

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Уфимский государственный авиационный технический университет»

Кафедра технической кибернетики

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В УПРАВЛЕНИИ БАЗОВЫЕ АЛГОРИТМЫ

Практикум
по дисциплинам «Методы искусственного интеллекта
в управлении», «Интеллектуальное управление сложными
объектами», «Интеллектуальное управление сложными
техническими объектами», «Методы искусственного интеллекта
в управлении техническими объектами», «Программные системы
и комплексы в управлении качеством»



Уфа 2021

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Уфимский государственный авиационный технический университет»
Кафедра технической кибернетики

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В УПРАВЛЕНИИ

БАЗОВЫЕ АЛГОРИТМЫ

Практикум

по дисциплинам «Методы искусственного интеллекта
в управлении», «Интеллектуальное управление сложными
объектами», «Интеллектуальное управление сложными
техническими объектами», «Методы искусственного интеллекта
в управлении техническими объектами», «Программные системы
и комплексы в управлении качеством»

Учебное электронное издание сетевого доступа

Уфа 2021

Авторы-составители: Б. Г. Ильясов, Е. А. Макарова, Е. Ш. Закиева,
Э. Р. Габдуллина

Методы искусственного интеллекта в управлении. Базовые алгоритмы : практикум по дисциплинам «Методы искусственного интеллекта в управлении», «Интеллектуальное управление сложными объектами», «Интеллектуальное управление сложными техническими объектами», «Методы искусственного интеллекта в управлении техническими объектами», «Программные системы и комплексы в управлении качеством» [Электронный ресурс] / Уфимск. гос. авиац. техн. ун-т ; [авт.-сост. : Б. Г. Ильясов, Е. А. Макарова, и др.]. – Уфа : УГАТУ, 2021. – URL: https://www.ugatu.su/media/uploads/MainSite/Ob%20universitete/Izdateli/El_izd/2021-53.pdf

Цель практикума – закрепление и совершенствование знаний студентов по дисциплинам «Методы искусственного интеллекта в управлении», «Интеллектуальное управление сложными объектами», «Интеллектуальное управление сложными техническими объектами», «Методы искусственного интеллекта в управлении техническими объектами», «Программные системы и комплексы в управлении качеством», формирования умений и навыков решения задач управления системами искусственного интеллекта, на основе применения которых осуществляется моделирование и управление поведением реальных объектов.

Предназначен для подготовки бакалавров по направлениям подготовки 27.03.03 Системный анализ и управление, 27.03.04 Управление в технических системах, для магистров по направлениям подготовки 27.04.02 Управление качеством.

Рецензент канд. техн. наук, доцент С. В. Сильнова

При подготовке электронного издания использовались следующие программные средства:

- Adobe Acrobat – текстовый редактор;
- Microsoft Word – текстовый редактор.

Авторы-составители: *Ильясов Барый Галеевич,
Макарова Елена Анатольевна,
Закиева Елена Шавкатовна,
Габдуллина Эльвира Риятовна*

Компьютерная верстка: *О. А. Соколова*
Программирование и компьютерный дизайн: *А. П. Меркулова*

ФГБОУ ВО «Уфимский государственный авиационный технический университет»
450008, Уфа, ул. К. Маркса, 12.
Тел.: +7-908-35-05-007
e-mail: rik.ugatu@yandex.ru

Все права на размножение, распространение в любой форме остаются за разработчиком.
Нелегальное копирование, использование данного продукта запрещено.

ВВЕДЕНИЕ

В настоящее время все актуальнее становится реализация проектов Интернета вещей (*Industrie 4.0*), которые требуют разработки социо-кибер-физических систем (*SCPS – socio-cyber-physical system*), состоящих из различных природных объектов, искусственных подсистем, распределенных сенсоров и управляющих контроллеров. Для этого необходимы знания и умения работы с интеллектуальными технологиями и системами искусственного интеллекта, на основе применения которых осуществляется моделирование и управление поведением реальных объектов, в том числе и уточнение свойств объекта исследования и особенностей управления им с помощью обучения по имеющимся данным и обобщения информации, а также генерирования правила вывода для анализа и принятия решений.

Материалы данного практикума предназначены для закрепления знаний методов искусственного интеллекта, а также умений и навыков их применения. В практикуме рассматриваются:

- искусственные нейронные сети, представляющие собой стохастические алгоритмы для обучения модели с учителем и без учителя;

- нечеткие множества и алгоритмы нечеткого логического вывода, являющиеся инструментом моделирования неопределенностей естественного языка;

- генетические алгоритмы – эволюционные алгоритмы, построенные на принципах естественного отбора и генетики, позволяющие эффективно исследовать пространство поиска оптимальных решений.

Содержание практикума:

- соответствует основной профессиональной образовательной программе (ОПОП) 27.03.03 *Системный анализ и управление*, содержанию рабочих программ дисциплин вариативной части учебного плана «Методы искусственного интеллекта в управлении» (180/5), «Интеллектуальное управление сложными объектами» (180/5) и направлено на освоение компетенций ПК-1 – способность принимать научно-обоснованные решения на основе математики, физики, химии, информатики, экологии, методов системного анализа и теории управления, теории знаний, осуществлять постановку и

выполнять эксперименты по проверке их корректности и эффективности; ПК-5 – способность разрабатывать методы моделирования, анализа и технологии синтеза процессов и систем в области техники, технологии и организационных систем (*разделы 2,3,4, 109 ч*);

– соответствует ОПОП 27.03.04 *Управление в технических системах*, содержанию рабочей программы дисциплины вариативной части учебного плана «Интеллектуальное управление сложными техническими объектами» (144/4), «Методы искусственного интеллекта в управлении техническими объектами» (144/4) и направлено на освоение компетенции ПК-5 – способность осуществлять сбор и анализ исходных данных для расчета и проектирования систем и средств автоматизации и управления (*разделы 2,3,4, 109 ч*);

– соответствует ОПОП 27.04.02 *Управление качеством*, содержанию рабочей программы дисциплины вариативной части учебного плана «Программные системы и комплексы в управлении качеством» (144/4) и направлено на освоение компетенций ОПК-5 – способность к профессиональной эксплуатации современного оборудования и приборов; ПК-6 – способность осуществлять постановку задачи исследования, формирование плана его реализации (*раздел 5, 34 ч*).

Практическая работа № 1

НЕЧЕТКИЕ МНОЖЕСТВА

1. Цель и задачи работы

Целью работы является закрепление знаний о нечетких множествах, приобретение навыков проведения нечетких операций и построения нечетких отношений.

Задачами работы являются формирование умений выполнения нечетких операций дополнения, пересечения и объединения, а также выполнения композиции нечетких отношений аналитическим способом свертки.

2. Теоретические сведения

2.1. Основные понятия нечетких множеств

Предложенная Л. Заде в 1965 г. *теория нечетких множеств* (*fuzzy set theory*) представляет собой формализм, предназначенный для описания таких явлений и понятий, которые имеют многозначный и неточный характер.

Лингвистической переменной (ЛП) называется переменная, значениями которой могут быть слова или словосочетания естественного языка (скорость, температура, напряжение и т.д.).

Термом (лингвистическим значением) называется значение ЛП, выраженное в словесной форме (высокий, положительный, небольшой). Термы всегда присутствуют в модели вместе со связанной ЛП (например, высокая скорость, низкая температура и т.д.).

Лингвистическим терм-множеством называется множество всех возможных значений ЛП. Например,

$$x_A = \{\text{малый, средний, большой}\} = \{x_{A1}, x_{A2}, x_{A3}\}.$$

Для сокращения записей могут использоваться *символические обозначения* значений отдельных термов ЛП (табл. 1.1).

Нечетким множеством A в некотором пространстве X ($A \subseteq X$) называется множество пар

$$A = \{(x, \mu_A(x)); x \in X\},$$

где $\mu_A(x): X \rightarrow [0,1]$ – функция принадлежности (ФП) нечеткого множества A , которая характеризует степень принадлежности

каждого элемента x нечеткому множеству A , причем $\mu_A(x) = 1$ означает полную принадлежность элемента нечеткому множеству; $\mu_A(x) = 0$ – отсутствие принадлежности, т.е. $x \notin A$; $0 < \mu_A(x) < 1$ – частичную принадлежность элемента x к нечеткому множеству A .

Таблица 1.1

Символические обозначения значений термов ЛП

Символическое обозначение	Англоязычная нотация	Русскоязычная нотация
<i>NB</i>	<i>Negative Big</i>	Отрицательное большое
<i>NM</i>	<i>Negative Middle</i>	Отрицательное среднее
<i>NS</i>	<i>Negative Small</i>	Отрицательное малое
<i>ZN</i>	<i>Zero Negative</i>	Отрицательное близкое к нулю
<i>Z</i>	<i>Zero</i>	Нуль, близкое к нулю
<i>ZP</i>	<i>Zero Positive</i>	Положительное близкое к нулю
<i>PS</i>	<i>Positive Small</i>	Положительное малое
<i>PM</i>	<i>Positive Middle</i>	Положительное среднее
<i>PB</i>	<i>Positive Big</i>	Положительное большое

Дискретное нечеткое множество A может быть записано в форме объединения (а не арифметической суммы) пар $(\mu_A(x_i)/x_i)$:

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i},$$

где знак «+» означает не деление, а приписывание конкретным элементам x_1, \dots, x_n степеней принадлежности $\mu_A(x_1), \dots, \mu_A(x_n)$.

Таким образом, запись $\frac{\mu_A(x_i)}{x_i}$ равносильна $(x_i, \mu_A(x_i))$.

Непрерывное нечеткое множество может быть записано в интегральной форме $A = \int_X \frac{\mu_A(x)}{x}$.

Нечеткое множество называется *нормальным* (нормированным), если его ФП принимает значения в интервале от 0 до 1, и при этом существуют элементы, у которых степень принадлежности нечеткому множеству равна 1. Нечеткое множество называется *субнормальным*, если максимальное значение его ФП меньше 1.

Высота нечеткого множества A определяется как максимальное из значений, принимаемых ФП нечеткого множества на всей области определения X : $h = \sup_{x \in X} (\mu_A(x))$.

Носитель нечеткого множества A представляет собой четкое подмножество области определения X , содержащее все элементы, степени принадлежности которых множеству A отличны от нуля: $S(A) = \text{supp}(A) = \{x: \mu_A(x) > 0, x \in X\}$.

Ядро нечеткого множества A – четкое подмножество области определения X , содержащее все элементы, принадлежащие множеству A со степенью, равной 1: $C(A) = \text{core}(A) = \{x: \mu_A(x) = 1, x \in X\}$. Ядро субнормального множества является пустым.

Горизонтальная форма представления нечеткого множества соответствует его заданию с помощью *-срезов* A^α , введение которых обусловлено упрощением процедуры извлечения экспертных знаний для построения ФП. Так, если эксперт затрудняется задать конкретные значения степеней принадлежности элементов нечеткому множеству, он может обозначить элементы, принадлежащие нечеткому множеству со степенью, не меньшей α , следующим образом:

$$A^{>\alpha} = \{x: x \in X, \mu_A(x) > \alpha\} \text{ или } A^{\geq\alpha} = \{x: x \in X, \mu_A(x) \geq \alpha\}.$$

При $\alpha = 0$ α -срез совпадает с носителем множества $S(A)$, а при $\alpha = 1$ – с его ядром $C(A)$.

На рис. 1.1 представлены некоторые характеристики нечетких множеств.

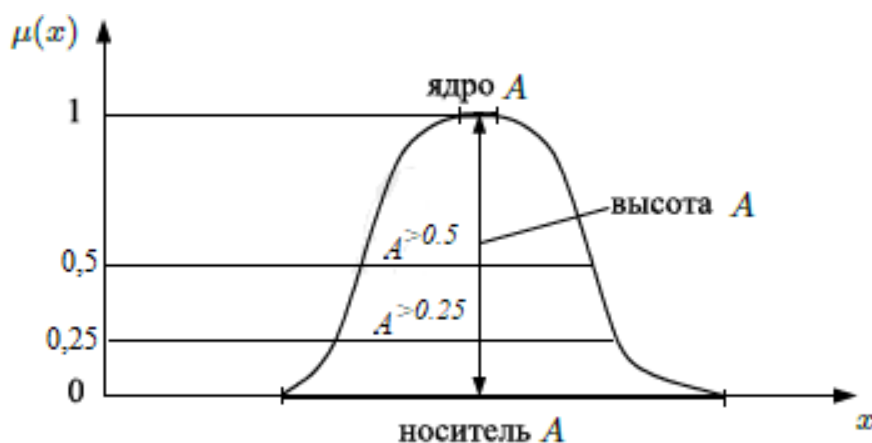


Рис. 1.1. Характеристические показатели нечеткого множества

Выпуклое нечеткое множество обладает тем свойством, что все его α -срезы являются связными одноинтервальными подмножествами области определения X . У невыпуклого множества имеются

несвязные α -срезы, состоящие из нескольких частей (рис. 1.2). Для выпуклых нечетких множеств справедливы следующие условия:
 $x_1 \leq x_2 \leq x_3 \Rightarrow \min(\mu_A(x_1), \mu_A(x_3)), \forall x_1, x_2, x_3 \in X$.

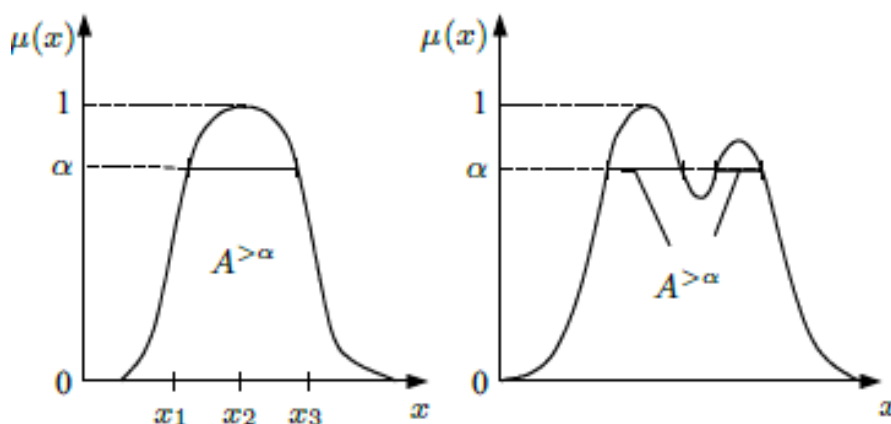


Рис. 1.2. Примеры выпуклого и невыпуклого нечетких множеств

Для создания нечетких множеств, являющихся производными от некоторых ранее заданных, используются лингвистические модификаторы. Например, если имеется нечеткое множество «холодный», то на его основе с помощью лингвистических модификаторов можно получить множества «очень холодный» или «более или менее холодный».

Выделяют два основных модификатора (рис. 1.3):

1) оператор концентрирования (*concentration*), который позволяет получить производное значение «очень»:

$$\mu_{CON(A)}(x) = CON(\mu_A(x)) = (\mu_A(x))^2, \forall x \in X.$$

2) оператор растяжения, значение которого соответствует характеристикам «довольно», «более или менее»:

$$\mu_{DIL(A)}(x) = DIL(\mu_A(x)) = \sqrt{\mu_A(x)}, \forall x \in X.$$

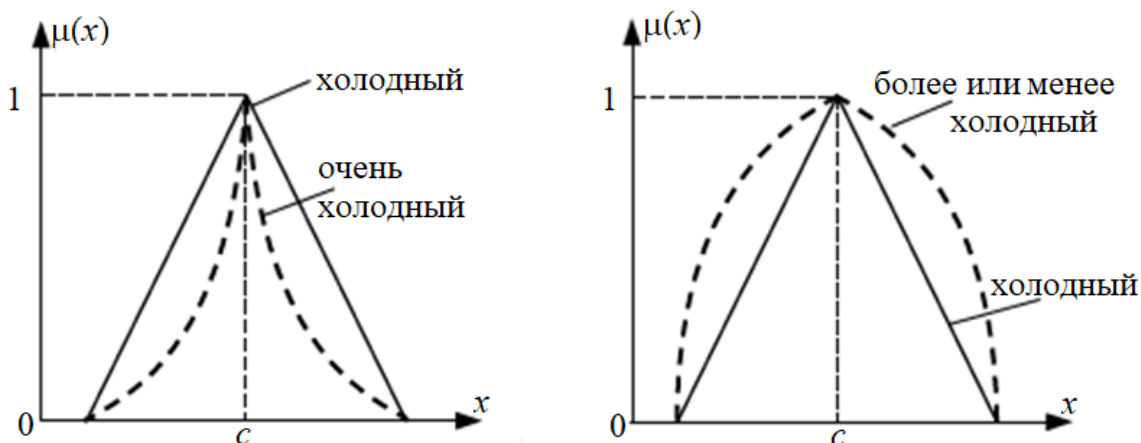


Рис. 1.3. Примеры операторов концентрирования и растяжения

2.2. Построение характеристических функций принадлежности

В практических приложениях теории нечетких множеств используется большое количество различных типов ФП. Рассмотрим некоторые из них.

2.2.1. *Треугольная* ФП (рис. 1.4) – кусочно-линейная функция, состоящая из четырех участков. На I ($x \leq a$) и IV ($x \geq c$) участках ФП равны нулю: $\mu_I(x) = 0, \mu_{IV}(x) = 0$. На II участке ($a \leq x \leq e$) функция принадлежности μ_{II} строится исходя из отношений сторон, полученных на основе подобия треугольников: $\operatorname{tg}(\alpha) = \frac{\mu_{II}(x)}{x-a} = \frac{1}{e-a}$, отсюда:

$$\mu_{II}(x) = \frac{x-a}{e-a}.$$

ФП III участка ($e \leq x \leq c$) строится аналогичным образом:

$$\operatorname{tg}(\beta) = \frac{\mu_{III}(x)}{c-x} = \frac{1}{c-e} \Rightarrow \mu_{III}(x) = \frac{c-x}{c-e}.$$

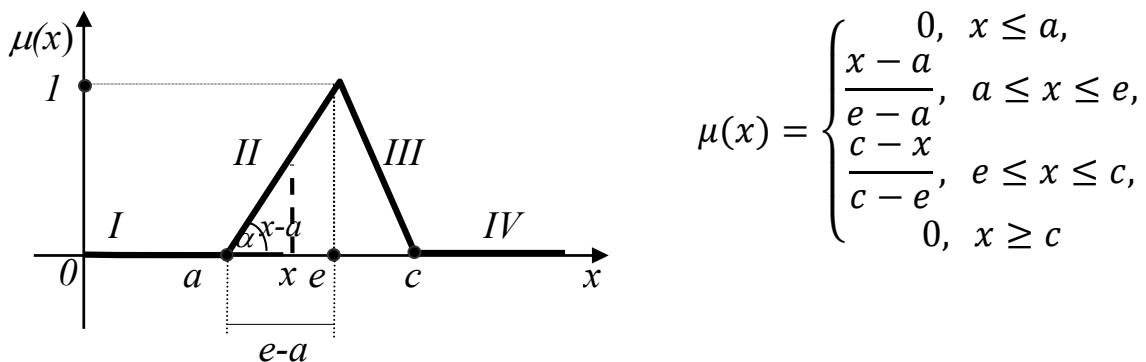


Рис. 1.4. Треугольная ФП в графическом и в аналитическом виде

2.2.2. *Трапецевидная* ФП. В аналитическом виде ФП может быть представлена так же, как и в п. 2.1.1, в виде кусочно-линейной функции. Существует и другой способ математической записи ФП, который заключается во введении дополнительных переменных. Например, для ассиметричной *трапецевидной* ФП с тремя участками (рис. 1.5) вводятся дополнительные логические переменные:

$$w_1 = \begin{cases} 1 \text{ для } a \leq x < b, \\ 0 \text{ в других случаях} \end{cases}$$

$$w_2 = \begin{cases} 1 \text{ для } b \leq x < c, \\ 0 \text{ в других случаях} \end{cases}$$

$$w_3 = \begin{cases} 1 \text{ для } c \leq x \leq d \\ 0 \text{ в других случаях} \end{cases}$$

Отсюда, ФП, имеющая форму асимметричной трапеции, может быть представлена в виде: $\mu(x) = w_1 \left(\frac{x-a}{b-a} \right) + w_2 + w_3 \left(\frac{d-x}{d-c} \right)$.

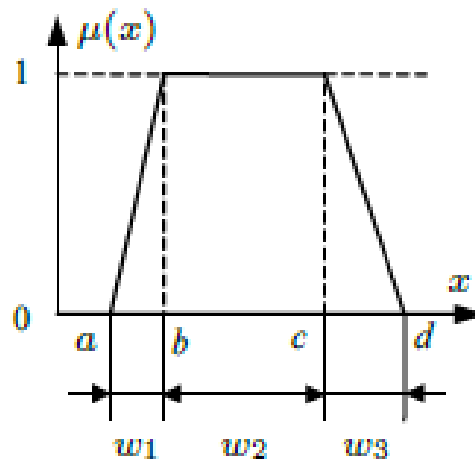


Рис. 1.5. Пример асимметричной трапецевидной ФП

2.2.3. Гауссова (экспоненциальная) ФП. Недостатком многоугольных ФП является то, что они не являются непрерывно дифференцируемыми. Примером непрерывной ФП является гауссовый колокол, формула которого имеет вид: $\mu(x) = \exp \left[- \left(\frac{x-b}{a} \right)^2 \right]$, где параметр b задает модальное значение функции, параметр a – ширину.

2.2.4. Z- и S-образные ФП. Z-образные ФП (рис. 1.6, а) используются для представления таких свойств нечетких множеств, которые характеризуются неопределенностью типа «малый», «низкий», т.е. со слабой степенью проявления того или иного качественного признака. S-образные ФП (рис. 1.6, б), напротив, используются для представления таких характеристик как «большой», «высокий» и т.д.

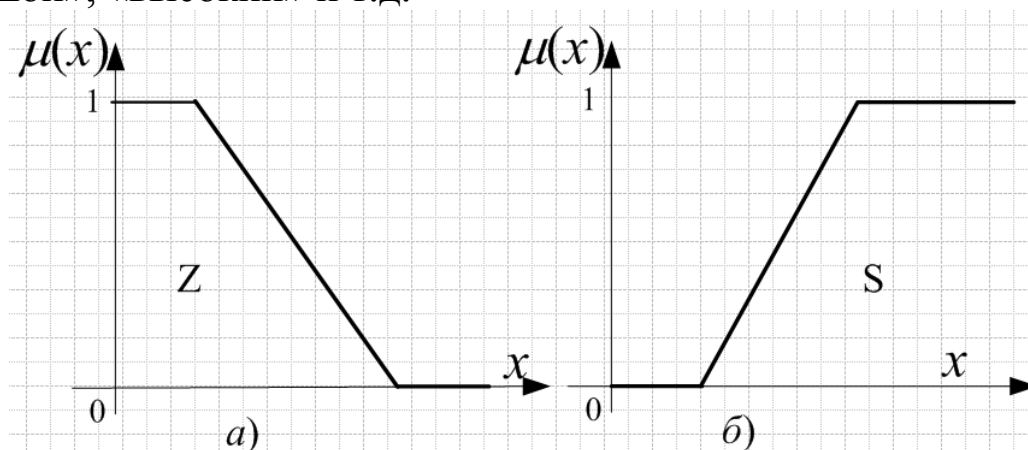


Рис. 1.6. Графики ФП Z- и S-типа

2.2.5. ФП с *ограниченным диапазоном*. ФП может иметь форму (рис. 1.7), в которой диапазон изменения параметров ограничен с обеих сторон. В этом случае для полного задания трех ФП достаточно трех (вместо девяти) значений: x_{\min} , x_{med} , x_{\max} .

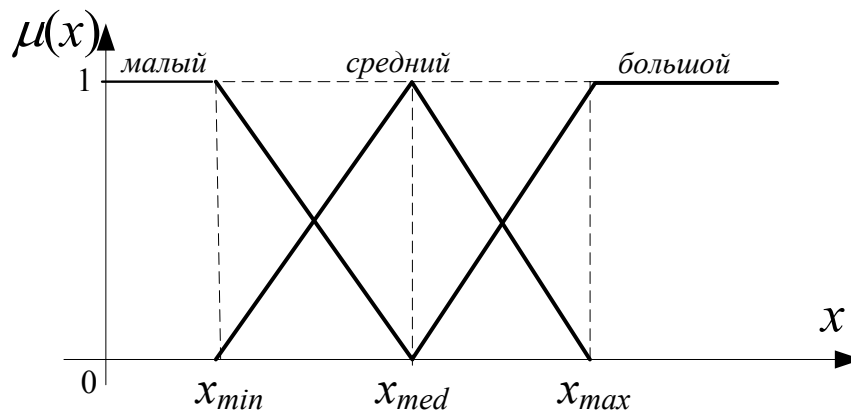


Рис. 1.7. Пример ФП с ограниченным диапазоном изменения параметров

2.3. Операции с нечеткими множествами

2.3.1. Нечеткая операция «НЕ» (дополнение).

В теории нечетких множеств, в отличие от классической теории множеств, возможно несколько видов операций дополнения, обозначаемых \bar{A} :

а) нечеткое «НЕ» по Л. Заде $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ (рис. 1.8);

б) нечеткое «НЕ» по Сугено $\mu_{\bar{A}}(x) = \frac{1 - \mu_A(x)}{1 + \lambda \cdot \mu_A(x)}$.

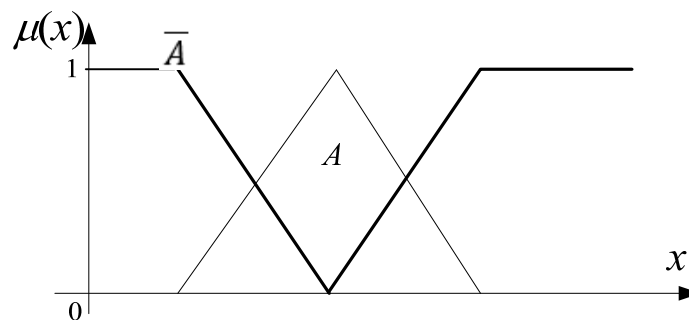


Рис. 1.8. Нечеткое множество и его дополнение

2.3.2. Нечеткая операция И – пересечение нечетких множеств.

Операция пересечения нечетких множеств является расширением операции пересечения над *обычными* множествами.

Рассмотрим операции пересечения на примере двух нечетких множеств автомобилей x_i :

$$A = \left\{ \frac{1}{x_1}; \frac{0,8}{x_2}; \frac{0,6}{x_3}; \frac{0,4}{x_4}; \frac{0,2}{x_5}; \frac{0}{x_6} \right\}, \quad B = \left\{ \frac{0}{x_1}; \frac{0,2}{x_2}; \frac{0,4}{x_3}; \frac{0,6}{x_4}; \frac{0,8}{x_5}; \frac{1,0}{x_6} \right\};$$

где A – множество дешевых автомобилей, B – множество комфортабельных автомобилей.

Автомобиль x_4 может быть отнесен к дешевым со степенью $\mu_A(x_4) = 0,4$ и к комфортабельным – со степенью $\mu_B(x_4) = 0,6$ (рис. 1.9). В какой степени его можно одновременно считать дешевым и комфортабельным, и как определить эту степень, используя степени его принадлежности $\mu_A(x_4)$ и $\mu_B(x_4)$ соответствующим множествам?

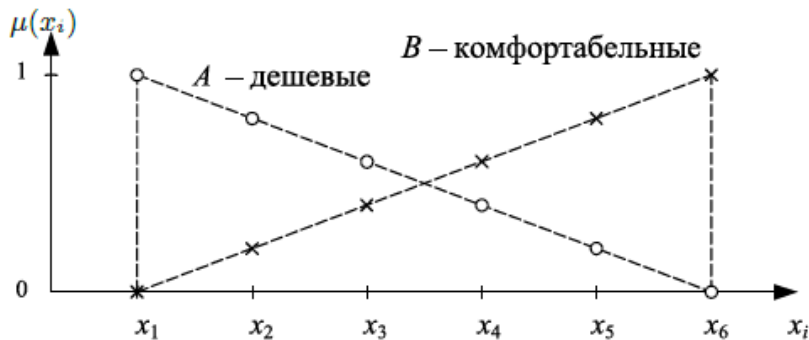


Рис. 1.9. ФП множеств дешевых A и комфортабельных B автомобилей

Л. Заде предложил вычислять значения ФП произведения множеств по формуле с использованием оператора MIN :

$$\mu_{A \cap B}(x) = MIN(\mu_A(x), \mu_B(x)), \forall x \in X.$$

Применяя оператор MIN для нахождения множества $A \cap B$ дешевых и, вместе с тем, комфортабельных автомобилей, получаем выражение:

$$C = A \cap B = \left\{ \frac{0}{x_1}; \frac{0,2}{x_2}; \frac{0,4}{x_3}; \frac{0,4}{x_4}; \frac{0,2}{x_5}; \frac{0}{x_6} \right\}.$$

Нечеткое множество, являющееся результатом данной операции, представлено на рис. 1.10, а.

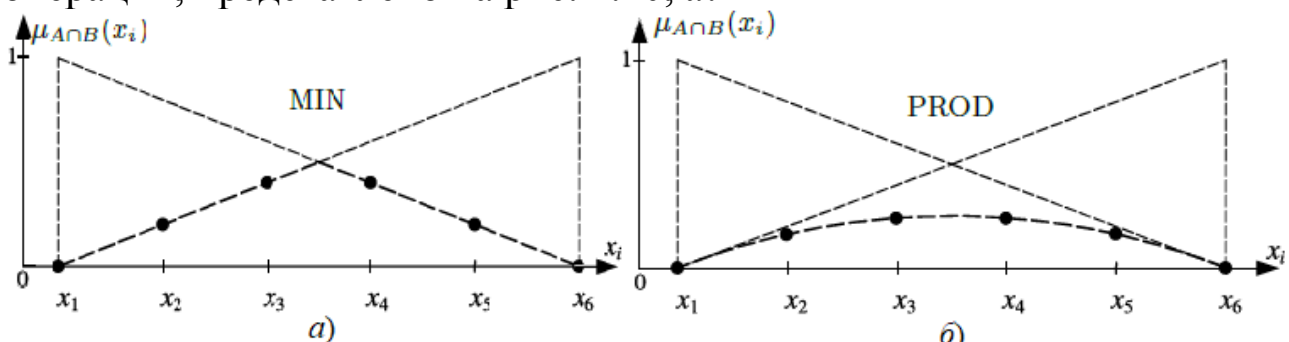


Рис. 1.10. Произведение нечетких множеств $A \cap B$ с использованием операторов MIN и $PROD$

Использование оператора MIN для операции пересечения множеств приводит к потере части информации, поскольку данный оператор учитывает только то, что одна степень принадлежности меньше другой, без учета значения их разности. По этой причине для моделей систем, использующих оператор MIN , обычно характерны нечувствительность к малым изменениям значений входных величин, а также резкие изменения выходного значения при превышении некоторого порогового уровня входных значений.

Менее существенна потеря информации при использовании оператора $PROD$, основанном на алгебраическом произведении ФП.

$$\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x), \forall x \in X.$$

Применяя оператор $PROD$ для нахождения множества $A \cap B$ автомобилей, получаем выражение (рис. 1.10, б):

$$C = A \cap B = \left\{ \frac{0}{x_1}; \frac{0,16}{x_2}; \frac{0,24}{x_3}; \frac{0,24}{x_4}; \frac{0,16}{x_5}; \frac{0}{x_6} \right\}.$$

Наиболее часто в качестве операторов пересечения $A \cap B$ используются так называемые T -нормы, определяющие различные формы реализации данной операции.

Оператор T -нормы представляет собой функцию T , моделирующую операцию \cap пересечения двух нечетких множеств A и B , удовлетворяющую перечисленным ниже свойствам:

- 1) пространство отображения: $T: [0,1] \times [0,1] \rightarrow [0,1]$;
- 2) свойство обнуления: $T(0,0) = 0$;
- 3) свойство коммутативности: $T(\mu_A(x), \mu_B(x)) = T(\mu_B(x), \mu_A(x))$;
- 4) свойство ассоциативности:

$$T(\mu_A(x), T(\mu_B(x), \mu_C(x))) = T[T(\mu_A(x), \mu_B(x)), \mu_C(x)];$$

- 5) свойство монотонности:

$$\mu_A(x) \leq \mu_C(x), \mu_B(x) \leq \mu_D(x) \implies T(\mu_A(x), \mu_B(x)) \leq T(\mu_C(x), \mu_D(x)),$$

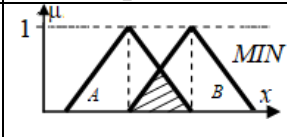
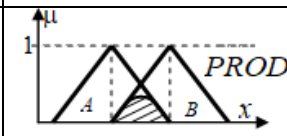
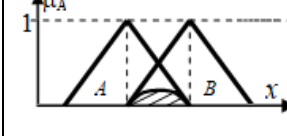
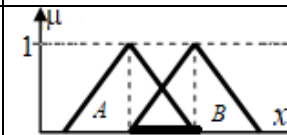
- 6) свойство ограниченности:

$$T(0,0) = 0, T(\mu_A(x), 1) = \mu_A(x), T(\mu_B(x), 1) = \mu_B(x).$$

В табл. 1.2 перечислены наиболее распространенные T -нормы. Как видно из таблицы, использование оператора MIN приводит к наиболее высоким значениям ФП, по этой причине другие операторы T -норм иногда называют *sub-MIN*-операторами – они являются более строгими, требующими более высокой степени выполнения условий A и B , входящих в нечеткое произведение. Поэтому об операторе MIN говорят, как о наиболее оптимистичном среди T -норм.

Таблица 1.2

Некоторые операторы T-нормы

Название оператора	Формула $\mu_{A \cap B}(x)$	Графическое изображение
Минимум (логическое произведение по Л.Заде)	$T_{min}(\mu_A(x), \mu_B(x)) = \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$	
Произведение (алгебраическое произведение)	$T_{prod}(\mu_A(x), \mu_B(x)) = \mu_A(x) \cdot \mu_B(x)$	
Ограниченная разность (граничное произведение)	$T_{max}(\mu_A(x), \mu_B(x)) = \mu_A(x) \otimes \mu_B(x) = \max(0, \mu_A(x) + \mu_B(x) - 1)$	
Усиленное (драстическое) произведение	$T_{\Delta} = \begin{cases} \mu_A(x), & \text{если } \mu_B(x) = 1 \\ \mu_B(x), & \text{если } \mu_A(x) = 1 \\ 0 & \text{в других случаях} \end{cases}$	

2.3.3. *Нечеткая операция ИЛИ– объединение* нечетких множеств.

Результат объединения *четких* множеств является однозначным, поскольку объединение выполняется всегда одним и тем же способом. В случае *нечетких* множеств возможен ряд способов выполнения объединения, и тем самым результат его неоднозначен.

Первыми операторами, предложенными в качестве основы для выполнения операции *объединения* нечетких множеств, являлись оператор *MAX* и алгебраическая сумма.

Применяя для объединения оператор логической суммы по Л. Заде, т.е. с помощью оператора *MAX*, получим:

$$C = A \cup B = \left\{ \frac{1}{x_1}; \frac{0,8}{x_2}; \frac{0,6}{x_3}; \frac{0,6}{x_4}; \frac{0,8}{x_5}; \frac{1}{x_6} \right\}.$$

Если для нахождения объединения *AUB* использовать оператор алгебраической суммы *SUM* по формуле:

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x), \forall x \in X,$$

то результатом будет множество:

$$C = A \cup B = \left\{ \frac{1}{x_1}; \frac{0,84}{x_2}; \frac{0,76}{x_3}; \frac{0,76}{x_4}; \frac{0,84}{x_5}; \frac{1}{x_6} \right\}.$$

В качестве операторов объединения $A \cup B$ множеств выступают T -конормы, также называемые S -нормами.

Оператор S -нормы представляет собой функцию S , реализующую функцию ИЛИ объединения двух нечетких множеств A и B , удовлетворяющую перечисленным ниже свойствам:

1) пространство отображения: $S: [0,1] \times [0,1] \rightarrow [0,1]$;

2) свойство обнуления: $S(0,0) = 0$;

3) свойство коммутативности:

$$S(\mu_A(x), \mu_B(x)) = S(\mu_B(x), \mu_A(x));$$

4) свойство ассоциативности:

$$S(\mu_A(x), S(\mu_B(x), \mu_C(x))) = S[S(\mu_A(x), \mu_B(x)), \mu_C(x)];$$

5) свойство монотонности:

$$\mu_A(x) \leq \mu_C(x), \mu_B(x) \leq \mu_D(x) \Rightarrow S(\mu_A(x), \mu_B(x)) \leq S(\mu_C(x), \mu_D(x)),$$

6) свойство ограниченности:

$$S(0,0) = 0, S(\mu_A(x),0) = S(0,\mu_A(x)) = \mu_A(x).$$

Каждой T -норме соответствует своя S -норма:

$$S(\mu_A, \mu_B) = 1 - T(1 - \mu_A, 1 - \mu_B),$$

поэтому S -норму часто называют T -конормой или дополняющей нормой. В табл. 1.3 перечислены наиболее распространенные S -нормы.

Таблица 1.3

Некоторые операторы S -нормы

Название оператора	Формула $\mu_{A \cup B}(x)$	Графическое изображение
Максимум (логическая сумма по Л.Заде)	$S_{max}(\mu_A(x), \mu_B(x)) = \mu_{A \vee B}(x) = \text{MAX}(\mu_A(x), \mu_B(x))$	
Алгебраическая сумма	$S_{sum}(\mu_A(x), \mu_B(x)) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$	
Ограниченная (граничная) сумма	$S_{min}(\mu_A(x), \mu_B(x)) = \mu_A(x) \oplus \mu_B(x) = \text{MIN}(\mu_A(x) + \mu_B(x); 1)$	
Усиленная (драстическая) сумма	$S_{\Delta} = \begin{cases} \mu_A(x), & \text{если } \mu_B(x) = 0 \\ \mu_B(x), & \text{если } \mu_A(x) = 0 \\ 1 & \text{в других случаях} \end{cases}$	

2.4. Нечеткие отношения

Одним из важнейших понятий математической логики является классическое (двухместное, бинарное) отношение – свойство *пар* объектов, которое описывает определенную взаимосвязь между объектами.

Пример. Пусть имеются одномерные множества X_1 и X_2 , где $X_1 = \{c_1, c_2, \dots, c_5\}$ – множество граждан, $X_2 = \{b_1, b_2, \dots, b_5\}$ – множество банков. Примером классического отношения на множестве определения $X = X_1 \times X_2$ является отношение R «иметь счет в банке», которое состоит из пар (c_i, b_j) и, таким образом, является бинарным отношением, сопоставляющим граждан c_i с банками b_j , в которых у них открыты счета. Данное отношение может быть описано с помощью ФП $\mu_R(c_i, b_j)$, представленной в трехмерном пространстве, и с помощью матрицы отношений R .

На рисунке 1.11 представлены трехмерная ФП и матрица отношения для $R = \{(c_1, b_2), (c_3, b_4), (c_4, b_1), (c_5, b_3)\}$. Отметим, что поскольку у гражданина c_2 нет счета ни в каком банке, то второй столбец матрицы R целиком состоит из нулей.

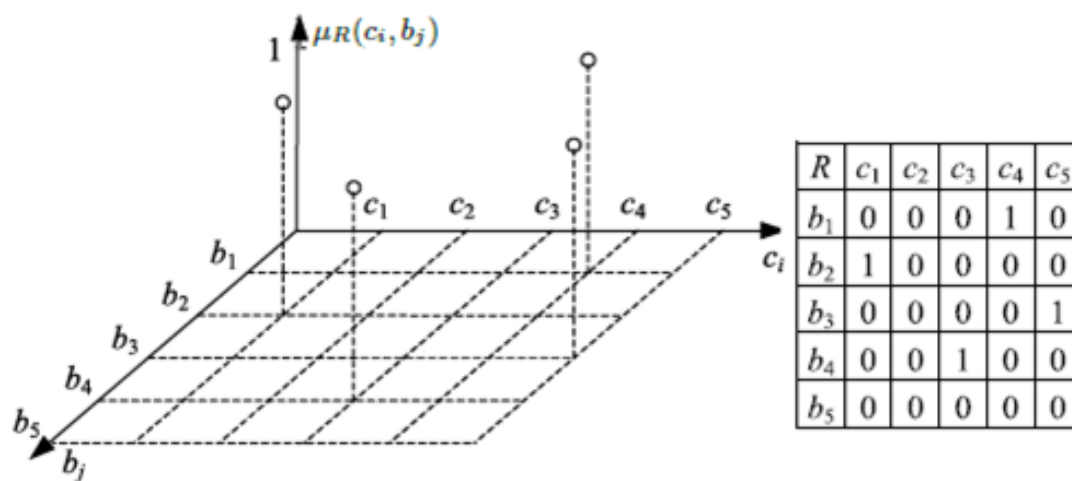


Рис. 1.11. Отношение R в виде трехмерной ФП $\mu_R(c_i, b_j)$
и в виде матрицы отношений R

Так же, как и в случае обычных множеств, нечеткое отношение между элементами из двух универсальных множеств принято называть *бинарным*, между элементами трех множеств – *тернарным*, а в общем случае *n-арным* отношением. При этом на форму и вид ФП нечеткого отношения предварительно не накладывается никаких ограничений.

Нечетким n -арным отношением R , заданным на области определения $X = X_1 \times \dots \times X_n$, называется упорядоченное множество кортежей из n элементов, имеющее вид:

$$R = \{((x_1, \dots, x_n), \mu_R(x_1, \dots, x_n)) | (x_1, \dots, x_n) \in X\},$$

где $\mu_R(x_1, \dots, x_n): X_1 \times \dots \times X_n \rightarrow [0,1]$ представляет собой функцию принадлежности отношения R , которая отображает область определения X на непрерывный интервал $[0, 1]$.

В общем случае ФП μ_R отношения R представляет собой гиперповерхность в $(n+1)$ -мерном пространстве.

Пример. Предположим, необходимо построить нечеткое отношение, описывающее в упрощенном виде диагностику неисправности в автомобиле.

Введем входное множество причин неисправностей (предпосылок) $X = \{x_1, x_2, x_3, x_4\}$, где x_1 – «неисправность аккумулятора», x_2 – «неисправность карбюратора», x_3 – «низкое качество бензина», x_4 – «неисправность системы зажигания».

Множество $Y = \{y_1, y_2, y_3\}$ представляет собой проявления следующих неисправностей: y_1 – «двигатель не запускается», y_2 – «двигатель работает неустойчиво», y_3 – «двигатель не развивает полной мощности».

Между каждым элементом множества предпосылок и каждым элементом множества следствий существует некоторая взаимосвязь. Исходя из экспертных знаний конкретного механика, марки автомобиля и т.д. взаимосвязь причин неисправностей и проявлений неисправностей в автомобиле описывается в виде нечеткого отношения $\mu_R(x_i, y_j)$ (табл. 1.4).

Таблица 1.4

Нечеткое отношение диагностики неисправности в автомобиле

$\mu_R(x_i, y_j)$	Двигатель		
	y_1 – не запускается	y_2 – работает неустойчиво	y_3 – не развивает полной мощности
x_1 – неисправность аккумулятора	1	0,1	0,3
x_2 – неисправность карбюратора	0,8	0,9	1
x_3 – низкое качество бензина	0,7	0,8	0,5
x_4 – неисправность системы зажигания	1	0,5	0,2

В нечетком моделировании и управлении часто приходится иметь дело с нечеткими отношениями, полученными путем *агрегирования нечетких множеств*, заданных на различных одномерных областях.

Например, в правиле *ЕСЛИ–ТО* компоненты условия $\{x_1, x_2, \dots, x_k\}$ агрегируются с помощью логических связок *И* / *ИЛИ*, образуя бинарное нечеткое отношение R с ФП $\mu_R(x_l, x_m)$. Агрегирование нечетких множеств x_l и x_m можно выполнять, используя операторы T -норм в случае связки *И* $\mu_R(x_l, x_m) = T(\mu(x_l), \mu(x_m))$ и операторы S -норм в случае связки *ИЛИ*: $\mu_R(x_l, x_m) = S(\mu(x_l), \mu(x_m))$.

Нечеткая импликация представляет собой правило R , простейшая форма которого выражается в виде: *ЕСЛИ* $(x = A)$ *ТО* $(y = B)$, где $(x = A)$ – условие (антецедент), а $(y = B)$ – заключение (консеквент). Здесь A и B – нечеткие множества, заданные своими ФП $\mu_A(x)$, $\mu_B(y)$, и областями определения X , Y соответственно. Обозначение нечеткой импликации имеет вид: $A \rightarrow B$.

ФП импликации $\mu_{A \rightarrow B}(x, y)$ лежит в основе так называемых *нечетких рассуждений*, обеспечивающих возможность вычисления выходного значения нечеткой модели (регулятора) для заданных входных значений. Чтобы определить данную функцию на основе ФП условия $\mu_A(x)$ и заключения $\mu_B(y)$, следует использовать подходящий оператор импликации.

Оператор импликации *Мамдани* основан на предположении, что степень истинности заключения $\mu_B(y)$ не может быть выше, чем степень выполнения условия $\mu_A(x)$:

$$\mu_{A \rightarrow B}(x, y) = \text{MIN}(\mu_A(x), \mu_B(y)).$$

В нечетком управлении также используется оператор алгебраического произведения *PROD*:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cdot \mu_B(y).$$

Если известны нечеткие отношения между множествами $R_1(X, Y): X \rightarrow Y$ и $R_2(Y, Z): Y \rightarrow Z$, то нечёткое отношение $R_1 \cdot R_2: X \rightarrow Z$, где $\mu_{R_1 \cdot R_2}(X, Z) = \cup[\mu_{R_1}(X, Y) \wedge \mu_{R_2}(Y, Z)]$, называется (max-min)-композицией (сверткой) множеств X и Z .

Операция (max-min) ассоциативна: $R_1 \cdot (R_2 \cdot R_3) = (R_1 \cdot R_2) \cdot R_3$.

Операция (max-min) не дистрибутивна относительно пересечения: $R_1 \cdot (R_2 \wedge R_3) \neq (R_1 \cdot R_2) \wedge (R_1 \cdot R_3)$.

Нечеткое отношение между X и Y , обозначаемое через $R_1 \cdot R_2$, может быть также задано (max-prod)-композицией отношений $R_1 \cdot R_2$ $\mu_{R_1 \cdot R_2}(X, Z) = \cup [\mu_{R_1}(X, Y) \cdot \mu_{R_2}(Y, Z)]$.

Операция свертки является композиционным правилом нечеткого логического вывода.

Композиционными правилами вида (max-T) для нечеткой импликации ЕСЛИ A ТО B являются:

- (max-min)-композиция $\mu_B(y) = \max\{\min(\mu_A(x), \mu_R(x, y))\}$.
- (max-prod)-композиция $\mu_B(y) = \max\{\mu_A(x) \cdot \mu_R(x, y)\}$.

3. Методика выполнения работы

3.1. Построение нечетких множеств

Пример. В качестве примера рассматривается универсальное множество $X = \{1, 2, \dots, 10\}$ и два его подмножества: A – множество чисел, меньших 7, и B – множество чисел, немного меньших 7.

Запишем подмножества в форме объединения одноэлементных подмножеств X :

$$A = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{0}{7} + \frac{0}{8} + \frac{0}{9} + \frac{0}{10}, \text{ или}$$

$$A = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6}.$$

$$B = \frac{0}{1} + \frac{0}{2} + \frac{0,5}{3} + \frac{0,6}{4} + \frac{0,8}{5} + \frac{0,9}{6} + \frac{0}{7} + \frac{0}{8} + \frac{0}{9} + \frac{0}{10}, \text{ или}$$

$$B = \frac{0,5}{3} + \frac{0,6}{4} + \frac{0,8}{5} + \frac{0,9}{6}.$$

Несущее множество (носитель) для нечёткого множества B состоит из чисел $\{3, 4, 5, 6\}$, в которых значения ФП $\mu_B(x)$ больше нуля.

Точкой перехода для множества B является $x = 3$, при котором значение функции принадлежности $\mu_B(x) = 0,5$.

Поскольку ни одно из значений $\mu_B(x) = 3$ не достигло своего возможного максимального значения, равного 1, то множество B является субнормальным. Для преобразования его в нормальное необходимо разделить все значения ФП на ее наибольшее значение. Нечеткое множество B после нормирования примет следующий вид:

$$B = \frac{5/9}{3} + \frac{2/3}{4} + \frac{8/9}{5} + \frac{1}{6}.$$

3.2. Выполнение операций с нечеткими множествами.

Пример. Имеется универсальное множество $X = \{1, 2, \dots, 10\}$ и два его подмножества:

$$A = \frac{0,8}{3} + \frac{1}{5} + \frac{0,6}{6} \text{ и } B = \frac{0,7}{3} + \frac{0,5}{6}.$$

Дополнения нечетких множеств A и B :

$$\bar{A} = \frac{1-0}{1} + \frac{1-0}{2} + \frac{1-0,8}{3} + \frac{1-0}{4} + \frac{1-1}{5} + \frac{1-0,6}{6} + \frac{1-0}{7} + \frac{1-0}{8} + \frac{1-0}{9} + \frac{1-0}{10} = \frac{1}{1} + \frac{1}{2} + \frac{0,2}{3} + \frac{1}{4} + \frac{0,4}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10}.$$

$$\bar{B} = \frac{1}{1} + \frac{1}{2} + \frac{0,3}{3} + \frac{1}{5} + \frac{0,5}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10}.$$

Выполним операцию \min -пересечения нечетких множеств A и B :

$$A \cap B = \frac{0,7}{3} + \frac{0,5}{6}.$$

Выполним операцию объединения нечетких множеств A и B :

$$A \cup B = \frac{0,8}{3} + \frac{1}{5} + \frac{0,6}{6}.$$

Отметим, что пересечение нечеткого множества со своим дополнением может не быть пустым:

$$A \cap \bar{A} = \frac{0,2}{3} + \frac{0,4}{6} \neq \emptyset.$$

Объединение нечеткого множества со своим дополнением может не составлять универсальное множество X .

$$B \cap \bar{B} = \frac{1}{1} + \frac{1}{2} + \frac{0,7}{3} + \frac{1}{4} + \frac{1}{5} + \frac{0,5}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} \neq X.$$

Пример. На отрезке $[0, 6]$ заданы множества A – «величина x велика» и B – «величина x мала» с функциями принадлежности:

$$\mu_A(x) = \begin{cases} 0, & \text{если } 0 \leq x < 1; \\ \frac{x-1}{3}, & \text{если } 1 \leq x \leq 4, \\ 1, & \text{если } 4 < x \leq 6 \end{cases} \quad \mu_B(x) = \begin{cases} 1, & \text{если } 0 \leq x < 1; \\ \frac{3-x}{2}, & \text{если } 1 \leq x \leq 3, \\ 0, & \text{если } 3 < x \leq 6 \end{cases}$$

Построим дополнения нечетких множеств A и B :

$$\mu_{\bar{A}}(x) = \begin{cases} 1, & \text{если } 0 \leq x < 1; \\ \frac{4-x}{3}, & \text{если } 1 \leq x \leq 4, \\ 0, & \text{если } 4 < x \leq 6 \end{cases} \quad \mu_{\bar{B}}(x) = \begin{cases} 0, & \text{если } 0 \leq x < 1; \\ \frac{x-1}{2}, & \text{если } 1 \leq x \leq 3, \\ 1, & \text{если } 3 < x \leq 6 \end{cases}$$

Объединение нечетких множеств A и B имеет вид:

$$\mu_{A \cup B}(x) = \begin{cases} \max(1,0), & \text{если } 0 \leq x < 1 \\ \max\left(\frac{x-1}{3}, \frac{3-x}{2}\right), & \text{если } 1 \leq x \leq 2,2 \\ \max\left(\frac{x-1}{3}, \frac{3-x}{2}\right), & \text{если } 2,2 \leq x \leq 4 \\ \max(1,0), & \text{если } 4 < x \leq 6 \end{cases} = \begin{cases} 1, & \text{если } 0 \leq x < 1 \\ \frac{3-x}{2} & \text{если } 1 \leq x \leq 2,2 \\ \frac{x-1}{3} & \text{если } 2,2 \leq x \leq 4 \\ 1, & \text{если } 4 < x \leq 6. \end{cases}$$

3.3. Нечеткие отношения и их свойства

Пример. В результате каких-то внешних воздействий (например, землетрясения) строительные конструкции получают различные повреждения. Экспертам необходимо оценить состояние здания.

Характеристика повреждений отражается в множестве признаков $X = \{x_1, x_2, \dots, x_k\}$, например, x_1 – «много трещин», x_2 – «большие трещины», x_3 – «сильная деформация».

Множество видов потенциально возможных деформаций выражается в признаках $Y = \{y_1, y_2, \dots, y_l\}$: например, y_1 – «усталостное или разрывное повреждение», y_2 – «пластическая деформация», y_3 – «нестабильность», y_4 – «прогрессирующее разрушение».

Пусть $Z = \{z_1, z_2, \dots, z_m\}$ – результаты оценки степени повреждения сооружения, которые изменяются от 0 до 1.

На основе профессиональных знаний эксперты могут указать, во-первых, соответствие между характеристиками повреждений X и видами деформаций Y : $R_1(X, Y)$ (табл. 1.5); во-вторых, соответствие между видами деформаций Y и степенью повреждения Z : $R_2(Y, Z)$ (табл. 1.6).

Таблица 1.5

Нечеткое отношение $R_1(X, Y)$ «повреждения – деформации»

Признаки повреждения X	Виды деформации Y			
	y_1 – усталость и разрыв	y_2 – пластическая деформация	y_3 – нестабильность	y_4 – прогрессирующее разрушение
x_1 – «много трещин»	0,9	0,2	0,6	0,4
x_2 – «большие трещины»	0,8	0,3	0,7	0,8
x_3 – «сильная деформация»	0,3	0,8	0,9	0,7

Таблица 1.6

Нечеткое отношение $R_2(Y, Z)$ «деформации – степень повреждения»

Виды деформации Y	Степень повреждения Z
y_1 – усталость и разрыв	0,4
y_2 – пластическая деформация	0,3
y_3 – нестабильность	0,8
y_4 – прогрессирующее разрушение	1

Для установления отношений между характеристиками повреждений X и степенью повреждения Z строится «max-min»-композиция нечетких отношений, представленная в табл. 1.7.

$$\begin{aligned} \mu_{R_1 \cdot R_2}(x_1, z_1) &= [\mu_{R_1}(x_1, y_1) \wedge \mu_{R_2}(y_1, z_1)] \vee [\mu_{R_1}(x_1, y_2) \wedge \mu_{R_2}(y_2, z_1)] \vee \\ &\vee [\mu_{R_1}(x_1, y_3) \wedge \mu_{R_2}(y_3, z_1)] \vee [\mu_{R_1}(x_1, y_4) \wedge \mu_{R_2}(y_4, z_1)] = \\ &= [0,9 \wedge 0,4] \vee [0,2 \wedge 0,3] \vee [0,6 \wedge 0,8] \vee [0,4 \wedge 1] = 0,6. \end{aligned}$$

Таблица 1.7

Нечеткое отношение $R_1 \cdot R_2$ «повреждения – степень повреждения»

$\mu_{R_1 \cdot R_2}(X, Z)$	Степень повреждения Z
«max-min»-композиция	
x_1 – «много трещин»	0,6
x_2 – «большие трещины»	0,8
x_3 – «сильная деформация»	0,8
«max-prod»-композиция	
x_1 – «много трещин»	0,4
x_2 – «большие трещины»	0,8
x_3 – «сильная деформация»	0,7

По результатам построения нечеткой композиции определяются наиболее возможные степени повреждения конструкции здания в зависимости от характеристик повреждений: так, например, наибольшие степени повреждения здания получают при наличии больших трещин и сильной деформации.

Порядок выполнения работы

1. По варианту задания 1 представить три неточные формулировки некоторого понятия, представленного на естественном языке, в виде нечетких подмножеств A_1, A_2, A_3 . В качестве области рассуждений определить диапазон $[0; x_{\max}]$.

2. Изобразить на одном графике функции принадлежности нечетких подмножеств A_1, A_2, A_3 с учетом того, что ФП подмножества A_1 относится к классу Z , A_2 – трапецевидная ФП, A_3 – ФП относится к классу S . Записать математические выражения указанных ФП в виде кусочно-линейных функций. Записать нечеткие подмножества A_1, A_2, A_3 в дискретном виде в форме объединения пар вида $(\mu_A(x_i)/x_i)$.

3. Определить в фиксированной точке x_0 значение функции принадлежности $\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{A_3}(x_0)$.

4. Определить свойства нечеткого подмножества A_2 (носитель, точка перехода, ядро). Используя операторы концентрации и растяжения, провести лингвистическую модификацию нечеткого подмножества A_2 .

5. Выполнить операции дополнения нечеткого подмножества A_1 , операции пересечения и объединения нечетких подмножеств A_1 и A_2 и заполнить табл. 1.8. Провести сравнительный анализ результатов.

Таблица 1.8

Операции с нечеткими множествами

Вид операции	Выполнение операции	
	Математическая запись операции	Графический вид
$\mu_{\overline{A_1}}(x)$		
$T_{\min}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$T_{prod}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$T_{\max}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$T_{\Delta}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$S_{\max}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$S_{sum}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$S_{\min}(\mu_{A_1}(x), \mu_{A_2}(x))$		
$S_{\Delta}(\mu_{A_1}(x), \mu_{A_2}(x))$		

6. По варианту задания 2 выполнить композицию двух нечетких отношений аналитическим способом свертки методами «max-min» и «max-prod». Провести сравнительный анализ результатов композиции разными методами.

7. Сделать выводы по результатам практической работы.

Требования к содержанию и оформлению отчета

Отчет по практической работе должен содержать:

1. Название и цель работы.

2. Исходные данные варианта задания 1, графики функций принадлежности нечетких подмножеств A_1 , A_2 , A_3 , математические выражения указанных ФП в виде кусочно-линейных функций, в дискретном виде в форме объединения пар вида $(\mu_A(x_i)/x_i)$.

Значение функции принадлежности $\mu_{A_1}(x_0)$, $\mu_{A_2}(x_0)$, $\mu_{A_3}(x_0)$ в определенной точке. Свойства нечеткого подмножества A_2 (носитель, точка перехода, ядро). Результаты лингвистической модификации («очень», «довольно») нечетких подмножеств по варианту задания.

Результаты операции дополнения нечеткого подмножества A_1 , операции пересечения и объединения нечетких подмножеств A_1 и A_2 , заполненная табл. 1.8.

3. Исходные данные варианта задания 2. Этапы проведения композиции нечетких отношений методами «max-min» и «max-prod», математическое обоснование. Результаты сравнительного анализа результатов композиции.

4. Выводы по результатам практической работы.

Варианты заданий

Задание 1. Варианты заданий представлены в табл. 1.9.

Таблица 1.9

№	Понятие	Характеристики		
1	2	3		
1, 11	Скорость автомобиля	Довольно низкая	Средняя	Высокая
2, 12	Температура для купания	Некомфортная	Нейтральная	Очень комфортная
3, 13	Скорость ветра	Слабая	Средняя	Довольно сильный
4, 14	Температура у больного	Пониженная	Нормальная	Очень высокая
5, 15	Температура воды в реке	Холодная	Средняя	Довольно теплая
6, 16	Рост юноши	Невысокий	Средний	Очень высокий
7, 17	Рост девушки	Невысокий	Средний	Очень высокий

1	2	3		
8, 18	Температура на улице летом	Прохладная	Нормальная	Довольно жарко
9, 19	Температура на улице зимой	Очень морозно	Обычная зимняя погода	Тепло
10, 20	Возраст человека	Очень молодой	Средний возраст	Пожилой

Задание 2.

Рассматривается ситуация, связанная с консалтингом в области выбора профессии для последующего обучения и получения соответствующей специальности.

Пусть $X = \{x_1, x_2, x_3, x_4, x_5\}$ – множество кандидатов на обучение: x_1 – Петров, x_2 – Иванов, x_3 – Сидоров, x_4 – Васильева, x_5 – Григорьева.

Множество $Y = \{y_1, y_2, \dots, y_{10}\}$ представляет собой различные психофизиологические характеристики человека: y_1 – быстрота и гибкость мышления, y_2 – умение быстро принимать решения, y_3 – устойчивость и концентрация внимания, y_4 – зрительная память, y_5 – быстрота реакции, y_6 – двигательная память, y_7 – физическая выносливость, y_8 – координация движений, y_9 – эмоционально-волевая устойчивость, y_{10} – ответственность.

Множество $Z = \{z_1, z_2, z_3, z_4, z_5\}$ составляет множество специальностей, по которым проводится набор на обучение: z_1 – менеджер, z_2 – программист, z_3 – водитель, z_4 – секретарь-референт, z_5 – переводчик.

Строится нечеткая модель, основанная на двух бинарных нечетких отношениях R_1 и R_2 . Первое из этих нечетких отношений R_1 строится на двух базисных множествах Y и X , а второе R_2 – на двух базисных множествах Y и Z .

Нечеткое отношение R_1 содержательно описывает психофизиологическое профилирование специальностей, а R_2 – психофизиологическое профилирование кандидатов на обучение.

Требуется определить, какой из кандидатов наилучшим образом соответствует определенной профессии. Для этого необходимо построить композицию нечетких отношений аналитическим способом свертки методами «max-min» и «max-prod».

Конкретные значения ФП рассматриваемых нечетких отношений R_1 и R_2 представлены в табл. 1.10 и 1.11.

Таблица 1.10

Нечеткое отношение R_1

Психофизиологическая характеристика Y	Специальность Z				
	Менеджер	Программист	Водитель	Секретарь	Переводчик
Быстрота и гибкость мышления	0,9	0,8	0,3	0,5	0,7
Умение быстро принимать решения	0,9	0,5	0,9	0,4	0,8
Устойчивость и концентрация внимания	0,8	0,9	0,6	0,5	0,8
Зрительная память	0,4	0,3	0,5	0,5	0,2
Быстрота реакции	0,5	0,1	0,9	0,2	0,6
Двигательная память	0,3	0,2	0,8	0,2	0,2
Физическая выносливость	0,6	0,2	0,9	0,3	0,2
Координация движений	0,2	0,2	0,8	0,3	0,3
Эмоционально-волевая устойчивость	0,9	0,5	0,6	0,9	0,3
Ответственность	0,8	0,5	0,3	0,8	0,2

Таблица 1.11

Нечеткое отношение R_2

Психофизиологическая характеристика Y	Кандидаты на обучение X				
	Петров	Иванов	Сидоров	Васильева	Григорьева
Быстрота и гибкость мышления	0,9	0,8	0,7	0,9	1
Умение быстро принимать решения	0,6	0,4	0,8	0,5	0,6
Устойчивость и концентрация внимания	0,5	0,2	0,3	0,8	0,7
Зрительная память	0,5	0,9	0,5	0,8	0,4
Быстрота реакции	1	0,6	0,5	0,7	0,4
Двигательная память	0,4	0,5	1	0,7	0,8
Физическая выносливость	0,5	0,8	0,9	0,5	0,4
Координация движений	0,5	0,6	0,7	0,6	0,5
Эмоционально-волевая устойчивость	0,8	1	0,2	0,5	0,6
Ответственность	0,3	0,5	0,9	0,6	0,8

Контрольные вопросы и задания

1. Что такое нечеткое множество, функция принадлежности, лингвистическая переменная, терм-множество?
2. Назовите основные свойства нечетких множеств.
3. Какие операции пересечения и объединения нечетких множеств Вы знаете? Какие операторы обобщают операции пересечения и объединения, для чего они предназначены?
4. Что представляют собой нечеткие отношения?
5. Как строятся композиционные правила нечеткого логического вывода?

Список литературы

1. Пегат А. Нечеткое моделирование и управление / А. Пегат; под редакцией Ю. В. Тюменцева; перевод с английского А. Г. Подвесовского, Ю. В. Тюменцева. 3-е изд. М.: Лаборатория знаний, 2015. – 801 с.
2. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2003. 736 с.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И.Д. Рудинского. М.: Горячая линия – Телеком, 2004. 452 с.
4. Ярушкина Н. Г. Основы теории нечетких и гибридных систем: учеб. пособие. – М.: Финансы и статистика, 2004. 320с.
5. Флегонтов А. В. Моделирование задач принятия решений при нечетких исходных данных: монография / А. В. Флегонтов, В. Б. Вилков, А. К. Черных. СПб.: Лань, 2020. 332 с.
6. Чернов В. Г. Нечеткие множества. Основы теории и применения: учеб. пособие / В. Г. Чернов; Владим. гос. ун-т им. А. Г. и Н.Г. Столетовых. Владимир: Изд-во ВлГУ, 2018. 156 с.

Практическая работа № 2

АЛГОРИТМЫ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА

1. Цель и задачи работы

Целью работы является закрепление знаний о структуре системы нечеткого управления, приобретение навыков применения алгоритмов нечеткого логического вывода.

Задачами работы являются формирование умений выполнения процедур фаззификации, нечеткого логического вывода, дефаззификации, применения алгоритма Мамдани, Ларсена и Такаги-Сугено на базе нечетких правил.

2. Теоретические сведения

Понятие нечеткого вывода занимает центральное место в теории нечеткого управления. На рис. 2.1 представлена типовая структура модуля нечеткого управления, который состоит из следующих компонентов: база правил, блоки фаззификации, выработки решений (нечеткого логического вывода), дефаззификации.

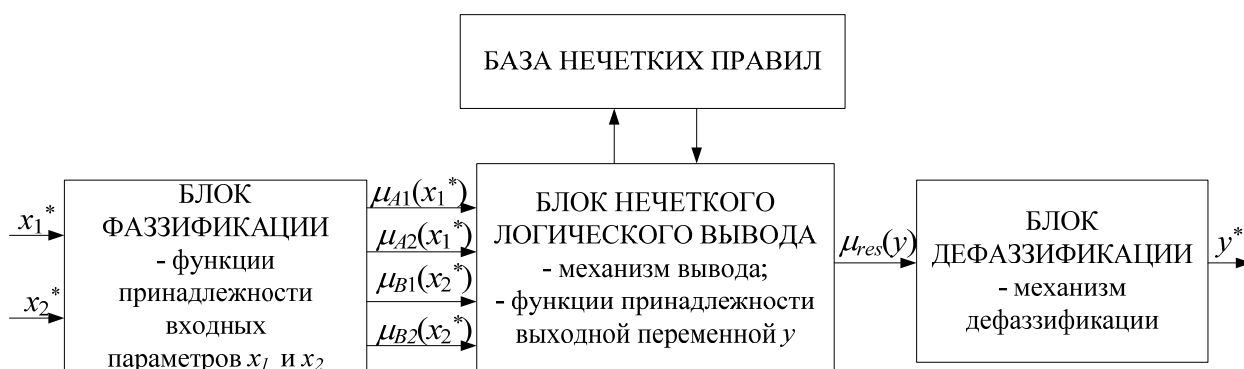


Рис. 2.1. Блочная структура модуля нечеткого управления

Блок *фаззификации* вычисляет степени принадлежности двух четких числовых значений x_1^* и x_2^* входным нечетким подмножествам A_i и B_j . Для выполнения указанной операции блок фаззификации должен иметь доступ к заранее определенным функциям принадлежности (ФП) $\mu_{A_i}(x_1)$, $\mu_{B_j}(x_2)$.

Вычисленные степени принадлежности $\mu_{A_i}(x_1^*)$, $\mu_{B_j}(x_2^*)$ показывают, в какой степени входные значения x_1^* и x_2^* принадлежат конкретным нечетким множествам.

Блок *нечёткого логического вывода* (НЛВ) – это процесс формирования нечетких заключений о требуемом управляющем воздействии на объект. Блок НЛВ на основе нечетких условий в виде степеней принадлежности $\mu_{A_i}(x_1^*)$, $\mu_{B_j}(x_2^*)$ вычисляет результирующую ФП выходного значения модели $\mu_{res}(y)$. Данная функция обычно имеет сложную форму и определяется посредством *логического вывода*.

База правил предназначена для формального представления эмпирических знаний экспертов в той или иной предметной области в форме нечетких продукционных правил.

База правил может, например, иметь следующий вид:

Правило 1: ЕСЛИ ($x_1 = A_1$) И ($x_2 = B_1$) ТО ($y = C_1$);

Правило 2: ЕСЛИ ($x_1 = A_1$) И ($x_2 = B_2$) ТО ($y = C_2$);

Правило 3: ЕСЛИ ($x_1 = A_2$) И ($x_2 = B_1$) ТО ($y = C_2$);

Правило 4: ЕСЛИ ($x_1 = A_2$) И ($x_2 = B_2$) ТО ($y = C_3$).

Таким образом, база нечетких продукционных правил системы НЛВ – это согласованное множество *нечетких продукционных правил* вида «ЕСЛИ – ТО», отражающая знания экспертов о методах управления объектом и характере его функционирования в различных условиях.

Блок *дефаззификации* на основе результирующей ФП $\mu_{res}(y)$ вычисляет четкое числовое значение y^* выходного параметра на основе механизма дефаззификации.

2.1. Фаззификация

В блоке фаззификации происходит введение нечёткости путем выполнения процедуры вычисления значений ФП нечетких множеств на основе обычных (не нечетких) исходных данных. Целью фаззификации является установление соответствия между конкретным (численным) значением отдельной входной переменной системы НЛВ и значением ФП соответствующего ей терма входной лингвистической переменной (ЛП). После завершения этого этапа для всех входных переменных должны быть определены конкретные

значения ФП по каждому из лингвистических термов, которые используются в подусловиях базы правил системы НЛВ.

На рис. 2.2 показан пример процедуры фаззификации. Предположим, имеются два входных значения $x_1^* = 1,4$ и $x_2^* = 1,6$. Вычисленные степени принадлежности $\mu_{A_i}(x_1^*)$ и $\mu_{B_j}(x_2^*)$ показывают, в какой степени входные значения x_1^* и x_2^* принадлежат конкретным нечетким подмножествам, т. е. насколько эти величины являются малыми S (*small*) (A_1, B_1) или большими L (*large*) (A_2, B_2).

С одной стороны, входное значение $x_1^* = 1,4$ относится к подмножеству A_1 со степенью принадлежности, равной 0,3 (т.е. к малым S величинам): $\mu_{A_1}(x_1^*)|_{x_1^*=1,4} = 0,3$; с другой стороны, это же входное значение относится к подмножеству A_2 со степенью, равной 0,7 (т.е. к большим L величинам): $\mu_{A_2}(x_1^*)|_{x_1^*=1,4} = 0,7$. Таким образом, указанное значение x_1^* в большей степени соответствует типичному большому значению, чем типичному малому.

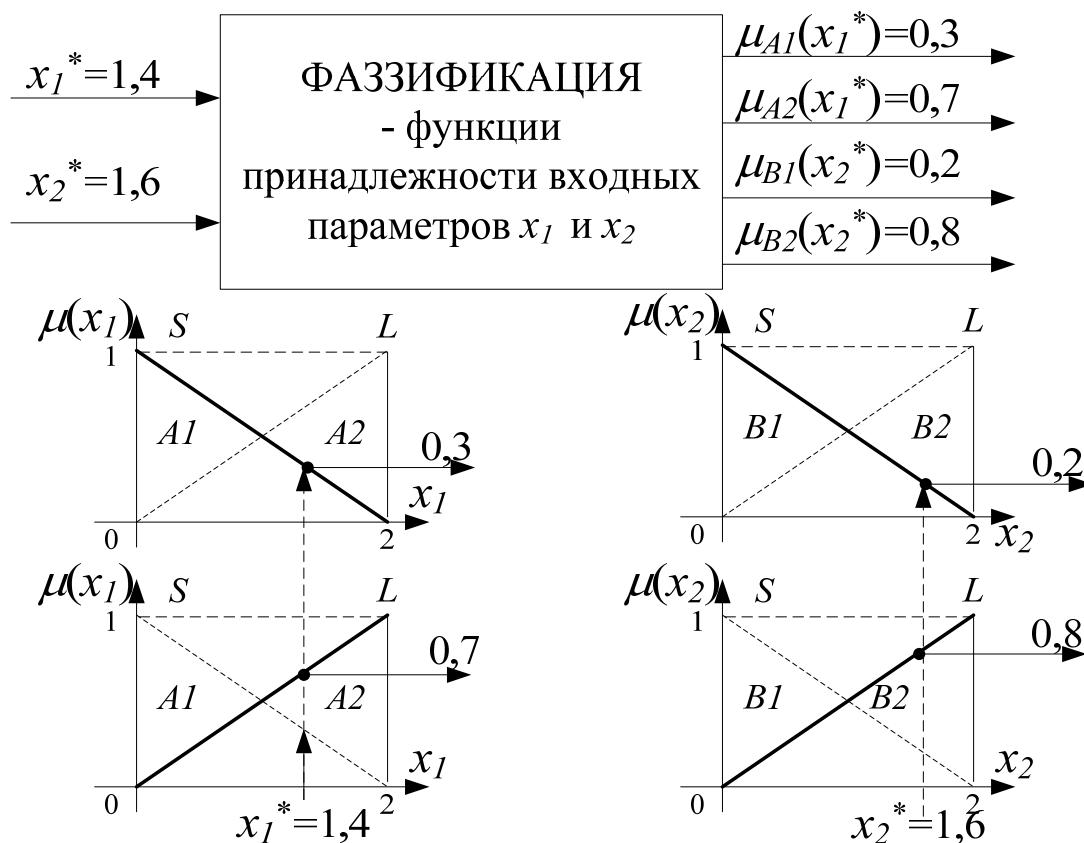


Рис. 2.2. Пример работы блока фаззификации

2.2. Нечеткий логический вывод

Блок НЛВ на основе степеней принадлежности $\mu_{A_i}(x_1)$ и $\mu_{B_j}(x_2)$ входных значений определяет результирующую ФП $\mu_{res}(y)$ выходного значения модели.

Операция НЛВ включает в себя следующие этапы:

- этап агрегирования;
- этап активизации;
- этап аккумуляции.

Отметим, что процедуры НЛВ реализуются для систем двух типов: *SISO* (*single input single output*) с одним входом и одним выходом, в основе которых лежат *простые* правила; *MISO* (*many input single output*) с несколькими входами и одним выходом, т.е. *сложными* (*составными*) правилами.

2.2.1. *Агрегирование* – процедура определения степени выполнения (истинности) условий каждого отдельного правила.

Если условие нечеткого продукционного правила типа *SISO* является *простым* нечетким высказыванием, то степень его истинности в процедуре агрегирования равна значению ФП соответствующего термина ЛП (рис. 2.3).

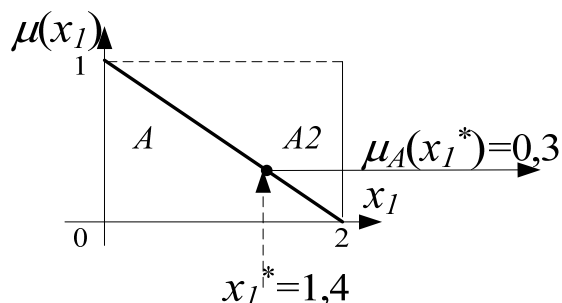


Рис. 2.3. Агрегирование для простого условия

Это означает, что в случае простого условия вида *Если* ($x = A$) для $x = x^*$ степень выполнения условия равна степени принадлежности значения x^* множеству A : $\mu_A(x_1^*) = 0,3$.

Для правил типа *MISO* условие представляет собой *составное* высказывание, состоящее из двух простых подусловий; степень истинности такого высказывания в процедуре агрегирования определяется на основе известных значений истинности составляющих его элементарных высказываний при помощи нечетких логических операций (рис. 2.4).

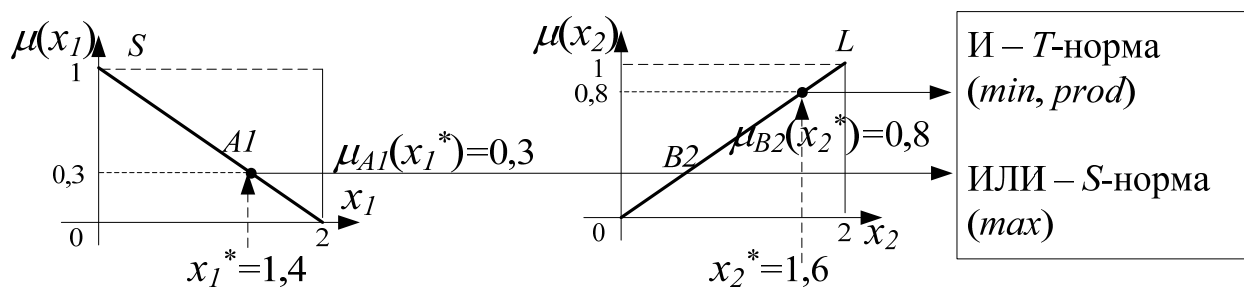


Рис. 2.4. Агрегирование для сложного условия

Если в сложном высказывании *Если* ($x_1 = A_1$) *И* ($x_2 = B_2$) два простых подусловия связаны логическим союзом *И* (конъюнктивное условие), то степень выполнения условия для числовых значений аргументов $x_1 = x_1^*$ и $x_2 = x_2^*$ определяется с помощью любого из операторов *T*-нормы, например, *min* или *prod*:

$$\mu_{A_1 \cap B_2}(x_1^*, x_2^*) = T(\mu_{A_1}(x_1^*), \mu_{B_2}(x_2^*));$$

$$T_{min}(\mu_{A_1}(x_1^*), \mu_{B_2}(x_2^*)) = 0,3;$$

$$T_{prod}(\mu_{A_1}(x_1^*), \mu_{B_2}(x_2^*)) = 0,24.$$

Если в сложном высказывании *Если* ($x_1 = A_1$) *ИЛИ* ($x_2 = B_2$) два простых подусловия связаны логическим союзом *ИЛИ* (альтернативное условие), то степень выполнения условия для числовых значений аргументов $x_1 = x_1^*$ и $x_2 = x_2^*$ определяется с помощью любого из операторов *S*-нормы, например, *max*:

$$\mu_{A_1 \cup B_2}(x_1^*, x_2^*) = S(\mu_{A_1}(x_1^*), \mu_{B_2}(x_2^*)),$$

$$S_{max}(\mu_{A_1}(x_1^*), \mu_{B_2}(x_2^*)) = 0,8.$$

Условия могут иметь и более сложную форму – состоять из множества подусловий, связанных союзами *И*, *ИЛИ*, например, как в выражении: *Если* ($x_1 = A_1$) *И* ($x_2 = B_2$) *ИЛИ* ($x_1 = A_2$) *И* ($x_2 = B_1$). В этом случае вначале следует выполнять все операции пересечения *И*, а затем все операции объединения *ИЛИ*.

После нахождения степеней выполнения условий для отдельных правил становится ясным, какие правила будут участвовать в процессе логического вывода, и появляется возможность определения активизированных ФП заключений отдельных правил при заданных входных значениях x_i^* нечеткой модели.

2.2.2. *Активизация* в системах НЛВ – операция, которая может быть названа *выводом на правилах* и заключается в определении активизированных ФП заключений отдельных правил.

Функции принадлежности $\mu(y)$ каждого из элементарных подзаключений консеквентов всех продукционных правил определяются при помощи одного из методов нечеткой композиции:

– min-активизация $\mu_{C_i}^*(y) = \min\{\alpha_{C_i}, \mu_{C_i}(y)\}$;

– prod-активизация $\mu_{C_i}^*(y) = \alpha_{C_i} \cdot \mu_{C_i}(y)$,

где $\mu_{C_i}(y)$ – ФП терма, который является значением выходной переменной нечеткого правила; α_{C_i} – степени истинности нечетких высказываний, образующих соответствующие следствия ядер нечетких продукционных правил.

При min-активизации ФП вывода в правой части правила «отсекается» по высоте, соответствующей вычисленной степени истинности предпосылки в левой части правила.

На рис. 2.5 представлена min-активизация для *Правила 1 ЕСЛИ* ($x_1 = A_1$) И ($x_2 = B_1$) ТО ($y = C_1$).

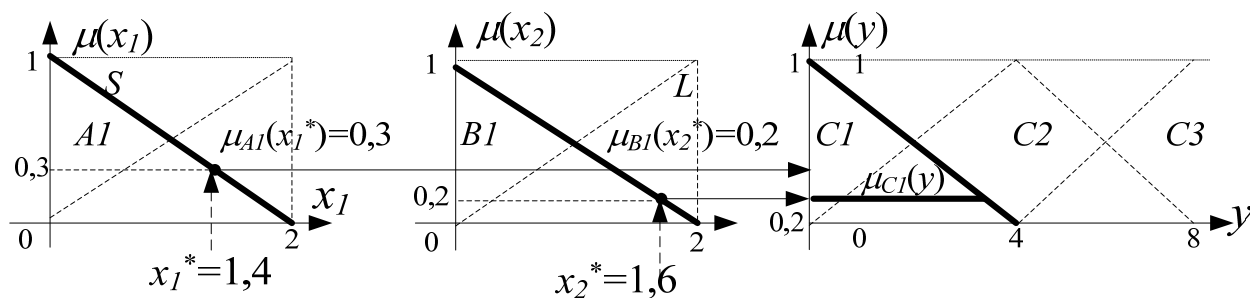


Рис. 2.5. Min-активизация для сложного правила

В *prod-активизации* ФП вывода масштабируется в соответствии с вычисленной степенью истинности предпосылки левой части правила. На рис. 2.6 представлена *prod-активизация* для *Правила 1*.

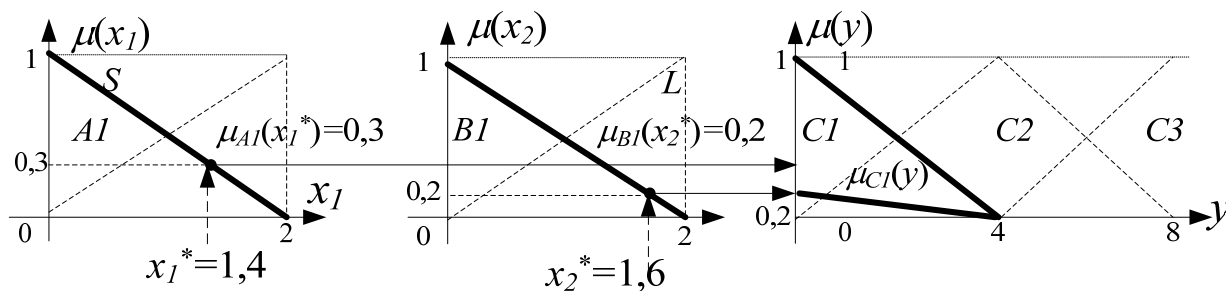


Рис. 2.6. Prod-активизация для сложного правила

2.2.3. *Аккумуляция* в системах НЛВ – это процедура объединения, или аккумуляции, всех степеней истинности заключений правил; формирования результирующего нечеткого множества и нахождения результирующей ФП $\mu_{res}(y)$.

Результирующая ФП $\mu_{res}(y)$ определяется с использованием одной из *S*-норм, например, с помощью операции максимума (*max*) или суммы (*sum*). На рис. 2.7 представлена результирующая ФП для правил нечеткой базы правил, построенная с помощью операции *max*.

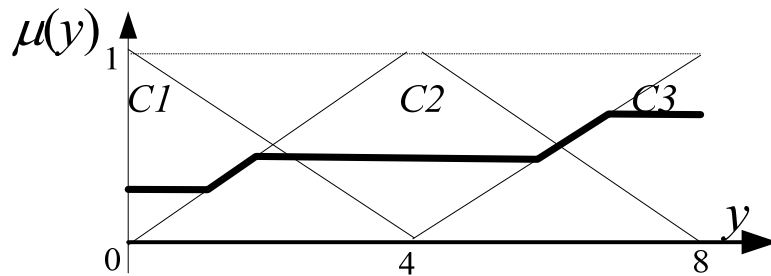


Рис. 2.7. Результирующая ФП, построенная с помощью операции *max*

2.3. Дефаззификация

Под *дефаззификацией* нечеткого множества, являющегося результатом нечеткого логического вывода, понимается операция нахождения четкого (числового) значения y^* , которое бы наиболее «рациональным» образом представляло это множество. Цель дефаззификации заключается в том, чтобы, используя результаты аккумуляции всех выходных ЛП, получить количественное значение каждой из выходных переменных, которое может быть использовано исполнительными механизмами системы управления, внешними по отношению к системе нечеткого вывода.

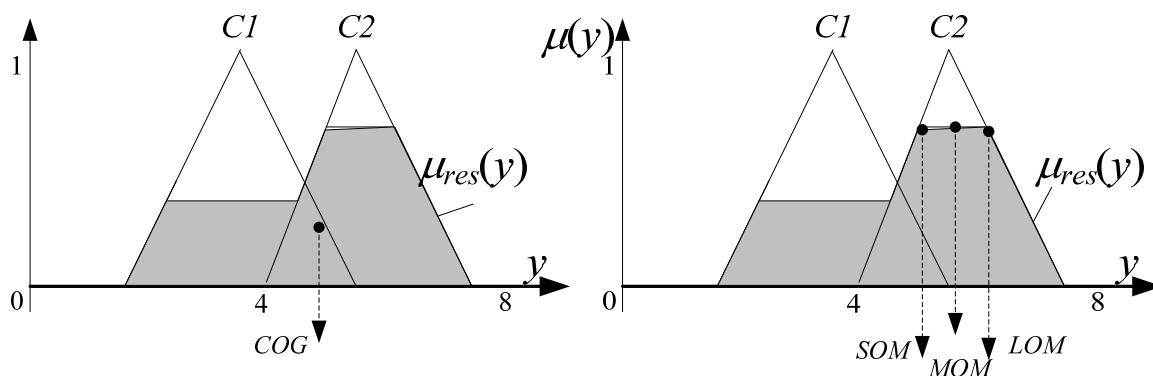


Рис. 2.8. Иллюстрация методов дефаззификации

Существуют различные методы дефаззификации, наиболее известными среди которых являются (см. рис. 2.8):

– *COG (Center of Gravity)* – метод центра тяжести, в котором определяется абсцисса центра тяжести фигуры, ограниченной графиком ФП. Значение координаты центра тяжести может быть найдено как отношение момента фигуры под кривой $\mu_{res}(y)$ относительно вертикальной оси $\mu(y)$ к площади этой фигуры:

$$y^* = \frac{\int y \mu_{res}(y) dy}{\int \mu_{res}(y) dy},$$

пределы интегрирования задаются областью определения результирующего нечеткого множества-вывода. Для интегрирования необходимо определить точки пересечения отдельных составляющих ФП, разбить поверхность на секторы и выполнить интегрирование в пределах каждого из секторов.

Для дискретного нечеткого множества значение координаты центра тяжести определяется по формуле:

$$y^* = \frac{\sum_{i=1}^n y_i \cdot \mu_{res}(y)}{\sum_{i=1}^n \mu_{res}(y)};$$

– *SOM (Smallest Of Maximums)* – метод первого из максимумов, в котором в качестве четкого значения y , представляющего результирующее нечеткое множество, выбирается наименьшее значение, максимизирующее его ФП;

– *LOM (Largest Of Maximums)* – метод последнего максимума, где в качестве четкого значения y выбирается наибольшее значение, соответствующее максимуму ФП;

– *MOM (Mean Of Maximums)* – метод центра максимума, в котором ФП рассматривается как функция, представляющая результирующее множество средним значением, соответствующим максимуму ФП.

2.4. Алгоритмы нечеткого логического вывода

Рассмотрим алгоритмы нечеткого логического вывода. Положим, что базу правил образуют два нечетких правила вида:

П 1: ЕСЛИ ($x_1 = A_1$) И ($x_2 = B_1$) ТО ($y = C_1$);

П 2: ЕСЛИ ($x_1 = A_2$) И ($x_2 = B_2$) ТО ($y = C_2$),

где x_1 и x_2 – имена входных переменных; y – имя переменной вывода; $A_1, A_2, B_1, B_2, C_1, C_2$ – некоторые значения нечетких подмножеств.

Необходимо определить на основе четких значений x_1^* и x_2^* и приведенной базы нечетких правил четкое значение выхода y_0 .

2.4.1. *Алгоритм Мамдани* был предложен в 1975 г. английским математиком Э. Мамдани. Данный алгоритм получил наибольшее применение в системах НЛВ. Этапы нечеткого логического вывода в алгоритме Мамдани описываются в табл. 2.1.

2.4.2. *Алгоритм Ларсена* (табл. 2.1) отличается от алгоритма Мамдани тем, что в нем нечеткая импликация моделируется с использованием операции алгебраического произведения. В случае немонотонных входных нечётких множеств данный алгоритм оказывается точнее алгоритма Мамдани.

Таблица 2.1

Алгоритмы Мамдани и Ларсена

№	Шаг алгоритма	Описание
1	Фаззификация	определяются степени истинности предпосылок каждого правила: $\mu_{A1}(x_1^*), \mu_{A2}(x_1^*), \mu_{B1}(x_2^*), \mu_{B2}(x_2^*)$
2	Агрегирование	определяются уровни «отсечения» для предпосылок каждого из правил с использованием операции «min»: $\alpha_1 = \mu_{A1}(x_1^*) \wedge \mu_{B1}(x_2^*); \quad \alpha_2 = \mu_{A2}(x_1^*) \wedge \mu_{B2}(x_2^*)$
3	Активизация по Мамдани	определяются «усеченные» результирующие ФП на основе min-активизации: $\mu_{C1}^*(y) = \alpha_1 \wedge \mu_{C1}(y); \quad \mu_{C2}^*(y) = \alpha_2 \wedge \mu_{C2}(y)$
	Активизация по Ларсену	определяются «усеченные» результирующие ФП на основе <i>prod</i> -активизации: $\mu_{C1}^*(y) = \alpha_1 \cdot \mu_{C1}(y); \quad \mu_{C2}^*(y) = \alpha_2 \cdot \mu_{C2}(y)$
4	Аккумуляция:	с использованием операции max производится объединение найденных усеченных ФП, что приводит к получению итогового нечеткого подмножества для выходной переменной: $\mu_{\Sigma}(y) = \mu_{C1}^*(y) \vee \mu_{C2}^*(y) = (\alpha_1 \wedge \mu_{C1}) \vee (\alpha_2 \wedge \mu_{C2}(y))$
5	Дефаззификация	нахождение выходной переменной y_0 проводится любым из методов дефаззификации, например, методом центра тяжести

2.4.3. *Алгоритм Сугено*. Модели Такаги-Сугено (*TS*-модели) еще называют квазилинейными моделями и нечеткими линейными моделями. От моделей Мамдани, в которых правила представлены в виде *ЕСЛИ* ($x = A$) *ТО* ($y = B$), *TS*-модели отличаются формой правил:

$$\text{ЕСЛИ } (x = A) \text{ ТО } (y = f(x)),$$

т.е. вместо нечетких термов заключение каждого правила содержит функцию $f(x)$, которая может быть представлена как линейной функцией от входов вида $(y = ax + b)$, так и нелинейной ФП.

Если в модели нечеткой системы база правил имеет вид:

Правило 1: ЕСЛИ $(x_1 = A_1)$ ТО $(y = f_1(x))$;

Правило m : ЕСЛИ $(x_m = A_m)$ ТО $(y = f_m(x))$,

то значение выхода модели вычисляется на основе степеней активизации отдельных заключений $f_i(x)$, $i = 1, \dots, m$, в соответствии с формулой:

$$y = \frac{\sum_{i=1}^m \mu_{Ai}(x) \cdot f_i(x)}{\sum_{i=1}^m \mu_{Ai}(x)}.$$

Этапы выполнения алгоритма Сугено описываются в табл. 2.2; при этом используется набор из двух правил:

П 1: ЕСЛИ $(x_1 = A_1)$ И $(x_2 = B_1)$ ТО $(y = a_1 \cdot x_1 + b_1 \cdot x_2)$;

П 2: ЕСЛИ $(x_1 = A_2)$ И $(x_2 = B_2)$ ТО $(y = a_2 \cdot x_1 + b_2 \cdot x_2)$.

Таблица 2.2

Алгоритм Такаги-Сугено

№	Шаг	Описание
1	Фаззификация	определяются степени истинности предпосылок каждого правила: $\mu_{A1}(x_1^*), \mu_{A2}(x_1^*), \mu_{B1}(x_2^*), \mu_{B2}(x_2^*)$
2	Агрегирование	определяются уровни «отсечения» для предпосылок каждого из правил с использованием операции «min»: $\alpha_1 = \mu_{A1}(x_1^*) \wedge \mu_{B1}(x_2^*)$; $\alpha_2 = \mu_{A2}(x_1^*) \wedge \mu_{B2}(x_2^*)$
3	Активизация	определяются значения степеней истинности всех заключений правил нечетких продукций: $y_1^* = a_1 \cdot x_1^* + b_1 \cdot x_2^*$; $y_2^* = a_2 \cdot x_1^* + b_2 \cdot x_2^*$
4	Аккумуляция:	практически отсутствует, т.к. для расчетов используются действительные числа
5	Дефаззификация	используется модифицированный вариант расчета четкого значения переменной: $y_0 = \frac{a_1 \cdot y_1^* + a_2 \cdot y_2^*}{a_1 + a_2}$

2.5. Системы нечеткого вывода в задачах управления

Одним из основных направлений практического использования систем нечеткого вывода (СНВ) является решение задач управления различными объектами или процессами. В этом случае построение нечеткой модели основывается на формальном представлении

характеристик исследуемой системы в терминах ЛП. Поскольку кроме алгоритма управления, основными понятиями систем управления (СУ) являются входные и выходные переменные, то именно они рассматриваются как ЛП при формировании базы правил в СНВ.

В общем случае *цель управления* заключается в том, чтобы на основе анализа текущего состояния объекта управления (ОУ) определить значения *управляющих* переменных, реализация которых позволяет обеспечить желаемое поведение или состояние ОУ. Для решения этих задач используется классическая теория управления, которая основывается на представлении объекта и процесса управления в форме систем с обратной связью (рис. 2.9).

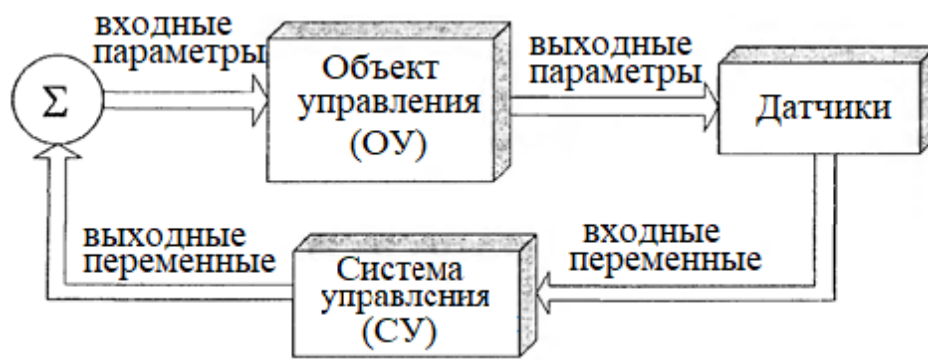


Рис. 2.9. Архитектура компонентов процесса управления с обратной связью

С датчиков, преобразующих контролируемые параметры ОУ в сигналы, входные переменные поступают на вход СУ. На выходе СУ на основе некоторого алгоритма управления формируется множество значений выходных переменных, которые еще называют управляющими переменными. Значения переменных процесса управления поступают на вход ОУ и, совместно со значениями входных параметров ОУ, изменяют его поведение в желаемом направлении. В такой системе используемые для управления техническими объектами СУ называются *контроллерами*.

Модель *нечеткого управления* основана на замене классической СУ системой нечеткого управления, в качестве которой используются СНВ. В этом случае модель нечеткого управления (рис. 2.10) строится с учетом необходимости реализации всех этапов нечеткого вывода, а сам процесс вывода реализуется на основе одного из ранее рассмотренных алгоритмов нечеткого вывода.



Рис. 2.10. Архитектура компонентов процесса нечеткого управления

3. Методика выполнения работы

3.1. Построение модели системы нечеткого управления смесителем воды в душе на основе алгоритмов нечеткого логического вывода Мамдани и Ларсена с простыми правилами вида *SISO*

При принятии душа на вход смесителя подается холодная и горячая вода. Наиболее комфортные условия для душа создаются при наличии на выходе смесителя теплой воды постоянной температуры. Во время принятия душа возможен неравномерный расход воды, поэтому температура воды на выходе смесителя может колебаться, приводя к необходимости ручного изменения подачи холодной или горячей воды. Задача состоит в том, чтобы сделать регулировку температуры воды автоматической, обеспечивая постоянную температуру воды на выходе смесителя (рис. 2.11).

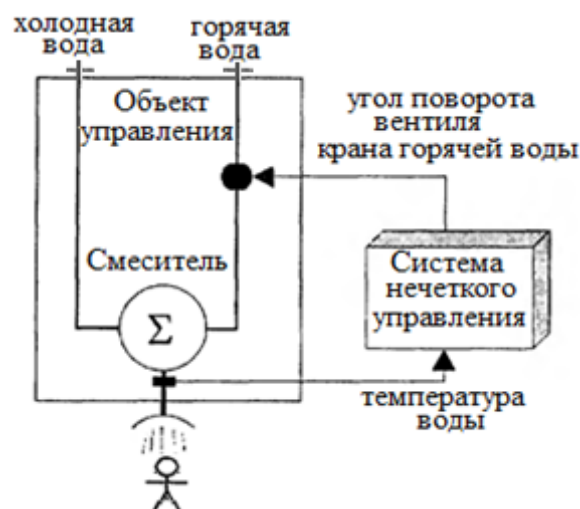


Рис. 2.11. Иллюстрация модели нечеткого управления смесителем воды в душе

Опыт принятия душа позволяет сформулировать несколько эвристических правил, которые применяются каждым человеком при регулировании температуры воды на выходе смесителя. Примем, что поворот вентиля вправо ведет к уменьшению температуры поступающей воды, а поворот влево – к увеличению.

1. Если вода холодная, то следует повернуть вентиль крана горячей воды на большой угол влево.

2. Если вода прохладная, то следует повернуть вентиль крана горячей воды на небольшой угол влево.

3. Если вода теплая, то оставить вентиль крана горячей воды без воздействия.

4. Если вода не очень горячая, то следует повернуть вентиль крана горячей воды на небольшой угол вправо.

5. Если вода горячая, то следует повернуть вентиль крана горячей воды на большой угол вправо.

Представленные правила будут основой базы правил СНВ модели нечеткого управления. Для формирования базы правил необходимо сначала определить входные и выходные ЛП.

Входная ЛП x – «температура воды» на выходе смесителя ($^{\circ}\text{C}$) использует терм-множество $A = \{\text{«холодная» } A_1, \text{ «прохладная» } A_2, \text{ «теплая» } A_3, \text{ «не очень горячая» } A_4, \text{ «горячая» } A_5\}$, с ФП, изображенными на рис. 2.12.

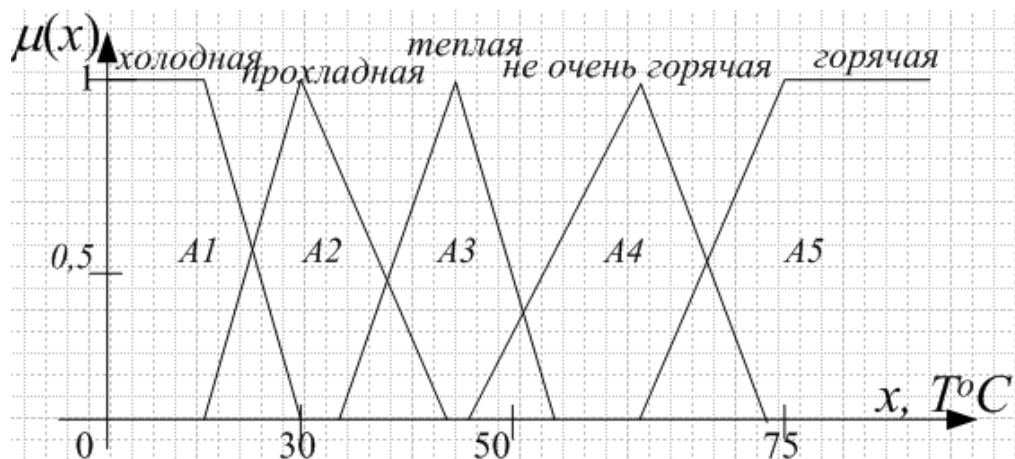


Рис. 2.12. График ФП для термов ЛП «Температура воды»

Выходная ЛП y – «угол поворота вентиля крана горячей воды», измеряемый в угловых градусах: поворот вправо означает положительное направление отсчета, а поворот влево – отрицательное. Терм-множеством ЛП y является множество $B = \{\text{«большой»}$

угол влево» B_1 , «небольшой угол влево» B_2 , «нуль» B_3 , «небольшой угол вправо» B_4 , «большой угол вправо» B_5 с кусочно-линейными ФП, изображенными на рис. 2.13.

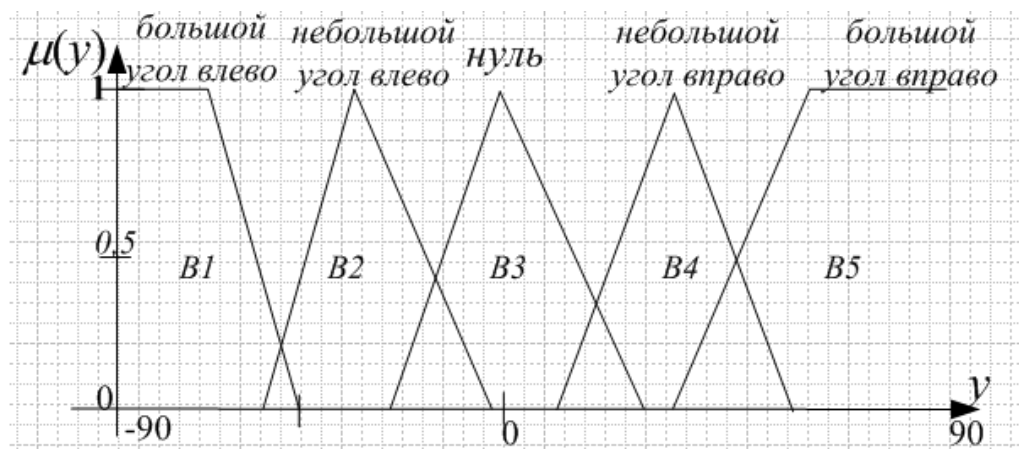


Рис. 2.13 График ФП для термов ЛП «Угол поворота вентиля крана»

Итак, база нечетких правил имеет вид:

Правило 1: ЕСЛИ $x = A_1$ ТО $y = B_1$.

Правило 2: ЕСЛИ $x = A_2$ ТО $y = B_2$.

Правило 3: ЕСЛИ $x = A_3$ ТО $y = B_3$.

Правило 4: ЕСЛИ $x = A_4$ ТО $y = B_4$.

Правило 5: ЕСЛИ $x = A_5$ ТО $y = B_5$.

Необходимо определить угол поворота вентиля крана горячей воды для случая, когда текущая температура воды на выходе смесителя равна 55°C . Для решения задачи используется алгоритм НЛВ Мамдани (рис. 2.14).

Фаззификация входной ЛП приводит к срабатыванию 3-го и 4-го правил нечетких продукций, поэтому эти правила считаются *активными* и используются в текущем процессе нечеткого вывода. Степени истинности для предпосылок активных правил равны 0,2 и 0,5: $\mu_{A_3}(x)|_{x^*=55} = 0,2$; $\mu_{A_4}(x)|_{x^*=55} = 0,5$.

Поскольку все условия в правилах заданы в форме *простых* высказываний типа *SISO*, этап их агрегирования тривиален и оставляет степени истинности 0,2 и 0,5 без изменения.

На этапе активизации заключений в нечетких правилах продукций определяются «усеченные» результирующие ФП:

Правило 3: $\mu_{B_3}(y)|_{x^*=55} = \min\{0,2; \mu_{B_3}(y)\}$;

Правило 4: $\mu_{B_4}(y)|_{x^*=55} = \min\{0,5; \mu_{B_4}(y)\}$.

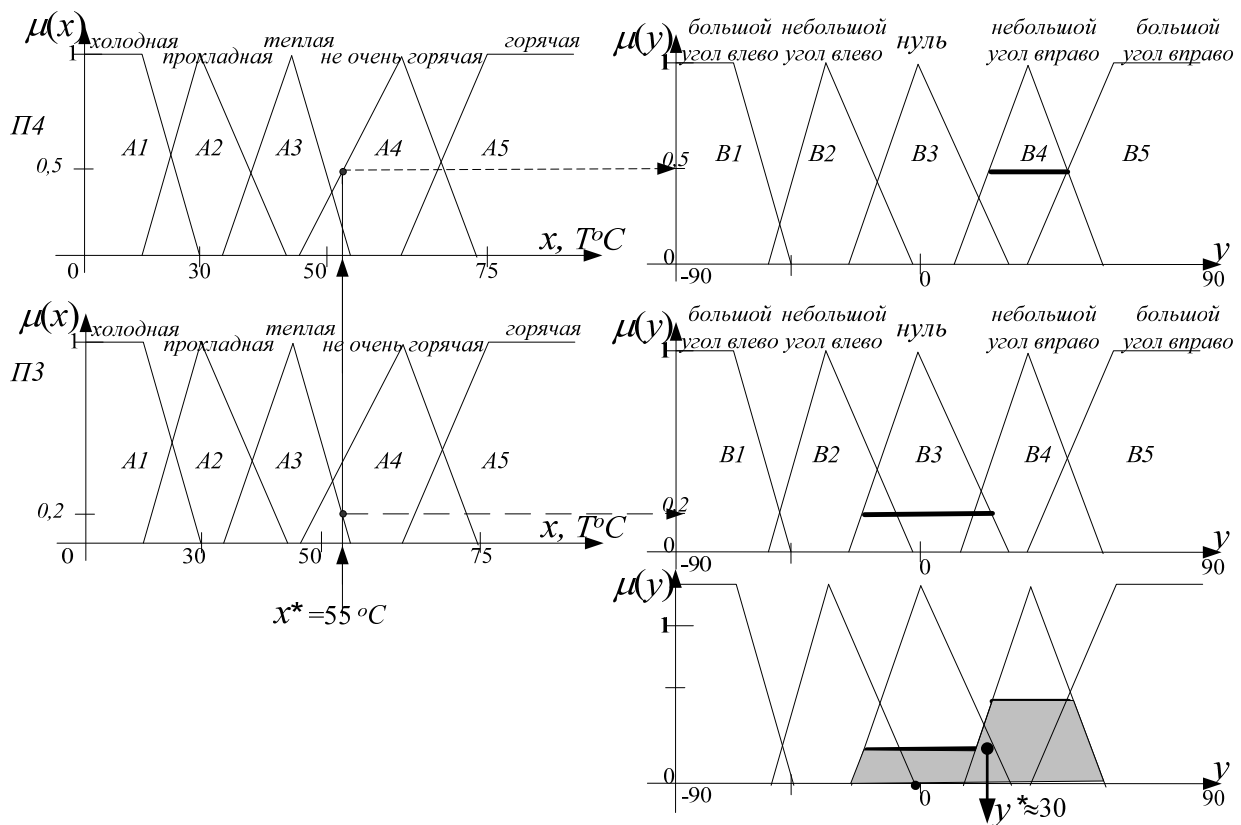


Рис. 2.14. Процедура логического вывода по алгоритму Мамдани

Отметим, что в алгоритме Ларсена (рис. 2.15) этапы фаззификации и агрегирования выполняются так же, как в алгоритме Мамдани, а на этапе активизации заключений нечеткая импликация моделируется с помощью оператора умножения:

Правило 3: $\mu_{B3}(y)|_{x^*=55} = \text{prod}\{0,2; \mu_{B3}(y)\}$.

Правило 4: $\mu_{B4}(y)|_{x^*=55} = \text{prod}\{0,5; \mu_{B4}(y)\}$;

После аккумулялирования заключений нечетких правил продукций с использованием операции max-дизъюнкции получим нечеткое множество с ФП: $\mu_{res}(y) = \text{max}\{\mu_{B2}(y)|_{x^*=55}; \mu_{B3}(y)|_{x^*=55}\}$.

Заключительным этапом является дефаззификация выходной ЛП. Если допустить, что результирующее нечеткое множество является дискретным, то можно определить приближенное значение управляющей переменной методом центра тяжести по формуле:

$$y^* = \frac{\sum_{i=1}^n y_i \cdot \mu_{res}(y)}{\sum_{i=1}^n \mu_{res}(y)}.$$

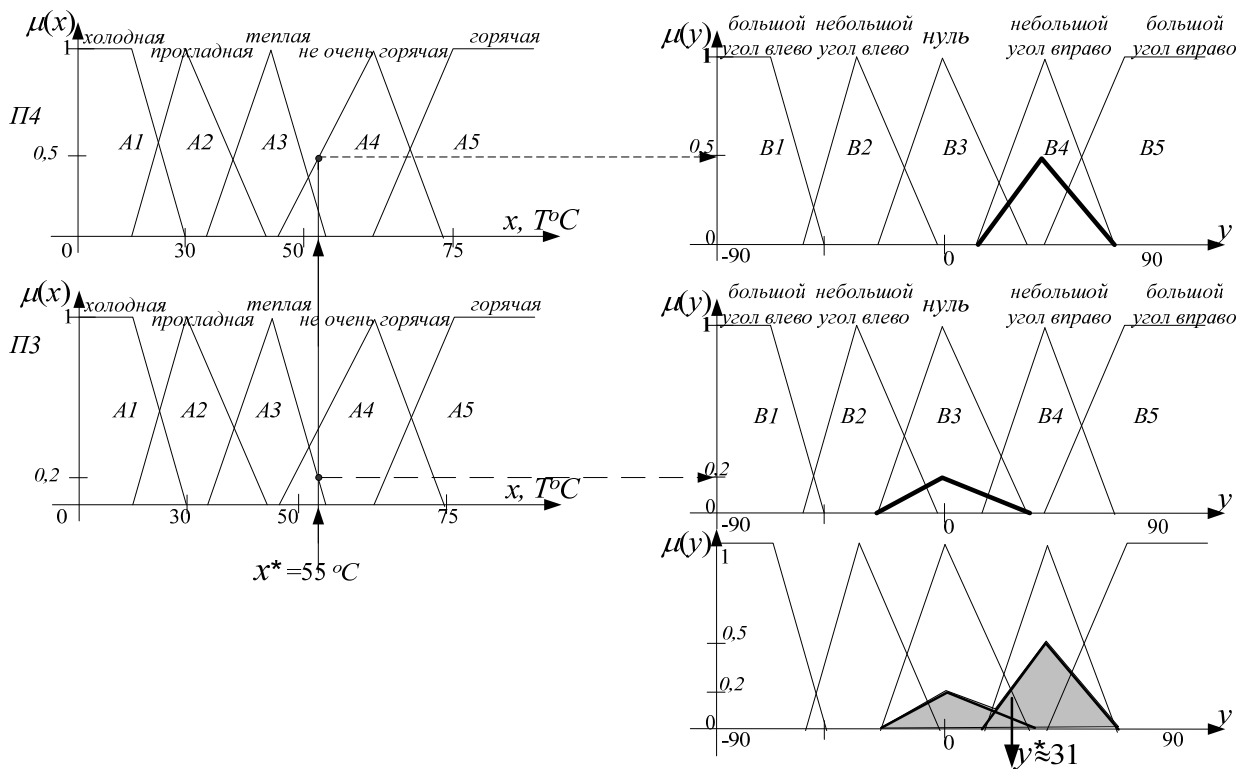


Рис. 2.15. Процедура логического вывода по алгоритму Ларсена

Процедура дефаззификации нечеткого множества, полученного по алгоритму Мамдани методом центра тяжести, представлена ниже:

$$\sum_{i=1}^n y_i \cdot \mu_{res}(y) = (-25) \cdot 0 + (-20) \cdot 0,1 + (-15 - 10 - 5 + 0 + 5 + 10 + 15) \cdot 0,2 + 20 \cdot 0,4 + (25 + 30 + 35 + 40 + 45 + 50 + 55 + 60) \cdot 0,5 + 65 \cdot 0,2 + 70 \cdot 0 = 196,5;$$

$$\sum_{i=1}^n \mu_{res}(y) = 0 + 0,1 + 0,2 \cdot 6 + 0,4 + 0,5 \cdot 8 + 0,2 + 0 = 6,7;$$

$$y^* = \frac{196,5}{6,7} = 29,3 \approx 29^\circ.$$

Процедура дефаззификации по алгоритму Ларсена:

$$\sum_{i=1}^n y_i \cdot \mu_{res}(y) = (-25) \cdot 0 + (-20) \cdot 0,01 + (-15) \cdot 0,05 + (-10) \cdot 0,1 + (-5) \cdot 0,15 + 0 \cdot 0,2 + 5 \cdot 0,15 + 10 \cdot 0,1 + 15 \cdot 0,05 + 20 \cdot 0,1 + 25 \cdot 0,2 + 30 \cdot 0,3 + 35 \cdot 0,4 + 40 \cdot 0,5 + 45 \cdot 0,4 + 50 \cdot 0,3 + 55 \cdot 0,2 + 60 \cdot 0,1 + 65 \cdot 0,05 + 70 \cdot 0 = 103,05;$$

$$\sum_{i=1}^n \mu_{res}(y) = 0 + 0,01 + 0,05 + 0,1 + 0,15 + 0,2 + 0,15 + 0,1 + 0,05 + 0,1 + 0,2 + 0,3 + 0,4 + 0,5 + 0,4 + 0,3 + 0,2 + 0,1 + 0,05 + 0 = 3,36;$$

$$y^* = \frac{103,05}{3,36} = 30,67 \approx 31^\circ.$$

Таким образом, на основе двух алгоритма НЛВ Мамдани и Ларсена были рассчитаны значения угла поворота вентиля крана горячей воды, равного приблизительно 30° . Полученный результат означает, что когда текущая температура воды на выходе смесителя становится высокой и равной 55°C , вентиль горячей воды необходимо повернуть на 30° вправо – это уменьшит температуру поступающей из смесителя воды.

Для практической реализации рассмотренной системы нечеткого управления смесителем воды в душе необходимо организовать периодическое измерение температуры воды на выходе смесителя в некоторые дискретные моменты времени; при этом, чем меньше будет интервал измерения температуры, тем выше станет точность регулирования температуры воды.

3.2. Построение модели системы нечеткого управления кондиционером воздуха в помещении на основе алгоритма нечеткого логического вывода Мамдани со сложными правилами вида *MISO*

В помещении установлен бытовой кондиционер, который позволяет охлаждать или нагревать воздух. Очевидно, что наиболее комфортные условия создаются при некоторой стабильной температуре воздуха, однако поскольку температура окружающей среды вне помещения изменяется в течение суток, температура воздуха в помещении может колебаться. Задача состоит в том, чтобы сделать регулировку кондиционера автоматической, обеспечивая постоянную температуру воздуха в помещении (рис. 2.16).

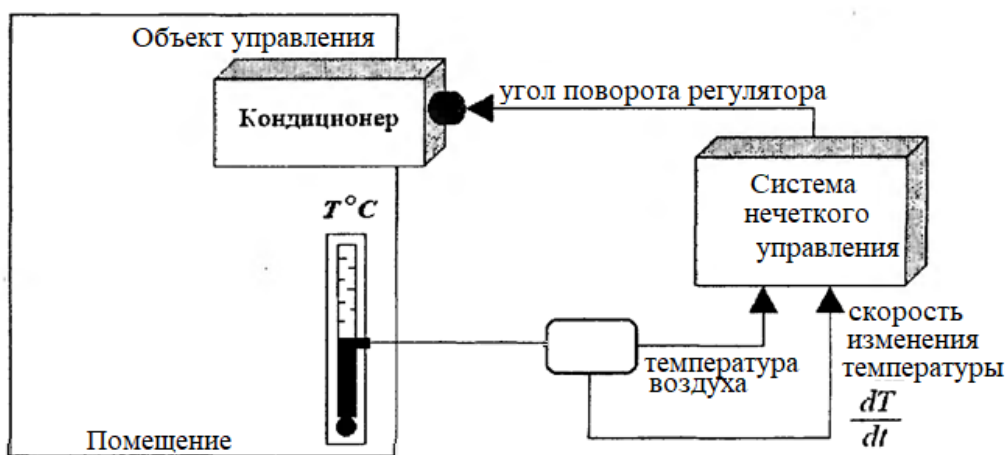


Рис. 2.16. Иллюстрация модели нечеткого управления кондиционером воздуха в помещении

Включение режима «холод» осуществляется поворотом регулятора кондиционера влево, а включение режима «тепло» – вправо относительно некоторой точки текущего состояния кондиционера.

При использовании бытовых кондиционеров было замечено, что процесс охлаждения или нагревания воздуха в помещении обладает некоторой инерционностью; например, после включения режима «холод» происходит нагнетание холодного воздуха, в связи с чем температура постепенно падает, и даже в момент отключения этого режима температура продолжает падать в течение небольшого конечного промежутка времени. Аналогичная ситуация наблюдается при включении и отключении режима «тепло». Поэтому необходимо рассматривать в качестве параметров объекта управления не только температуру воздуха в помещении, но и скорость изменения температуры.

Опыт ручного регулирования температуры воздуха в помещении с кондиционером позволил экспертам сформулировать следующие эвристические правила:

1. Если температура воздуха в помещении очень теплая, а скорость изменения температуры положительная, то следует включить режим «холод», повернув регулятор кондиционера на очень большой угол влево.

2. Если температура воздуха в помещении очень теплая, а скорость изменения температуры отрицательная, то следует включить режим «холод», повернув регулятор кондиционера на небольшой угол влево.

3. Если температура воздуха в помещении теплая, а скорость изменения температуры положительная, то следует включить режим «холод», повернув регулятор кондиционера на большой угол влево.

4. Если температура воздуха в помещении теплая, а скорость изменения температуры отрицательная, то кондиционер следует выключить.

5. Если температура воздуха в помещении очень холодная, а скорость изменения температуры отрицательная, то следует включить режим «тепло», повернув регулятор кондиционера на очень большой угол вправо.

6. Если температура воздуха в помещении очень холодная, а скорость изменения температуры положительная, то следует

включить режим «тепло», повернув регулятор кондиционера на небольшой угол вправо.

7. Если температура воздуха в помещении холодная, а скорость изменения температуры отрицательная, то следует включить режим «тепло», повернув регулятор кондиционера на большой угол вправо.

8. Если температура воздуха в помещении холодная, а скорость изменения температуры положительная, то кондиционер следует выключить.

9. Если температура воздуха в помещении очень теплая, а скорость изменения температуры равна нулю, то следует включить режим «холод», повернув регулятор кондиционера на большой угол влево.

10. Если температура воздуха в помещении теплая, а скорость изменения температуры равна нулю, то следует включить режим «холод», повернув регулятор кондиционера на небольшой угол влево.

11. Если температура воздуха в помещении очень холодная, а скорость изменения температуры равна нулю, то следует включить режим «тепло», повернув регулятор кондиционера на большой угол вправо.

12. Если температура воздуха в помещении холодная, а скорость изменения температуры равна нулю, то следует включить режим «тепло», повернув регулятор кондиционера на небольшой угол вправо.

13. Если температура воздуха в помещении в пределах нормы, а скорость изменения температуры положительная, то следует включить режим «холод», повернув регулятор кондиционера на небольшой угол влево.

14. Если температура воздуха в помещении в пределах нормы, а скорость изменения температуры отрицательная, то следует включить режим «тепло», повернув регулятор кондиционера на небольшой угол вправо.

15. Если температура воздуха в помещении в пределах нормы, а скорость изменения температуры равна нулю, то кондиционер следует выключить.

Сформулированные правила становятся основой базы правил СНВ модели нечеткого управления кондиционером.

Входными переменными модели являются температура воздуха в помещении (измеряемая в $^{\circ}\text{C}$), и скорость изменения температуры воздуха (измеряемая в $^{\circ}\text{C}/\text{мин}$).

Для первой входной ЛП x_1 «Температура воздуха» используется терм-множество $A = \{\text{«очень холодная» } A_1, \text{ «холодная» } A_2, \text{ «в пределах нормы» } A_3, \text{ «теплая» } A_4, \text{ «очень теплая» } A_5\}$, которое записывается в символическом виде: $A = \{NB, NS, Z, PS, PB\}$, с ФП, изображенными на рис. 2.17.

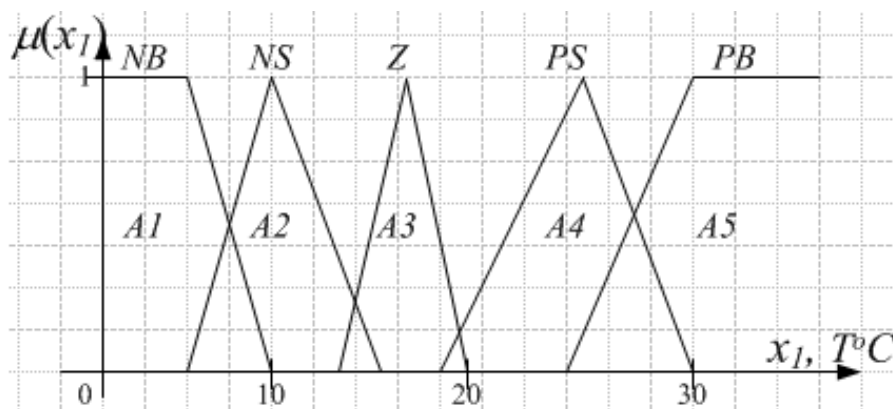


Рис. 2.17. График ФП для термов ЛП «Температура воздуха»

Для второй входной ЛП x_2 «Скорость изменения температуры воздуха» используется терм-множество $B = \{\text{«отрицательная» } B_1, \text{ «равна нулю» } B_2, \text{ «положительная» } B_3\}$, или $B = \{NS, Z, PS\}$, с ФП, изображенными на рис. 2.18.

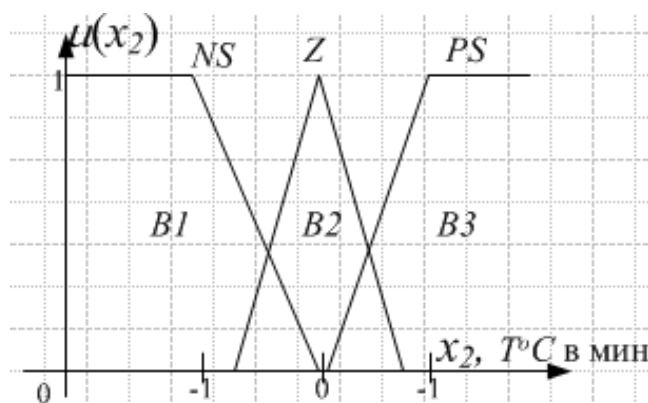


Рис. 2.18. График ФП для термов ЛП «Скорость изменения температуры»

Выходной ЛП является y – «Угол поворота регулятора» включения режимов «холод» и «тепло» кондиционера, измеряемый в угловых градусах: поворот регулятора вправо означает включение режима «тепло» и положительное направление отсчета, а поворот

влево – включение режима «холод» и отрицательное направление отсчета. Для переменной выхода используется терм-множество $C = \{\text{«очень большой угол влево» } C_1, \text{ «большой угол влево» } C_2, \text{ «небольшой угол влево» } C_3, \text{ «выключить кондиционер» } C_4, \text{ «небольшой угол вправо» } C_5, \text{ «большой угол вправо» } C_6, \text{ «очень большой угол вправо» } C_7\}$ или в символическом виде $C = \{NB, NM, NS, Z, PS, PM, PB\}$ с кусочно-линейными ФП, изображенными на рис. 2.19.

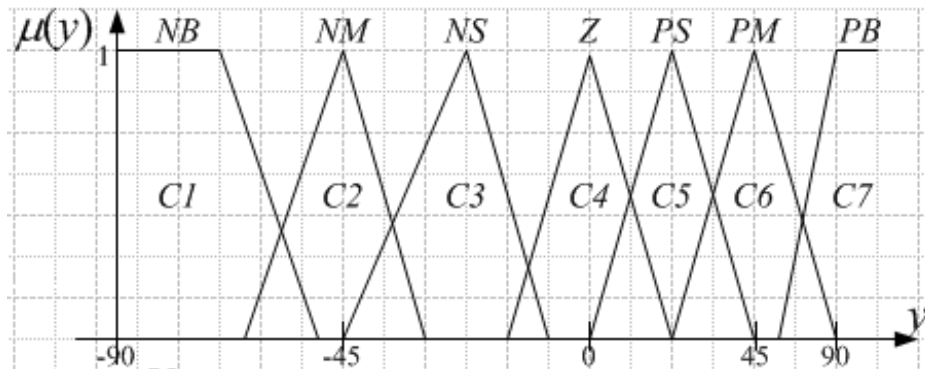


Рис. 2.19. График ФП для термов ЛП «Угол поворота регулятора»

База нечетких правил в матричном виде и в виде продукционных правил представлена в табл. 2.3 и 2.4.

Таблица 2.3

База нечетких правил в матричном виде

	<i>NS</i>	<i>Z</i>	<i>PS</i>
<i>NB</i>	<i>PB</i>	<i>PM</i>	<i>PS</i>
<i>NS</i>	<i>PM</i>	<i>PS</i>	<i>Z</i>
<i>Z</i>	<i>PS</i>	<i>Z</i>	<i>NS</i>
<i>PS</i>	<i>Z</i>	<i>NS</i>	<i>NM</i>
<i>PB</i>	<i>NS</i>	<i>NM</i>	<i>NB</i>

Таблица 2.4

База нечетких правил в виде продукционных правил

№	Правила
1	2
1	ЕСЛИ $x_1 = PB$ И $x_2 = PS$, ТО $y = NB$
2	ЕСЛИ $x_1 = PB$ И $x_2 = NS$, ТО $y = NS$
3	ЕСЛИ $x_1 = PS$ И $x_2 = PS$, ТО $y = NM$
4	ЕСЛИ $x_1 = PS$ И $x_2 = NS$, ТО $y = Z$
5	ЕСЛИ $x_1 = NB$ И $x_2 = NS$, ТО $y = PB$
6	ЕСЛИ $x_1 = NB$ И $x_2 = PS$, ТО $y = PS$
7	ЕСЛИ $x_1 = NS$ И $x_2 = NS$, ТО $y = PM$

1	2
8	ЕСЛИ $x_1 = NS$ И $x_2 = PS$, ТО $y = Z$
9	ЕСЛИ $x_1 = PB$ И $x_2 = Z$, ТО $y = NM$
10	ЕСЛИ $x_1 = PS$ И $x_2 = Z$, ТО $y = NS$
11	ЕСЛИ $x_1 = NB$ И $x_2 = Z$, ТО $y = PM$
12	ЕСЛИ $x_1 = NS$ И $x_2 = Z$, ТО $y = PS$
13	ЕСЛИ $x_1 = Z$ И $x_2 = PS$, ТО $y = NS$
14	ЕСЛИ $x_1 = Z$ И $x_2 = NS$, ТО $y = PS$
15	ЕСЛИ $x_1 = Z$ И $x_2 = Z$, ТО $y = Z$

Рассматривается пример работы алгоритма НЛВ Мамдани для случая, когда текущая температура воздуха равна $x_1^* = 19,5$ °С, а скорость ее положительная и равна $x_2^* = 0,2$ °С/мин (рис. 2.20). Соответствующие подусловия используются в правилах нечетких продукций с номерами 3, 10, 13, 15 – эти правила считаются активными и используются в текущем процессе нечеткого вывода.

Этап фаззификации переменных. На этом этапе определяются степени истинности предпосылок правил:

$$\begin{aligned} \mu_{PS}(x_1)|_{x_1^*=19,5} &= 0,2; & \mu_{PS}(x_2)|_{x_2^*=0,2} &= 0,3; \\ \mu_Z(x_1)|_{x_1^*=19,5} &= 0,5; & \mu_Z(x_2)|_{x_2^*=0,2} &= 0,7. \end{aligned}$$

Этап агрегирования. Поскольку все условия в правилах заданы в форме *сложных* высказываний типа *MISO*, а подусловия во всех правилах соединены логической связкой *И*, то операция агрегирования будет выполняться с помощью оператора *T*-нормы. В методе Мамдани в качестве *T*-нормы выступает операция *min*:

Правило 3: $T \min \{ \mu_{PS}(x_1^*); \mu_{PS}(x_2^*) \} = 0,2;$

Правило 10: $T \min \{ \mu_{PS}(x_1^*); \mu_Z(x_2^*) \} = 0,2;$

Правило 13: $T \min \{ \mu_Z(x_1^*); \mu_{PS}(x_2^*) \} = 0,3;$

Правило 15: $T \min \{ \mu_Z(x_1^*); \mu_Z(x_2^*) \} = 0,5.$

Этап активизации заключений. На этом этапе в нечетких правилах продукций формируются «усеченные» ФП выходной переменной для каждого из правил; степени истинности для предпосылок каждого правила представлены ниже:

Правило 3: $\mu_{NM}^{\Pi 3}(y) = \min \{ \min \{ \mu_{PS}(x_1^*); \mu_{PS}(x_2^*) \}; \mu_{NM}(y) \} = \min(0,2; \mu_{NM}(y));$

Правило 10: $\mu_{NS}^{\Pi 10}(y) = \min \{ \min \{ \mu_{PS}(x_1^*); \mu_Z(x_2^*) \}; \mu_{NS}(y) \} = \min(0,2; \mu_{NS}(y));$

Правило 13: $\mu_{NS}^{\Pi 13}(y) = \min \{ \min \{ \mu_Z(x_1^*); \mu_{PS}(x_2^*) \}; \mu_{NS}(y) \} = \min(0,3; \mu_{NS}(y));$

Правило 15: $\mu_Z^{\Pi 15}(y) = \min \{ \min \{ \mu_Z(x_1^*); \mu_Z(x_2^*) \}; \mu_Z(y) \} = \min(0,5; \mu_Z(y)).$

Этап аккумуляирования заключений нечетких правил. На этапе аккумуляирования заключений нечетких правил продукции с использованием операции *max-дизъюнкции* образуется нечеткое множество с ФП:

$$\mu_{res}(y) = \max\{\mu_{NM}^{\Pi 3}(y); \mu_{NS}^{\Pi 10}(y); \mu_{NS}^{\Pi 13}(y); \mu_Z^{\Pi 15}(y)\}.$$

Этап дефаззификации. Проведем дефаззификацию выходной ЛП методом центра тяжести, допустив, что результирующее нечеткое множество является дискретным, и определим приближенное значение управляющей переменной по формуле: $y^* = \frac{\sum_{i=1}^n y_i \cdot \mu_{res}(y)}{\sum_{i=1}^n \mu_{res}(y)}$.

$$\sum_{i=1}^n y_i \cdot \mu_{res}(y) = (-70) \cdot 0 + (-65 - 60 - 55 - 50 - 45) \cdot 0,2 +$$

$$+ (-35 - 30 - 25 - 20) \cdot 0,3 + (-15) \cdot 0,4 + (-10 - 5 + 0 + 5 + 15) \cdot 0,5 +$$

$$+ 15 \cdot 0 = -106;$$

$$\sum_{i=1}^n \mu_{res}(y) = 0 + 0,15 \cdot 5 + 0,3 \cdot 5 + 0,4 + 0,5 \cdot 4 + 0 = 5,4;$$

$$y^* = \frac{-106}{5,4} = -19,6296 \approx -20.$$

При дефаззификации выходной ЛП методом центра тяжести определяется значение управляющей переменной: при температуре воздуха $x_1^* = 19,5^\circ\text{C}$ и скорости изменения температуры, равной $x_2^* = 0,2^\circ\text{C}/\text{мин}$ угол поворота регулятора кондиционера влево должен быть приблизительно равен 20° . Это значение соответствует включению режима «холод» на пятую часть своей мощности и является результатом решения задачи нечеткого вывода.

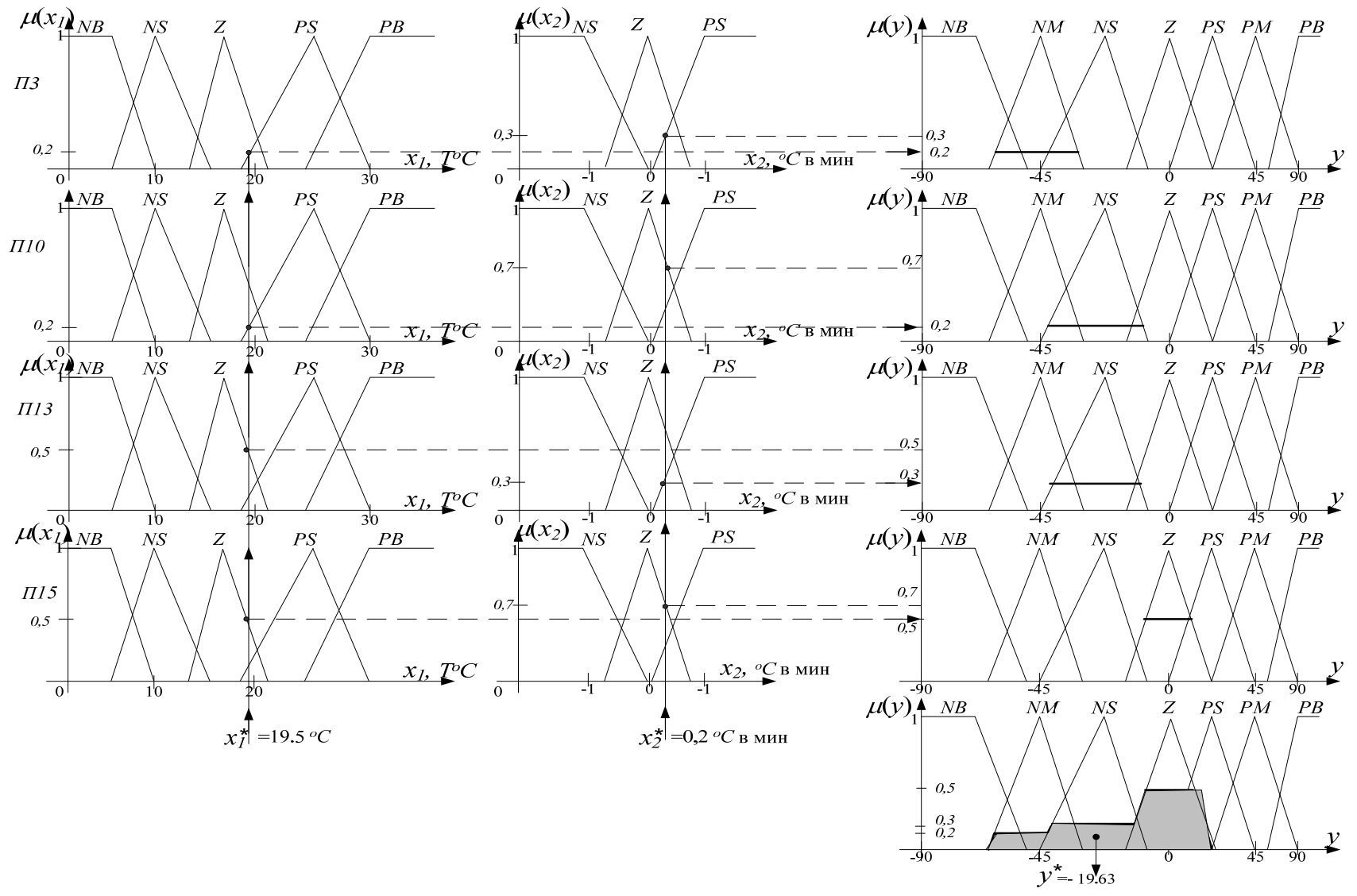


Рис. 2.20. Процедура нечеткого логического вывода по алгоритму Мамдани

3.3. Построение нечеткой модели Такаги-Сугено

3.3.1 Модель Такаги-Сугено с простыми правилами вида *SISO*

Пусть имеется три простых правила вида *SISO*:

Правило 1: ЕСЛИ ($x = A_1$) ТО ($y = -x + 3$),

Правило 2: ЕСЛИ ($x = A_2$) ТО ($y = \frac{4x-10}{3}$),

Правило 3: ЕСЛИ ($x = A_3$) ТО ($y = \frac{-x+24}{3}$).

Трапецевидные ФП и поверхность отображения нечеткой *TS*-модели представлены на рис. 2.21.

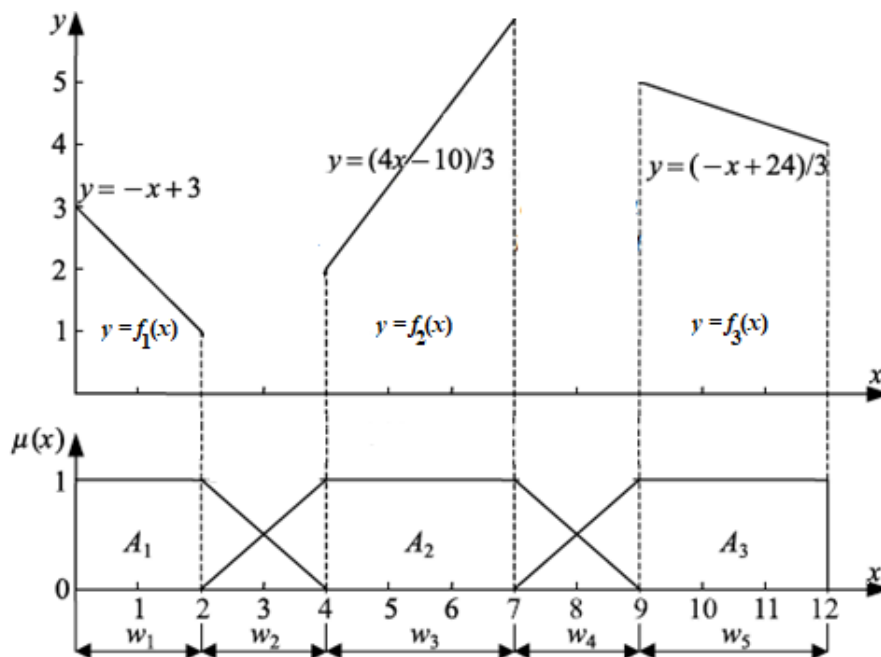


Рис. 2.21. ФП входных параметров и поверхность отображения *TS*-модели

Строится обобщенная поверхность нечеткой модели $y(x)$. Для этого вводятся дополнительные логические переменные для пяти участков графика ФП $\mu(x)$.

$$w_1 = \begin{cases} 1 & \text{для } 0 \leq x < 2, \\ 0 & \text{в других случаях;} \end{cases} \quad w_2 = \begin{cases} 1 & \text{для } 2 \leq x < 4, \\ 0 & \text{в других случаях;} \end{cases}$$

$$w_3 = \begin{cases} 1 & \text{для } 4 \leq x < 7 \\ 0 & \text{в других случаях;} \end{cases} \quad w_4 = \begin{cases} 1 & \text{для } 7 \leq x < 9 \\ 0 & \text{в других случаях.} \end{cases}$$

$$w_5 = \begin{cases} 1 & \text{для } 9 \leq x \leq 12 \\ 0 & \text{в других случаях;} \end{cases}$$

В верхней части рисунка показана поверхность нечеткой модели $y(x)$, которая соответствует заключениям правил, представленным в виде линейных функций, но только на тех участках входного пространства, степени принадлежности элементов которых

соответствуют множествам A_i с единичной высотой ФП, т.е. удовлетворяющих условию $\mu_{A_i}(x) = 1$ (участки w_1, w_3, w_5).

На участках w_2 и w_4 поверхность модели должна *переходить* из одной линейной формы, задаваемой соответствующим заключением, в другую. Математическое выражение *функций перехода* зависит от типа используемых ФП.

С помощью логических переменных, записывается ФП нечетких подмножеств A_1, A_2, A_3 в виде кусочно-линейных функций:

$$\mu_{A_1}(x) = w_1 - 0,5 \cdot (x - 4) \cdot w_2.$$

$$\mu_{A_2}(x) = 0,5 \cdot (x - 2) \cdot w_2 + w_3 - 0,5 \cdot (x - 9) \cdot w_4.$$

$$\mu_{A_3}(x) = 0,5 \cdot (x - 7) \cdot w_4 + w_5.$$

Рассматриваемые ФП удовлетворяют условию разбиения единицы: $\sum_{i=1}^3 \mu_{A_i}(x) = 1$.

Результирующая функция поверхности модели рассчитывается по формуле:

$$y(x) = \sum_{i=1}^3 \mu_{A_i}(x) \cdot f_i(x).$$

На участках w_1, w_3, w_5 с единичной высотой функция поверхности равна:

$$y_{w_1}(x) = \mu_{A_1}(x) \cdot f_1(x) = w_1 \cdot (-x + 3);$$

$$y_{w_3}(x) = \mu_{A_2}(x) \cdot f_2(x) = w_3 \cdot \left(\frac{4x-10}{3}\right);$$

$$y_{w_5}(x) = \mu_{A_3}(x) \cdot f_3(x) = w_5 \cdot \left(\frac{-x+24}{3}\right).$$

На участках w_2 и w_4 строятся функции перехода, которые имеют квадратичную форму:

$$y_{w_2}(x) = \mu_{A_1}(x) \cdot f_1(x) + \mu_{A_2}(x) \cdot f_2(x) = [-0,5 \cdot (x - 4) \cdot w_2] \cdot (-x + 3) + [0,5 \cdot (x - 2) \cdot w_2] \cdot \left(\frac{4x - 10}{3}\right) = \frac{w_2(7x^2 - 39x + 56)}{6}.$$

$$y_{w_4}(x) = \mu_{A_2}(x) \cdot f_2(x) + \mu_{A_3}(x) \cdot f_3(x) = [-0,5 \cdot (x - 9) \cdot w_4] \cdot \left(\frac{4x-10}{3}\right) + [0,5 \cdot (x - 7) \cdot w_4] \cdot \frac{-x + 24}{3} = \frac{w_4(-5x^2 + 77x - 258)}{6}.$$

Результирующая функция обобщенной поверхности нечеткой модели равна:

$$y(x) = w_1 \cdot (-x + 3) + \frac{w_2 \cdot (7x^2 - 39x + 56)}{6} + \frac{w_3 \cdot (4x - 10)}{3} + \frac{w_4 \cdot (-5x^2 + 77x - 258)}{6} + \frac{w_5 \cdot (-x + 24)}{3}.$$

Графическое изображение результирующей функции обобщающей поверхности показано на рис. 2.22.

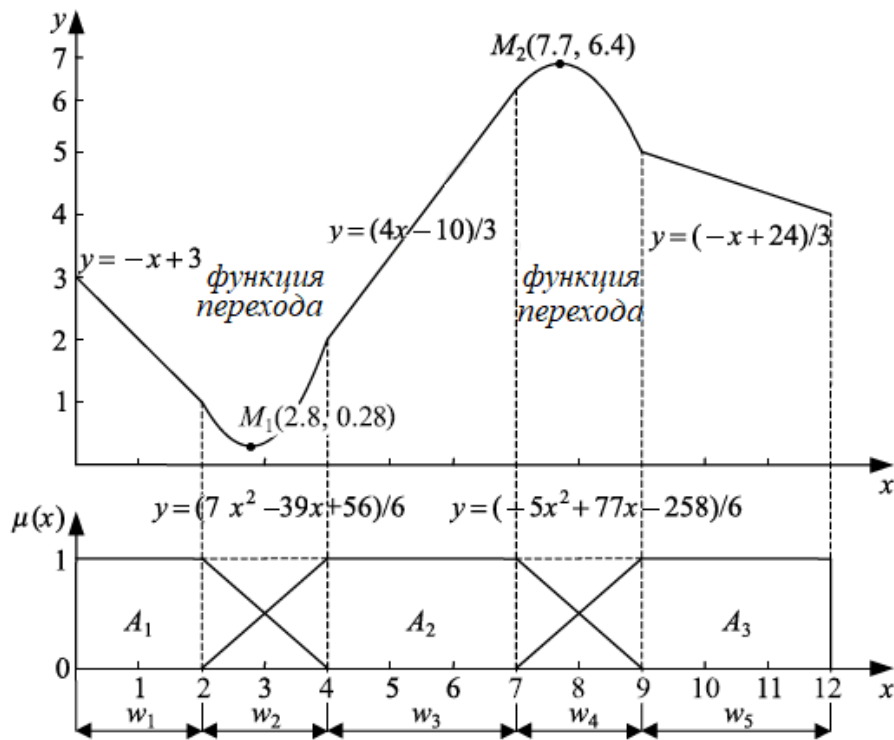


Рис. 2.22. ФП входных параметров и результирующая поверхность TS -модели

Теперь может быть решена задача нахождения выходного результата нечеткой модели Такаги-Сугено. Пусть четкое значение входной переменной равно $x^* = 2,8$. По рисунку значение выходной функции должно находиться во второй области поверхности нечеткой модели. Определяются уровни отсечения α_i и значения степеней истинности y_i^* заключений правил:

$$\alpha_1 = -0,5 \cdot (x - 4) \cdot w_2 = 0,6 \cdot w_2; \quad y_1^* = -x + 3 = 2,8 + 3 = 0,2.$$

$$\alpha_2 = 0,5 \cdot (x - 2) \cdot w_2 = 0,4 \cdot w_2; \quad y_2^* = \frac{4 \cdot x - 10}{3} = 0,4.$$

Дефаззификация проводится либо с использованием модифицированного варианта расчета четкого значения по формуле:

$$y_0 = \frac{\alpha_1 \cdot y_1^* + \alpha_2 \cdot y_2^*}{\alpha_1 + \alpha_2} = \frac{0,6 \cdot w_2 \cdot 0,2 + 0,4 \cdot w_2 \cdot 0,4}{0,6 \cdot w_2 + 0,4 \cdot w_2} = 0,28;$$

либо путем подстановки значения входной переменной x^* в результирующую функцию $y(x)$:

$$y_0 = \frac{w_2 \cdot (7x^2 - 39x + 56)}{6} = \frac{1 \cdot (7 \cdot 2,8^2 - 39 \cdot 2,8 + 56)}{6} = 0,28.$$

Результирующая функция обобщающей поверхности непрерывна и не имеет скачков на границах областей. Вместе с тем, производная этой функции непрерывной не является и изменяется на границах областей скачкообразно, что является следствием типа используемых ФП (их кусочной линейности). Непрерывность

и большую гладкость поверхности модели можно обеспечить путем использования непрерывно дифференцируемых ФП, например, гауссовых функций.

В данном примере использованы трапецевидные ФП, которые имеют зоны четкости, характеризующиеся тем, что их элементы полностью (со степенью, равной 1) принадлежат соответствующему множеству. Вообще применять *TS*-модели следует главным образом в тех случаях, когда ФП имеют трапецевидную или подобную ей форму; не рекомендуется использовать треугольные ФП.

3.3.2. Модель Такаги-Сугено со сложными правилами вида *MISO*

Пусть имеется два правила база правил вида *MISO*:

Правило 1:

ЕСЛИ $x_1 = \text{малое}$ (A_1) *И* $x_2 = \text{малое}$ (B_1) *ТО* $y_1 = -2x_1 + 5x_2$.

Правило 2:

ЕСЛИ $x_1 = \text{большое}$ (A_2) *И* $x_2 = \text{среднее}$ (B_2) *ТО* $y_2 = 2 + 7x_1 - 3x_2$.

Входные сигналы, полученные с датчиков объекта управления, равны: $x_1^* = 2$ и $x_2^* = 3$.

Процедура нечеткого логического вывода *TS*-модели включает следующие этапы (рис. 2.23).

На этапе *фаззификации* определяются степени истинности предпосылок нечетких правил:

$$\mu_{A_1}(x_1)|_{x_1^*=2}=0,75; \quad \mu_{B_1}(x_2)|_{x_2^*=3}=0,2;$$

$$\mu_{A_2}(x_1)|_{x_1^*=2}=0,3; \quad \mu_{B_2}(x_2)|_{x_2^*=3}=0,7.$$

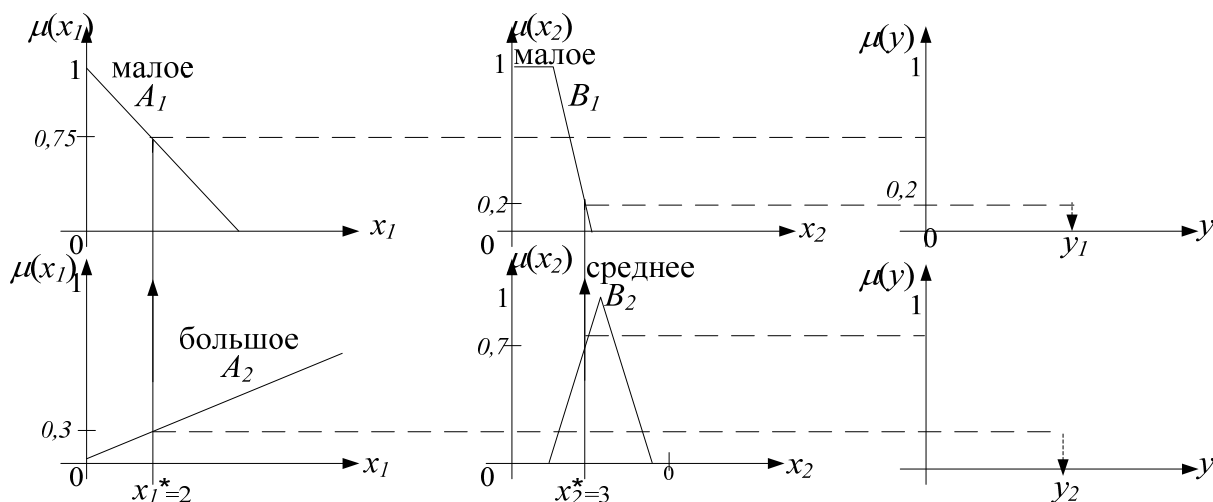


Рис. 2.23. Процедура нечеткого логического вывода по алгоритму Такаги-Сугено

На этапе *агрегирования* определяются уровни «отсечения» для предпосылок каждого из правил с использованием операции *min*:

$$\alpha_1 = \min(\mu_{A1}(x_1)|_{x_1^*=2}; \mu_{B1}(x_2)|_{x_2^*=3}) = \min(0,75; 0,2) = 0,2;$$

$$\alpha_2 = \min(\mu_{A2}(x_1)|_{x_1^*=2}; \mu_{B2}(x_2)|_{x_2^*=3}) = \min(0,3; 0,7) = 0,3.$$

На этапе *активизации* правил определяются значения истинности всех заключений правил нечетких продукций:

$$y_1^* = f_1(x_1^*; x_2^*) = -2 \cdot 2 + 5 \cdot 3 = 11;$$

$$y_2^* = f_2(x_1^*; x_2^*) = 2 + 7 \cdot 2 - 3 \cdot 3 = 7.$$

При дефаззификации используется модифицированный вариант расчета четкого значения:

$$y_0 = \frac{\alpha_1 \cdot y_1^* + \alpha_2 \cdot y_2^*}{\alpha_1 + \alpha_2} = \frac{0,2 \cdot 11 + 0,3 \cdot 7}{0,2 + 0,3} = 8,6.$$

Таким образом, полученное значение $y_0 = 8,6$ и является результирующей переменной нечеткой модели.

При решении практических задач нечеткого моделирования могут одновременно использоваться несколько алгоритмов нечеткого вывода с целью получения наиболее адекватных результатов.

Порядок выполнения работы

1. Выполнить нечеткий логический вывод для модели системы нечеткого управления кондиционером воздуха в помещении (п.3.2).

Входные значения модели СНВ использовать по выданному варианту задания 1. Определить активные правила модели, затем алгоритмами Мамдани и Ларсена выполнить этапы фаззификации, агрегирования, активизации, аккумуляирования; методами дефаззификации найти точные значения на выходе модуля нечеткого управления. Провести исследования по поиску результирующих значений, применяя различные методы дефаззификации. Обосновать окончательное решение. Процедуру вывода провести в графическом и математическом виде.

2. Выполнить нечеткий логический вывод алгоритмом Такаги-Сугено (п.3.3.2) по выданному варианту задания 2. Выполнить этапы фаззификации, агрегирования, активизации, затем методом модифицированного варианта расчета четкого значения переменной выполнить процедуру дефаззификации и найти точные значения на выходе модуля нечеткого управления. Процедуру вывода провести в графическом и математическом виде.

Требования к содержанию и оформлению отчета

Отчет по практической работе должен содержать:

1. Название и цель работы.

2. Исходные данные варианта задания 1. Процедура нечеткого логического вывода в графическом и математическом виде. Таблица точных значений для алгоритмов Мамдани и Ларсена и методов различных методов дефаззификации.

3. Исходные данные варианта задания 2. Процедура нечеткого логического вывода по алгоритму Такаги-Сугено с графическим изображением и математическим представлением.

4. Выводы.

Варианты заданий

Задание 1. Варианты заданий представлены в табл. 2.5.

Таблица 2.5

№ варианта	1	2	3	4	5	6	7	8	9	10
x_1^*	6	7	8	9	14	15	19	26	27	28
x_2^*	0,5	-0,3	0,3	0,25	-0,2	-0,3	0,25	-0,4	0,1	0,5
№ варианта	11	12	13	14	15	16	17	18	19	20
x_1^*	6	7	8	9	14	15	19	26	27	28
x_2^*	0,4	-0,1	0,25	0,2	-0,25	0,1	-0,3	0,5	0,4	-0,2

Задание 2.

Модуль нечеткого управления построен на базе нечетких правил, фрагмент которой представлен ниже:

Правило 1:

ЕСЛИ x_1 = среднее (A_1) И x_2 = среднее (B_1) ТО $y_1 = 2 + x_1 + 2x_2$;

Правило 2:

ЕСЛИ x_1 = малое (A_2) И x_2 = высокое (B_2) ТО $y_1 = 2 + 2x_1 + 3x_2$.

Заданы термы нечетких подмножеств A_1, B_1, A_2, B_2 :

$$\mu_{A_1}(x) = \begin{cases} \frac{x-2}{2} & \text{при } 2 \leq x \leq 4, \\ \frac{6-x}{2} & \text{при } 4 < x \leq 6 \end{cases} \quad \mu_{A_2}(x) = \begin{cases} \frac{x-0}{2} & \text{при } 0 \leq x \leq 2, \\ \frac{4-x}{2} & \text{при } 2 < x \leq 4 \end{cases}$$

$$\mu_{B1}(y) = \begin{cases} \frac{y-2}{2} & \text{при } 2 \leq y \leq 4, \\ \frac{6-y}{2} & \text{при } 4 < y \leq 6 \end{cases} \quad \mu_{B2}(y) = \begin{cases} \frac{y-4}{4} & \text{при } 4 \leq y \leq 8, \\ \frac{12-y}{4} & \text{при } 8 < y \leq 12 \end{cases}$$

Входные четкие величины x^* и y^* , считываемые с датчиков, заданы в табл. 2.6. Необходимо определить четкое значение выходной переменной z_0 .

Таблица 2.6

№ варианта	1,11	2,12	3,13	4,14	5,15	6, 16	7, 17	8, 18	9, 19	10,20
x^*	3	2,5	3	2,2	2,5	3,5	3,5	3,2	2,7	2,5
y	4,2	4,3	5	5,5	5	4,5	5	5,5	4,7	5

Контрольные вопросы и задания

1. В чем состоит суть нечеткого управления, раскройте структуру модуля нечеткого управления.
2. Назовите этапы нечеткого логического вывода.
3. Чем отличается алгоритм Мамдани от алгоритма Ларсена? Приведите пример.
4. Поясните содержание алгоритма Такаги-Сугено.
5. Что такое фаззификация и дефаззификация?
6. Раскройте содержание методов дефаззификации.

Список литературы

1. Пегат А. Нечеткое моделирование и управление / А. Пегат; под ред. Ю. В. Тюменцева; перевод с английского А. Г. Подвесовского, Ю. В. Тюменцева. 3-е изд. М.: Лаборатория знаний, 2015. 801 с.
2. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2003. 736 с.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И.Д. Рудинского. М.: Горячая линия - Телеком, 2004. 452 с.
4. Ярушкина Н. Г. Основы теории нечетких и гибридных систем: учеб. пособие. М.: Финансы и статистика, 2004. 320с.
5. Флегонтов, А.В. Моделирование задач принятия решений при нечетких исходных данных: монография / А. В. Флегонтов, В. Б. Вилков, А. К. Черных. СПб.: Лань, 2020. 332 с.

Практическая работа № 3

ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ С УЧИТЕЛЕМ. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

1. Цель и задачи работы

Целью работы является закрепление знаний об обучении искусственных нейронных сетей, приобретение навыков применения алгоритма обучения нейронных сетей с учителем.

Задачами работы являются: формирование умений расчета выхода нейронной сети при прямом распространении входного примера; расчета среднеквадратичной ошибки нейросети, корректировки весовых коэффициентов синаптических связей нейросети по алгоритму обратного распространения ошибки.

2. Теоретические сведения

2.1. Модель искусственного нейрона

Искусственные нейронные сети (ИНС) – это направление искусственного интеллекта, в котором для обработки сигналов моделируются процессы, аналогичные происходящим в нейронах живых существ. ИНС – элементарные *нейроподобные* преобразователи информации (нейроны), соединенные каналами обмена информацией. Единицей обработки информации в ИНС является искусственный нейрон – абстрактный логический элемент, отражающий лишь те свойства биологического нейрона, которые связаны с переработкой информации. Модель искусственного нейрона представлена на рис. 3.1.

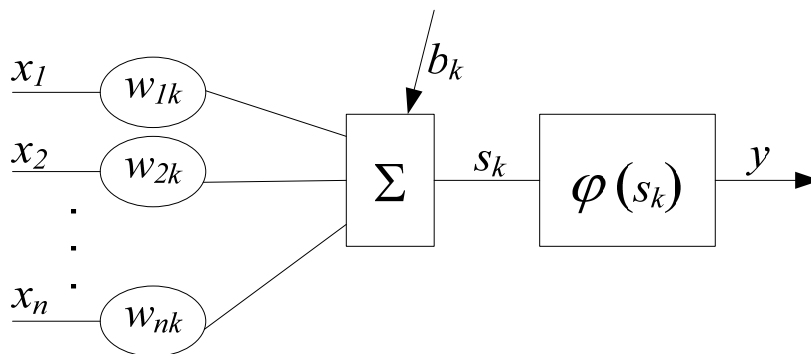


Рис. 3.1. Модель искусственного нейрона

В модели искусственного нейрона выделяют следующие элементы:

– вектор *входных* сигналов $X = \{x_1, x_2, \dots, x_n\}$, поступающих от других нейронов;

– вектор коэффициентов *синаптических связей* (*weight*) между нейронами $W = \{w_{1k}, w_{2k}, \dots, w_{nk}\}$;

– *сумматор* s_k , который складывает входные сигналы X , взвешенные относительно соответствующих синапсов W нейрона; на выходе сумматора формируется линейная комбинация входных воздействий;

– *функция активации* $\varphi(\cdot)$, ограничивающая амплитуду выходного сигнала нейрона;

– *выходной* сигнал нейрона y_k ;

– *пороговый* элемент b_k (*bias*), который является внешним параметром искусственного нейрона.

Математически функционирование искусственного нейрона k описывается парой уравнений:

$$\begin{aligned} s_k &= \sum_{i=1}^n w_{ik} \cdot x_i; \\ y_k &= \varphi(s_k + b_k). \end{aligned} \quad (3.1)$$

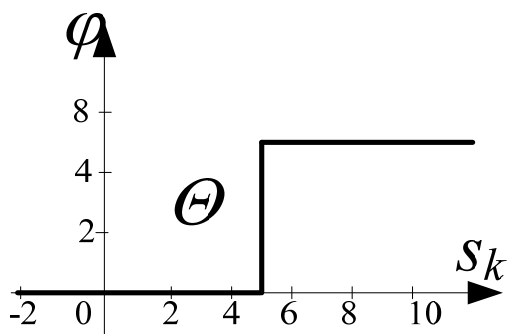
Формальное представление нейрона демонстрирует *двухтактную* работу нейрона: на первом такте вычисляется скалярное произведение векторов X и W ; на втором – рассчитывается выходной сигнал нейрона Y в соответствии с заданными порогом и функцией активации.

Функции активации $\varphi(\cdot)$, которые также называют нелинейным преобразователем, определяют выходной сигнал нейрона: в результате нормализованный диапазон амплитуд выхода нейрона должен находиться в интервале $[0, 1]$ или $[-1, 1]$. Выделяют следующие типы функций активации:

– *функция единичного скачка*, или *пороговая функция* (рис. 3.2), используется в классическом искусственном нейроне У. МакКалока и У. Питтса для обработки двоичных сигналов. К сожалению, такой вид функции активации не предоставляет ИНС достаточную гибкость при обучении и настройке на заданную задачу, так как функция является не дифференцируемой;

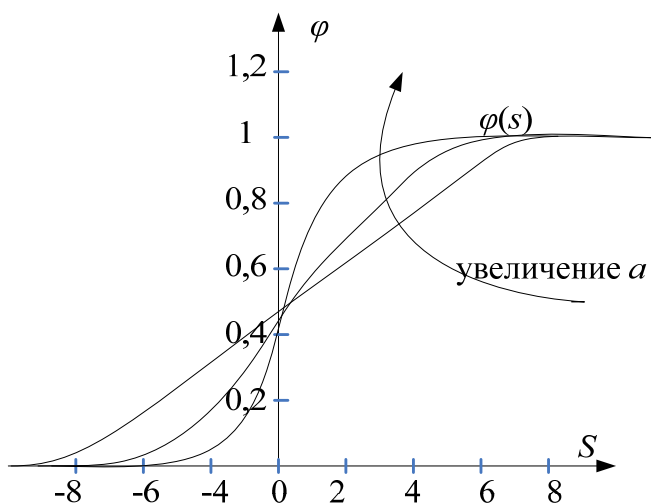
– *сигмоидальная функция* (рис. 3.3) дифференцируема на всей оси абсцисс, что широко используется во многих алгоритмах

обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем сильные, и предотвращает насыщение от сильных сигналов.



$$\varphi(s_k) = \begin{cases} 1, & \text{если } s_k \geq \Theta \\ 0, & \text{если } s_k < \Theta \end{cases}; \quad (3.2)$$

Рис. 3.2. Пороговая функция активации



$$\varphi(s_k) = \frac{1}{1+e^{-as_k}}; \quad (3.3)$$

a — коэффициент крутизны функции активации (параметр наклона):

$a \rightarrow \infty$ — вырождается в пороговую

$a \rightarrow 0$ — становится полой

Область значений (0;1)

$$\varphi'(s_k) = a \cdot \varphi(s_k) \cdot (1 - \varphi(s_k))$$

Рис. 3.3. Сигмоидальная функция активации

2.2. Обучение искусственных нейронных сетей

Искусственные нейросети — это множество искусственных нейронов, соединенных определенным образом между собой синаптическими соединениями. Как правило, активационные функции всех нейронов ИНС фиксированы, а синаптические веса являются параметрами ИНС и могут изменяться.

Цель обучения ИНС — настройка синаптической матрицы на заданное поведение. Выделяют две основные парадигмы обучения: «с учителем» и «без учителя».

При обучении «с учителем» роль учителя проявляется в виде источника знаний об объекте, представленного в виде пар входных и выходных координат. Одновременно с подачей входных координат обучающего вектора «учитель» передает нейросети желаемый отклик, соответствующий данному входному вектору. В процессе обучения параметры ИНС корректируются на основе значений сигнала ошибки на выходе сети, т.е. разности между желаемым сигналом и текущим откликом сети. Данная парадигма обучения используется при решении задач классификации, аппроксимации и прогнозирования.

Обучение «без учителя», или *самообучение*, использует значения входных координат обучающего вектора и не требует знания правильных ответов для примеров обучающей выборки. В процессе самообучения определяется внутренняя структура данных, обнаруживаются корреляции между входными примерами, что позволяет разбить примеры по категориям. Данная парадигма обучения применяется при решении задач кластеризации.

Форма обучения ИНС «с учителем» рассматривается как задача минимизации эмпирической ошибки E между желаемым выходом d_j и фактической реакцией нейросети y_j^N ($j = \overline{1, n_N}$) по всем нейронам выходного слоя.

$$E = \frac{1}{2} \sum_{j=1}^{n_N} (y_j^N - d_j)^2.$$

Большинство методов обучения на основе *коррекции ошибок* используют идею Хебба, которая заключается том, что *синаптический вес увеличивается, если 2 нейрона (источник и приемник) активизированы*; т.е. увеличение весового коэффициента Δw_{ij} зависит от уровней возбуждения нейронов. На основе идеи Хебба и формулы вычисления вектора градиента в методе наискорейшего спуска формулируется алгоритм минимизации среднеквадратической ошибки, известный также под названием *дельта-правило* (3.4); переходя от итерации (t) к $(t + 1)$ -й, алгоритм выполняет коррекцию весовых коэффициентов:

$$W(t + 1) = W(t) - \eta \cdot \nabla E(W), \quad (3.4)$$

где $W(t + 1)$, $W(t)$ – вектор весов синаптических связей между нейронами на предыдущей и текущей итерациях обучения; $\nabla E(W)$ – вектор градиента функции ошибки в точке $W(t)$;

η – положительная константа, называемая параметром скорости обучения ($0 < \eta \leq 1$), которая оказывает влияние на скорость сходимости алгоритма: если параметр η мал, то алгоритм замедляется, и траектория изменения $W(t)$ соответствует гладкой кривой; напротив, если параметр η велик, то алгоритм ускоряется, и траектория изменения $W(t)$ принимает зигзагообразный вид.

Обучение ИНС проводится с использованием обучающей выборки, представляющей совокупность пар векторов входных X и желаемых выходных D значений. Входной вектор нейронной сети имеет вид: $X = [x_1, x_2, \dots, x_m]^p$, где m – число признаков, p – количество примеров обучающего множества. Выходной вектор желаемого отклика сети $D = [d_1, d_2, \dots, d_r]^p$ имеет размерность r , соответствующую количеству выходных координат p .

На рис. 3.4 представлена структура многослойного персептрона – многослойной нейронной сети прямого распространения с последовательными связями, в которых входной сигнал распространяется по сети от слоя к слою. Количество слоев нейросети равно N ; в каждом слое разное число нейронов n_k , $k = 1, \dots, N$. На вход нейросети подаются внешние сигналы $x_i, i = \overline{1, n_0}$. Все выходы нейронов k -го слоя подаются на каждый нейрон $(k+1)$ -го слоя.

В обозначении выходных сигналов нейронов y_i^k верхний индекс k – номер слоя, а нижний индекс i – номер нейрона в этом слое. Весовой коэффициент синаптической связи w_{ij}^k соединяет i -й нейрон $(k-1)$ -го слоя с j -м нейроном слоя (k) .

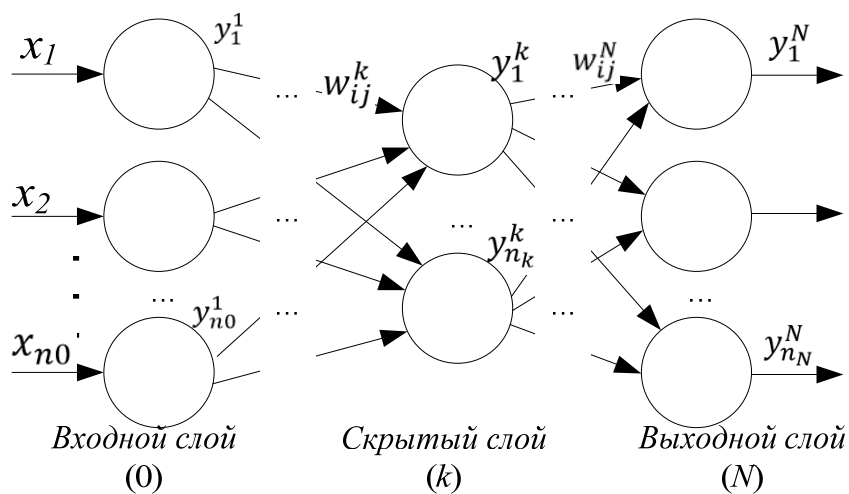


Рис. 3.4. Структура многослойного персептрона

В многослойных ИНС желаемые выходные значения нейронов всех слоев, кроме последнего, неизвестны, и поэтому трех- или более слойный персептрон невозможно обучить, руководствуясь только величинами ошибок на выходах ИНС.

Наиболее известным и чаще всего применяемым для многослойных ИНС алгоритмом обучения с учителем является *метод обратного распространения ошибки* – итеративный градиентный алгоритм обучения, который используется с целью минимизации функции ошибки многослойных ИНС, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Функция ошибки представляет собой сумму квадратов рассогласования (ошибки) желаемого и фактического выхода ИНС. При вычислении элементов вектора-градиента используется активационная функция сигмоидального типа.

2.3. Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки (ОРО) (*error back propagation algorithm*) в настоящее время считается одним из наиболее эффективных алгоритмов обучения и определяет стратегию подбора весов многослойной нейронной сети с применением градиентных методов оптимизации. Алгоритм ОРО представляет собой развитие обобщенного *дельта-правила* и является алгоритмом градиентного спуска, минимизирующим суммарную среднеквадратичную ошибку. В соответствии с дельта-правилом корректировка весов на текущем шаге обучения осуществляется в направлении антиградиента функции ошибки.

Обучение ИНС с использованием алгоритма ОРО проводится в несколько этапов:

Шаг 1. Начальная инициализация весов ИНС w_{ij} малыми случайными значениями.

Прямой проход по ИНС

Шаг 2. Выбирается очередной обучающий пример (X, D) из обучающего множества. Вектор X предъявляется на вход нейросети.

Шаг 3. Входные нейроны ИНС используются для разветвления сигнала X и передают сигналы на нейроны вычислительного слоя, в которых, во-первых, осуществляется взвешенное суммирование,

а во-вторых, производится нелинейное преобразование с помощью функции активации:

$$s_j^k = \sum_{i=1}^{n_{k-1}} w_{ij}^k \cdot x_i^k = \sum_{i=1}^{n_{k-1}} w_{ij}^k \cdot y_i^{k-1}, \quad (3.5)$$

где s_j^k – взвешенная сумма входных сигналов j -го нейрона в слое k , а y_i^{k-1} – выход i -го нейрона в предыдущем слое ($k-1$). Выходной сигнал j -го нейрона k -го слоя:

$$y_j^k = f(s_j^k). \quad (3.6)$$

В общем случае рекуррентное соотношение, определяющее выход нейрона в произвольном слое, записывается следующим образом:

$$y_j^k = f\left(\sum_{i=1}^{n_{k-1}} w_{ij}^k \cdot y_i^{k-1} + b_j^k\right). \quad (3.7)$$

Таким образом, осуществляется последовательное *прямое распространение* входного обучающего примера X по нейронной сети.

Обратный проход по ИНС

Шаг 4. Определяется суммарная среднеквадратичная ошибка E по всем нейронам выходного слоя нейросети, которая вычисляется как разность между *требуемым* (целевым, желаемым, эталонным) выходом D и *реальным* (фактическим) выходом Y последнего N -го слоя нейросети:

$$E = \frac{1}{2} \sum_{j=1}^{n_N} (y_j^N - d_j)^2. \quad (3.8)$$

Шаг 5. Обучение ИНС основано на адаптивной коррекции весовых коэффициентов $w_{ij}^k(t)$ таким образом, чтобы минимизировать значение среднеквадратичной ошибки. Минимизация функции ошибки $E(w)$ проводится на основе дельта-правила (3.4), согласно которому подстройка весовых коэффициентов осуществляется по формуле:

$$w_{ij}^k(t+1) = w_{ij}^k(t) - \eta \cdot \frac{\partial E}{\partial w_{ij}^k};$$

$$\Delta w_{ij}^k = -\eta \cdot \frac{\partial E}{\partial w_{ij}^k}, \quad (3.9)$$

где η – коэффициент скорости обучения, определяющий величину шага коррекции, $0 < \eta \leq 1$; t – номер итерации обучения.

Поскольку суммарная среднеквадратичная ошибка E зависит от результатов работы ИНС (3.8), а выход ИНС формируется с помощью нелинейной функции активации (3.6) от взвешенной суммы входных сигналов $f(s_j^k) = f(\sum_{i=1}^{N_{k-1}} w_{ij}^k \cdot x_i^k)$, то частная производная функции ошибки $\frac{\partial E}{\partial w_{ij}^k}$ записывается как производная сложной функции в виде:

$$\frac{\partial E}{\partial w_{ij}^k} = \frac{\partial E}{\partial y_j^k} \cdot \frac{\partial y_j^k}{\partial s_j^k} \cdot \frac{\partial s_j^k}{\partial w_{ij}^k}. \quad (3.10)$$

В соответствии с формулой (3.4) взвешенная сумма входных сигналов s_j^k является функцией от синаптических весов; а выражение частной производной $\frac{\partial s_j^k}{\partial w_{ij}^k}$ равно значению входного сигнала текущего слоя многослойной нейронной сети, который одновременно является выходом нейрона предыдущего слоя:

$$\frac{\partial s_j^k}{\partial w_{ij}^k} = x_i^k = y_i^{k-1}. \quad (3.11)$$

Отсюда следует, что,

$$\frac{\partial E}{\partial w_{ij}^k} = \frac{\partial E}{\partial s_j^k} \cdot x_i^k. \quad (3.12)$$

Если ввести обозначение $\delta_j^k = \frac{\partial E}{\partial s_j^k}$, то образуется следующее равенство:

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_j^k \cdot x_i^k.$$

$$\delta_j^k = \frac{\partial E}{\partial s_j^k} = -(y_j^k - d_j) \cdot y'(s) = -(y_j^k - d_j) \cdot \frac{\partial y_j^k}{\partial s_j^k}. \quad (3.13)$$

Подставив выражение (3.12) в (3.8), а также используя равенство (3.10), запишем выражение для коррекции весов (3.8) в следующем виде:

$$\Delta w_{ij}^k = -\eta \cdot \delta_j^k \cdot y_i^{k-1}. \quad (3.14)$$

По этой формуле и проводится коррекция весовых коэффициентов, начиная с выходного слоя сети и по направлению ко входу.

Шаг 6. Проверка критерия останова алгоритма обучения. Если выполняется хотя бы один из следующих критериев, то обучение останавливается:

- ошибка обучения достигла заранее заданной величины;
- ошибка обучения не уменьшается или уменьшается незначительно;
- ошибка обобщения начинает возрастать, указывая на начало переобучения.

Если же критерий останова алгоритма обучения не выполняется, то осуществляется переход к шагу 2 и проводится следующая итерация обучения для каждой пары обучающего множества.

По окончании работы алгоритма ОРО нейронная сеть считается обученной (натренированной) и готовой к применению.

3. Методика выполнения работы

3.1. Прямое распространение входного примера по искусственной нейронной сети. Расчет среднеквадратичной ошибки

Пример 1. Имеется двухвходовая однослойная ИНС прямого распространения (однослойный персептрон) (рис. 3.5), которая состоит из двух нейронов.

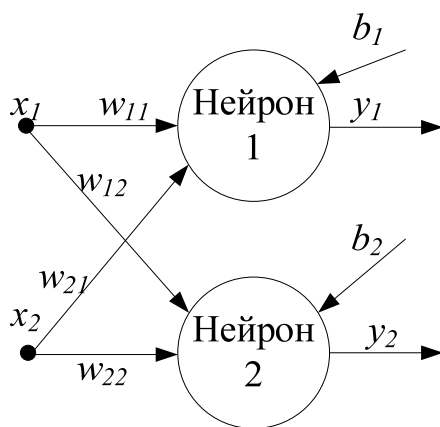


Рис. 3.5. Однослойный персептрон

Входом нейросети является пример X : $x_1 = 1$, $x_2 = -1$. Целевые (желаемые) значения выходов ИНС равны $d_1^0 = 0,4$, $d_2^0 = 0,25$. Пороговые значения (смещения) для нейронов равны $b_1 = b_2 = 0,2$. Весовые коэффициенты синаптических связей ИНС w_{ij} показаны в табл. 3.1.

Таблица 3.1

Таблица входных значений и весов нейросети

Нейрон Вход	Нейрон 1	Нейрон 2
$x_1 = 1$	$w_{11} = 1$	$w_{12} = 3$
$x_2 = -1$	$w_{21} = 2$	$w_{22} = -1$

Множество весовых коэффициентов в виде матрицы синаптических связей имеет вид: $W = \begin{vmatrix} 1 & 3 \\ 2 & -1 \end{vmatrix}$.

В качестве нелинейного преобразователя используется сигмоидальная функция активации $f(s) = \frac{1}{1+e^{-as}}$ с коэффициентом крутизны $a = 1$.

Рассматривается процесс прямого распространения входного примера X по нейросети.

Взвешенная сумма входных сигналов и выходной сигнал 1-го нейрона равны:

$$s_1 = x_1 \cdot w_{11} + x_2 \cdot w_{21} + b_1 = 1 \cdot 1 + (-1) \cdot 2 + 0,2 = -0,8;$$

$$y_1 = f(s_1) = \frac{1}{1 + e^{-as_1}} = \frac{1}{1 + e^{0,8}} = 0,31.$$

Взвешенная сумма входных сигналов и выходной сигнал 2-го нейрона равны:

$$s_2 = x_1 \cdot w_{12} + x_2 \cdot w_{22} + b_2 = 1 \cdot 3 + (-1) \cdot (-1) + 0,2 = 4,2;$$

$$y_2 = f(s_2) = \frac{1}{1 + e^{-as_2}} = \frac{1}{1 + e^{-4,2}} = 0,985.$$

Определяются эталонные (целевые) значения выходов нейросети, преобразованные нелинейной функции возбуждения:

$$d_1 = f(d_1^0) = \frac{1}{1 + e^{-d_1^0}} = \frac{1}{1 + e^{-0,4}} = 0,59;$$

$$d_2 = f(d_2^0) = \frac{1}{1 + e^{-d_2^0}} = \frac{1}{1 + e^{-0,25}} = 0,56.$$

Среднеквадратичная ошибка ИНС (3.7) равна:

$$E = \frac{1}{2} \cdot ((y_1 - d_1)^2 + (y_2 - d_2)^2) = \frac{1}{2} \cdot ((0,31 - 0,59)^2 + (0,985 - -0,56)^2) = 0,129.$$

3.2. Обучение искусственной нейронной сети методом обратного распространения ошибки

Пример 2. Рассматривается процесс обучения многослойного персептрона методом обратного распространения ошибки на примере сложения чисел $1+2=3$.

В соответствии с условием задачи определяется структура многослойного персептрона (рис. 3.6): ИНС имеет два входа, один выход. Пусть ИНС также содержит дополнительный скрытый слой, состоящий из двух нейронов.

Входом ИНС является вектор $X = [x_1; x_2] = [1; 2]$, а целевое значение выхода нейросети равно $d^0 = 3$. Таким образом, обучающее множество для сложения чисел представляет собой примеры, состоящие из двух входов и одного выхода.

Вводятся следующие обозначения: $Y_{(t)}^1, Y_{(t)}^2$ – выходные векторы 1-го (скрытого) и 2-го (выходного) слоев нейросети на t -й итерации обучения; $W_{(t)}^1$ и $W_{(t)}^2$ – синаптические матрицы весовых коэффициентов 1-го и 2-го слоев на t -й итерации обучения:

$$W_{(1)} = \begin{bmatrix} 0,1 & 0,2 \\ 0,1 & 0,2 \end{bmatrix}; W_{(1)}^2 = \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix}.$$

Нейроны возбуждаются сигмоидальной функцией активации с единичным коэффициентом крутизны $a = 0$.

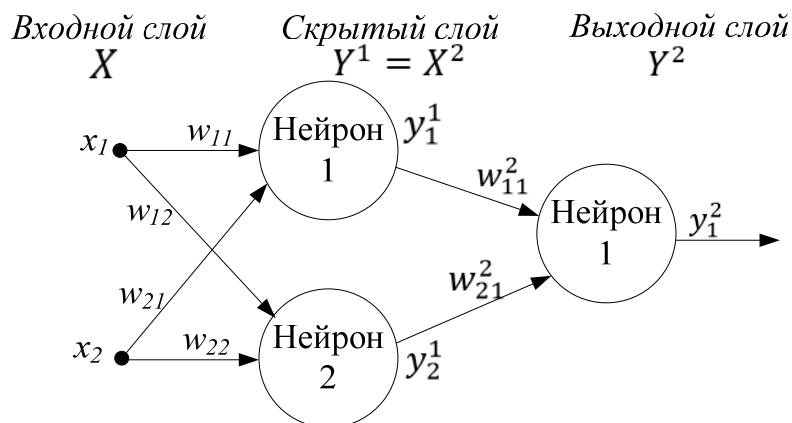


Рис. 3.6. Нейронная сеть с одним скрытым слоем

Обучение нейросети включает следующие этапы:
 1) осуществляется прямое распространение входного примера по ИНС (*прямой проход*, первая итерация $t = 1$); 2) определяется среднеквадратичная ошибка нейросети; 3) проводится корректировка коэффициентов синаптических связей в направлении, обратном прямому распространению входных сигналов (*обратный проход*); 4) рассчитывается выход нейросети путем прямого распространения входного примера по ИНС (*прямой проход*, вторая итерация $t = 2$); 5) определяется среднеквадратичная ошибка нейросети; уменьшение ошибки будет свидетельствовать о том, что обучение сети прошло успешно.

1 итерация, первый прямой проход

Шаг 1. Осуществляется прямое распространения входного примера по ИНС.

Рассчитывается вектор взвешенных входов нейронов 1-го скрытого слоя:

$$S_{(t=1)}^1 = [s_{1(1)}^1; s_{2(1)}^1] = X_{(1)}^1 \cdot W_{(1)}^1 = [1; 2] \cdot \begin{bmatrix} 0,1 & 0,2 \\ 0,1 & 0,2 \end{bmatrix} = [0,3; 0,6].$$

Выход скрытого слоя ИНС $Y_{(1)}^1$, преобразованный функцией активации, равен:

$$Y_{(1)}^1 = [y_{1(1)}^1; y_{2(1)}^1] = [f(s_{1(1)}^1); f(s_{2(1)}^1)];$$

$$Y_{(1)}^1 = [f(0,3); f(0,6)] = \left[\frac{1}{1 + e^{-0,3}}; \frac{1}{1 + e^{-0,6}} \right] = [0,574; 0,646].$$

Для дальнейшего распространения входного примера выходной вектор скрытого слоя $Y_{(1)}^1$ принимают в качестве входного вектора следующего (выходного) слоя нейронов $Y_{(1)}^1 = X_{(1)}^2$, т.е. $y_{1(1)}^1 = x_{1(1)}^2, y_{2(1)}^1 = x_{2(1)}^2$. Тогда вектор взвешенных входов нейронов второго (выходного) слоя равен:

$$S_{(1)}^2 = [s_{1(1)}^2; s_{2(1)}^2] = X_{(1)}^2 \cdot W_{(1)}^2 = [0,574; 0,646] \cdot \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} = 0,179.$$

Выход нейросети равен:

$$Y_{(1)}^2 = f(S_{(1)}^2) = f(0,179) = \frac{1}{1 + e^{-0,179}} = 0,545.$$

Шаг 2. Определение среднеквадратичной ошибки ИНС.

Целевое (желаемое) значение ИНС $d^0 = 3$, преобразованное нелинейной функцией возбуждения для обучающего примера, равно:

$$d = f(d^0) = f(3) = \frac{1}{1 + e^{-3}} = 0,953.$$

Среднеквадратичная ошибка (3.7) на первой итерации работы нейросети равна:

$$E_{(1)} = \frac{1}{2} \cdot (y_{(1)}^2 - d)^2 = \frac{1}{2} \cdot (0,545 - 0,953)^2 = 0,083.$$

Поскольку желаемые и фактические значения выходов ИНС не совпадают, необходима коррекция (адаптация) весовых коэффициентов синаптических связей нейросети.

1 итерация, обратный проход

Шаг 3. Коррекция коэффициентов синаптических связей в направлении, обратном прямому распространению входных сигналов.

Проводится анализ влияния изменения весовых коэффициентов связей на величину среднеквадратичной ошибки нейронной сети, начиная с выходного слоя сети и по направлению к входу:

Шаг 3.1. Определение скорости изменения ошибки нейросети при изменении значения нейрона выходного (второго) слоя $y_{(1)}^2$ через производную функции ошибки:

$$Ea^2 = \frac{\partial E}{\partial y_{(1)}^2} = y_{(1)}^2 - d = 0,545 - 0,953 = -0,407.$$

Поскольку скорость изменения ошибки Ea^2 имеет отрицательный знак, то целью обучения нейросети является увеличение значения ее выхода $y_{(1)}^2$.

Шаг 3.2. Расчет чувствительности изменения ошибки нейросети при изменении средневзвешенного входа выходного слоя путем нахождения произведения скорости изменения ошибки при изменении выхода сети Ea^2 и скорости изменения выходного нейрона:

$$Eq^2 = \frac{\partial E}{\partial s_{(1)}^2} = \frac{\partial E}{\partial y_{(1)}^2} \cdot \frac{\partial y_{(1)}^2}{\partial s_{(1)}^2} = Ea^2 \cdot \frac{\partial y_{(1)}^2}{\partial s_{(1)}^2} = EA^2 \cdot y_{(1)}^2 \cdot (1 - y_{(1)}^2);$$

$$Eq^2 = -0,407 \cdot 0,545 \cdot (1 - 0,545) = -0,101.$$

Значение Eq^2 показывает, что скорость изменения ошибки нейросети при изменении средневзвешенного входа выходного нейрона существенно ниже скорости реакции ИНС на изменение ее выхода.

При обратном продвижении ко входу определяется влияние каждого из входных нейронов выходного слоя на ошибку нейросети:

$$\frac{\partial E}{\partial w_{ij(1)}^2} = \frac{\partial E}{\partial y_{(1)}^2} \cdot \frac{\partial y_{(1)}^2}{\partial s_{(1)}^2} \cdot \frac{\partial s_{(1)}^2}{\partial w_{ij(1)}^2} = Eq^2 \cdot \frac{\partial s_{(1)}^2}{\partial w_{ij(1)}^2}.$$

Для рассматриваемой нейронной сети вектор значений для корректировки синаптической матрицы второго (выходного) слоя ИНС EW^2 равен:

$$EW^2 = [EW_{1(1)}^2; EW_{2(1)}^2]^T = \left[\frac{\partial E}{\partial w_{1(1)}^2}; \frac{\partial E}{\partial w_{2(1)}^2} \right]^T = [Eq^2 \cdot y_{1(1)}^1; Eq^2 \cdot y_{2(1)}^1]^T;$$

$$EW^2 = [-0,101 \cdot 0,574; -0,101 \cdot 0,646]^T = [-0,058; -0,065]^T,$$

где выражение $[a; b]^T$ является результатом транспонирования матрицы.

Шаг 3.3. Определяется скорость изменения ошибки Ea^1 нейросети при изменении активности нейронов *скрытого* слоя:

$$Ea_i^1 = \frac{\partial E}{\partial y_{i(1)}^1} \cdot \frac{y_{i(1)}^1}{\partial x_{i(1)}^2} = Eq^2 \cdot W_{i(1)}^1;$$

$$Ea_i^1 = [-0,101 \cdot 0,2; -0,101 \cdot 0,1] = [-0,020; -0,010].$$

Шаг 3.4. Расчет весовых коэффициентов синаптической матрицы скрытого (первого) слоя EW^1 :

$$\begin{aligned} EW^1 &= \begin{bmatrix} EW_{11}^1 & EW_{12}^1 \\ EW_{21}^1 & EW_{22}^1 \end{bmatrix} = \begin{bmatrix} Eq_1^1 \cdot x_{1(1)} & Eq_1^1 \cdot x_{1(1)} \\ Eq_1^1 \cdot x_{2(1)} & Eq_1^1 \cdot x_{2(1)} \end{bmatrix} = \\ &= \begin{bmatrix} Ea_1^1 \cdot y_{1(1)}^1 \cdot (1 - y_{1(1)}^1) \cdot x_{1(1)} & Ea_2^1 \cdot y_{2(1)}^1 \cdot (1 - y_{2(1)}^1) \cdot x_{1(1)} \\ Ea_1^1 \cdot y_{1(1)}^1 \cdot (1 - y_{1(1)}^1) \cdot x_{2(1)} & Ea_2^1 \cdot y_{2(1)}^1 \cdot (1 - y_{2(1)}^1) \cdot x_{2(1)} \end{bmatrix}; \\ EW^1 &= \begin{bmatrix} -0,02 \cdot 0,574 \cdot (1 - 0,574) \cdot 1 & -0,01 \cdot 0,646 \cdot (1 - 0,646) \cdot 1 \\ -0,02 \cdot 0,574 \cdot (1 - 0,574) \cdot 2 & -0,01 \cdot 0,646 \cdot (1 - 0,646) \cdot 2 \end{bmatrix} = \\ &= \begin{bmatrix} -0,005 & -0,002 \\ -0,010 & -0,004 \end{bmatrix}. \end{aligned}$$

Шаг 3.5. Корректировка весов синаптической матрицы скрытого слоя с учетом, что скорость обучения равна единице ($\eta = 1$):

$$W_{(2)}^1 = W_{(1)}^1 - EW = \begin{bmatrix} 0,1 & 0,2 \\ 0,1 & 0,2 \end{bmatrix} - \begin{bmatrix} -0,005 & -0,002 \\ -0,010 & -0,004 \end{bmatrix} = \begin{bmatrix} 0,105 & 0,202 \\ 0,110 & 0,204 \end{bmatrix}.$$

Корректировка значений весовых коэффициентов нейронов выходного слоя:

$$W_{(2)}^2 = W_{(1)}^2 - EW^2 = \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} - \begin{bmatrix} -0,058 \\ -0,065 \end{bmatrix} = \begin{bmatrix} 0,258 \\ 0,165 \end{bmatrix}.$$

2 итерация

Шаг 4. Прямое распространения входного примера по ИНС (*второй прямой проход*).

Второй прямой проход выполняется с целью определения качества обучения нейронной сети, то есть насколько ИНС близка к поставленной цели – правильному суммированию цифр 1 и 2.

Вектор взвешенных входов и выходов нейронов скрытого слоя равны:

$$S_{(2)}^1 = [s_{1(2)}^1; s_{2(2)}^1] = X_{(2)}^1 \cdot W_{(2)}^1;$$

$$Y_{(2)}^1 = [y_{1(2)}^1; y_{2(2)}^1] = [f(s_{1(2)}^1); f(s_{2(2)}^1)];$$

$$S_{(2)}^1 = [1; 2] \cdot \begin{bmatrix} 0,105 & 0,202 \\ 0,110 & 0,204 \end{bmatrix} = [0,325; 0,610].$$

$$Y_{(2)}^1 = [f(0,325); f(0,61)] = \left[\frac{1}{1 + e^{-0,325}}; \frac{1}{1 + e^{-0,61}} \right] = [0,581; 0,648].$$

Вектор взвешенных входов и выходов нейронов выходного слоя:

$$S_{(2)}^2 = [s_{1(2)}^2; s_{2(2)}^2] = Y_{(2)}^1 \cdot W_{(2)}^2 = [0,581; 0,648] \cdot \begin{bmatrix} 0,258 \\ 0,165 \end{bmatrix} = 0,257.$$

Выход нейросети:

$$y_{(2)}^2 = f(S_{(2)}^2) = f(0,257) = \frac{1}{1 + e^{-0,257}} = 0,564.$$

Шаг 5. Определение среднеквадратичной ошибки ИНС.

Целевое (желаемое) значение ИНС $d^0 = 3$, преобразованное нелинейной функцией возбуждения для обучающего примера, равно:

Среднеквадратичная ошибка ИНС на второй итерации работы нейросети равна:

$$E_{(2)} = \frac{1}{2} \cdot (y_{(2)}^2 - d)^2 = \frac{1}{2} \cdot (0,564 - 0,953)^2 = 0,076.$$

Сравнение среднеквадратичных ошибок ИНС, определенных на 1-й и 2-й итерациях обучения ИНС, позволяет сделать вывод о том, что обучение ИНС было успешным: ИНС уже более точно выполняет свою функцию, так как среднеквадратичная ошибка ИНС уменьшилась с 0,083 до 0,076.

Порядок выполнения работы

1. Для выданного варианта задания 1 решить задачу прямого распространения входного примера по искусственной нейронной сети (п.3.1); рассчитать среднеквадратичную ошибку; в качестве функции активации принять сигмоиду с единичным коэффициентом крутизны.

2. Для выданного варианта задания 2 решить задачу обучения искусственной нейронной сети, используя алгоритм обратного распространения ошибки (п.3.2); в качестве функции активации принять сигмоиду с единичным коэффициентом крутизны, скорость обучения равна 1.

При решении задачи выполнить 2 итерации: на первой итерации определить выход нейронной сети путем распространения входного примера по ИНС; определить среднеквадратичную ошибку нейросети; затем провести корректировку коэффициентов синаптических связей в направлении, обратном прямому распространению входных сигналов; на второй итерации вновь рассчитать выход нейросети путем прямого распространения входного примера по ИНС и определить среднеквадратичную ошибку нейросети. Сделать вывод о качестве обучения ИНС по уменьшению ошибки.

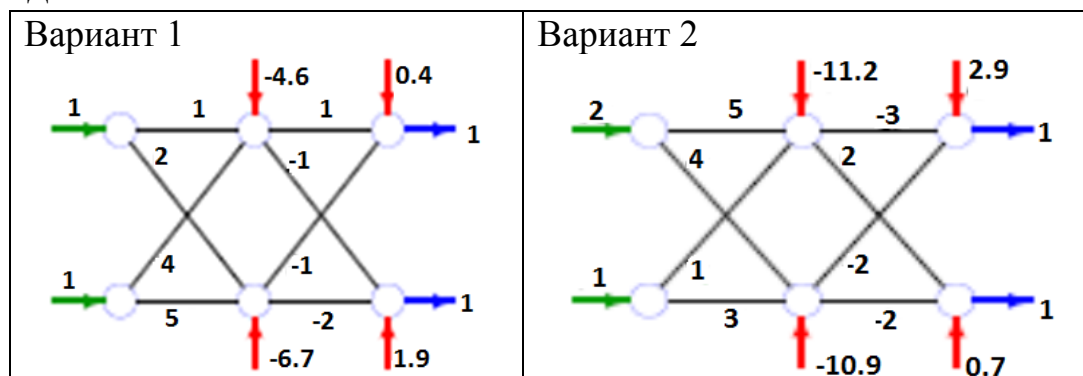
Требования к содержанию и оформлению отчета

Отчет по практической работе должен содержать:

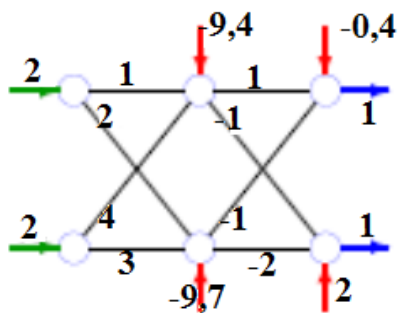
1. Название, цель работы, исходные данные варианта задания.
2. Ход решения задачи прямого распространения входного примера по искусственной нейронной сети; найденная среднеквадратичная ошибка.
3. Ход решения задачи обучения нейронной сети методом обратного распространения ошибки. Выполненные две итерации обучения нейронной сети; по каждой итерации представить результаты распространения входного примера по ИНС; рассчитанные среднеквадратичные ошибки; показать результаты корректировки коэффициентов синаптических связей. Вывод о качестве обучения ИНС по уменьшению ошибки.
4. Выводы по выполненной работе.

Варианты заданий

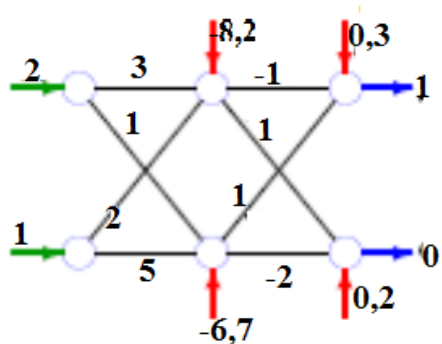
Задание 1



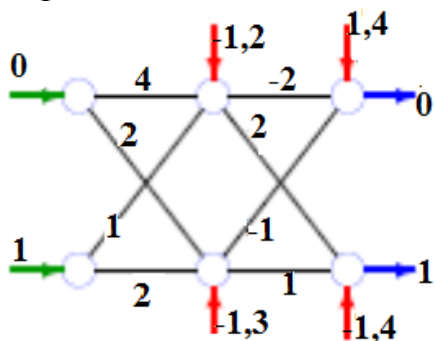
Вариант 3



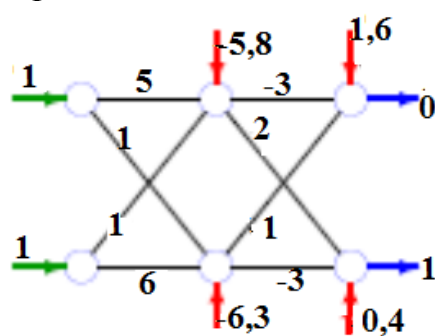
Вариант 4



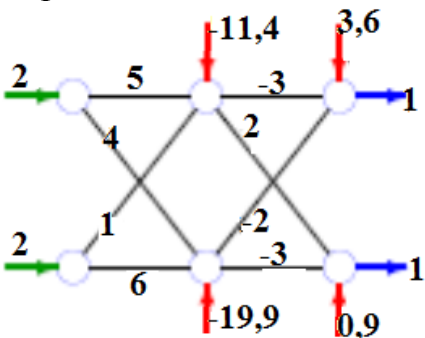
Вариант 5



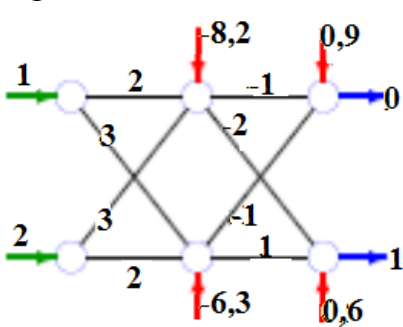
Вариант 6



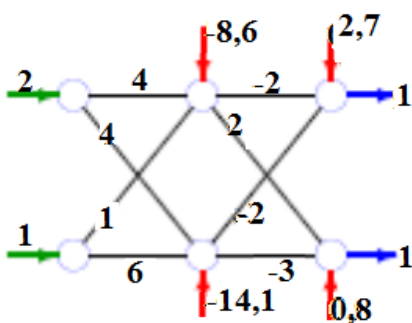
Вариант 7



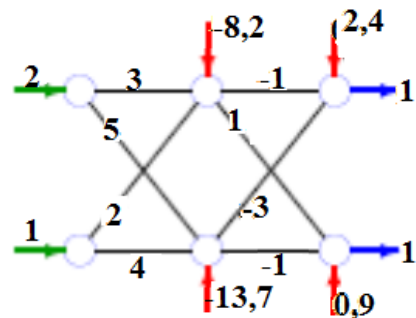
Вариант 8



Вариант 9

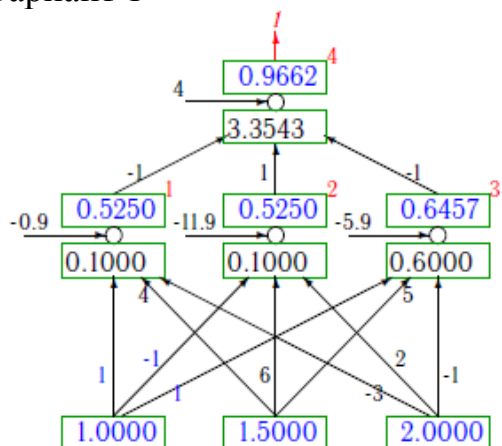


Вариант 10

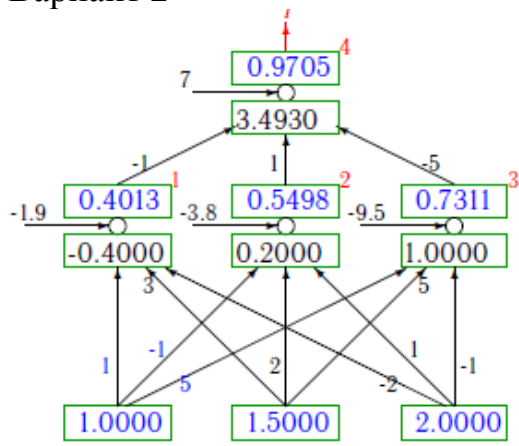


Задание 2

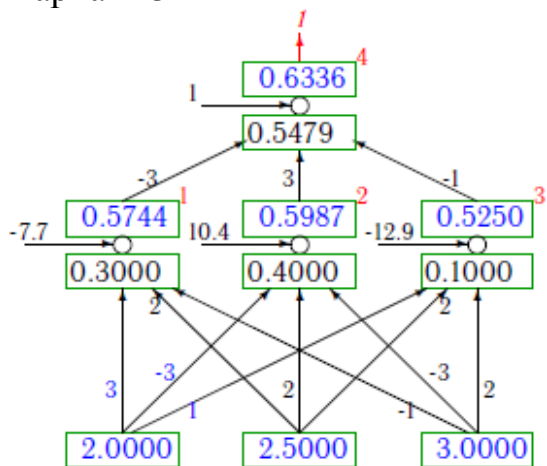
Вариант 1



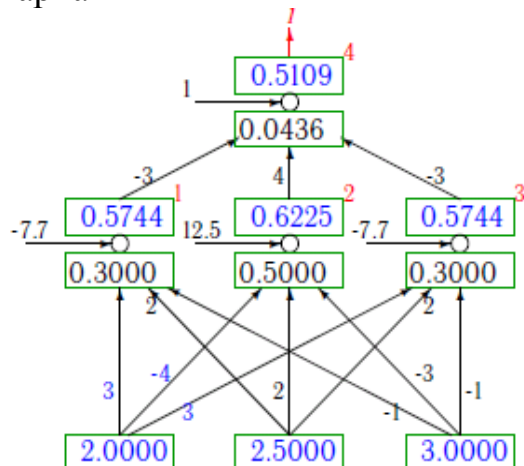
Вариант 2



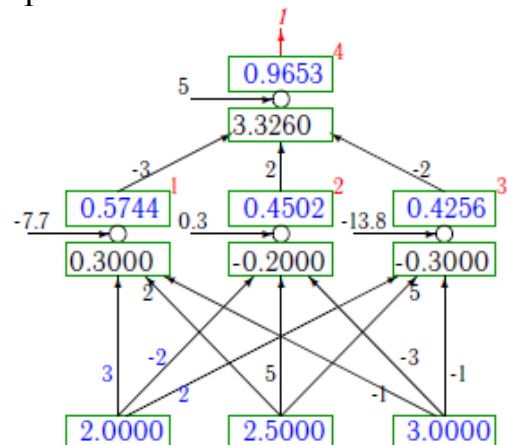
Вариант 3



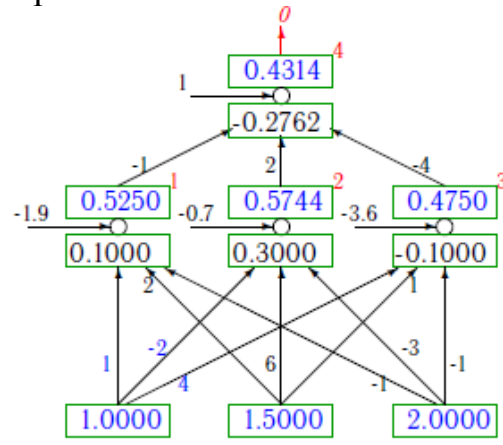
Вариант 4



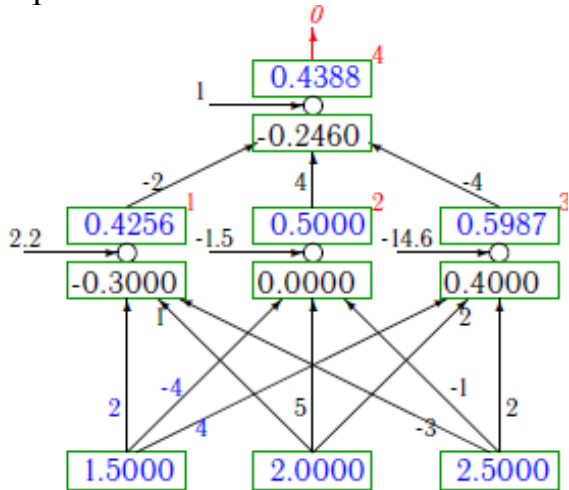
Вариант 5



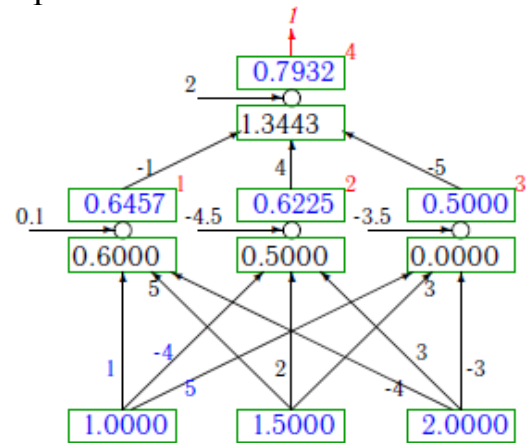
Вариант 6



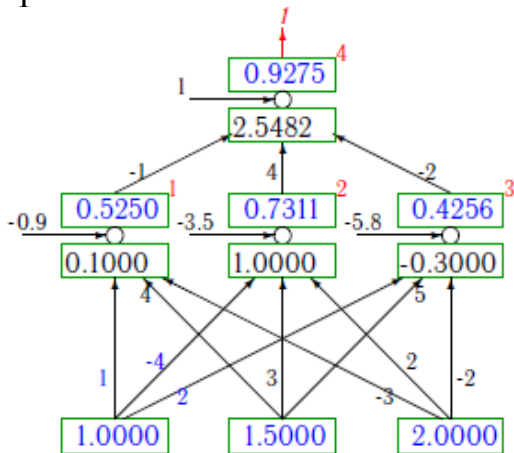
Вариант 7



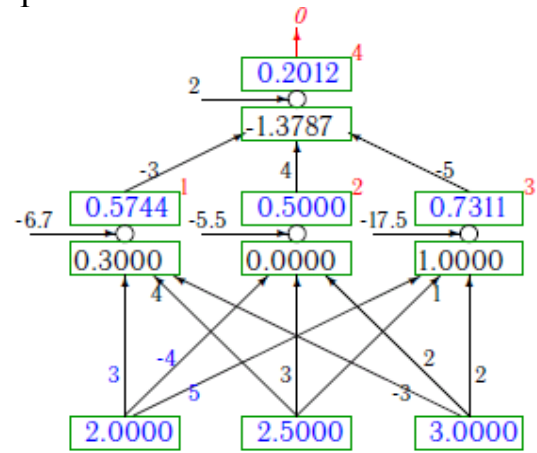
Вариант 8



Вариант 9



Вариант 10



Контрольные вопросы и задания

1. Какие существуют парадигмы обучения нейронных сетей?
2. Что представляет собой модель искусственного нейрона?
3. Чем отличаются пороговая и сигмоидальная функция активации?
4. В чем заключается алгоритм обратного распространения ошибки?
5. Запишите выражение для коррекции весовых коэффициентов нейронной сети.

Список литературы

1. Хайкин С. Нейронные сети: полный курс, 2-е изд.: Пер. с англ. М. Издательский дом «Вильямс», 2006. 1104 с.
2. Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям: учеб. пособие. 2-е изд., перераб. и доп. СПб.: Питер, 2010. 704 с.
3. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы = Sieci neuronowe algorytmy genetyczne i systemy rozmyte / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польск. И. Д. Рудинского. М.: Горячая линия-Телеком, 2008. 383 с.
4. Круглов В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. 2-е изд. М.: Горячая линия-Телеком, 2002. 382 с.
5. Матвеев М. Г. Модели и методы искусственного интеллекта. Применение в экономике: учеб. пособие / М. Г. Матвеев, А. С. Свиридов, Н. А. Алейникова. М.: Финансы и статистика: ИНФРА-М, 2008. 446 с.
6. Анализ данных и процессов: учеб. пособие / А. А. Барсегян, М.С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2009. 512 с.
7. Васильев В. И., Ильясов Б. Г. Интеллектуальные системы управления. Теория и практика: учебное пособие. М.: Радиотехника, 2009. 392 с.
8. Ростовцев В. С. Искусственные нейронные сети: учебник / В.С. Ростовцев. СПб.: Лань, 2019. 216 с.

Практическая работа № 4

ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ БЕЗ УЧИТЕЛЯ. СЕТИ КОХОНЕНА

1. Цель и задачи работы

Целью работы является закрепление знаний об обучении искусственных нейронных сетей без учителя, приобретение навыков применения алгоритма самоорганизующегося обучения.

Задачами работы являются приобретение умений проектирования сети Кохонена, навыков выполнения процедуры нейросетевой кластеризации на основе алгоритма *WTA* и определения принадлежности входных примеров заданным кластерам.

2. Теоретические сведения

2.1. Обучение сетей Кохонена

Парадигма обучения нейронных сетей без учителя (*unsupervised learning*) своим названием подчеркивает отсутствие внешнего учителя, контролирующего процесс настройки весовых коэффициентов синаптических связей между нейронами ИНС. Данная парадигма предполагает отсутствие маркированных примеров с эталонными выходными значениями для входных векторов, по которым проводится обучение.

Целью алгоритмов без учителя, или алгоритмов самоорганизующегося обучения, является выявление в множестве входных данных существенных образов или признаков без участия учителя. Для этого алгоритм реализует множество правил вычисления отображения входного сигнала на входной с требуемыми свойствами.

Одной из моделей нейронных сетей, обучаемых на основе принципа самоорганизации, являются *нейронные сети Кохонена* – прямонаправленные нейронные сети, основанные на *конкурентном* обучении. Сети Кохонена представляют собой разновидность *самоорганизующихся карт признаков (Self-organizing map – SOM)*, предложенных финским исследователем Тойво Кохоненом, и используются для решения задач кластеризации.

Искусственная нейронная сеть Кохонена представляет собой двухслойную сеть (рис. 4.1), в которой каждый нейрон первого (распределительного) слоя соединен со всеми нейронами второго (выходного) слоя, которые расположены в виде двумерной решетки. Нейроны выходного слоя называются *кластерными элементами*; их количество определяет максимальное количество групп, на которые алгоритм может разделить входные данные.

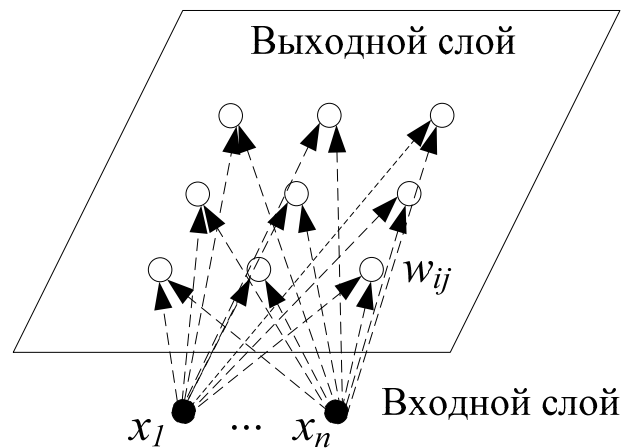


Рис. 4.1. Структура сети Кохонена

Сети Кохонена основаны на конкурентном обучении, которое заключается в следующем. Конкурирующие между собой нейроны (рис. 4.2) получают одни и те же входные сигналы x_j .

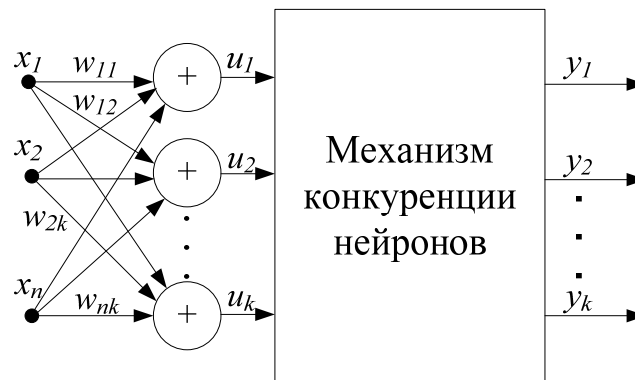


Рис. 4.2. Механизм конкуренции нейронов в сети Кохонена

В зависимости от фактических значений весовых коэффициентов суммарные сигналы u_i отдельных нейронов могут различаться. Выходные сигналы нейронов сравниваются между собой – в процессе этой конкуренции *побеждает* нейрон с самым сильным выходным сигналом, то есть тот, для которого взвешенная сумма его входов $u_i = \sum_{j=1}^n x_j \cdot w_{ij}$ максимальна, что для

нормализованных векторов равнозначно наименьшему евклидову расстоянию между входным вектором и вектором весов нейронов.

Этот нейрон называется «нейроном-победителем», и он получает право уточнить свои веса в направлении входного вектора x по правилу:

$$w_{ij}(t + 1) = w_{ij}(t) + \eta \cdot [x_j - w_{ij}(t)] \cdot G(j), \quad (4.1)$$

где w_{ij} – весовой коэффициент, соединяющий i -вход с j -м выходом; t – номер итерации обучения; η – коэффициент скорости обучения; $G(j)$ – функция окрестности; причем единичная функция окрестности $G(j)$ рассчитывается по формуле:

$$G(j) = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$

где j – номер выходного нейрона, k – номер нейрона-победителя, имеющего максимальное значение выходного сигнала.

Представленный алгоритм обучения, использующий единичную функцию окрестности, называется «победитель получает все» (*WTA* – англ. *Winner Takes All*). Это значит, что нейрон-победитель вырабатывает на своем выходе состояние 1, а остальные (проигравшие) нейроны переходят в состояние 0.

Подстройка весов связей в алгоритме *WTA* осуществляется не для всех нейронов слоя, а только для «нейрона-победителя», значение выхода которого в слое нейронов оказывается максимальным.

Когда «победивший в соревновании» нейрон переходит в состояние 1, он получает возможность уточнения своих весовых коэффициентов, значения которых приближаются к значениям текущего обучающего вектора x . Напротив, у проигравших нейронов, сформировавших на выходах состояние 0, блокируется процесс уточнения весовых коэффициентов: проигравшие нейроны не изменяют свои веса.

Таким образом, только победа при очередном представлении входного вектора позволит нейронам скорректировать весовые коэффициенты и продолжить процесс обучения в случае еще одной победы.

Нейроны в ходе конкурентного процесса избирательно настраиваются на различные входные образы. Следствием процесса конкуренции становится самоорганизация процесса обучения.

Нейроны уточняют свои веса таким образом, что при предъявлении группы близких по значениям входных векторов победителем всегда оказывается один и тот же нейрон. В этом случае весовые коэффициенты этого нейрона становятся равными усредненным значениям тех входных векторов, благодаря которым данный нейрон оказался победителем.

Серьезной проблемой при обучении *WTA* является появление «мёртвых» нейронов, которые после инициализации никогда не побеждают в конкурентной борьбе и остаются в состоянии, однажды сформированном в начальный момент времени. Каждый «мертвый» нейрон уменьшает эффективное количество элементов, прошедших обучение, и соответственно увеличивает общую погрешность распознавания данных. Для разрешения этой проблемы применяется модифицированное обучение, основанное на учете прошлых побед каждого нейрона и штрафования (временной дисквалификации) тех из них, которые побеждали чаще всего.

2.2. Алгоритм обучения без учителя *WTA*

Шаг 1. Обучающий вектор X подается на вход сети.

Шаг 2. Определяются расстояния d_j (в n -мерном пространстве) между X и весовыми векторами W_j каждого нейрона. В евклидовом пространстве это расстояние вычисляется по следующей формуле:

$$d_j = \sqrt{\sum_{i=1}^n (x_i - w_{ij})^2}, \quad (4.2)$$

где x_i – компонента i -го входного вектора X , w_{ij} – весовой коэффициент, связывающий входной вектор i с нейрона j .

Шаг 3. Нейрон, который имеет весовой вектор, самый близкий к X , объявляется победителем. Этот весовой вектор, называемый W_c , становится основным в группе весовых векторов, которые лежат в пределах расстояния d от W_c .

Шаг 4. Весовые векторы в пределах расстояния d от W_c настраиваются в соответствии с выражением (4.1).

Шаг 5. Повторяются шаги с 1 по 4 для каждого входного вектора до тех пор, пока принадлежность объектов к кластерам не перестанет изменяться и веса сети не станут стабильными.

3. Методика выполнения работы

Пример. Рассматривается процедура кластеризации с помощью нейронной сети Кохонена (рис. 4.3), имеющей 4 входных нейрона в распределительном слое и 2 нейрона (кластерных элемента) в выходном слое.

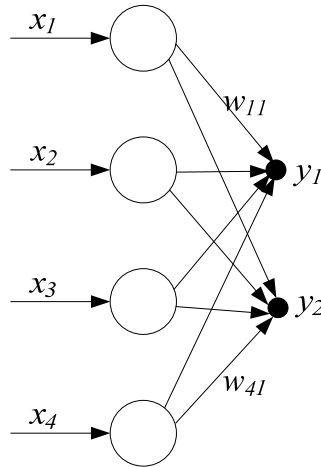
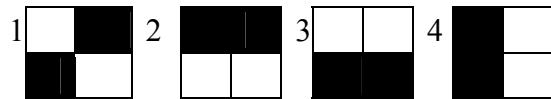


Рис. 4.3. Пример нейронной сети Кохонена

Для обучения используется обучающая выборка, состоящая из четырех примеров, закодированных в виде черно-белого точечного изображения:



Имеется матрица начальных значений весовых коэффициентов нейронов $W = \begin{bmatrix} 0,2 & 0,4 & 0,1 & 0,1 \\ 0,5 & 0,6 & 0,6 & 0,3 \end{bmatrix}$ и коэффициент скорости обучения, равный $\eta = 0,8$.

На основе евклидовой метрики определяется принадлежность входных примеров $X_i, i = \overline{1,4}$ кластерным элементам и вычисляются весовые коэффициенты после первого цикла обработки данных.

В соответствии с принятым правилом кодировки обучающая выборка преобразуется в четыре входных примера:

$$X_1 = \{0, 1, 1, 0\},$$

$$X_2 = \{1, 1, 0, 0\},$$

$$X_3 = \{0, 0, 1, 1\},$$

$$X_4 = \{1, 0, 1, 0\}.$$

Матрица весовых коэффициентов выходных нейронов (кластерных элементов) в виде двух векторов имеет вид:

$$W_1 = \{0,2, 0,4, 0,1, 0,1\},$$

$$W_2 = \{0,5, 0,6, 0,6, 0,3\}.$$

Через d_{ij} обозначается мера близости между i -м входным примером и j -м кластерным элементом.

Шаг 1. Входной пример X_1 поочередно сравнивается с кластерными элементами W_1 и W_2 – для этого осуществляется расчет меры близости на основе евклидовой метрики:

$$d_{1,1}^2 = (0 - 0,2)^2 + (1 - 0,4)^2 + (1 - 0,1)^2 + (0 - 0,1)^2 = 1,22;$$

$$d_{1,2}^2 = (0 - 0,5)^2 + (1 - 0,6)^2 + (1 - 0,6)^2 + (0 - 0,3)^2 = 0,66.$$

Расчет мер близости показал, что первый пример X_1 ближе к кластерному элементу 2, так как $d_{1,2} < d_{1,1}$. Следовательно, нейроном-победителем является второй выходной нейрон, и его весовые значения необходимо приблизить к входному вектору X_1 по формуле: $W_2 = W_2 - \eta \cdot (W_2 - X_1)$:

$$w'_{21} = 0,5 - 0,8 \cdot (0,5 - 0) = 0,1;$$

$$w'_{22} = 0,6 - 0,8 \cdot (0,6 - 1) = 0,92;$$

$$w'_{23} = 0,6 - 0,8 \cdot (0,6 - 1) = 0,92;$$

$$w'_{24} = 0,3 - 0,8 \cdot (0,3 - 0) = 0,06.$$

Таким образом, матрица весовых коэффициентов примет вид:

$$W = \begin{bmatrix} 0,2 & 0,4 & 0,1 & 0,1 \\ 0,1 & 0,92 & 0,92 & 0,06 \end{bmatrix}.$$

Шаг 2. Рассчитываются меры близости входного примера X_2 с кластерными элементами W_1 и W_2 :

$$d_{2,1}^2 = (1 - 0,2)^2 + (1 - 0,4)^2 + (0 - 0,1)^2 + (0 - 0,1)^2 = 1,02;$$

$$d_{2,2}^2 = (1 - 0,1)^2 + (1 - 0,92)^2 + (0 - 0,92)^2 + (0 - 0,06)^2 = 1,67.$$

Очевидно, что поскольку $d_{2,1} < d_{2,2}$, то входной пример X_2 ближе к кластеру 1. Группа весовых коэффициентов первого кластерного элемента W_1 приближается к входному вектору X_2 по формуле: $W_1 = W_1 - \eta \cdot (W_1 - X_2)$:

$$w'_{11} = 0,2 - 0,8 \cdot (0,2 - 1) = 0,84;$$

$$w'_{12} = 0,4 - 0,8 \cdot (0,4 - 1) = 0,88;$$

$$w'_{13} = 0,1 - 0,8 \cdot (0,1 - 0) = 0,02;$$

$$w'_{14} = 0,1 - 0,8 \cdot (0,1 - 0) = 0,02.$$

Матрица весовых коэффициентов принимает вид:

$$W = \begin{bmatrix} 0,84 & 0,88 & 0,02 & 0,02 \\ 0,1 & 0,92 & 0,92 & 0,06 \end{bmatrix}.$$

Шаг 3. Рассчитываются меры близости примера X_3 с кластерными элементами W_1 и W_2 :

$$d_{3,1}^2 = (0 - 0,84)^2 + (0 - 0,88)^2 + (1 - 0,02)^2 + (1 - 0,02)^2 = 3,4;$$

$$d_{3,2}^2 = (0 - 0,1)^2 + (0 - 0,92)^2 + (1 - 0,92)^2 + (1 - 0,06)^2 = 1,75.$$

Пример X_3 ближе к кластеру 2. Весовые коэффициенты нейрона-победителя уточняются по направлению к значениям третьего примера.

$$w'_{21} = 0,1 - 0,8 \cdot (0,1 - 0) = 0,02;$$

$$w'_{22} = 0,92 - 0,8 \cdot (0,92 - 0) = 0,184;$$

$$w'_{23} = 0,92 - 0,8 \cdot (0,92 - 1) = 0,984;$$

$$w'_{24} = 0,06 - 0,8 \cdot (0,06 - 1) = 0,812.$$

Матрица весовых коэффициентов:

$$W = \begin{bmatrix} 0,84 & 0,88 & 0,02 & 0,02 \\ 0,02 & 0,184 & 0,98 & 0,812 \end{bmatrix}.$$

Шаг 4. И, наконец, последний пример оказывается ближе ко второму кластеру:

$$d_{4,1}^2 = (1 - 0,84)^2 + (0 - 0,88)^2 + (1 - 0,02)^2 + (0 - 0,02)^2 = 2,72;$$

$$d_{4,2}^2 = (1 - 0,02)^2 + (0 - 0,18)^2 + (1 - 0,98)^2 + (0 - 0,81)^2 = 1,03.$$

Уточняются весовые коэффициенты кластерного элемента 2:

$$w'_{21} = 0,02 - 0,8 \cdot (0,02 - 1) = 0,804;$$

$$w'_{22} = 0,184 - 0,8 \cdot (0,184 - 0) = 0,037;$$

$$w'_{23} = 0,98 - 0,8 \cdot (0,98 - 1) = 0,997;$$

$$w'_{24} = 0,812 - 0,8 \cdot (0,812 - 0) = 0,96.$$

Матрица весовых коэффициентов:

$$W = \begin{bmatrix} 0,84 & 0,88 & 0,02 & 0,02 \\ 0,804 & 0,037 & 0,997 & 0,96 \end{bmatrix}.$$

Последовательность кластеров-победителей и скорректированные весовые коэффициенты заносятся в табл. 4.1.

Таблица 4.1

Результат кластеризации на основе алгоритма *WTA*

№ кластера	X	W			
		w_{11}	w_{12}	w_{13}	w_{14}
1	X_2	0,840	0,880	0,02	0,02
2	X_1, X_3, X_4	0,804	0,037	0,997	0,96

Порядок выполнения работы

1. Выполнить процедуру нейросетевой кластеризации методом *WTA* (п.3) по выданному варианту задания 1. Вычислить весовые коэффициенты кластерных элементов после первого цикла обработки данных.

2. Решить задачу кластеризации объектов по выданному варианту задания 2. Изобразить расположение исходных объектов на графике.

Необходимо разбить объекты на 3 кластера. Для нейросетевой кластеризации строится сеть Кохонена со следующей структурой: число входных нейронов равно 2 (в распределительном слое), число выходных нейронов равно 3. Структуру сети Кохонена изобразить графически.

Провести кластеризацию объектов методом *WTA*, если первоначальные весовые коэффициенты нейронов $W = \begin{bmatrix} 0,1 & 0,2 \\ 0,2 & 0,3 \\ 0,3 & 0,1 \end{bmatrix}$ и коэффициент скорости обучения $\eta = 0,5$. При вычислении меры близости использовать евклидову метрику. Итоги нейросетевой кластеризации оформить в виде таблицы.

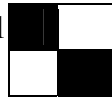
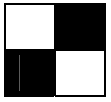
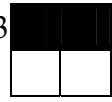
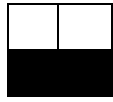



















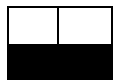
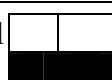




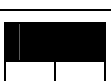
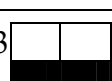
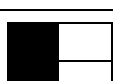
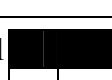

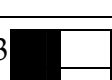
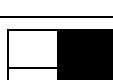


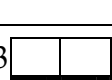
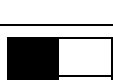
Требования к содержанию и оформлению отчета

Отчет по практической работе должен содержать:

1. Название и цель работы.
2. Исходные данные варианта задания 1. Этапы нейросетевой кластеризации методом *WTA*. Итоговая таблица с результатами кластеризации после первого цикла обработки данных: составом кластеров и весовыми коэффициентами нейронной сети.
3. Исходные данные варианта задания 2. Графическое изображение объектов в пространстве признаков. Структура нейронной сети Кохонена в графическом виде. Этапы нейросетевой кластеризации методом *WTA*. Итоговая таблица с результатами кластеризации.
4. Выводы.

Варианты заданий

Задание 1

№	η	W	Входные образцы для обучения нейросети
1	0,8	$W = \begin{bmatrix} 0 & 0,3 & 0,1 & 0,9 \\ 0,1 & 0,5 & 0,9 & 0,1 \end{bmatrix}$	1  2  3  4 
2	0,8	$W = \begin{bmatrix} 0,4 & 0,4 & 0,6 & 0,9 \\ 0,2 & 0,2 & 0,7 & 0,5 \end{bmatrix}$	1  2  3  4 
3	0,4	$W = \begin{bmatrix} 0,3 & 0,1 & 0,3 & 0,7 \\ 0,2 & 0,4 & 0,5 & 0,4 \end{bmatrix}$	1  2  3  4 
4	0,4	$W = \begin{bmatrix} 0,9 & 0,1 & 0,1 & 0,6 \\ 0,4 & 1 & 0,1 & 1 \end{bmatrix}$	1  2  3  4 
5	0,5	$W = \begin{bmatrix} 0,1 & 0,7 & 0,2 & 0,6 \\ 0,3 & 0,2 & 0,7 & 0,2 \end{bmatrix}$	1  2  3  4 
6	0,7	$W = \begin{bmatrix} 0 & 0,3 & 1 & 0,3 \\ 1 & 0,6 & 0,4 & 0,1 \end{bmatrix}$	1  2  3  4 
7	0,5	$W = \begin{bmatrix} 0,5 & 0,8 & 0,8 & 0,5 \\ 0,9 & 0,5 & 0,9 & 0,6 \end{bmatrix}$	1  2  3  4 
8	0,4	$W = \begin{bmatrix} 0,2 & 0,1 & 0,7 & 0,5 \\ 0,3 & 0,6 & 0,6 & 0,3 \end{bmatrix}$	1  2  3  4 
9	0,5	$W = \begin{bmatrix} 0,4 & 0,7 & 0,2 & 0,3 \\ 0,1 & 0,2 & 0,9 & 0,5 \end{bmatrix}$	1  2  3  4 
10	0,5	$W = \begin{bmatrix} 0,2 & 0,9 & 0,3 & 0,4 \\ 0,4 & 0,9 & 0,4 & 0,3 \end{bmatrix}$	1  2  3  4 

Задание 2

№ варианта	Координаты входных объектов
1	(21; 0,3); (16; 0,8); (15; 0,6); (13; 0,7); (25; 0,7)
2	(22; 0,2); (15; 0,7); (16; 0,5); (12; 0,8); (24; 0,8)
3	(23; 0,1); (14; 0,6); (17; 0,4); (14; 0,9); (25; 0,5)
4	(24; 0,4); (15; 0,7); (14; 0,6); (15; 0,6); (24; 0,4)
5	(25; 0,5); (16; 0,8); (15; 0,6); (13; 0,5); (25; 0,9)
6	(26; 0,4); (11; 0,7); (16; 0,5); (12; 0,7); (24; 0,5)
7	(27; 0,3); (17; 0,8); (17; 0,4); (14; 0,8); (25; 0,6)
8	(20; 0,2); (18; 0,6); (18; 0,6); (13; 0,6); (24; 0,3)
9	(21; 0,1); (19; 0,7); (14; 0,4); (14; 0,7); (25; 0,7)
10	(22; 0,5); (17; 0,8); (15; 0,5); (12; 0,8); (24; 0,3)
11	(23; 0,4); (16; 0,6); (16; 0,4); (14; 0,5); (25; 0,5)
12	(24; 0,3); (15; 0,7); (17; 0,7); (15; 0,6); (24; 0,4)
13	(25; 0,2); (14; 0,8); (14; 0,6); (13; 0,7); (25; 0,9)
14	(26; 0,1); (13; 0,6); (15; 0,5); (12; 0,8); (24; 0,5)
15	(27; 0,5); (12; 0,7); (16; 0,4); (14; 0,9); (25; 0,6)
16	(20; 0,4); (11; 0,8); (17; 0,3); (13; 0,7); (24; 0,3)
17	(21; 0,3); (12; 0,6); (18; 0,4); (14; 0,9); (23; 0,7)
18	(22; 0,2); (13; 0,7); (14; 0,5); (12; 0,6); (24; 0,3)
19	(23; 0,1); (14; 0,8); (15; 0,4); (14; 0,8); (25; 0,5)
20	(24; 0,5); (15; 0,6); (16; 0,7); (15; 0,6); (25; 0,4)

Контрольные вопросы и задания

1. В чем заключается конкуренция при функционировании сети Кохонена?
2. Какие процедуры используются для определения нейрона-победителя в сети Кохонена?
3. Для какой задачи используется сеть Кохонена?
4. Опишите структуру сети Кохонена и основные моменты алгоритма ее функционирования.
5. Какие проблемы возникают при обучении нейронной сети Кохонена?
6. Как работает алгоритм обучения *WTA* – «победитель получает все»?

Список литературы

1. Осовский С. Нейронные сети для обработки информации / Пер. с польского И. Д. Рудинского. М.: Финансы и статистика, 2002. 344 с.
2. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям: Питер, 2013. 706 с.
3. Каллан Р. Основные концепции нейронных сетей: Пер. с англ. М.: Издательский дом «Вильямс», 2001. 287 с.
4. Хайкин С. Нейронные сети: Полный курс: Пер. с англ. 2-е изд. М.: Вильямс, 2006. 1104 с.

Практическая работа № 5

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

1. Цель и задачи работы

Целью работы является приобретение умений и навыков поиска оптимального значения нелинейной функции на заданном интервале с помощью генетических алгоритмов.

Задачами работы являются: приобретение навыков кодирования значений параметров функции в двоичной системе; выбора исходной популяции хромосом, оценки функции приспособленности, селекции хромосом, применения генетических операторов (скрещивания и мутации) и определения наилучшей хромосомы.

2. Теоретические сведения

2.1. Реализация идеи биологической эволюции в рамках генетических алгоритмов

Генетические алгоритмы (ГА) относятся к одному из направлений *эволюционных вычислений* – бионическому направлению, использующему существующие в природе закономерности, упрощенные таким образом, чтобы их можно было реализовать в компьютерных моделях.

Основным механизмом эволюции является естественный отбор. Все биологические виды развиваются с различной степенью приспособленности к окружающей среде; при этом более приспособленные особи получают больше возможностей для выживания. При генерации новых поколений биологических видов работает механизм *генетического наследования*, согласно которому потомки наследуют от родителей некоторые свойства. Очевидно, что количество потомков более приспособленных родителей будет увеличиваться, поэтому через несколько десятков или сотен новых поколений средняя приспособленность такой группы возрастает.

Научное направление, изучающее законы и механизмы наследственности, называется *генетикой*. Генетическая информация хранится в виде набора молекул ДНК. Каждая молекула ДНК – это цепочка из молекул нуклеотидов четырех типов: *A, T, C, G*.

Собственно генетическая информация хранится в виде порядка следования нуклеотидов в молекуле ДНК (в виде кода). Ген – это единица наследственной информации, которая представляет собой отрезок цепи ДНК, ответственный за определенное свойство особи, например за цвет глаз, тип волос, цвет кожи и т.д. В живых организмах клетки содержат важные группы специальной материи, содержащей наследственную информацию – *хромосомы*. Гены образуют хромосомы. В каждой клетке организма есть набор хромосом, несущих информацию обо всей особи.

2.2. Генетические алгоритмы

Генетические алгоритмы – это адаптивные методы поиска, которые часто используются для решения задач функциональной и структурной *оптимизации*. ГА могут использоваться для проектирования структуры механизмов, поиска оптимальной формы детали, раскроя ткани и др.

Основные принципы ГА были сформулированы американским математиком Дж. Холландом в 1975 г. ГА моделируют биологические процессы и по аналогии с эволюционным механизмом работают с популяцией, каждая из хромосом которой представляет собой возможное решение данной задачи. Каждая хромосома оценивается мерой ее «приспособленности», которую также называют функцией *оптимальности*. Наиболее приспособленные особи получают возможность «воспроизвести» потомство с помощью «перекрестного скрещивания» с другими особями популяции. В результате появляются новые особи, сочетающие в себе характеристики, наследуемые от родителей. Наименее приспособленные особи постепенно исчезают из популяции в процессе эволюции. Новое поколение обладает лучшими характеристиками по сравнению с предыдущим. Скрещивание наиболее приспособленных особей приводит к тому, что эволюция отыскивает перспективные решения в широком пространстве поиска. В конечном итоге популяция сходится к оптимальному решению задачи.

В настоящее время термином «генетические алгоритмы» называют широкий класс алгоритмов поиска оптимального решения с различными представлениями хромосом, операторами скрещивания, мутации и т.д.

2.3. Основные понятия генетических алгоритмов

ГА используют словарь, заимствованный из естественной генетики:

– *хромосомы (цепочки, или кодовые последовательности)* – это упорядоченные последовательности *генов*. Хромосома – битовая строка 0101...101, или вектор из нулей и единиц, которая определяет точку пространства поиска и представляет собой потенциальное решение задачи;

– *гены* – элементы, из которых состоит хромосома;

– *популяция* – это конечное множество особей;

– *особи*, входящие в популяцию, в ГА представляются хромосомами с закодированным в них множествами параметров задачи, т.е. решений, которые иначе называются точками в пространстве поиска;

– *генотип*, или структура, – это набор хромосом определенной особи. Следовательно, особями популяции могут быть генотипы либо единичные хромосомы (генотип может состоять даже из одной хромосомы);

– *фенотип* – это набор значений, соответствующих данному генотипу; иначе его называют либо декодированной структурой, либо множеством параметров задачи;

– *аллель* – значение конкретного гена, также определяемое как значение *свойства*;

– *локус*, или позиция, – место размещения данного гена в хромосоме (цепочке). Множество позиций генов называются локами;

– *кроссовер* – операция скрещивания хромосом, при котором хромосомы обмениваются своими частями;

– *мутация* – случайное изменение одной или нескольких позиций в хромосоме;

– *функция приспособленности*, или функция пригодности (*fitness function*), – функция оценки, мера приспособленности особи в популяции, которая позволяет выбирать наиболее приспособленные особи;

– механизм *селекции* заключается в выборе хромосом с наивысшей оценкой (т.е. наиболее приспособленных), которые репродуцируют чаще, чем особи с более низкой оценкой;

– *репродукция* – создание новых хромосом в результате *рекомбинации* генов родительских хромосом;

– *рекомбинация* – это процесс, в результате которого возникают новые комбинации генов. Для этого используется две операции: *скрещивание*, позволяющее создать две совершенно новые хромосомы потомков путем комбинирования генетического материала пары родителей, а также *мутация*, которая может вызвать изменения в отдельных хромосомах.

2.4. Стандартный генетический алгоритм

Стандартный ГА состоит из следующих шагов: 1) инициализация, или выбор исходной популяции хромосом; 2) оценка приспособленности хромосом в популяции; 3) проверка условия остановки алгоритма; 4) селекция хромосом; 5) применение генетических операторов; 6) формирование новой популяции; 7) выбор «наилучшей» хромосомы.

На рис. 5.1 представлена блок-схема ГА, включающего следующие этапы:

Шаг 1. Инициализация, т.е. формирование исходной популяции хромосом, заключается в случайном выборе заданного количества хромосом (особей), представляемых двоичными последовательностями фиксированной длины.

Шаг 2. Оценивание приспособленности хромосом в популяции состоит в расчете функции приспособленности для каждой хромосомы этой популяции. Чем больше значение функции приспособленности, тем выше «качество» хромосомы.

Шаг 3. Проверка условия остановки алгоритма. Остановка работы ГА определяется следующими условиями:

1) в оптимизационных задачах, если известно максимальное (или минимальное) значение функции приспособленности, то остановка алгоритма может произойти после достижения ожидаемого оптимального значения;

2) остановка алгоритма также может произойти в случае, когда его выполнение не приводит к улучшению уже достигнутого значения;

3) алгоритм может быть остановлен по истечении определенного времени выполнения либо после выполнения заданного количества итераций.

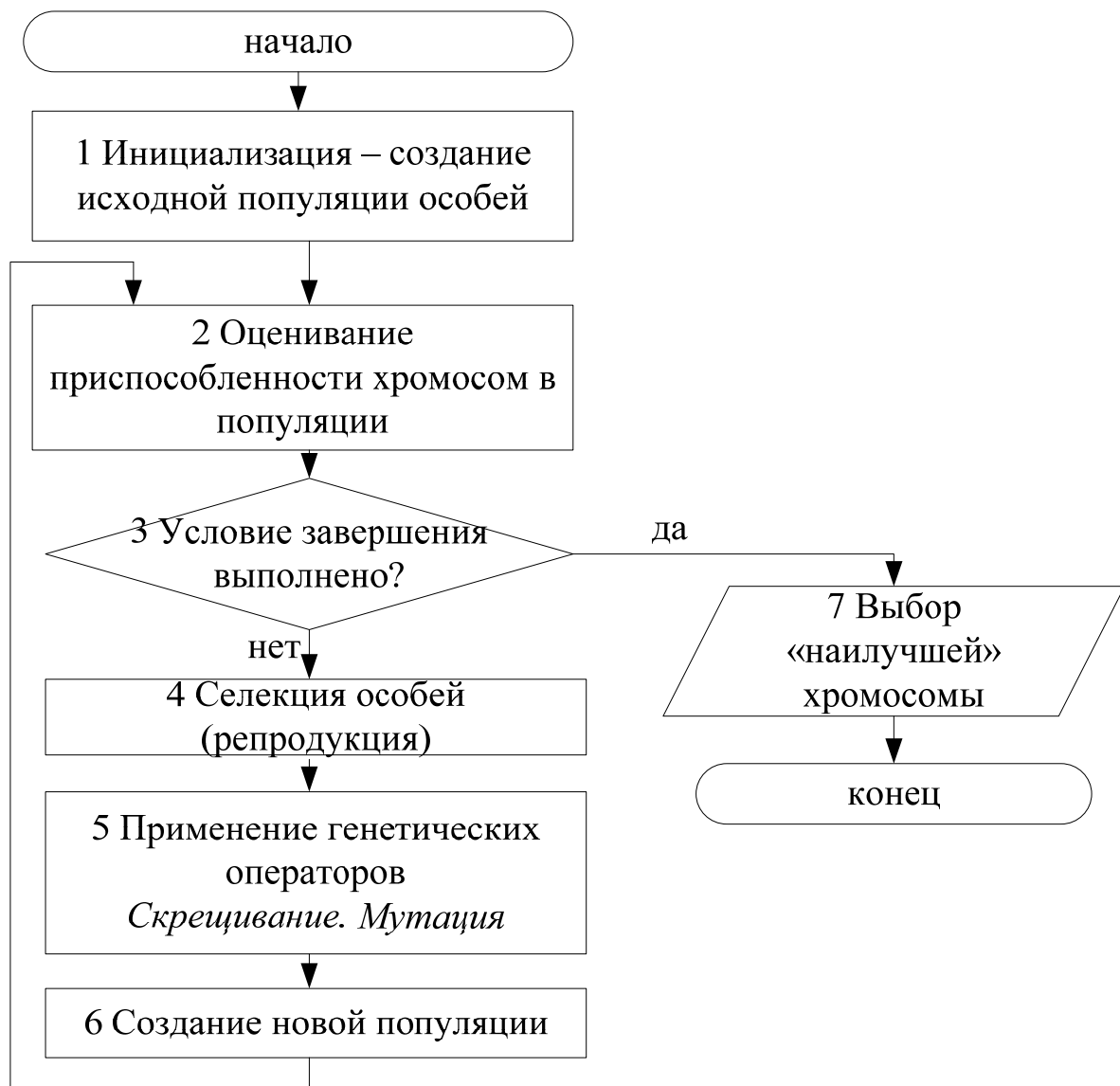


Рис. 5.1. Блок-схема генетического алгоритма

Если условие остановки выполнено, то осуществляется переход к этапу *выбора «наилучшей» хромосомы*, вывода результата алгоритма и завершение работы алгоритма. В противном случае – переход к следующему шагу.

Шаг 4. Селекция хромосом, которая заключается в выборе тех хромосом, которые будут участвовать в создании потомков для следующей популяции, т.е. для очередного поколения. Селекция хромосом нужна для создания условий преимущества более приспособленным хромосомам в процессе репродукции новой

популяции. Такой выбор соответствует принципу естественного отбора, согласно которому наибольшие шансы на участие в создании новых особей имеют хромосомы с наибольшими значениями функции приспособленности.

Наиболее популярным методом селекции считается *метод рулетки*, который свое название получил по аналогии с азартной игрой: каждой хромосоме сопоставляется сектор колеса рулетки, величина которого устанавливается пропорционально значению функции приспособленности данной хромосомы. Чем больше значение функции приспособленности, тем больше сектор на колесе рулетки. Очевидно, что все колесо рулетки соответствует сумме значений функции приспособленности всех хромосом рассматриваемой популяции. Итак, каждой хромосоме ch_i , $i = 1, 2, \dots, N$, соответствует сектор колеса $v(ch_i^*)$, выраженный в процентах:

$$v(ch_i^*) = p_S(ch_i^*) \cdot 100\%, \quad (5.1)$$

где $p_S(ch_i^*)$ — вероятность селекции i -й хромосомы, которая рассчитывается с использованием функции приспособленности каждой хромосомы $F(ch_i^*)$:

$$p_S(ch_i^*) = \frac{F(ch_i^*)}{\sum_{i=1}^N F(ch_i^*)}. \quad (5.2)$$

Селекция хромосомы, таким образом, осуществляется на основе моделирования поворота колеса рулетки, где «выигравшая» (т.е. выбранная) хромосома определяется по выпавшему сектору колеса.

Разумеется, чем больше такой сектор, тем больше вероятность «победы» соответствующей хромосомы. Поэтому вероятность выбора данной хромосомы оказывается пропорциональной значению ее функции приспособленности.

В результате процесса селекции создается *родительская популяция*, также называемая родительским пулом, с численностью N , равной численности текущей популяции.

Шаг 5. Применение генетических операторов к хромосомам, отобраным с помощью селекции. Формируется новая популяция потомков от созданной на предыдущем шаге родительской популяции.

Применяются два основных генетических оператора: оператор скрещивания (*crossover*) и оператор мутации (*mutation*).

Оператор скрещивания моделирует процесс скрещивания особей путем обмена частями хромосом между двумя или более

особями в популяции следующим образом. На первом этапе скрещивания выбираются пары хромосом из родительской популяции (родительского пула). Это временная популяция, состоящая из хромосом, отобранных в результате селекции и предназначенных для дальнейших преобразований операторами скрещивания и мутации с целью формирования новой популяции потомков. Хромосомы из родительской популяции случайным способом объединяются в пары. Для каждой пары отобранных родителей разыгрывается позиция гена (локус) в хромосоме, определяющая точку скрещивания. Если хромосома каждого из родителей состоит из L генов, то очевидно, что точка скрещивания l_k представляет собой натуральное число, меньшее L . Поэтому фиксация точки скрещивания сводится к случайному выбору числа из интервала $[1, L - 1]$. В результате скрещивания пары родительских хромосом получается следующая пара потомков:

1) потомок, хромосома которого на позициях от 1 до l_k состоит из генов первого родителя, а на позициях от $l_k + 1$ до L – из генов второго родителя;

2) потомок, хромосома которого на позициях от 1 до l_k состоит из генов второго родителя, а на позициях от $l_k + 1$ до L – из генов первого родителя.

Оператор мутации – оператор, моделирующий стохастическое изменение части хромосом. Оператор с вероятностью p_m изменяет значение гена в хромосоме на противоположное (т.е. с 0 на 1 или обратно). Например, если в хромосоме $[100110101010]$ мутации подвергается ген на позиции 7, то его значение, равное 1, изменяется на 0, что приводит к образованию хромосомы $[100110001010]$. Оператор мутации играет второстепенную роль по сравнению с оператором скрещивания, т.к. мутация осуществляется достаточно редко (из аналогии с миром живых организмов) и составляет примерно 1%. Оператор мутации необходим, чтобы, с одной стороны, ввести в популяцию некоторое разнообразие и расширить область поиска, а с другой – чтобы предупредить потери, которые могли бы произойти вследствие исключения какого-нибудь значимого гена в результате скрещивания и не привести к таким изменениям потомков, которые будут далеки от приемлемых решений.

В ГА мутация хромосом может выполняться на популяции родителей перед скрещиванием либо на популяции потомков, образованных в результате скрещивания.

Шаг 6. Формирование новой популяции. Хромосомы, полученные в результате применения генетических операторов к хромосомам временной родительской популяции, включаются в состав *новой популяции*, которая становится так называемой текущей популяцией для данной итерации ГА. Далее осуществляется переход к шагу 2, на котором определяются оценки приспособленности хромосом в носой популяции.

Шаг 7. Выбор «наилучшей» хромосомы. Если условие остановки алгоритма выполнено, то следует вывести результат работы, т.е. представить искомое решение задачи. Лучшим решением считается хромосома с наибольшим значением функции приспособленности.

3. Методика выполнения работы

Пример. Необходимо найти максимум функции $f(x) = 2x^2 + 1$ на отрезке $0 < x < 31$ с помощью генетических алгоритмов.

Очевидно, что эта задача легко решается с использованием необходимого условия экстремума, однако на этом примере удобно проиллюстрировать принципы работы ГА.

Допустим, что x принимает целые значения из заданного диапазона. Задача оптимизации этой функции заключается в перемещении по пространству, состоящему из 32 точек со значениями $0, 1, \dots, 31$, для обнаружения той точки, в которой функция принимает максимальное (или минимальное) значение. В этом случае в качестве *параметра задачи* выступает переменная x . Множество $\{0, 1, \dots, 31\}$ составляет *пространство поиска* и одновременно – множество потенциальных *решений задачи*. Каждое из 32 чисел, принадлежащих этому множеству, называется либо точкой пространства поиска решений, либо решением, либо значением параметра, либо фенотипом. Решение, оптимизирующее функцию, называется *наилучшим*, или оптимальным, решением.

В качестве функции пригодности (приспособленности) выступает сама исходная функция. Чем больше ее значение, тем лучше пригодность хромосомы.

3.1. Инициализация, или выбор исходной популяции хромосом

3.1.1. Кодирование фенотипов в хромосомы.

Очевидно, что для кодирования в двоичную систему значений аргументов функции x , изменяющихся в диапазоне $0 < x < 31$, потребуется 5 двоичных разрядов. Соответствие фенотипов и генотипов (хромосом) показано в табл. 5.1.

Таблица 5.1

Кодирование значений параметра в двоичной системе

Фенотип ch^*	0	1	2	3	4	5	6	7
Генотип ch	00000	00001	00010	00011	00100	00101	00110	00111
Фенотип ch^*	8	9	10	11	12	13	14	15
Генотип ch	01000	01001	01010	01011	01100	01101	01110	01111
Фенотип ch^*	16	17	18	19	20	21	22	23
Генотип ch	10000	10001	10010	10011	10100	10101	10110	10111
Фенотип ch^*	24	25	26	27	28	29	30	31
Генотип ch	11000	11001	11010	11011	11100	11101	11110	11111

Представленные кодовые последовательности называются *хромосомами*, которые выступают в роли *генотипов*. Каждая из хромосом состоит из 5 генов (длиной 5 битов). Значение гена в конкретной позиции называется аллелью, принимающей в данном случае значения 0 или 1.

3.1.2. Генерация начальной выборочной популяции особей, или генотипов.

Итак, *популяция* состоит из *особей*, выбираемых среди исходных 32 хромосом. Встает вопрос: какого размера должна быть начальная популяция и каким методом ее определять. В классическом генетическом алгоритме принято определять выборочную популяцию случайным образом.

Устанавливается размер исходной популяции с численностью, равной шести хромосомам $N = 6$; а затем случайным образом генерируется популяция. Предположим, что случайный выбор определил множество хромосом $\{ch_1, ch_2, \dots, ch_6\}$, представляющих собой закодированную форму фенотипов: $\{ch_1^*, ch_2^*, \dots, ch_6^*\}$ (табл. 5.2).

Таблица 5.2

Хромосомы исходной популяции и их фенотипы

Хромосома	Фенотип	Хромосома	Фенотип
$ch_1 = [10011]$	$ch_1^* = 19$	$ch_4 = [10101]$	$ch_4^* = 21$
$ch_2 = [00011]$	$ch_2^* = 3$	$ch_5 = [01000]$	$ch_5^* = 8$
$ch_3 = [00111]$	$ch_3^* = 7$	$ch_6 = [11101]$	$ch_6^* = 29$

3.2. Оценка приспособленности хромосом в популяции

Функция приспособленности отдельных хромосом в популяции определяется значением исходной функции $f(x) = 2x^2 + 1$ для значений x , соответствующих этим хромосомам, т.е. для фенотипов, соответствующих определенным *генотипам* (табл. 5.3).

Таблица 5.3

Значения функции приспособленности для каждой хромосомы

Хромосома	Фенотип	Функция приспособленности
$ch_1 = [10011]$	$ch_1^* = 19$	$F(ch_1^*) = 723$
$ch_2 = [00011]$	$ch_2^* = 3$	$F(ch_2^*) = 19$
$ch_3 = [00111]$	$ch_3^* = 7$	$F(ch_3^*) = 99$
$ch_4 = [10101]$	$ch_4^* = 21$	$F(ch_4^*) = 883$
$ch_5 = [01000]$	$ch_5^* = 8$	$F(ch_5^*) = 129$
$ch_6 = [11101]$	$ch_6^* = 29$	$F(ch_6^*) = 1683$

3.3. Проверка условия остановки алгоритма

Наибольшим значением функции приспособленности характеризуется хромосома ch_6 ($F(ch_6^*) = 1683$), поэтому в этой популяции она считается наилучшим кандидатом на решение задачи.

Очевидным является то, что в данной задаче максимум исходной функции достигается на интервале $0 < x < 31$ в точке $x = 31$ ($f(x) = 1923$), поэтому можно сделать вывод о том, что лучшее значение F еще не найдено. Это означает, что условие остановки алгоритма не выполняется и осуществляется переход к следующему шагу алгоритма.

3.4. Селекция хромосом

Селекция хромосом осуществляется *методом рулетки*. Для отбора наиболее приспособленных хромосом для каждой из 6 хромосом текущей популяции задается вероятность выбора

$p_S(ch_i^*)$ по формуле (5.2) пропорционально относительной приспособленности. Затем числовой промежуток $[0, 100]$ разбивается на интервалы длиной $v(ch_i^*)$ (5.1) и формируются секторы колеса рулетки, выраженные в процентах (табл. 5.4).

Таблица 5.4

Секторы колеса рулетки

Фенотип ch_i^*	Значения функции приспособленности $F(ch_i^*)$	Вероятность выбора $p_S(ch_i^*)$, % – сектор колеса рулетки	$\Sigma p_S(ch_i^*)$, %
$ch_1^* = 19$	$F(ch_1^*) = 723$	20,45	20,45
$ch_2^* = 3$	$F(ch_2^*) = 19$	0,54	20,99
$ch_3^* = 7$	$F(ch_3^*) = 99$	2,80	23,79
$ch_4^* = 21$	$F(ch_4^*) = 883$	24,97	48,76
$ch_5^* = 8$	$F(ch_5^*) = 129$	3,64	52,41
$ch_6^* = 29$	$F(ch_6^*) = 1683$	47,59	100
	$\Sigma F(ch_i^*) = 3536$	100	

На рис. 5.2. представлено колесо рулетки, построенное на основе табл. 5.4. Чем больше сектор колеса рулетки, тем больше вероятность победы соответствующей хромосомы, поэтому в среднем функция приспособленности от поколения к поколению будет возрастать.

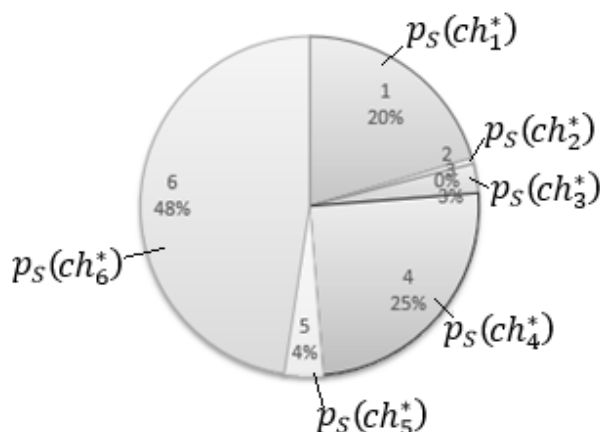


Рис. 5.2. Колесо рулетки для селекции

Розыгрыш с помощью колеса рулетки сводится к случайному выбору числа из интервала $[0, 100]$, указывающего на соответствующий сектор на колесе, т.е. на конкретную хромосому. Процесс отбора основан на вращении колеса рулетки в количестве раз, равном размеру популяции. Так, для рассматриваемого примера

колесо рулетки вращается 6 раз, при этом случайным образом выбирается число из интервала $[0, 100]$, фиксируется попадание этого числа в один из промежутков, соответствующих хромосоме начальной популяции.

Пусть разыграны следующие 8 чисел:

97, 26, 54, 13, 31, 88.

Тогда на основе данных таблицы 5.4 определяются секторы колеса рулетки, в которые попали эти числа, а затем устанавливается, какие хромосомы начальной популяции соответствуют этим числам (табл. 5.5).

Таблица 5.5

Результаты розыгрыша и соответствующие хромосомы

Результаты розыгрыша		Хромосомы
97	→	$ch_6 = [11101]$
26	→	$ch_4 = [10101]$
54	→	$ch_6 = [11101]$
13	→	$ch_1 = [10011]$
31	→	$ch_4 = [10101]$
88	→	$ch_6 = [11101]$

Анализ результатов розыгрыша показали, что хромосома ch_6 была выбрана трижды, а хромосома ch_4 – дважды, что связано с тем, что именно эти хромосомы имеют наибольшие значения функции приспособленности.

Таким образом, образуется так называемый *родительский пул* – набор хромосом, на основе которых формируются родительские пары хромосом. Родительский пул – временная популяция, необходимая для формирования потомков.

3.5. Применение генетических операторов

3.5.1. Операция скрещивания (кроссовер)

Множество всех возможных родительских пар формируется путем прямого произведения хромосом из родительского пула. В рассматриваемом примере образуется 36 пар хромосом:

$$\{ch_6, ch_4, ch_6, ch_1, ch_4, ch_6\} \times \{ch_6, ch_4, ch_6, ch_1, ch_4, ch_6\} = \\ = \{(ch_6, ch_6); (ch_4, ch_6); (ch_6, ch_6); (ch_1, ch_6); \dots; (ch_6, ch_6)\}$$

Предположим, что из полученных 36 пар хромосом случайным образом установлены три пары родителей:

ch_1 и ch_4 , ch_4 и ch_6 , ch_6 и ch_6 .

К каждой родительской паре хромосом применяется оператор скрещивания (*кроссовер*), в результате которого порождаются новые хромосомы, у которых первая половина аллелей заимствуется у первого родителя, а вторая – у второго.

В случае одноточечного скрещивания для каждой родительской пары случайным образом задается точка скрещивания (локус) l_k . Пусть, например, для первой пары $l_k = 3$, для второй $l_k = 2$, для третьей – $l_k = 2$. Далее в каждой родительской паре производится обмен частями хромосом и образуется 2 новых потомка. В результате выполнения оператора скрещивания формируется 3 пары потомков (табл. 5.6).

Таблица 5.6

Скрещивание хромосом				
№	Родительские пары	локус	скрещивание	Пары потомков
1	$ch_1 = [1\ 0\ 0\ 1\ 1]$ $ch_4 = [1\ 0\ 1\ 0\ 1]$	$l_k = 3$	→	$Ch_1 = [1\ 0\ 0\ 0\ 1]$ $Ch_2 = [1\ 0\ 1\ 1\ 1]$
2	$ch_4 = [1\ 0\ 1\ 0\ 1]$ $ch_6 = [1\ 1\ 1\ 0\ 1]$	$l_k = 2$	→	$Ch_3 = [1\ 0\ 1\ 0\ 1]$ $Ch_4 = [1\ 1\ 1\ 0\ 1]$
3	$ch_6 = [1\ 1\ 1\ 0\ 1]$ $ch_6 = [1\ 1\ 1\ 0\ 1]$	$l_k = 2$	→	$Ch_5 = [1\ 1\ 1\ 0\ 1]$ $Ch_6 = [1\ 1\ 1\ 0\ 1]$

Таким образом, применение оператора скрещивания к родительскому пулу порождает новую популяцию.

3.5.2. Операция мутации

Оператор мутации случайным образом с заданной вероятностью p_m , где $0 \leq p_m \leq 0,1$, изменяет значение гена в хромосоме на противоположный. В табл. 5.7 показано, как применяется оператор мутации в локусе $l_k = 5$ хромосомы Ch_2 : тогда $Ch_2 = [1\ 0\ 1\ 1\ 1]$ преобразуется в $Ch_2 = [1\ 0\ 1\ 1\ 0]$.

Таблица 5.7

Мутация хромосом				
№	Хромосома	локус	мутация	Новая хромосома
1	$Ch_2 = [1\ 0\ 1\ 1\ 1]$	$l_k = 5$	→	$Ch_2 = [1\ 0\ 1\ 1\ 0]$

Таким образом, операция мутации, видоизменяя хромосомы, вносит разнообразие в популяцию.

3.6. Формирование новой популяции

Операторы скрещивания и мутации формируют новую популяцию. Оценки функции приспособленности хромосом новой популяции и ее фенотипы представлены в таблице 5.8. Для отличия от хромосом исходной популяции обозначения вновь сформированных хромосом начинаются с заглавной *C*.

Таблица 5.8

Хромосомы новой популяции и их фенотипы

Хромосома	Фенотип	Функция приспособленности
$Ch_1 = [1\ 0\ 0\ 0\ 1]$	$Ch_1^* = 17$	$F(Ch_1^*) = 579$
$Ch_2 = [1\ 0\ 1\ 1\ 0]$	$Ch_2^* = 22$	$F(Ch_1^*) = 969$
$Ch_3 = [1\ 0\ 1\ 0\ 1]$	$Ch_3^* = 21$	$F(Ch_1^*) = 883$
$Ch_4 = [1\ 1\ 1\ 0\ 1]$	$Ch_4^* = 29$	$F(Ch_1^*) = 1683$
$Ch_5 = [1\ 1\ 1\ 0\ 1]$	$Ch_5^* = 29$	$F(Ch_1^*) = 1683$
$Ch_6 = [1\ 1\ 1\ 0\ 1]$	$Ch_6^* = 29$	$F(Ch_1^*) = 1683$

Анализ результатов расчета новых функций приспособленности показывает, что новая популяция (потомки) характеризуется большей средней приспособленностью по сравнению с исходной (родительской) популяцией (см. табл. 5.3).

Дальнейшие итерации генетического алгоритма формируют новые хромосомы популяции, значения функции приспособленности которых будут выше, чем у предыдущей популяции.

3.7. Выбор «наилучшей» хромосомы

Наибольшим значением функции принадлежности характеризуются хромосомы Ch_4 , Ch_5 и Ch_6 . Если на следующих итерациях генетического алгоритма будут разыграны для скрещивания пары хромосом, например, Ch_4 и Ch_2 , Ch_5 и Ch_2 , Ch_6 и Ch_2 с точками скрещивания $l_k=2$ или $l_k=3$, то среди прочих может быть образована хромосома $Ch=[11111]$ с фенотипом $Ch^* = 31$, при котором функция приспособленности достигнет своего максимума, равного $F(Ch^*) = 1923$.

Порядок выполнения работы

1. Согласно варианту задания найти экстремум функции на заданном интервале с помощью генетического алгоритма. Принять размер начальной популяции, равным 6.

Закодировать значения параметров функции в двоичной системе. Выбрать исходную популяцию хромосом. Оценить функцию приспособленности. Провести селекцию хромосом. Применить генетические операторы скрещивания и мутации. Определить экстремум функции.

2. Проанализировать найденное решение путем построения графика целевой функции.

3. Оценить среднее количество шагов генетического алгоритма, необходимое для нахождения оптимального решения, выполняя запуск с разными начальными условиями.

Требования к содержанию и оформлению отчета

Отчет по практической работе должен содержать:

1. Название и цель работы, исходные данные варианта задания.
2. Этапы генетического алгоритма.
3. Найденное решение функции и сравнение с оптимальным решением для заданной функции (по графику).
4. Выводы.

Варианты заданий

№	Функция	Вид экстремума	Интервал
1	2	3	4
1	$f(x) = 5 - 2x + 15x^2$	max	$0 < x < 15$
2	$f(x) = 6 + 3x - 12x^2$	min	$0 < x < 15$
3	$f(x) = 15 - 4x + 5x^2$	max	$0 < x < 15$
4	$f(x) = 2 + 5x - 10x^2$	min	$0 < x < 15$
5	$f(x) = 12 - 6x + 11x^2$	max	$0 < x < 15$
6	$f(x) = 10 + 7x - 14x^2$	min	$0 < x < 15$
7	$f(x) = 12 - 8x + 2x^2$	max	$0 < x < 15$
8	$f(x) = 15 + 9x - 10x^2$	min	$0 < x < 15$
9	$f(x) = 2 - 3x + 15x^2$	max	$0 < x < 15$
10	$f(x) = 10 + 4x - 5x^2$	min	$0 < x < 15$

Окончание табл.

1	2	3	4
11	$f(x) = 12 - 5x + 2x^2$	max	$0 < x < 15$
12	$f(x) = 5 + 6x - 3x^2$	min	$0 < x < 15$
13	$f(x) = 2 - 7x + 4x^2$	max	$0 < x < 15$
14	$f(x) = 15 + 8x - 5x^2$	min	$0 < x < 15$
15	$f(x) = 5 - 9x + 8x^2$	max	$0 < x < 15$
16	$f(x) = 10 + 3x - 2x^2$	min	$0 < x < 15$
17	$f(x) = 12 - 6x + x^2$	max	$0 < x < 15$
18	$f(x) = 5 + x - 2x^2$	min	$0 < x < 15$
19	$f(x) = 2 - 12x + 5x^2$	max	$0 < x < 15$
20	$f(x) = 15 + 10x - 5x^2$	min	$0 < x < 15$

Контрольные вопросы и задания

1. Опишите работу генетического алгоритма.
2. Как происходит кодирование фенотипов в хромосомы?
3. Что такое функция приспособленности?
4. Что определяет сходимость генетического алгоритма?
5. В чем заключается метод рулетки? Опишите процесс селекции.
6. Каким образом происходит формирование начальной популяции?
7. Какие задачи решают с помощью генетических алгоритмов?

Список литературы

1. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия - Телеком, 2004. 452 с.
2. Модели и методы искусственного интеллекта. Применение в экономике / Матвеев М. Г., Свиридов А. С., Алейникова Н. А. М.: Финансы и статистика, 2008. 448 с.
3. Ярушкина Н. Г. Основы теории нечетких и гибридных систем: учеб. пособие. М.: Финансы и статистика, 2004. 320с.
4. Осипов Г.С. Лекции по искусственному интеллекту. М.: Книжный дом «ЛИБРОКОМ», 2016. 272 с.
5. Панченко Т.В. Генетические алгоритмы: учеб. пособие / под ред. Ю.Ю.Тарасевича. Астрахань: Издательский дом «Астраханский университет», 2007. 87 с.