицы, в качестве входных данных использовались концентрации газов углеводородного ряда и водорода. Поскольку в пакете *Fuzzy Logic Toolbox* изначально заложен алгоритм обучения с учителем, то помимо входных переменных на вход сети подавалась и выходная переменная *D*, которая означает диагноз трансформатора. Следует отметить, что в обучаемой выборке были учтены все возможные комбинации значе-

ний концентраций газов, соответствующие разным состояниям. Исходные данные для обучения сети отражены на рис. 4.60.

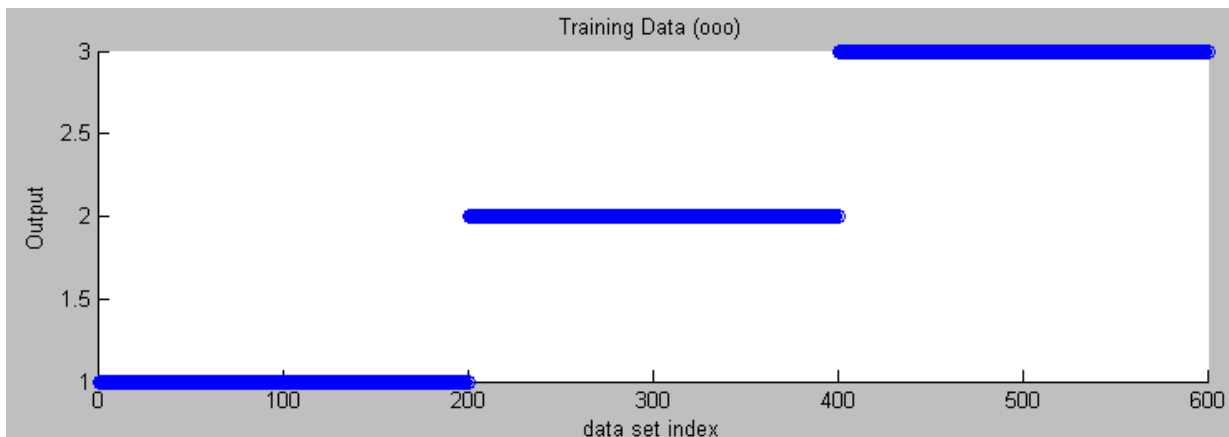


Рисунок 4.60 – Исходные данные для обучения сети

Далее была разработана топология нечеткой нейронной сети, которая приведена на рис. 4.61.

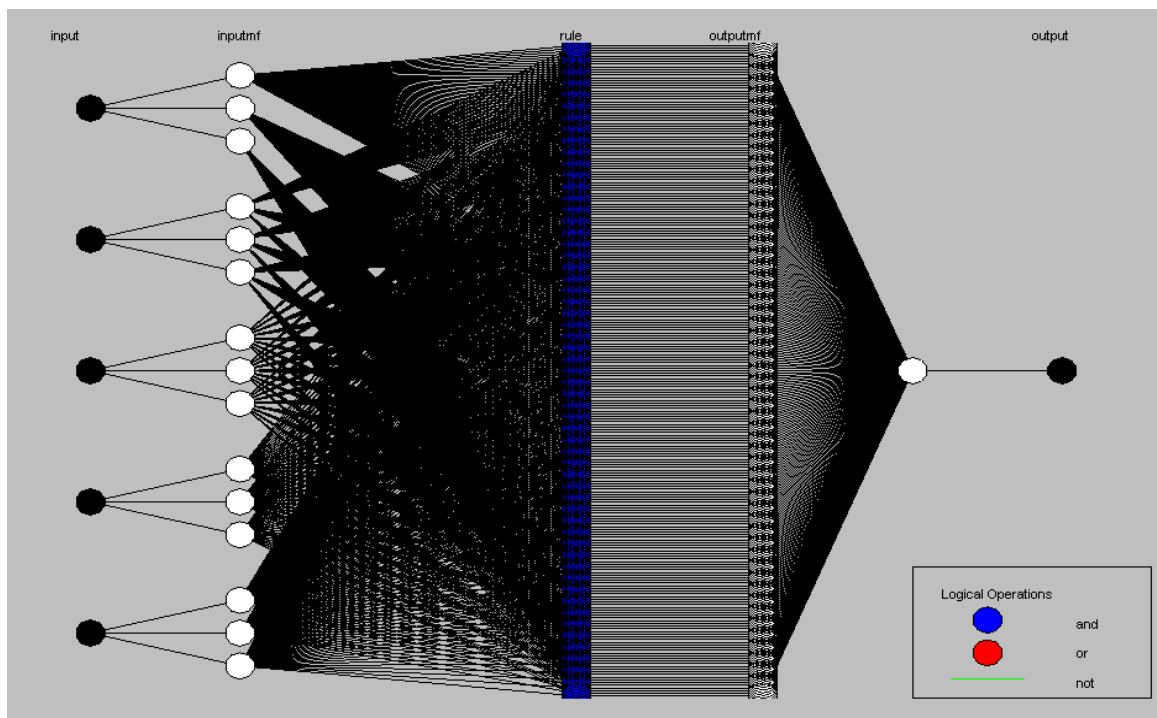


Рисунок 4.61 – Топология нечеткой нейронной сети

У созданной сети 15 входов, по три входа по каждому из газов.

Количество циклов обучения созданной нечеткой нейронной сети составило 300 эпох. На рис. 4.62 показана зависимость ошибки обучения от количества циклов обучения. Как видно из рисунка, снижение погреш-

ности обучения наблюдается примерно до середины интервала обучения. Дальнейшее увеличение числа циклов обучения не приводит к повышению достоверности распознавания.

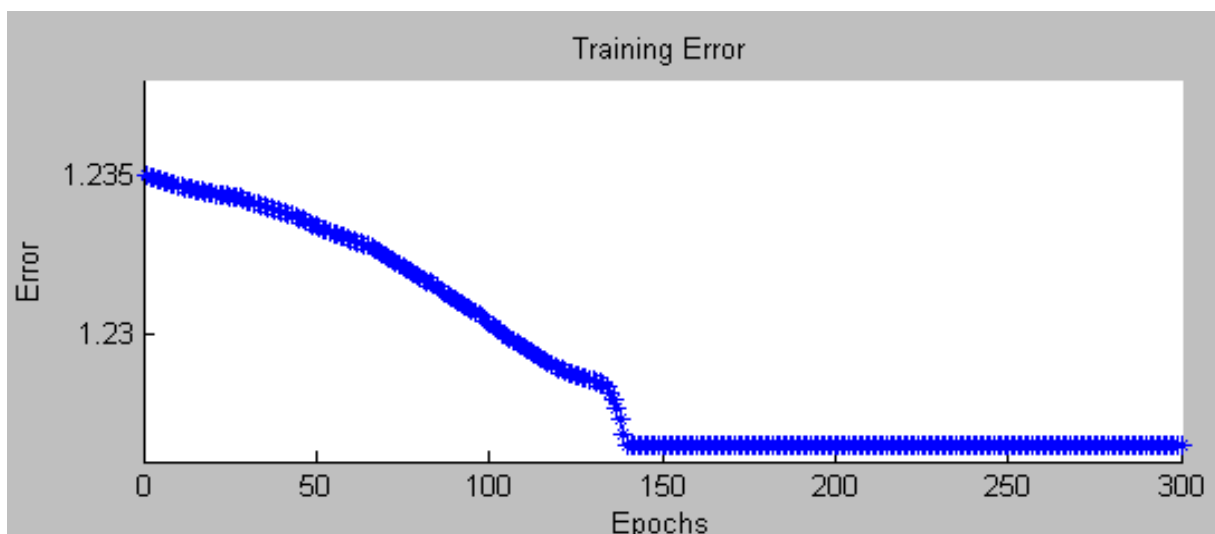


Рисунок 4.62 – Зависимость ошибки обучения нечеткой нейронной сети от количества циклов обучения

Окно нечеткого вывода нечеткой нейронной сети приведено на рис. 4.63. В левой части рисунка находятся входные переменные (концентрации растворенных в масле газов), в правой части – выходная переменная (диагноз), а в центре – блок решающих правил.

Этап фаззификации. На данном этапе выделяются входные лингвистические переменные:

$$\text{ЛП}_{\text{вх } i} \quad i = 1 \dots 5,$$

где i – номер коэффициента парной корреляции данного газа и времени.

Входные функции принадлежности будут иметь вид:

$\text{ЛП}_{\text{вх } 1}$: <концентрация водорода>

$\text{ЛП}_{\text{вх } 2}$: <концентрация метана>

$\text{ЛП}_{\text{вх } 3}$: < концентрация этана >

$\text{ЛП}_{\text{вх } 4}$: < концентрация этилена >

$\text{ЛП}_{\text{вх } 5}$: < концентрация ацетилена >

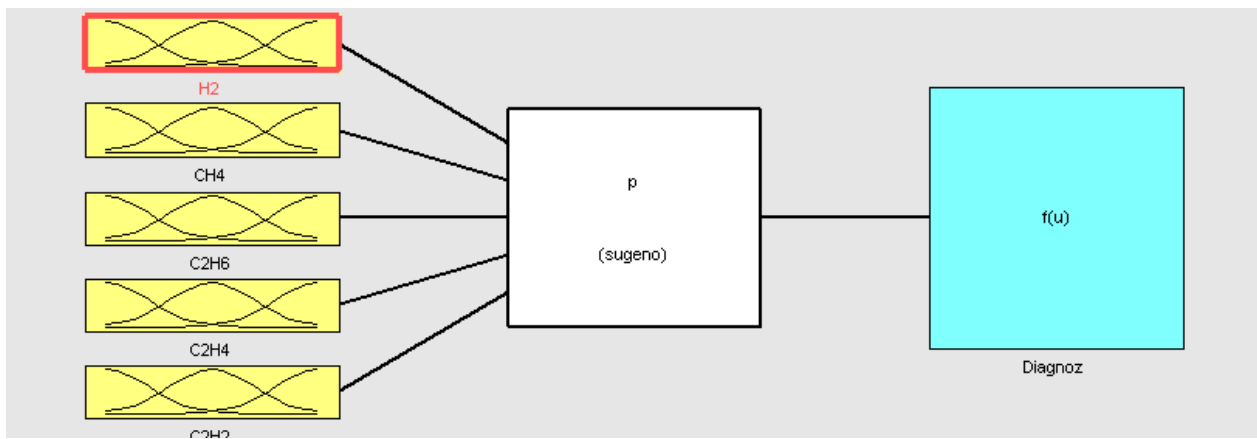


Рисунок 4.63 – Окно нечеткого вывода

Далее $\text{ЛП}_{\text{вх } i}$ делится на несколько лингвистических термов ЛТ_{ij} , характеризующих особенности состояния данной переменной:

$$\text{ЛТ}_{ij}, j = 1 \dots n,$$

где j – номер термина $\text{ЛП}_{\text{вх } i}$; n – количество термов, на которое разбито $\text{ЛП}_{\text{вх } i}$.

Разбиение на термы для каждого из газов проводилось следующим образом:

$\text{ЛП}_{\text{вх } 1}$ разделена на три ЛТ

ЛТ_{11} : < концентрация в пределах нормы – 1 >

ЛТ_{12} : < подозрительная концентрация – 2 >

ЛТ_{13} : < дефектная концентрация – 3 >

$\text{ЛП}_{\text{вх } 2}$ разделена на три ЛТ

ЛТ_{21} : < концентрация в пределах нормы – 1 >

ЛТ_{22} : < подозрительная концентрация – 2 >

ЛТ_{23} : < дефектная концентрация – 3 >

$\text{ЛП}_{\text{вх } 3}$ разделена на три ЛТ

ЛТ_{31} : < концентрация в пределах нормы – 1 >

ЛТ_{32} : < подозрительная концентрация – 2 >

ЛТ_{33} : < дефектная концентрация – 3 >

$\text{ЛП}_{\text{вх } 4}$ разделена на три ЛТ

ЛТ_{41} : < концентрация в пределах нормы – 1 >

ЛТ_{42} : < подозрительная концентрация – 2 >

ЛТ_{43} : < дефектная концентрация – 3 >

$\text{ЛП}_{\text{вх } 5}$ разделена на три ЛТ

ЛТ_{51} : < концентрация в пределах нормы – 1 >

ЛТ₅₂: < подозрительная концентрация – 2>

ЛТ₅₃: < дефектная концентрация – 3>

Затем проводится прием входного четкого значения для системы с нечеткой логикой, а также поочередное считывание из базы знаний всех функций принадлежности, которые имеют отношение к данному входному значению, и устанавливается значение достоверности измеренного значения каждому ЛТ_{ij} по функциям принадлежности.

Определение функций принадлежности является достаточно сложной задачей. Для того чтобы выбранные функции принадлежности и их границы наиболее адекватно описывали результаты хроматографического анализа, рассмотрим функции плотности распределения концентраций газов для нормального и дефектного состояний. Эти плотности распределений получены в [54] и приведены на рис. 4.64.

Как видно из рисунка, плотности распределений концентраций газов для дефектного и бездефектного состояний пересекаются, при этом с возрастанием значения концентраций газов в бездефектных трансформаторах вероятность их появления снижается. Также с уменьшением значения концентраций газов в дефектных трансформаторах наблюдается снижение вероятности их появления.

В качестве функции принадлежности очень удобно использовать трапецеидальную функцию, степень принадлежности которой имеет максимальное значение в середине интервала и снижается по его краям. В качестве границ для ЛТ₁ всех лингвистических переменных были выбраны границы уровней, рекомендуемые в [50] (см. табл. 4.1). Значения границ для ЛТ₂ и ЛТ₃ всех лингвистических переменных выбирались на основе функций плотностей распределения.

На рис. 4.65–4.69 приведены функции принадлежности для пяти лингвистических переменных. Как видно из рисунков, функции принадлежности выбраны таким образом, чтобы учесть как рекомендации СОУ - Н ЕЕ 46.501:2006, так и опыт эксплуатации, отраженный в виде функций плотности распределения.

Выходные лингвистические переменные имеют вид:

ЛП_{вых 1}: <Значения концентраций в пределах нормы – 1>

ЛП_{вых 2}: <Значения концентраций выше нормы – 2>

ЛП_{вых 3}: <Значения концентраций соответствуют наличию дефекта – 3>

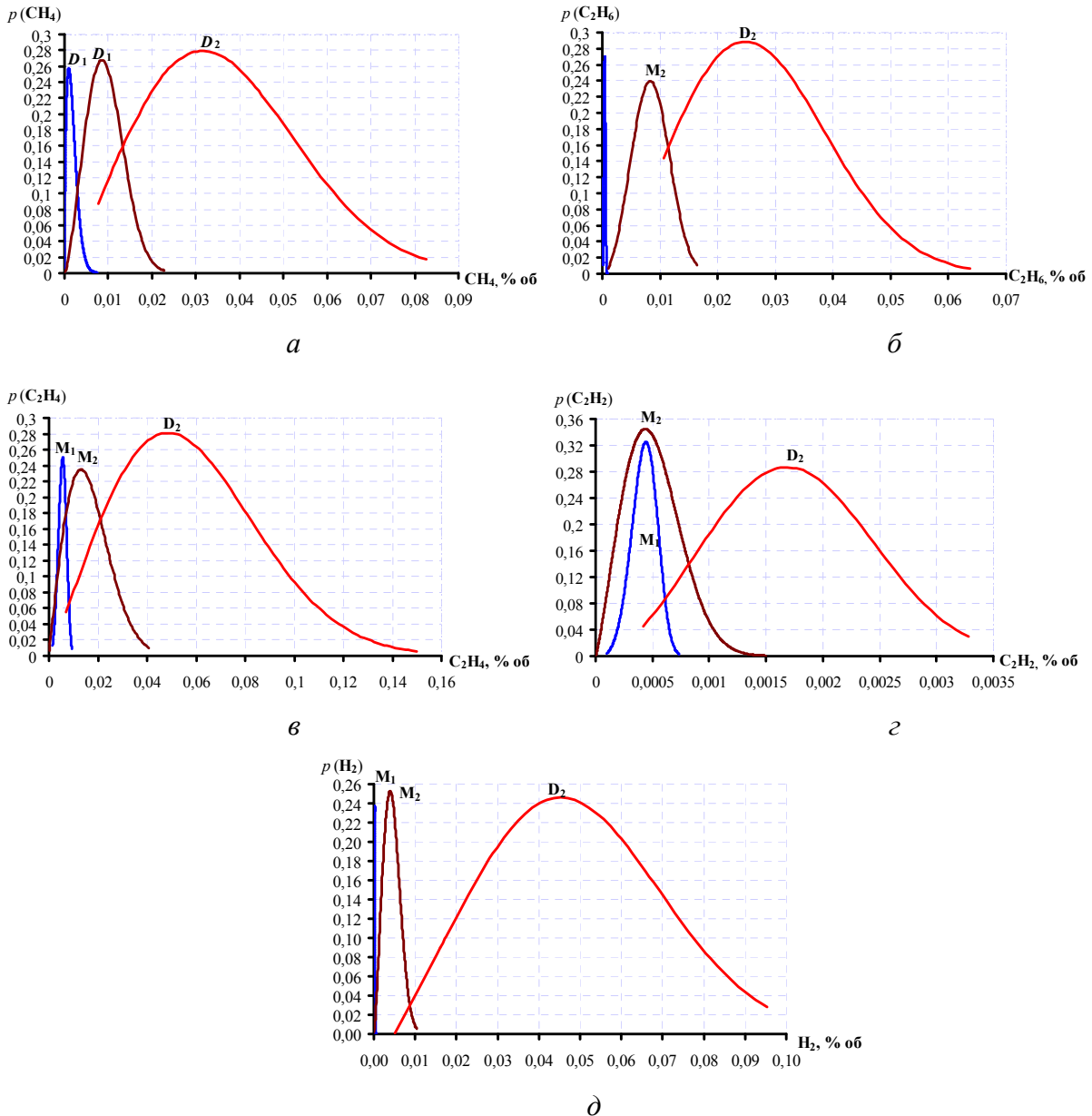


Рисунок 4.64 – Плотности теоретических распределений газов для бездефектного(D_1) и дефектного состояний (D_2):
a – содержание метана; *б* – содержание этана; *в* – содержание этилена;
г – содержание ацетилена; *д* – содержание водорода

Разбиение на термы проводилось следующим образом:

- ЛП_{ВЫХ 1} разделена на два ЛТ
- ЛТ₁₁: <Соответствует – Y >
- ЛТ₁₂: <Не соответствует – N >
- ЛП_{ВЫХ 2} разделена на два ЛТ

ЛТ₂₁: <Соответствует – Y>
 ЛТ₂₂: <Не соответствует – N>
 ЛП_{ВЫХ 3} разделена на два ЛТ
 ЛТ₃₁: <Соответствует – Y>
 ЛТ₃₂: <Не соответствует – N>

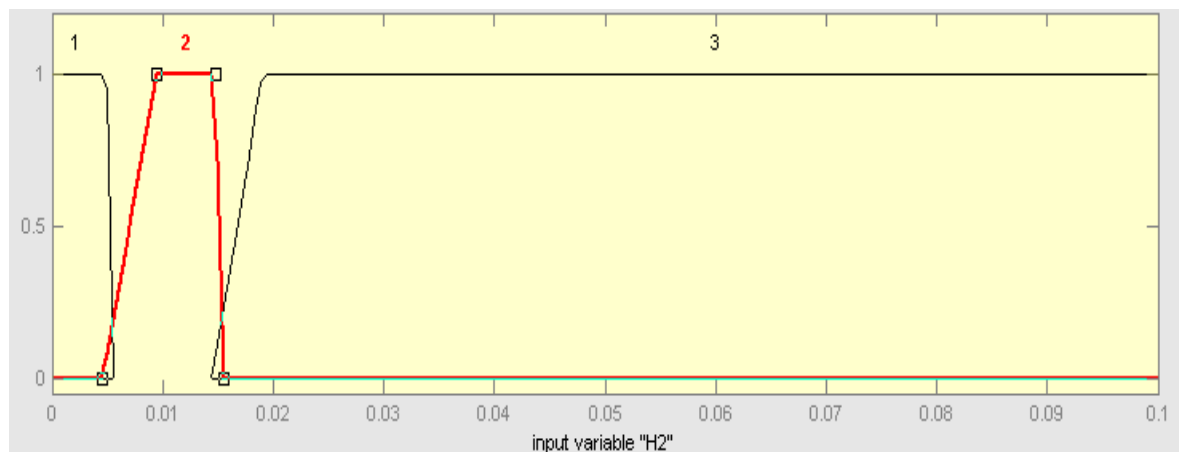


Рисунок 4.65 – Функции принадлежности трех лингвистических термов (для лингвистической переменной – содержание водорода в масле)

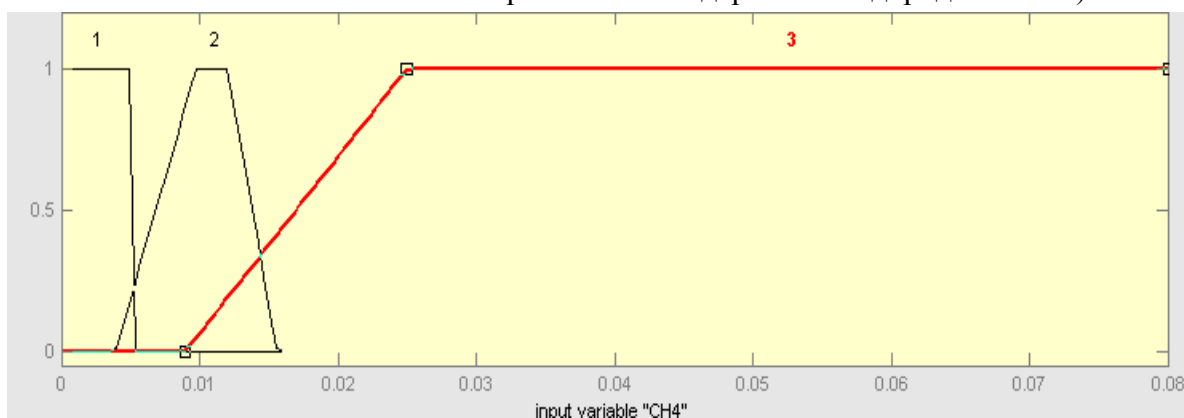


Рисунок 4.66 – Функции принадлежности трех лингвистических термов (для лингвистической переменной – содержание метана в масле)

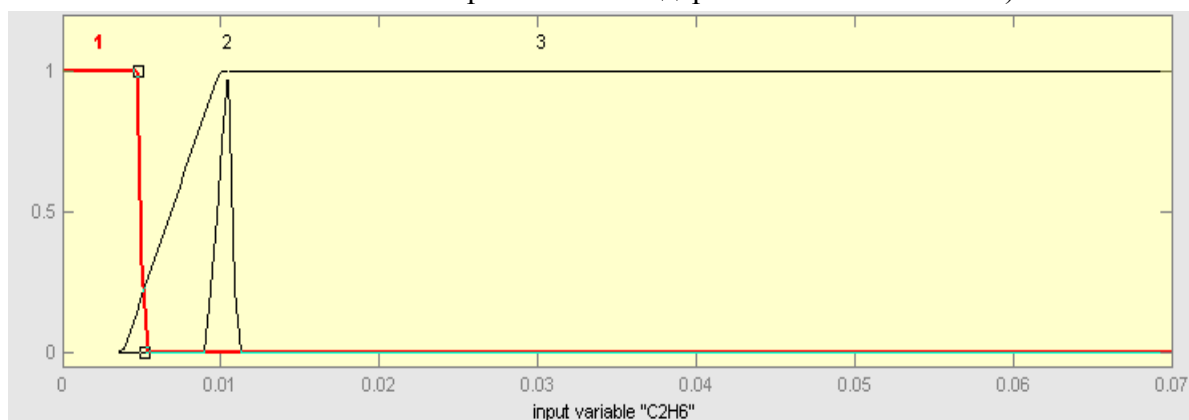


Рисунок 4.67 – Функции принадлежности трех лингвистических термов (для лингвистической переменной – содержание этана в масле)

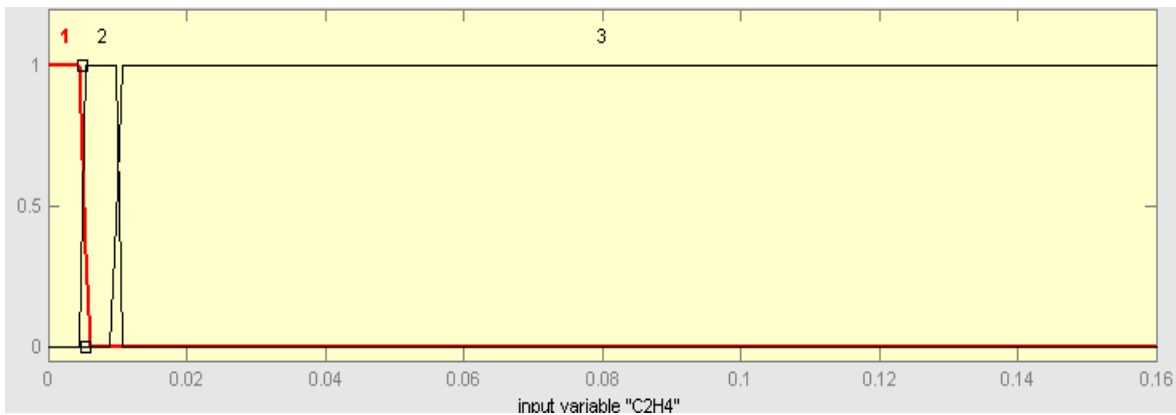


Рисунок 4.68 – Функции принадлежности трех лингвистических термов (для лингвистической переменной – содержание этилена в масле)

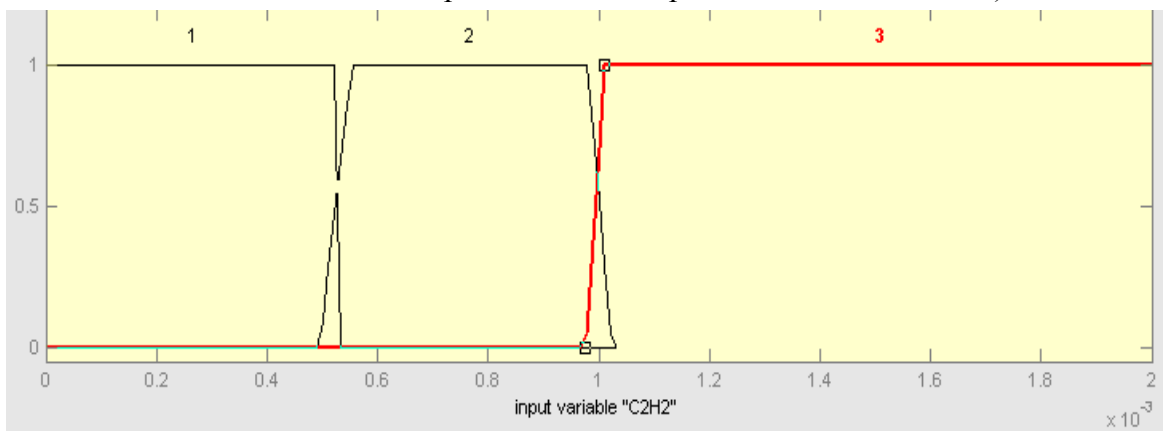


Рисунок 4.69 – Функции принадлежности трех лингвистических термов (для лингвистической переменной – содержание ацетилена в масле)

В качестве функций принадлежности для каждого из термов была выбрана треугольная функция.

Результатом работы блока фаззификации является массив нечетких выходов, которые представляют собой набор пар ЛТ_{ij} и его значение достоверности

$$\text{ЛП}_{\text{вх } i} = \left\{ z_i \mid \mu_{\text{ЛТ}_{ij}}(z_i) \right\}, \mu_{\text{ЛТ}_{ij}}(z_i) \in [0, 1],$$

где z_i – четкое входное значение коэффициента парной корреляции данного газа и времени.

Этап вычисления правила. Система с нечеткой логикой должна иметь базу правил, которая, по сути, представляет собой эмпирические знания эксперта о механизме управления. Для вычисления правил используются нечеткие входы, полученные от блока фаззификации, и правила, находящиеся в базе знаний. В левой части правил перебираются возможные си-

туации на входе системы, а в правой части указывается, какая ЛП описывает правильную реакцию системы.

В качестве решающего правила была реализована методика распознавания, регламентированная в [51]. Сокращенно базу правил можно представить в виде:

Правило 1: ЕСЛИ $[H_2 \in 1]$ И $[CH_4 \in 1]$, И $[C_2H_6 \in 1]$, И $[C_2H_4 \in 1]$, И $[C_2H_2 \in 1]$, ТО $[D \in 1]$;

Правило 2: ЕСЛИ $[H_2 \in 2]$ И $[CH_4 \in 1]$, И $[C_2H_6 \in 1]$, И $[C_2H_4 \in 1]$, И $[C_2H_2 \in 1]$, ТО $[D \in 2]$;

Правило 3: ЕСЛИ $[H_2 \in 1]$ И $[CH_4 \in 2]$, И $[C_2H_6 \in 1]$, И $[C_2H_4 \in 1]$, И $[C_2H_2 \in 1]$, ТО $[D \in 2]$;

...

Правило 6: ЕСЛИ $[H_2 \in 1]$ И $[CH_4 \in 1]$, И $[C_2H_6 \in 1]$, И $[C_2H_4 \in 1]$, И $[C_2H_2 \in 2]$, ТО $[D \in 2]$;

Правило 7: ЕСЛИ $[H_2 \in 2]$ И $[CH_4 \in 2]$, И $[C_2H_6 \in 1]$, И $[C_2H_4 \in 1]$, И $[C_2H_2 \in 1]$, ТО $[D \in 2]$;

...

Правило 17: ЕСЛИ $[H_2 \in 1]$ И $[CH_4 \in 1]$, И $[C_2H_6 \in 1]$, И $[C_2H_4 \in 2]$, И $[C_2H_2 \in 2]$, ТО $[D \in 2]$;

Правило 18: ЕСЛИ $[H_2 \in 2]$ И $[CH_4 \in 2]$, И $[C_2H_6 \in 2]$, И $[C_2H_4 \in 1]$, И $[C_2H_2 \in 1]$, ТО $[D \in 2]$;

...

Правило 27: ЕСЛИ $[H_2 \in 1]$ И $[CH_4 \in 1]$, И $[C_2H_6 \in 2]$, И $[C_2H_4 \in 2]$, И $[C_2H_2 \in 2]$, ТО $[D \in 2]$;

Правило 28: ЕСЛИ $[H_2 \in 2]$ И $[CH_4 \in 2]$, И $[C_2H_6 \in 2]$, И $[C_2H_4 \in 2]$, И $[C_2H_2 \in 1]$, ТО $[D \in 2]$;

...

Правило 32: ЕСЛИ $[H_2 \in 1]$ И $[CH_4 \in 2]$, И $[C_2H_6 \in 2]$, И $[C_2H_4 \in 2]$, И $[C_2H_2 \in 2]$, ТО $[D \in 2]$;

Правило 33: ЕСЛИ $[H_2 \in 2]$ И $[CH_4 \in 2]$, И $[C_2H_6 \in 2]$, И $[C_2H_4 \in 2]$, И $[C_2H_2 \in 2]$, ТО $[D \in 2]$;

Правило 34: ЕСЛИ $[H_2 \in 3]$ И $[CH_4 \in 1]$ ИЛИ $CH_4 \in 2]$, И $[C_2H_6 \in 1]$ ИЛИ $C_2H_6 \in 2]$, И $[C_2H_4 \in 1]$ ИЛИ $C_2H_4 \in 2]$, И $[C_2H_2 \in 1]$ ИЛИ $C_2H_2 \in 2]$, ТО $[D \in 3]$;

...

Правило 48: ЕСЛИ [$H_2 \in 1$ ИЛИ $H_2 \in 2$] И [$CH_4 \in 1$ ИЛИ $CH_4 \in 2$], И [$C_2H_6 \in 1$ ИЛИ $C_2H_6 \in 2$], И [$C_2H_4 \in 3$], И [$C_2H_2 \in 3$], ТО [$D \in 3$];

Правило 49: ЕСЛИ [$H_2 \in 3$] И [$CH_4 \in 3$], И [$C_2H_6 \in 3$], И [$C_2H_4 \in 1$ ИЛИ $C_2H_4 \in 2$], И [$C_2H_2 \in 1$ ИЛИ $C_2H_2 \in 2$], ТО [$D \in 3$];

...

Правило 58: ЕСЛИ [$H_2 \in 1$ ИЛИ $H_2 \in 2$] И [$CH_4 \in 1$ ИЛИ $CH_4 \in 2$], И [$C_2H_6 \in 3$] И [$C_2H_4 \in 3$] И [$C_2H_2 \in 3$] ТО [$D \in 3$];

Правило 59: ЕСЛИ [$H_2 \in 3$] И [$CH_4 \in 3$], И [$C_2H_6 \in 3$], И [$C_2H_4 \in 3$], И [$C_2H_2 \in 1$ ИЛИ $C_2H_2 \in 2$], ТО [$D \in 3$];

...

Правило 63: ЕСЛИ [$H_2 \in 1$ ИЛИ $H_2 \in 2$] И [$CH_4 \in 3$], И [$C_2H_6 \in 3$], И [$C_2H_4 \in 3$], И [$C_2H_2 \in 3$], ТО [$D \in 3$];

Правило 64: ЕСЛИ [$H_2 \in 3$] И [$CH_4 \in 3$], И [$C_2H_6 \in 3$], И [$C_2H_4 \in 3$], И [$C_2H_2 \in 3$], ТО [$D \in 3$];

Далее сеть была протестирована на независимой выборке (значения которой не использовались при обучении). Для удобства анализа тестовые данные были разбиты согласно известным диагнозам на три группы. Результаты тестирования для трансформаторов, не имеющих дефекта, значения концентраций которых соответствуют уровню 1 из табл. 4.1, приведены на рис. 4.70.

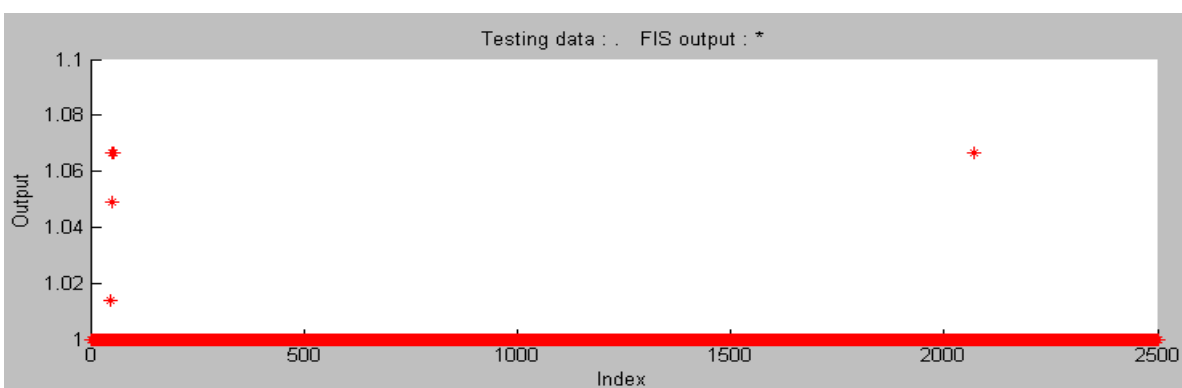
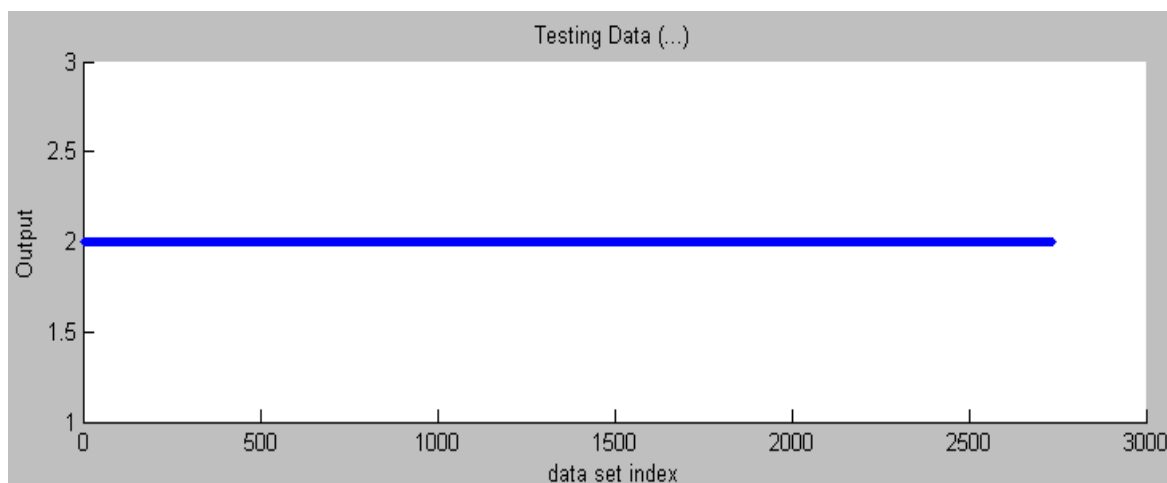


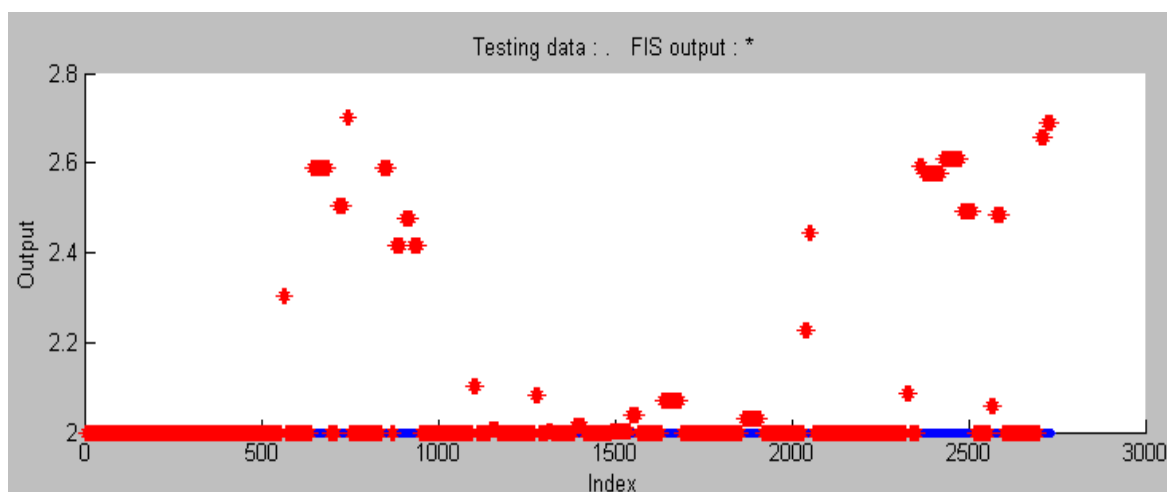
Рисунок 4.70 – Результаты тестирования нечеткой нейронной сети для результатов ХАРГ трансформаторов, не имеющих дефекта

Как видно из рисунка, нечеткая нейронная сеть безошибочно отнесла представленные к распознаванию данные к уровню 1.

Далее обученной нейронной сети были представлены к распознаванию значения концентраций растворенных газов, которые соответствуют уровню 2. Результаты тестирования приведены на рис. 4.71.



a



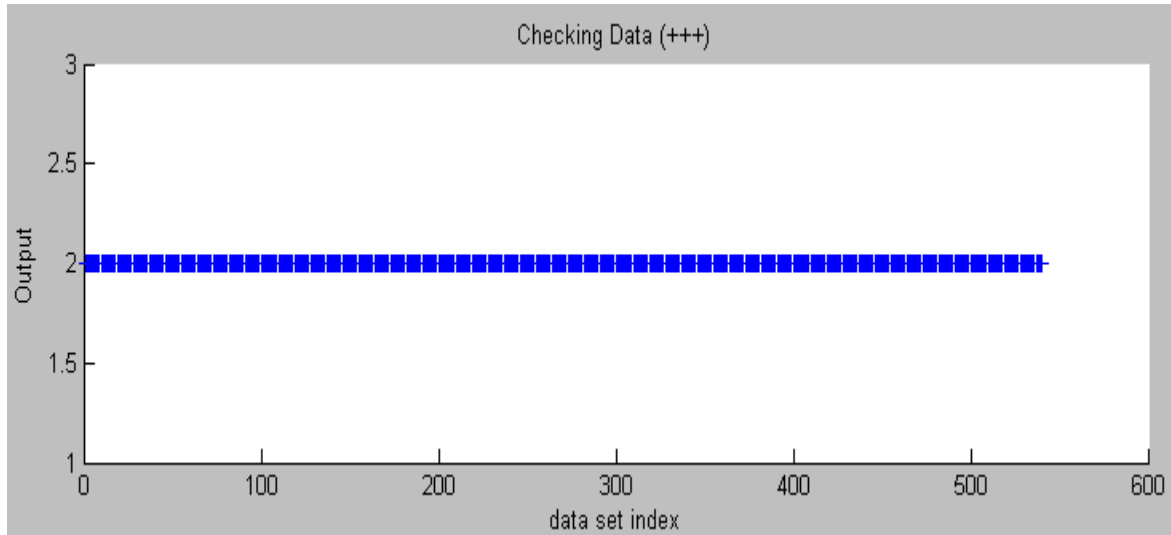
б

Рисунок 4.71 – Результаты тестирования нечеткой нейронной сети для результатов ХАРГ трансформаторов, значения концентраций газов которых соответствуют уровню 2:

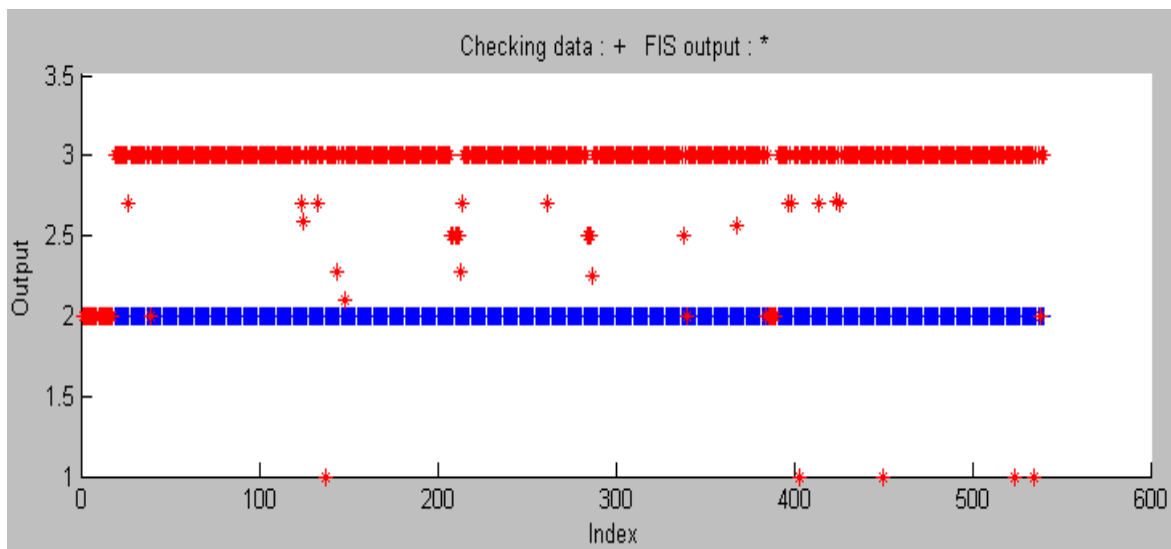
a – исходные данные; *б* – результаты тестирования

На последнем этапе тестирования на вход нечеткой нейронной сети были поданы результаты ХАРГ для оборудования, в котором были выявлены дефекты различного типа. Следует отметить, что в подаваемой на вход сети выборки значение концентрации как минимум одного из газов

ков короткого замыкания, перенапряжений, перегрузки трансформаторов [51–53]), так и нарушения оперативным персоналом в процессе отбора, транспортировки и хранения проб масел, а также при проведении испытаний [55]. Результаты работы сети приведены на рис. 4.73.



a



б

Рисунок 4.73 – Результаты тестирования нечеткой нейронной сети для результатов ХАРГ бездефектных трансформаторов, у которых концентрации газов соответствовали уровню 3:
a – исходные данные; *б* – результаты тестирования

Как видно из рисунка, из 541 поданного на вход результата ХАРГ к бездефектному состоянию сеть отнесла только 5. Выполненный анализ показал, что в этих пяти пробах масел уровни 3 соответствовали значениям

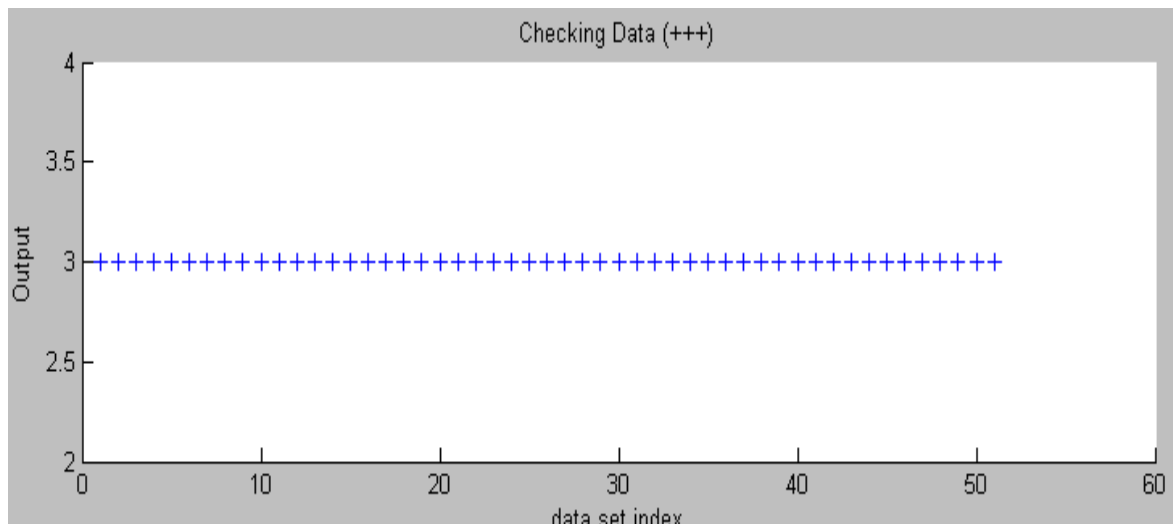
только одного из пяти газов. Значения концентраций остальных четырех не превышали предела обнаружения хроматографом (условно равнялись нулю). Подобное газосодержание не характерно для дефектного состояния, то есть в обучающей выборке такие данные отсутствовали, что, наверное, и позволило сети поставить данный диагноз. Еще 27 результатов ХАРГ были отнесены сетью к подозрительному состоянию, то есть уровню 2. В данных пробах масел концентрация только одного из пяти газов соответствовала уровню 3. Концентрации остальных четырех не превышали предела обнаружения хроматографом или соответствовали уровню 1. Все остальные результаты ХАРГ, предъявленные к распознаванию, были отнесены сетью к состоянию 3, то есть ошибочно признаны дефектными, что является ошибкой первого рода (ложная тревога).

Не менее интересными являются результаты работы нечеткой сети при распознавании состояния трансформаторов по результатам ХАРГ дефектного оборудования, которые получены на ранних стадиях развития дефекта, когда концентрации газов не превышают верхней границы уровня 2. Результаты работы электрической сети отражены на рис. 4.74.

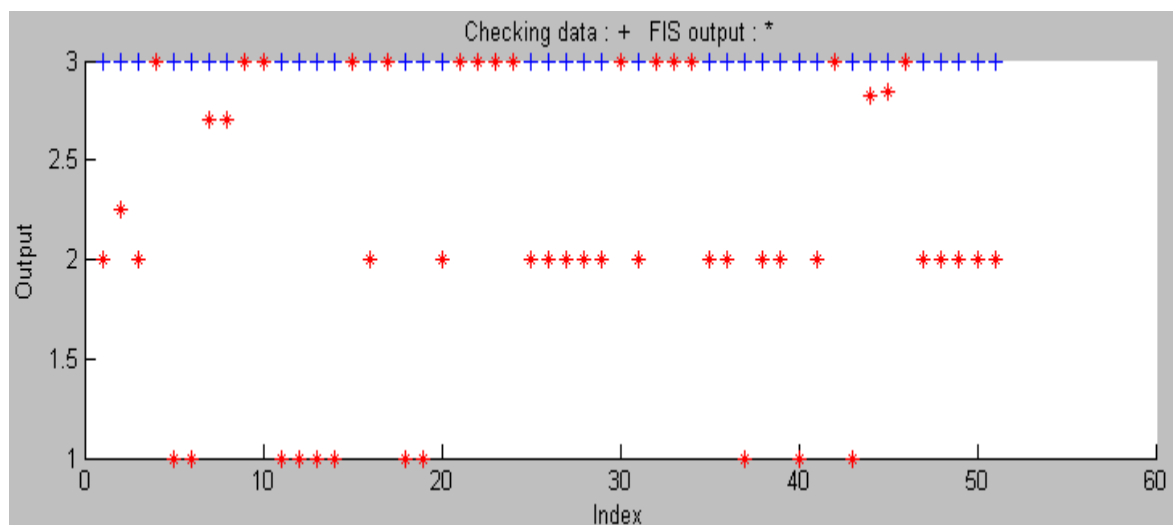
Как видно из рисунка, из 51 результата, поданного на вход сети, только для 15 сетью был поставлен диагноз «дефектное состояние».

Для 25 проб масел с разной степенью принадлежности сетью был поставлен диагноз «подозрительное состояние» (уровень 2). Примечательно, что к уровню 2 сетью были отнесены результаты ХАРГ трансформаторов гидростанции *Nelson river*, Северная Канада [56], которые были получены за 10 месяцев и 5 дней до их повреждения (соответственно 1 и 2 результаты). Следует отметить, что используемая компанией экспертная система не смогла предотвратить повреждение данных трансформаторов. Однако, по представленным результатам ХАРГ, нечеткая нейронная сеть однозначно не определила дефектное состояние в этих трансформаторах, а только отнесла полученные результаты к состоянию «подозрительных концентраций».

Для 11 результатов ХАРГ нечеткая нейронная сеть поставила диагноз «отсутствие дефекта», что является ошибкой второго рода (пропуск цели).



a



б

Рисунок 4.74 – Результаты тестирования нечеткой нейронной сети для результатов ХАРГ дефектных трансформаторов, у которых концентрации газов не превышали граничных значений уровня 2:
a – исходные данные; *б* – результаты тестирования

Анализируя полученные результаты, следует признать, что разработанной и обученной нейронной сети пришлось решать взаимоисключающую задачу. С одной стороны, для распознавания дефектов оборудования на ранней стадии их развития функция принадлежности для дефектного состояния должна быть смещена в область более низких значений, с другой стороны, для распознавания влияния аварийных режимов работы сети функция принадлежности для дефектного состояния должна быть смещена в область более высоких значений. Однако даже в такой ситуации нечеткая нейронная сеть достаточно успешно справилась с задачей, особенно если

учесть, что безошибочное распознавание аварийных воздействий и дефектов на ранней стадии развития на основе использования только значений концентраций газов принципиально невозможно. В то же время, как показано в разделе 3.6, использование в качестве критерия наличия дефекта характера изменений концентраций газов во времени позволяет решить данную задачу практически со 100 % достоверностью. Таким образом, на достоверность диагноза нейронной сети значительное влияние оказывают как диагностические признаки, используемые для постановки диагноза, так и правило принятия решений.

Кроме того, достоверность диагноза, который ставит нейронная сеть, существенно ограничивается особенностями метода ХАРГ. Наглядно это продемонстрировал пример с распознаванием состояния, вызванного аварийным воздействием со стороны электрической сети либо ошибками персонала. Поскольку развитие дефекта, а также воздействие аномально высоких токов, напряжений или температур вследствие аварийных режимов работы сети приводят к возрастанию концентраций газов, то на основании анализа только значений концентраций газов определить причину газовой выделености не представляется возможным. Поэтому ожидать безошибочной постановки диагноза со стороны нейронной сети было бы крайне некорректным.

Вопросы для самоконтроля

1. Биологический нейрон и его кибернетическая модель.
2. Нейронная биологическая сеть.
3. Модели искусственных нейронов. Дать краткую характеристику, перечислить основные достоинства и недостатки.
4. Нейрон Маккалоха – Питтса.
5. Квадратичный нейрон.
6. Обобщенный формальный нейрон.
7. КОГНИТРОН: самоорганизующаяся многослойная нейросеть.
8. НЕОКОГНИТРОН и инвариантное распознавание образов.
9. Стохастическая модель нейрона.
10. Динамические нейроны.
11. Виды активационных функций искусственного нейрона.

12. Биологический и искусственный нейрон: в чем сходства и в чем отличия?

13. Персептрон Адалина.

14. Персептрон Розенблатта.

15. Недостатки однослойных сетей

16. Многослойные персептроны. Основные конфигурации.

17. Радиально-базисные нейронные сети.

18. Основные парадигмы и правила обучения нейронных сетей.

19. Задачи обучения. Перечислить и дать краткую характеристику.

20. Теорема об обучении персептрона.

21. Линейная разделимость. Персептронная представляемость.

22. Обучение методом обратного распространения ошибок.

23. Алгоритмы самообучения. Правило обучения Хэбба.

24. Модель Кохонена.

25. Звезды Гроссберга.

26. Нейронные сети с обратными связями.

27. Модель Хопфилда.

28. Нейронная сеть Хемминга.

29. Нейронные сети ассоциативной памяти.

30. Двухнаправленная ассоциативная память.

31. Адаптивные резонансные нейронные сети.

32. Нейронная сеть встречного распространения.

33. Нечеткие нейронные сети.

34. Сеть Такаги – Сугено – Канга.

35. Гибридный алгоритм обучения сети ТСК.

36. Архитектура *ANFIS*.

37. Применение адаптивной системы нейро-нечеткого вывода на примере решения задачи прогнозирования временных рядов.

38. Особенности прогнозирования значений показателей качества трансформаторных масел с помощью нейро-нечеткой системы *ANFIS*.

39. Какие факторы ограничивают достоверность интерпретации результатов ХАРГ при использовании нечетких нейронных сетей?

Литература к разделу 4

Основная

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен. – М. : Мир, 1992. – 184 с.
2. Бодянский Е.В. Искусственные нейронные сети: архитектуры, обучение, применение / Е.В. Бодянский, О.Г. Руденко. – Харьков : ТЕЛЕ-ТЕХ, 2004. – 371 с.
3. Терехов С.А. Лекции по теории и приложениям искусственных нейронных сетей [Электронный ресурс]. – Режим доступа: http://alife.narod.ru/lectures/neural/Neu_index.htm
4. Галушкин А.И. Теория нейронных сетей / А.И. Галушкин. – М. : ИПРЖР, 2000. – 416 с.

Дополнительная

5. McCulloch W.S. logical calculus of ideas immanent in nervous activity // W.S. McCulloch, W. Pitts // Bull. Math. Biophysics. – 1943. – 5. – P. 115–133.
6. Hecht - Nielsen R. Kolmogorov's mapping neural network existence theorem / R. Hecht - Nielsen // IEEE First Ann. Int. Conf. on Neural Networks. – San Diego. – 1987. – 3. – P. 11–13.
7. Hecht - Nielsen R. Theory of the backpropagation neural network / R. Hecht - Nielsen // Int. Joint Conf. on Neural Networks. – Washington, D.C. – 1989. – 1. – P. 593–606.
8. Fukushima K. Cognitron: a self - organizing multilayered neural network // Biological Cybernetic. – 1975 – Volume 20. – P. 121–136.
9. Fukushima K. Cognitron: a self-organizing multilayer neural network model // NHK Technical Monograph. Available from Nippon Hoso Kyokai (Japanese Broadcasting Corp.), Technical Research Labs, Tokio, Japan. – 1981. – 30. – P. 1–25.
10. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. // Biological Cybernetics. – 1980. – 36(4). – P. 193–202.

11. Fukushima K., Miyake S. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. // Pattern recognition. –1982. – 15(6). – P. 455–469.
12. Изерман Р. Цифровые системы управления / Р. Изерман. – М. : Мир, 1984. – 541 с.
13. Widrow B. Adaptive switching circuits / B. Widrow, M. E. Hoff Jr. // 1960 IRE Western Electric Show and Connection Record. – 1960. – Part 4. – P. 96–104.
14. Widrow B. 30 years of adaptive neural networks: perceptron, adaline and backpropagation / B. Widrow, M. Lee // Proc. IEEE. – 1990. – 78. – P. 1415–1442.
15. Розенблатт Ф. Принципы нейродинамики / Ф. Розенблатт. – М. : Мир, 1965. – 387 с.
16. Минский М. Перцептроны / М. Минский, С. Пейперт. – М. : Мир, 1971. – 262 с.
17. Craddock R.J. Multi - layer radial basis function networks. An extension to the radial basis function / R.J. Craddock, K. Warwick // Proc. Int. Conf. on Neural Networks ICNN'96. – Washington, DC, June 3–6, 1996. – V.2 – P. 700–704.
18. Wilamowski B.M. Implementation of RBF type networks by MLP networks / B.M. Wilamowski, R.C. Jaeger // Proc. Int. Conf. on Neural Networks ICNN'96. – Washington, DC, June 3–6, 1996. – V.3 – P. 1670–1675.
19. Haykin S. Neural Networks. A Comprehensive Foundation / S. Haykin – Upper Saddle River, N.J. : Prentice Hall, Inc. – 1999. – P. 842.
20. Learning internal representations by error propagation / D.E. Rumelhart, G.E. Hinton, and R.J. Williams, D.E. Rumelhart and J.L. McClelland, eds. Parallel Data Processing. – Vol.1, Cambridge, MA: The M.I.T. Press, 1986. – P. 318–362.
21. Kohonen T. Self - organization Maps / T. Kohonen. – Berlin : Springer - Verlag, 1995. – 362 с.
22. Grossberg S. Nonlinear neural networks: principles, mechanisms and architectures / S. Grossberg // Neural networks. – 1988. – 1. – P.17–61.
23. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities / J.J. Hopfield // Proc. Nat. Academy of Sciences USA. – 1982. – 79. – P. 2554–2558.

24. Hopfield J.J. Neural computation of decisions in optimization problems / J.J. Hopfield, D.W. Tank // *Biolog. Cybernetics*. – 1985. – 52. – P. 141–152.
25. Heht - Nielsen R. Counterpropagation networks / R. Heht - Nielsen // *Applied Optics*. – 1987. – 26. – P. 4979–4984.
26. Heht - Nielsen R. Application counterpropagation networks / R. Heht - Nielsen // *Neural Networks*. – 1988. – 1. – P. 131–139.
27. Круглов В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – 2-е изд. – М. : Горячая линия - Телеком, 2002. – 382 с.
28. Леоненков А.В. Нечеткое моделирование в среде MATLAB FuzzyTECH / А.В. Леоненков – С.Пб. : БХВ – Петербург, 2003. – 736 с.
29. Takagi T. Fuzzy identification of system and its application for modeling and control. *IEEE Trans. Systems, Man and Cybernetics* / T. Takagi, M. Sugeno. – 1995 – V.15 N1. – P.116–132.
30. Шутенко О.В. Особенности дрейфа показателей качества трансформаторного масла в течение длительной эксплуатации / О.В. Шутенко // *Інтегровані технології та енергозбереження*. – 2007. – № 4. – С. 26–30.
31. Шутенко О.В. Исследование влияния режимов работы трансформаторов на интенсивность старения масла / О.В. Шутенко // *Енергетика та електрифікація*. – № 8. – С. 54–59.
32. Шутенко О.В. Комплексный корреляционный анализ показателей качества трансформаторного масла / О.В. Шутенко // *Вестник НТУ «ХПИ»*. – 2008. – № 45. – С. 156–167.
33. Шутенко О.В. Прогнозирование значений показателей качества трансформаторного масла с помощью нейро-нечеткой системы ANFIS / О.В. Шутенко // *Світлотехніка та електроенергетика*. – Харків : «ХНАМГ». – 2008. – № 4. – С. 49–56.
34. Оцінка технічного стану і прогнозування ресурсу роботоздатності силових трансформаторів на основі теорії нечітких множин / М.В. Костерев, Є.І. Бардик, Р.В. Вожаков, М.П. Болотний // *Вісник Вінницького політехнічного інституту*. – 2012. – № 2. – С. 83–87.
35. Костерев Н.В. Нечеткое моделирование электрооборудования для оценки технического состояния и принятия решений о стратегии дальнейшей эксплуатации / Н.В. Костерев, Е.И. Бардик // *Технічна*

электродинаміка. – 2006. – Тематичний випуск «Проблеми сучасної електротехніки». – Ч. 3. – С. 39–43.

36. Нечеткие алгоритмы оценки технического состояния и прогнозирования остаточного ресурса электрооборудования / Н.В. Костерев, Е.И. Бардик, Р.В. Вожаков, Т.Ю. Курач // Наукові праці ДонНТУ – Електротехніка і енергетика. – 2008. – В. 8 (140). – С. 65–70.

37. Костерев Н.В. Диагностика электрооборудования электростанций и подстанций с использованием нечеткой логики / Н.В. Костерев, Е.И. Бардик // Матер. другої міжнар. наук. - практ. конф. – Сучасні наукові дослідження – «2006»: Том 15. Технічні науки. – Дніпропетровськ. – 2006. – С. 75–78.

38. Шутенко О.В. Формирование процедуры нечеткого вывода для обнаружения развивающихся дефектов высоковольтных трансформаторов / О.В. Шутенко // Вісник НТУ «ХПІ». – 2008. – № 44. – С. 162–177.

39. Ahmed M.R. Power transformer Fault Diagnosis using Fuzzy Logic Technique Based on Dissolved Gas Analysis / M.R. Ahmed, M.A. Geliel, and A. Khalil – 21-st Mediterranean Conference on Control&Automation (MED). – Plataniass - Chania, Crete, Greece, June 25–28, 2013. – pp. 584–589.

40. Hooshmand R. Application of fuzzy logic in fault diagnosis in transformers using dissolved gas analysis based on different standards / R. Hooshmand, M. Banejad : Proceedings of World Academy of Science, Engineering and Technology. – December, 2006. – Vol. 17. – pp. 157–161.

41. Kunjal Jane, Dissolved Gas Analysis in Transformer using Three Gas Ratio Method and Fuzzy Logic based on IEC Standard / Jane Kunjal, S.A Borakhade // SSRG International Journal of Electrical and Electronics Engineering (SSRG - IJEEE). – Vol. 2, Issue. – 4 April, 2015 (<http://www.internationaljournalssrg.org/IJEEE /2015/Volume2 - Issue4/IJEEE - V2I4P101.pdf>).

42. Dhote N.K. Diagnosis of Power Transformer Faults based on Five Fuzzy Ratio Method / N.K. Dhote, J.B. Helonde // WSEAS Transaction on Power System. –Vol. 7, Issue 3. – 4 July, 2012. – P. 114–125.

43. Flores W. A novel algorithm for the diagnostics of power transformers using type - 2 fuzzy logic systems / W. Flores, E. Mombello, J. Jardini, G. Ratta // Transmission and Distribution Conference and Exposition. IEEE/PES – 2008. – P. 1–5.

44. Mostafa M. Diagnosis of power transformer incipient faults using fuzzy logic - IEC based approach / M. Mostafa, Ibrahim, M. M. Sayed, and E.E. Abou El - Zahab // Energy Conference (ENERGYCON), 2014 IEEE International. – 2014. – P. 242–245.

45. Sherif S.M. Ghoneim and Ibrahim B. Taha. Artificial Neural Networks for Power Transformers Fault Diagnosis Based on IEC Code Using Dissolved Gas Analysis / S.M. Sherif // International journal of control automation and systems. – 2015. – 2. – Vol. 4. – P. 18–21.

46. Mehrdad Beykverdi A New Approach for Transformer Incipient Fault Diagnosis Based on Dissolved Gas Analysis (DGA) / Mehrdad Beykverdi, F. Faghihi, and A. MoArefian pour. // Nova Journal of Engineering and Applied Sciences. – 2014. – Vol 2(3). – pp. 1– 8.

47. Chih - Hsuan Liu Integrated power transformer diagnosis using hybrid fuzzy dissolved gas analysis / Chih - Hsuan Liu, Tung - Bin Lin, Shun - Yuan Wang. // IEEJ Transactions on Electrical and Electronic Engineering. – 2015. – Vol. 10. – no.6. – P. 689–698.

48. Siddique M.A.A., Mehruz S. Artificial neural networks based incipient fault diagnosis for power transformers //India Conference (INDICON), 2015 Annual IEEE. – IEEE, 2015. – P. 1–6.

49. Liu C.H., Yao L.T., Lin, T.B., & Wang, S.Y. Innovated Fault Diagnosis for Power Transformer Using Hybrid Fuzzy Dissolved Gas Analysis //Applied Mechanics and Materials. – 2013. – Vols. 284–287. – P. 1082–1086.

50. Діагностика маслонаповненого трансформаторного обладнання за результатами хроматографічного аналізу вільних газів, відібраних із газового реле, і газів, розчинених у ізоляційному маслі СОУ - Н ЕЕ 46.501: 2006. – Київ. – 2007. – 92 с.

51. Шутенко О.В. Дослідження впливу експлуатаційних факторів на результати хроматографічного аналізу розчинених у маслі газів / О.В. Шутенко // Проблеми енергозабезпечення та енергозбереження в АПК України: Вісник Харківського національного технічного університету сільського господарства імені Петра Василенка. – Харків : ХНТУСГ. – 2008. – Т. 1, В 73. – С. 45–48.

52. Шутенко О.В. Анализ причин газовыделения в силовых трансформаторах, на основе исследования корреляционных связей между растворенными в масле газами / О.В. Шутенко, Д.Н. Баклай, Т.А. Острикова,

Н.Ю. Мельник // Світлотехніка та електроенергетика. – Харків : «ХНАМГ». – 2012. – № 3 – С. 72–81.

53. Шутенко О.В. Анализ факторов, влияющих на однородность массивов концентраций растворенных в масле газов / О.В. Шутенко, В.Б. Абрамов, Д.Н. Баклай // Энергетика та електрифікація. – 2013. – № 6. – С. 39–50.

54. Шутенко О.В. Аналіз законів розподілу концентрацій газів, розчинених в маслі високовольтних трансформаторів негерметичного виконання / О.В. Шутенко, Д.Н. Баклай // Вісник НТУ «ХПІ». – 2014. – № 60 (1033). – С. 136–150.

55. Шутенко О.В. Анализ проблем, возникающих при интерпретации результатов хроматографического анализа растворенных в масле газов / Шутенко О.В., Д.Н. Баклай // Вісник НТУ «ХПІ». – 2013. – № 59 (1032). – С. 164–180.

56. Аксенов Ю.П. Мониторинг технического состояния высоковольтной изоляции электрооборудования энергетического назначения в эксплуатации и при ремонтах / Ю.П. Аксенов. – М. : Научтехлитиздат, 2002. – С. 338.

57. Бондаренко В.Е. Разработка нечеткой нейронной сети для интерпретации результатов анализа растворенных в масле газов / В.Е. Бондаренко, О.В. Шутенко // Електротехніка і Електромеханіка. – Харків: НТУ «ХПІ». – 2017. – №2. – С. 49–56;

Предметный указатель

А

- Абсорбция, 11
- Аксон, 116
- Алгоритм
 - безусловный, 37
 - нечеткого вывода, 76
 - Sugeno, 82
 - Tsukamoto, 84
 - Larsen, 84
 - условный, 37
 - с условной остановкой, 38
- Аппроксимация функций, 158
- Ассоциация, 156
- Ассоциативность, 11
- Аттрактор, 171

Б

- Базис, 11
 - сокращенный, 15
- База
 - знаний, 81

В

- Включение нечетких множеств, 69
- Вычисление правил, 77

Г

- Граф
 - вершина, 33
 - начальная вершина, 36
 - вершина типа вход, 45
 - вершина типа выход, 45
 - обыкновенный, 33
 - ориентированный, 33

- ребра, 33
- связный, 33

Д

- Дендрит, 115
- Дерево, 34
- Дефаззификация, 77
- Дефекты опорных изоляторов, 53
- Диагностика
 - на основе анализа динамики изменений концентраций газов, 87
 - по значениям концентраций газов, 227
- Диагностическое устройство, 13
- Дизъюнкция, 8
- Дизъюнктивная сумма, 70
 - нечетких множеств, 70
 - нечетких отношений, 72
- Дистрибутивность
 - умножения, 11
 - сложения, 11
- Дополнение
 - нечетких множеств, 69
 - нечетких отношений, 72

И

- Изображающее число, 12
- Импликация,
 - четкая, 9
 - нечеткая, 73
- Исход проверки
 - отрицательный, 38
 - положительный, 38

К

Кластеризация, 156
Композиция
нечетких отношений, 73
нечеткого вывода, 81
Коммутативность, 11
Конъюнкция, 8
Корень, 34

Л

Лес, 34
Логическая
величина, 7
функция, 7

М

Матрица
смежности, 40
стоимостей, 48
путей, 40
Методы
нечеткого вывода,
левого модального значения, 95
правого модального значения,
95
центра площади, 94
центра тяжести, 94
построения функций принадлеж-
ности
прямые, 68
косвенные, 68
Модель
Гроссберга (звезда), 173
Кохонена, 171
функциональная, 17

Хопфилда, 177

Множество

высота нечеткого множества, 66
нечеткое, 65
нормальное нечеткое, 66
принадлежностей, 65
субнормальное нечеткое, 66
пустое, нечеткое, 66
универсальное, 65
унимодальное, 65

Н

Нейрон, 115
биологический, 116
Гроссберга, 137
динамический, 135
искусственный, 120
квадратичный, 125
Маккалоха – Питтса, 120
моторный, 120
нечеткий «И», 196
нечеткий «ИЛИ», 196
обобщенный формальный, 125
сенсорный, 120
стохастический, 135
Фукушимы, 130
Хопфилда, 136
Нейронная сеть
ANFIS-сети, 204
автоассоциативные сети, 184
адаптивные резонансные, 187
с ассоциативной памятью, 182
биологическая, 118
встречного распространения,
191

гетероассоциативные, 184
гибридная, 196
дивергентные, 118
дивергентные с одним входом,
119
с двунаправленной ассоциатив-
ной памятью, 185
иерархические, 118
клеточные, 142
конвергентные, 118
локальные, 118
многослойные,
нечеткие, 194
нечеткая стандартная, 195
нечеткая Такаги – Сугено –
Канга, 198
с обратными связями, 142
радиально-базисная, 145
сети для распознавания образов,
184
Хэмминга, 180
Нейромедиатор, 117
Неокогнитрон, 132
Неопределенность, 63
Нечеткость, 64
Нечеткое отношение, 71
Носитель нечеткого множества, 66

О

Объединение,
нечетких множеств, 69
нечетких отношений, 72
Обучающая выборка, 162
Обучение нейронной сети, 151
без учителя, 153

методом обратного
распространения ошибок, 165
с подкреплением, 153
по Розенблатту, 163
с учителем, 152
по Хэббу, 169
Отрицание, 8, 11
Оптимизация, 160
Отказ
неразличимый, 21

П

Переменная
нечеткая, 75
лингвистическая, 75
Пересечение,
нечетких множеств, 69
нечетких отношений, 72
Персептрон, 139
Адалина, 139
Розенблатта, 140
многослойный, 141
Потенциал
покоя, 117
действия, 117
Поглощение, 11
Проверка, 20
Прогнозирование, 161, 217
Произведение нечетких
отношений, 72
Процедура
синтаксическая, 75
семантическая, 75

Р

Равенство нечетких множеств, 69

Разность нечетких множеств, 70

Ранг вершины дерева

максимальный, 36

минимальный, 35

Распознавание

образов, 155

типа дефекта по отношениям пар газов, 105

С

Сглаживание, 161

Синапс, 117

Случайность, 63

Сома, 116

Спайк, 117

Сумма нечетких отношений, 72

Т

Таблица неисправностей, 13

Тавтологии, 10

Тест, 24

диагностический, 24

диагностический для данного графа, 43

длина теста, 46

локализирующий, 43

минимальный, 24

проверяющий, 43

элементарный, 24

Треугольная норма, 71

Треугольная конорма, 71

У

Управление, 160

Ф

Фаззификация, 76

Фильтрация, 161

Функция

активационная нейрона, 122

булева, 10

гауссова, 79, 147

гиперболического тангенса, 122

двойная гауссова, 79

двойная сигмоидальная, 79

линейная, 122

мультикватратичная, 148

обобщенная колоколообразная, 79;

обратная мультикватратичная, 147

принадлежности, 65

произведение двух сигмоидальных функций, 79;

сплайновая, 148

радиально-базисная, 147

сигмоидальная, 79, 122

трапецеидальная, 79

треугольная, 79

характеристическая, 65

Z-функция, 79

S-функция, 79

Pi-функция, 79

Э

Эквивалентность, 9

ДЛЯ ЗАМЕТОК

ДЛЯ ЗАМЕЧАНИЙ

Навчальне видання

БОНДАРЕНКО Володимир Омелянович

ШУТЕНКО Олег Володимирович

БАКЛАЙ Дмитро Миколайович

**МАТЕМАТИЧНІ ОСНОВИ ТЕХНІЧОЇ ДІАГНОСТИКИ
ОБ'ЄКТІВ ЕЛЕКТРИЧНИХ МЕРЕЖ**

Навчальний посібник для студентів спеціальності
«Електроенергетика, електротехніка й Електромеханіка»,
у тому числі для іноземних студентів

Частина друга

**ДІАГНОСТИКА СТАНУ ОБЛАДНАННЯ ЕЛЕКТРИЧНИХ МЕРЕЖ
НА ОСНОВІ ЛОГІЧНИХ МЕТОДІВ РОЗПІЗНАВАННЯ, ТЕОРІЇ ГРАФІВ,
НЕЧІТКОЇ ЛОГІКИ І НЕЙРОННИХ МЕРЕЖ**

Російською мовою

Відповідальний за випуск В.О. Бондаренко

Роботу до видання рекомендував Ю.М. Веприк

Редактор О.В. Козюк

План 2017 р., поз. 19а

Підп. до друку 05 11 18 Формат 60x84 1/16. Папір офсетний.

Друк – ризографія. Гарнітура Times New Roman. Ум. друк. арк. 15,2

Наклад 50 прим. Зам. № __. Ціна договірна

Видавничий центр НТУ «ХП».

Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.

61002, Харків, вул. Кирпичова, 2.
