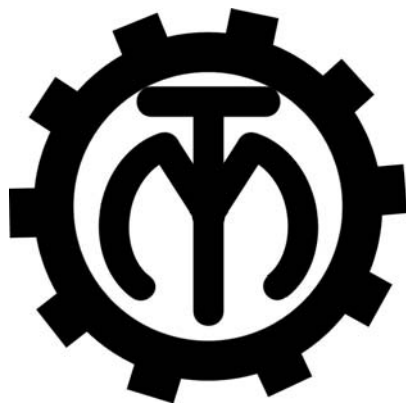


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В РОБОТОТЕХНИКЕ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.06 «Мехатроника и робототехника»
очной формы обучения*



Могилев 2023

УДК 621.01
ББК 65.304.15
М79

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «31» августа 2023 г.,
протокол № 1

Составитель канд. техн. наук, доц. М. Н. Миронова

Рецензент канд. техн. наук, доц. Д. М. Свирепа

Методические рекомендации предназначены для выполнения лабораторных работ по дисциплине «Методы искусственного интеллекта в робототехнике» студентами направления подготовки 15.03.06 «Мехатроника и робототехника» очной формы обучения. Изложены методики выполнения лабораторных работ.

Учебное издание

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В РОБОТОТЕХНИКЕ

| | |
|-------------------------|------------------|
| Ответственный за выпуск | В. М. Шеменков |
| Корректор | И. В. Голубцова |
| Компьютерная верстка | Н. П. Полевничая |

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

| | |
|---|----|
| Инструкция по охране труда при проведении лабораторных работ..... | 4 |
| 1 Лабораторная работа № 1. Проектирование экспертной системы на основе использования метода Байеса..... | 5 |
| 2 Лабораторная работа № 2. Построение решающих правил..... | 7 |
| 3 Лабораторная работа № 3. Построение нечетких множеств..... | 10 |
| 4 Лабораторная работа № 4. Нечеткий вывод..... | 14 |
| 5 Лабораторная работа № 5. Метрическое распознавание образов..... | 20 |
| 6 Лабораторная работа № 6. Обучение в интеллектуальных системах... | 25 |
| 7 Лабораторная работа № 7. Построение нейронных сетей | 29 |
| 8 Лабораторная работа № 8. Методы обучения сетей..... | 33 |
| Список литературы..... | 38 |

Инструкция по охране труда при проведении лабораторных работ

Общие требования безопасности

1 Для работы на ПЭВМ в компьютерном классе допускаются студенты, прошедшие обучение и проверку знаний по мерам безопасности.

2 Студенты должны соблюдать правила внутреннего распорядка. Не допускается находиться в классах в верхней одежде, в состоянии алкогольного, токсического или наркотического опьянения.

3 При проведении лабораторных работ необходимо соблюдать правила пожарной безопасности, знать места расположения первичных средств пожаротушения.

Требования безопасности перед началом работы

1 Внимательно изучить содержание и порядок проведения лабораторной работы, а также безопасные приемы ее выполнения.

2 В случае неисправности оборудования немедленно сообщить об этом преподавателю и до ее устранения к работе не приступать (работать на неисправном оборудовании запрещается).

Требования безопасности во время работы

Студенту при работе на ПЭВМ запрещается:

- прикасаться к задней стенке системного блока (процессора) при включенном питании;
- загромождать верхние панели устройств ненужными бумагами и посторонними предметами;
- допускать попадания влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры и другие устройства.

Требования безопасности по окончании работы

- 1 Произвести закрытие всех активных задач.
- 2 Отключить питание системного блока (процессора).
- 3 Осмотреть и привести в порядок рабочее место.
- 4 Предупредить преподавателя обо всех, даже малейших и незначительных, неисправностях оборудования.

1 Лабораторная работа № 1. Проектирование экспертной системы на основе использования метода Байеса

Цель работы: ознакомление с принципами проектирования экспертных систем на основе использования метода Байеса.

Теоретические сведения

Метод Байеса основан на вычислении условной вероятности появления такого события, как диагноз D_i при появлении конкретной реализации комплекса признаков S_j .

Априорную вероятность появления диагноза D_i $P(D_i)$ определяют по статистическим данным на начальном этапе применения метода. Если при обследовании N объектов диагноза установлено, что из них N_i имеют диагноз D_i , то вероятность появления этого диагноза определяется соотношением

$$P(D_i) = \frac{N_i}{N}.$$

Априорную условную вероятность появления признака S_j у объектов, имеющих техническое состояние (диагноз) D_i , $P(S_j/D_i)$ определяют на начальном этапе по имеющимся статистическим данным. Если из N обследованных объектов N_i находилось в диагнозе D_i , а из них N_{ij} объектов имели признак S_j , то условная вероятность появления признака S_j у объектов с диагнозом D_i вычисляется следующим образом:

$$P(S_j / D_i) = \frac{N_{ij}}{N_i}.$$

Априорная вероятность появления признака S_j у всех объектов независимо от их состояния определяется следующим соотношением:

$$P(S_j) = \frac{N_j}{N},$$

где N – количество объектов независимо от их технического состояния;

N_j – количество объектов, у которых был обнаружен признак S_j .

Условную вероятность существования диагноза D_i при обнаружении признака S_j можно определить, используя формулу Байеса

$$P(D_i | S_j) = P(D_i) \frac{P(S_j | D_i)}{P(S_j)}.$$

Для принятия решения о диагнозе при использовании набора (комплекса) признаков S^* применяется обобщенная формула Байеса

$$P(D_i / S^*) = P(D_i) \frac{\prod_{j=1}^k P(S_j / D_i)}{\sum_{i=1}^n P(D_i) \prod_{j=1}^k P(S_j / D_i)},$$

где k – количество признаков;
 n – количество диагнозов.

Задание

По выданным преподавателем результатам наблюдений за техническими объектами построить систему распознавания их состояний по методу Байеса.

Порог принятия гипотезы выбрать равным 0,7.

Порядок выполнения работы

1 Разместить исходные данные на рабочем листе *MS Excel* (рисунок 1.1).

| | | | | | | | | | |
|----|-------------------------|-------|----------|--------|-------|--------------------|-------------|-----------|-----------|
| 2 | Всего наблюдений | | | | | Результаты расчета | | | |
| 3 | 40 | | | | | | D1 | D2 | D3 |
| 4 | | | | | | P(Di/s1) | 0,333333333 | 0,6666667 | 0 |
| 5 | | | | | | P(Di/notS1) | 0,666666667 | 0,3333333 | 1 |
| 6 | n(D1) | n(D2) | n(D3) | n(S1) | n(S2) | P(Di/s2) | 0,142857143 | 0,5714286 | 0,2857143 |
| 7 | 2 | 6 | 32 | 3 | 7 | P(Di/notS2) | 0,857142857 | 0,4285714 | 0,7142857 |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | P(D1) | P(D2) | P(D3) | P(S1) | P(S2) | | D1 | D2 | D3 |
| 11 | 0,05 | 0,15 | 0,8 | 0,075 | 0,175 | P(Di/S1S2) | 0,272727273 | 0,7272727 | 0 |
| 12 | | | | | | P(Di/notS1 S2) | 0,096774194 | 0,516129 | 0,3870968 |
| 13 | Диагностическая матрица | | | | | P(Di/S1 notS2) | 0,428571429 | 0,5714286 | 0 |
| 14 | | D1 | D2 | D3 | | P(Di/notS1 notS2) | 0,942408377 | 0,0418848 | 0,9424084 |
| 15 | P(S1/Di) | 0,5 | 0,333333 | 0 | | | | | |
| 16 | P(notS1/Di) | 0,5 | 0,666667 | 1 | | | | | |
| 17 | P(S2/Di) | 0,5 | 0,666667 | 0,0625 | | | | | |
| 18 | P(notS2/Di) | 0,5 | 0,333333 | 0,9375 | | | | | |
| 19 | | | | | | | | | |

Рисунок 1.1 – Пример расчетного листа в *MS Excel*

2 Рассчитать априорные вероятности диагнозов и признаков, определив предварительно число объектов наблюдения, имеющих соответствующие признаки.

3 Рассчитать вероятности принадлежности объектов диагнозам $D_1 \dots D_n$ по наблюдаемым единичным признакам $S_1 \dots S_k$, а также по их отсутствию $\bar{S}_1 \dots \bar{S}_k$.

4 Рассчитать вероятности принадлежности объектов диагнозам $D_1 \dots D_n$ по наблюдаемым наборам признаков $S_1 \dots S_k$, а также по их отсутствию $\bar{S}_1 \dots \bar{S}_k$.

5 Определить наиболее информативные комплексы признаков, превышающие порог принятия гипотезы.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Априорные вероятности диагнозов и признаков.
- 4 Вероятности принадлежности объектов диагнозам $D_1 \dots D_n$ по наблюдаемым единичным и наборам признаков.
- 5 Рабочий лист *MS Excel* с результатами расчетов.
- 6 Выводы.

Контрольные вопросы

- 1 Каковы достоинства и недостатки метода Байеса?
- 2 Какие комплексы признаков наиболее информативны для метода Байеса?
- 3 Как строятся решающие правила по методу Байеса?

2 Лабораторная работа № 2. Построение решающих правил

Цель работы: ознакомление с принципами построения решающих правил на основе коэффициентов определенности.

Теоретические сведения

Решающее правило – это метод или модель, позволяющие выбрать решение, наиболее предпочтительное в каком-либо смысле.

Коэффициент определенности условия эквивалентен вероятности того, что посылка (условие) является истинной.

Рассмотрим правило

ЕСЛИ S , ТО D ,

где S – условие;

D – заключение.

Будем считать, что условие, правило и заключение не точны и заданы с некоторым коэффициентом определенности.

Тогда коэффициент определенности для заключения

$$ct(D) = ct(S) \cdot ct(\text{правила}),$$

где $ct(D)$ – коэффициент определенности заключения D ;
 $ct(S)$ – коэффициент определенности условия S ;
 $ct(\text{правила})$ – коэффициент определенности правила.

Все коэффициенты определенности (заключения, правила, условия) могут принимать значения от -1 до 1 :

1 – условие полностью определено (верно);

0 – отсутствуют знания об условии;

-1 – условие полностью неопределенно (неверно).

Условия в правиле могут быть связаны логическими операциями «и», «или», «не», например:

ЕСЛИ (S_1) или $(S_2$ и $S_3)$, ТО D ;

ЕСЛИ $(S_1$ и $S_2)$ и $((\text{не } S_3)$ или $S_4)$, ТО D .

При этом эквиваленты логических операций имеют вид:

$$ct(S_1 \text{ и } S_2) = \min \{ct(S_1); ct(S_2)\};$$

$$ct(S_1 \text{ или } S_2) = \max \{ct(S_1); ct(S_2)\};$$

$$ct(\text{не } S) = -ct(S).$$

Довольно часто одно и то же заключение поддерживается несколькими независимыми правилами. При этом расчет выполняется следующим образом.

Допустим, известно два правила:

ЕСЛИ (S_1) , ТО (D) ;

ЕСЛИ (S_2) , ТО (D) .

Общий коэффициент определенности заключения D

$$ct(D) = ct^1(D) + ct^2(D) - ct^1(D) \cdot ct^2(D),$$

где $ct^1(D)$ – коэффициент определенности для первого правила;

$ct^2(D)$ – коэффициент определенности для второго правила.

Существуют заключения, подтверждаемые не только двумя, но и большим количеством правил. Допустим, известно, что

ЕСЛИ (S_1) , ТО D ;

ЕСЛИ (S_2) , ТО D ;

ЕСЛИ (S_3) , ТО D .

Обозначим коэффициент определенности для первого заключения – $ct^1(D)$, для второго – $ct^2(D)$, для третьего – $ct^3(D)$.

Можно рассчитать коэффициент определенности для заключения исходя из первых двух правил:

$$ct\phi(D) = ct^1(D) + ct^2(D) - ct^1(D) \cdot ct^2(D).$$

Объединяя полученное выражение с последним, получаем окончательно

$$ct(D) = ct\phi(D) + ct^3(D) - ct\phi(D) \cdot ct^3(D).$$

Задание

По вариантам, выданным преподавателем, построить на основе коэффициентов определенности систему для распознавания состояний технических объектов.

Порядок выполнения работы

1 Разместить данные на рабочем листе *MS Excel* (рисунок 2.1).

| | A | B | C | D | E | F | G | H | I |
|----|-----|-----------|-----------|-----------|----------|----------|----------|----------|------|
| 1 | № | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | S1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | S2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | С |
| 4 | S3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | S4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | С |
| 6 | D1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | D2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | С |
| 8 | D-? | 0,9746875 | 0,9746875 | 0,9746875 | 0,974688 | 0,974688 | 0,974688 | 0,974688 | 0,97 |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |
| 11 | | | | | | | | | |
| 12 | | | | | | | | | |
| 13 | | S1 | S2 | S3 | S4 | п общ | | | |
| 14 | D1 | | | | | | | | |
| 15 | D2 | | | | | | | | |
| 16 | | | | | | | | | |
| 17 | МД | | | | | | | | |
| 18 | МНД | | | | | | | | |
| 19 | | | | | | | | | |
| 20 | CF | | | | | | | | |

Рисунок 2.1 – Пример расчетного листа в *MS Excel*

2 Рассчитать значения коэффициентов определенности для каждого из признаков.

3 Проверить качество полученной системы коэффициентов определенности, оценив с их помощью принадлежности объектов нескольким состояниям. В качестве порога для принятия гипотезы выбрать положительное (отрица-

тельное) значение коэффициента определенности для комбинации признаков.

4 Определить величину ошибки системы распознавания, если для принятия гипотезы используется значение коэффициента определенности для комбинации признаков, превышающее 0,7.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Исходные данные.
- 4 Рабочий лист *MS Excel* с полученными результатами.
- 5 Выводы.

Контрольные вопросы

- 1 Что такое решающее правило?
- 2 В каком случае решающее правило не может быть построено?
- 3 При каких условиях можно применять решающее правило?

3 Лабораторная работа № 3. Построение нечетких множеств

Цель работы: закрепление знаний о нечетких множествах; приобретение навыков проведения нечетких операций и построения нечетких отношений.

Теоретические сведения

Нечеткое множество представляет собой совокупность элементов произвольной природы, относительно которых нельзя с полной определенностью утверждать, принадлежит ли тот или иной элемент рассматриваемой совокупности данному множеству или нет. Другими словами, нечеткое множество отличается от обычного множества тем, что для всех или части его элементов не существует однозначного ответа на вопрос: «Принадлежит или не принадлежит тот или иной элемент рассматриваемому нечеткому множеству».

Функцией принадлежности нечеткого множества A называют функцию $\mu_A(x)$, которая:

– устанавливает значения степени принадлежности нечеткому множеству каждого из элементов универсального множества и принимает значения на отрезке $[0; 1]$;

– позволяет вычислить степень принадлежности произвольного элемента универсального множества к нечеткому множеству A .

Нечёткое множество можно записывать в виде $\{\mu_A(x) / x\}$.

В практических приложениях теории нечетких множеств используется большое количество различных типов функций принадлежности. Наиболее

распространенные получили треугольная, трапециевидная и гауссова функции принадлежности.

Треугольная функция принадлежности (рисунок 3.1) определяется тройкой чисел (a, b, c) и ее значение в точке x вычисляется согласно выражению

$$\mu(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b, \\ 1 - \frac{x-b}{c-b}, & b \leq x \leq c, \\ 0, & \text{в остальных случаях.} \end{cases}$$

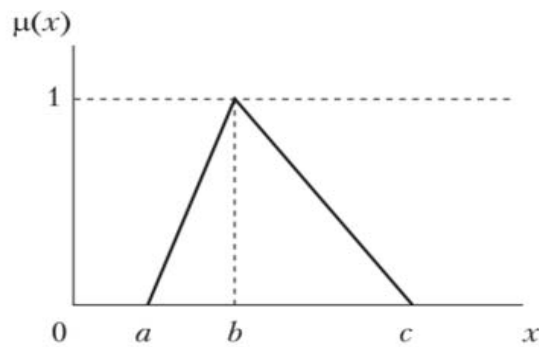


Рисунок 3.1 – График треугольной функции принадлежности

Для задания трапециевидной функции принадлежности (рисунок 3.2) используется четверка чисел (a, b, c, d) .

$$\mu(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ 1 - \frac{x-c}{d-c}, & c \leq x \leq d, \\ 0, & \text{в остальных случаях.} \end{cases}$$

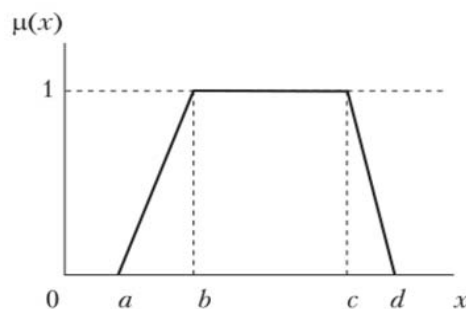


Рисунок 3.2 – График трапециевидной функции принадлежности

Функция принадлежности гауссова типа описывается зависимостью

$$\mu(x) = e^{-\left(\frac{x-c}{\sigma}\right)^2}.$$

Данная кривая оперирует двумя параметрами: c и σ . Параметр c обозначает центр нечеткого множества, а параметр σ отвечает за крутизну функции или ее рассеяние относительно центра.

На рисунке 3.3 приведены две гауссовы функции принадлежности. Центры обеих функций одинаковые ($c = 4$), а рассеяния разные.

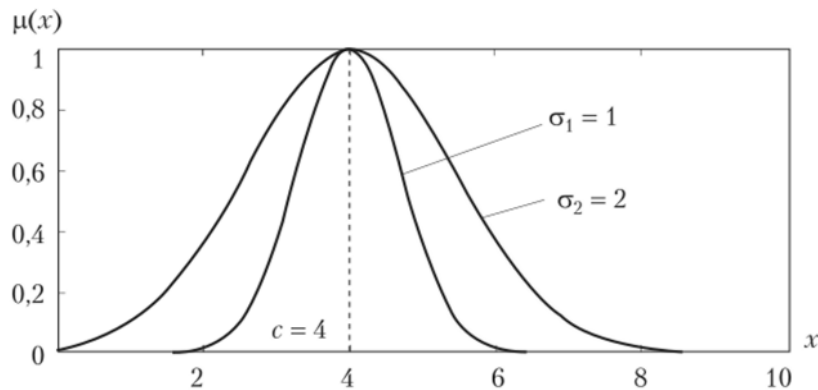


Рисунок 3.3 – Графики гауссовых функций принадлежности

Операции с нечеткими множествами:

- 1) включение;
- 2) равенство;
- 3) дополнение;
- 4) пересечение;
- 5) объединение и др.

Нечеткое множество A содержится в B ($A \subset B$), если функция принадлежности элементов $\mu_A(x) \leq \mu_B(x)$.

Нечеткое множество A равно B , если $\mu_A(x) = \mu_B(x)$.

Нечеткие множества A и B дополняют друг друга, если $\mu_A(x) = 1 - \mu_B(x)$ (рисунок 3.4).

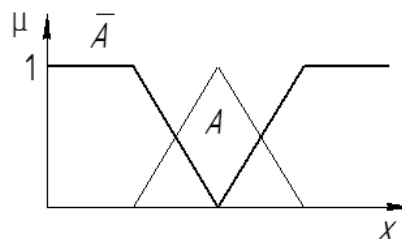


Рисунок 3.4 – Нечеткое множество и его дополнение

Нечеткое множество A пересекает B – нечеткое подмножество, содержащее одновременно в A и в B (рисунок 3.5).

$$\mu A \cap B(x) = \min(\mu A(x), \mu B(x)).$$

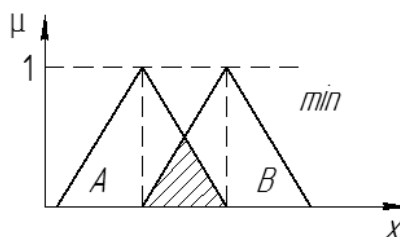


Рисунок 3.5 – Графическое изображение операции пересечения нечетких множеств

Объединение нечетких множеств $A \cup B$ – это нечеткое множество, включающее как A , так и B (рисунок 3.6).

$$\mu A \cup B(x) = \max(\mu A(x), \mu B(x)).$$

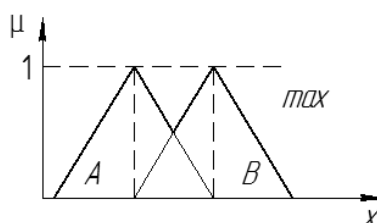


Рисунок 3.6 – Графическое изображение операции объединения нечетких множеств

Задание

По вариантам, выданным преподавателем, построить функции принадлежности нечетких множеств для системы управления технологической системой.

Порядок выполнения работы

- 1 Определить входные и выходные лингвистические переменные.
- 2 Для всех лингвистических переменных представить по три неточные формулировки в виде нечетких множеств.
- 3 Построить графики функций принадлежности нечетких множеств для всех лингвистических переменных. Изобразить на одном графике функции принадлежности нечетких подмножеств.
- 4 Записать математические выражения функций принадлежности в виде кусочно-линейных функций.
- 5 Записать нечеткие подмножества лингвистических переменных в дискретном виде в форме объединения пар вида $(\mu_A(x)/x)$.
- 6 Выполнить операции дополнения нечеткого множества, операции пересечения и объединения нечетких множеств одной из лингвистической переменной и заполнить таблицу 3.1.
- 7 Провести сравнительный анализ результатов.

Таблица 3.1 – Операции с нечеткими множествами

| Вид операции | Выполнение операции | |
|--------------------------------|--------------------------------|-----------------|
| | Математическая запись операции | Графический вид |
| Дополнение нечеткого множества | | |
| Пересечение нечетких множеств | | |
| Объединение нечетких множеств | | |

Содержание отчета

- 1 Название и цель работы.
- 2 Исходные данные.
- 3 Графики функций принадлежности нечетких подмножеств лингвистических переменных.
- 4 Математические выражения функций принадлежности в виде кусочно-линейных функций.
- 5 Результаты операции дополнения, пересечения и объединения нечетких подмножеств, заполненная таблица 3.1.
- 6 Выводы.

Контрольные вопросы

- 1 Чем отличается нечеткое множество от обычного?
- 2 Какие функции нечеткости используют на практике?
- 3 Что такое функция принадлежности?

4 Лабораторная работа № 4. Нечеткий вывод

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

Задание

Изучить и выполнить процедуры фаззификации, нечеткого логического вывода, дефаззификации, применяя алгоритмы Мамдани, Ларсена и функции принадлежности, построенные в лабораторной работе № 3.

Теоретические сведения

Понятие нечеткого вывода занимает центральное место в теории нечеткого управления. На рисунке 4.1 представлена типовая структура модуля нечеткого управления, который состоит из следующих компонентов: база правил, блоки

фаззификации, выработки решений (нечеткого логического вывода), дефаззификации.



Рисунок 4.1 – 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 нечеткого 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.

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 рисунке 4.2 показан пример процедуры фаззификации.



Рисунок 4.2 – Пример работы блока фаззификации

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.

Агрегирование – процедура определения степени выполнения (истинности) условий каждого отдельного правила. Если условие нечеткого продукционного правила типа SISO является простым нечетким высказыва-

нием, то степень его истинности в процедуре агрегирования равна значению ФП соответствующего термина ЛП (рисунок 4.3).

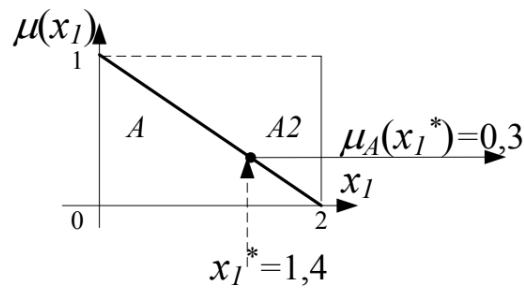


Рисунок 4.3 – Агрегирование для простого условия

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.

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 и x_2 определяется с помощью любого из операторов T -нормы, например, \min .

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

Условия могут иметь и более сложную форму – состоять из множества подусловий, связанных союзами И, ИЛИ. В этом случае вначале следует выполнять все операции пересечения И, а затем все операции объединения ИЛИ.

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

Аккумуляция в системах НЛВ – это процедура объединения, или аккумуляция, 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).

Под дефазификацией 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 лингвистических переменных, получить количественное значение каждой из выходных переменных, которое может быть использовано исполнительными механизмами системы управления, внешними по отношению к системе нечеткого вывода.

Существуют различные методы дефаззификации, наиболее известными среди которых являются (рисунок 4.4):

– COG (Center of Gravity) – метод центра тяжести, в котором определяется абсцисса центра тяжести фигуры, ограниченной графиком ФП;

– 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.

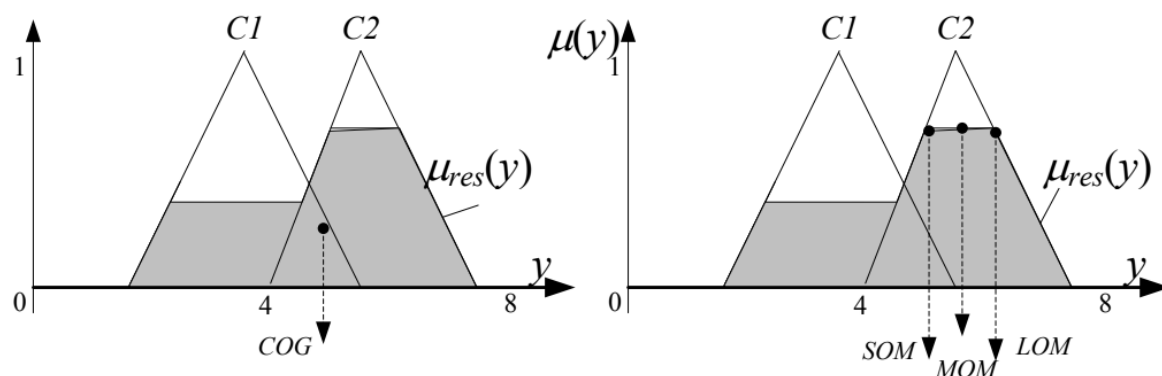


Рисунок 4.4 – Иллюстрация методов дефаззификации

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

P_1 : ЕСЛИ ($x_1 = A_1$) И ($x_2 = B_1$) ТО ($y = C_1$);

P_2 : ЕСЛИ ($x_1 = A_1$) И ($x_2 = B_2$) ТО ($y = C_2$),

где x_1, 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 .

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 таблице 4.1.

Алгоритм Ларсена (см. таблицу 4.1) 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 4.1 – Алгоритмы Мамдани и Ларсена

| Шаг алгоритма | Описание |
|------------------------|---|
| Фаззификация | Определяются степени истинности предпосылок каждого правила: $\mu_{A1}(x1^*), \mu_{A2}(x1^*), \mu_{B1}(x2^*), \mu_{B2}(x2^*)$ |
| Агрегирование | Определяются уровни «отсечения» для предпосылок каждого из правил с использованием операции «min»: $\alpha_1 = \mu_{A1}(x1^*) \wedge \mu_{B1}(x2^*);$ $\alpha_2 = \mu_{A2}(x1^*) \wedge \mu_{B2}(x2^*)$ |
| Активизация по Мамдани | Определяются «усеченные» результирующие ФП на основе min-активизации: $\mu_{C1^*}(y) = \alpha_1 \wedge \mu_{C1}(y);$ $\mu_{C2^*}(y) = \alpha_2 \wedge \mu_{C2}(y)$ |
| Активизация по Ларсену | Определяются «усеченные» результирующие ФП на основе prod-активизации: $\mu_{C1^*}(y) = \alpha_1 \cdot \mu_{C1}(y);$ $\mu_{C2^*}(y) = \alpha_2 \cdot \mu_{C2}(y)$ |
| Аккумуляция | С использованием операции max производится объединение найденных усеченных ФП, что приводит к получению итогового нечеткого подмножества для выходной переменной: $\mu_{\Sigma}(y) = \mu_{C1^*}(y) \vee \mu_{C2^*}(y) = (\alpha_1 \wedge \mu_{C1}) \vee (\alpha_2 \wedge \mu_{C2}(y))$ |
| Дефаззификация | Нахождение выходной переменной y_0 проводится любым из методов дефаззификации, например методом центра тяжести |

Порядок выполнения работы

- 1 Определить активные правила модели.
- 2 Алгоритмами Мамдани и Ларсена выполнить этапы фаззификации, агрегирования, активизации, аккумуляирования.
- 3 Методами дефаззификации найти точные значения на выходе модуля нечеткого управления.
- 4 Провести исследования по поиску результирующих значений, применяя различные методы дефаззификации.
- 5 Обосновать окончательное решение. Процедуру вывода провести в графическом и математическом виде.

Содержание отчета

- 1 Название и цель работы.
- 2 Исходные данные.

3 Процедура нечеткого логического вывода в графическом и математическом виде.

4 Таблицы точных значений для алгоритмов Мамдани и Ларсена.

5 Выводы.

Контрольные вопросы

1 В чем состоит суть нечеткого управления?

2 Назовите этапы нечеткого логического вывода.

3 Чем отличается алгоритм Мамдани от алгоритма Ларсена?

4 Поясните содержание алгоритма Такаги – Сугено.

5 Что такое фаззификация и дефаззификация?

6 Раскройте содержание методов дефаззификации.

5 Лабораторная работа № 5. Метрическое распознавание образов

Цель работы: ознакомление с методами метрического распознавания образов, приобретение навыков применения алгоритмов метрического распознавания образов.

Теоретические сведения

Распознавание образов – это отнесение исходных данных к определённому классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы данных.

Образ (класс) – классификационная группировка в системе классификации, объединяющая (выделяющая) определенную группу объектов по некоторому признаку.

Условно классические методы распознавания образов можно разделить на следующие группы:

- методы эталонов – объекты каждого класса занимают достаточно компактную область в пространстве признаков;

- методы ближайших соседей (lazy learning) – почти не учатся, а просто помнят все обучающие примеры;

- разделяющие поверхности – между объектами, которые отделяют одни классы от других;

- вероятностные методы – учитывают количественный состав обучающих примеров различных классов и плотности вероятностей распределения их в пространстве;

- деревья принятия решений – могут эффективно учитывать скачкообразный характер принадлежности к классу при изменении признаков;

- композиция методов – улучшение качества распознавания в результате объединения различных методов.

Метод эталонов. Центр масс всех объектов данного класса иногда характеризует типичный «образ» класса. Образ (точка в пространстве признаков) может не совпадать ни с одним из обучающих примеров (рисунок 5.1).

Для получения подобного эталонного представителя класса вычисляется простое среднее по всем обучающим примерам $i = 1, \dots, N_k$ данного класса k :

$$c_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i, \text{ для } x_i \in \text{классу } k.$$

Процесс классификации сводится к поиску эталона, который находится ближе всего к тестовой точке:

$$d(x, c_k) = \min \rightarrow x \in \text{классу } k. \quad (5.1)$$

В качестве $d(x, c_k)$ можно взять любое метрическое расстояние, например евклидово:

$$d(x, c_k) = |x - c_k| = \sqrt{(x - c_k)^2}.$$

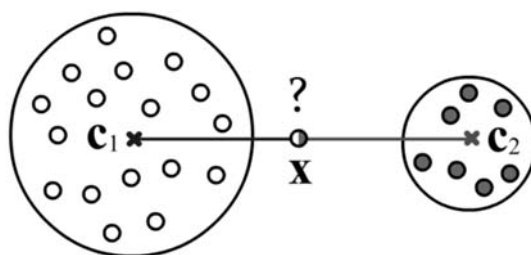


Рисунок 5.1 – Иллюстрация метода эталонов

Сферы эталонов обычно имеют различный размер. Радиус сферы R_k можно определять как среднее расстояние или как квадратный корень среднего квадратов расстояний объектов класса от центра эталона:

$$R_k^2 = \frac{1}{N_k} \sum_{i=1}^{N_k} (x_i - c_k)^2, \text{ для } x_i \in \text{классу } k.$$

В качестве радиуса сферы можно также использовать расстояние максимально удалённого от центра c_k объекта.

Если эталон k -го класса характеризуется положением c_k и радиусом сферы R_k , то можно уточнить решающее правило (5.1). Например, пусть объекты двух классов в пространстве признаков занимают сферические области различных радиусов. Если классифицируемый объект находится на равных расстояниях от центров сфер, то его стоит отнести к тому классу, расстояние до поверхности сферы которого минимально. Так, на рисунке точку x следует

считать белым, а не чёрным объектом. В связи с этим при равномерном распределении точек внутри n -мерной сферы, при больших n , случайно выбранная точка почти наверняка окажется возле поверхности сферы.

Возможны следующие меры близости к эталону, которые используются в классификационном правиле:

$$\frac{d(x, c_k)}{R_k} = \min,$$

или

$$d(x, c_k) = \begin{cases} 0, & |x - c_k| \leq R_k, \\ |x - c_k| - R_k, & |x - c_k| > R_k \end{cases} = \min.$$

Второе правило предполагает, что, если точка попала в сферу эталона (всё равно на каком расстоянии от центра), то она принадлежит этому классу (расстояние нулевое), иначе берётся расстояние до поверхности сферы.

Метод ближайших соседей. Цель методов эталонов состоит в уменьшении числа входных примеров (сжатие информации) путём получения небольшого количества интегральных, эталонных образов. Противоположными к методам эталонов (в смысле сжатия информации) является группа методов, в которых запоминают всю обучающую выборку.

При классификации данного объекта x изучают его непосредственное окружение в обучающей выборке и на его основании делают вывод о принадлежности примера к тому или иному классу.

Возможны различные модификации метода. В простейшем случае ищут ближайшую точку обучающего множества и берут при распознавании её класс. При наличии шума лучше отобрать k ближайших соседей

$$d(x, x_1) \leq d(x, x_2) \leq \dots \leq d(x, x_m) \quad d(x, x_1) \leq d(x, x_2) \leq \dots \leq d(x, x_m)$$

и использовать класс, встречающийся чаще всего.

Другая разновидность метода состоит в использовании соседей, попавших внутрь сферы некоторого радиуса (рисунок 5.2).

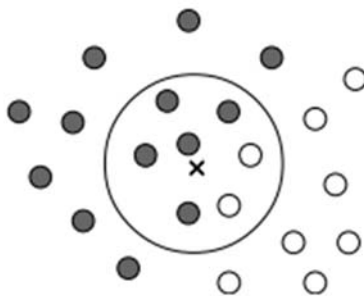


Рисунок 5.2 – Иллюстрация метода ближайших соседей

Подбор параметров (число соседей, радиус) можно проводить методом LOO (leave-one-out). Для этого метод соседей применяется по очереди к каждому объекту обучающей выборки и вычисляется ошибка классификации.

Метод ближайших соседей при отсутствии ограничений на память и быстродействие является достаточно мощным при очевидной простоте его реализации. Однако он не обладает способностью к обобщению (сглаживанию).

Метод потенциальных функций является разновидностью метода ближайших соседей, в котором вес класса соседа зависит от его расстояния до данной точки. Для k ближайших соседей или для всех примеров данного класса вычисляется значение функции

$$\phi_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \phi(x - x_i), \quad x \in \text{классу } k,$$

где суммирование ведётся по примерам k -го класса, а в качестве «потенциалов», создаваемых каждым примером $\phi(x - x_i)$, могут выступать функции

$$\phi(r) = \frac{\phi_0}{1 + (ar)^2}; \quad \phi(r) = \phi_0 e^{-(ar)^2}; \quad \phi(r) = \frac{\phi_0}{e^{-ar} + e^{ar}},$$

где a, ϕ_0 – параметры функций.

Подобные потенциальные функции строятся независимо для каждого класса. В качестве правила распознавания выбирается тот класс, чья потенциальная функция в данной точке имеет наибольшее значение.

Отбор эталонных соседей. Не все объекты обучающего множества одинаково важны при классификации методом ближайших соседей или потенциальных функций. Часть объектов может оказаться шумом. Другая часть – избыточно сконцентрированными в некоторой области пространства признаков. Поэтому имеет смысл уменьшать обучающую выборку, формируя значимое обучающее подмножество эталонных объектов.

Рассмотрим два жадных алгоритма формирования подмножества эталонных объектов.

Отбрасывание худших. Перебирают все объекты обучающего множества. Текущий объект выбрасывают и методом LOO (leave-one-out) вычисляют ошибку классификации остальных объектов. Выброшенный объект возвращают и повторяют процедуру. После окончания перебора исключают из обучающей выборки объект, отбрасывание которого приводило к минимальному изменению ошибки. Затем процедура повторяется, пока LOO-ошибка не начинает заметно расти. Оставшиеся в выборке объекты и являются эталонными.

Добавление лучших. Аналогично, но в множество эталонных объектов (первоначально пустое) добавляется объект, LOO-ошибка исключения которого была максимальна.

На рисунке 5.3 представлена модельная обучающая выборка для двух классов в двумерном пространстве признаков. Это пространство разбивается на две эталонные области, в которых в основном находятся объекты одного класса. В этих областях вкраплены шумовые объекты (один чёрный объект в классе белых и один белый в классе чёрных). Кроме этого, есть пограничная зона, в которой объекты двух классов сильно перемешаны. Такие объекты будут не информативны при распознавании. Шумовые вкрапления не только не информативны, но и вредны. Их исключение приведёт к уменьшению LOO-ошибки, т. к. их соседи перестанут неправильно классифицироваться.

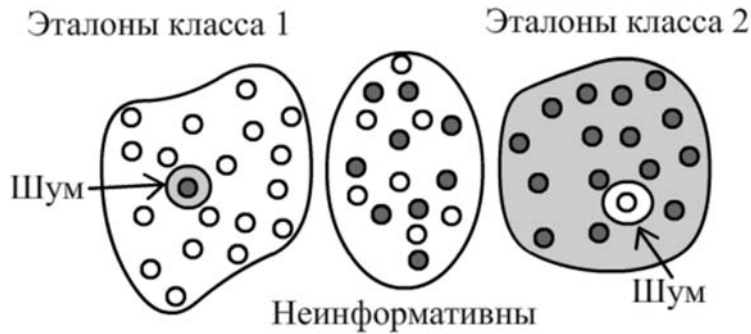


Рисунок 5.3 – Модельная обучающая выборка

На рисунке 5.4 представлен график LOO-ошибки как функции числа отброшенных объектов в алгоритме «отбрасывание худших». В начале отбрасываются те объекты, которые заведомо являются шумовыми, и ошибка снижается (перестают ошибаться соседи шумовых «вкраплений»). Затем очередь доходит до неинформативных объектов, присутствие которых в обучающей выборке пользы для распознавания не приносит. Для них ошибка классификации выходит на плато. Наконец, когда алгоритм добирается до значимых (эталонных) объектов, график ошибки резко начинает расти. В этот момент необходимо остановиться.



Рисунок 5.4 – График метода LOO (leave-one-out)

Если эталонных объектов сравнительно не много, быстрее будет работать метод «добавления лучших». В противном случае эффективнее метод «отбрасывания худших».

Задание

По вариантам, выданным преподавателем, определить состояние объектов на основе методики метрического распознавания образов – метода ближайших соседей.

Порядок выполнения работы

- 1 Изучить особенности методов распознавания образов.
- 2 Построить графическое изображение исследуемых объектов и обучающих множеств.
- 3 Изучить непосредственное окружение исследуемых объектов в обучающих выборках.
- 4 Определить расстояния до ближайших точек обучающих множеств.
- 5 Сделать вывод о принадлежности исследуемых объектов к тому или иному классу.

Содержание отчета

- 1 Название и цель работы.
- 2 Исходные данные.
- 3 Графическое изображение исследуемых объектов и обучающих множеств.
- 4 Результаты расчетов.
- 5 Выводы.

Контрольные вопросы

- 1 В чем заключается процедура метрического распознавания образов?
- 2 Что такое метрика пространства?
- 3 Как влияет метрика пространства на надежность распознавания?

6 Лабораторная работа № 6. Обучение в интеллектуальных системах

Цель работы: приобретение навыков применения алгоритмов обучения в интеллектуальных системах.

Теоретические сведения

Обучение осуществляется путем показа системе отдельных объектов. В результате распознающая система должна приобрести способность реагировать одинаковыми реакциями на все объекты одного класса и различными – на все объекты различных классов. Процесс обучения должен завершиться только путем показов конечного числа объектов без каких-либо

других подсказок.

За обучением следует процесс идентификации новых объектов, который характеризует действия уже обученной системы. Автоматизация этих процедур и составляет проблему обучения распознаванию образов.

Любое изображение, которое возникает в результате наблюдения какого-либо объекта в процессе обучения, можно представить в виде вектора, а значит, и в виде точки некоторого пространства признаков.

Если утверждается, что при показе изображений возможно однозначно отнести их к одному из двух (или нескольких) образов, то тем самым утверждается, что в некотором пространстве существует две (или несколько) области, не имеющие общих точек, и что изображения – точки из этих областей.

Проинтерпретируем в терминах геометрической картины процесс обучения распознаванию образов, ограничившись случаем распознавания только двух образов.

Заранее считается известным лишь то, что требуется разделить две области в некотором пространстве и что показывааются точки только из этих областей. Сами области заранее не определены, т. е. нет никаких сведений о расположении их границ или правил определения принадлежности точки к той или иной области.

В ходе обучения предъявляются точки, случайно выбранные из этих областей, и сообщается информация о том, к какой области принадлежат предъявляемые точки. Никакой дополнительной информации об этих областях, т. е. о расположении их границ, в ходе обучения не сообщается.

Цель обучения состоит либо в построении поверхности, которая разделяла бы не только показанные в процессе обучения точки, но и все остальные точки, принадлежащие этим областям, либо в построении поверхностей, ограничивающих эти области так, чтобы в каждой из них находились только точки одного образа.

Иначе говоря, цель обучения состоит в построении таких функций от векторов-изображений, которые были бы, например, положительны на всех точках одного и отрицательны на всех точках другого образа.

В связи с тем, что области не имеют общих точек, всегда существует целое множество таких разделяющих функций, а в результате обучения должна быть построена одна из них (рисунок 6.1).

Если предъявляемые изображения принадлежат не двум, а большему числу образов, то задача состоит в построении по показанным в ходе обучения точкам поверхности, которая отделила бы все области, соответствующие этим образам, друг от друга.

Такая задача может быть решена, например, путем построения функции, принимающей над точками каждой из областей одинаковое значение, а над точками из разных областей значение этой функции должно быть различно.

Алгоритм метода обучения распознаванию образов. Рассмотрим метод обучения, состоящий в итерационной подстройке матрицы весов, последовательно уменьшающей ошибку в выходных векторах.

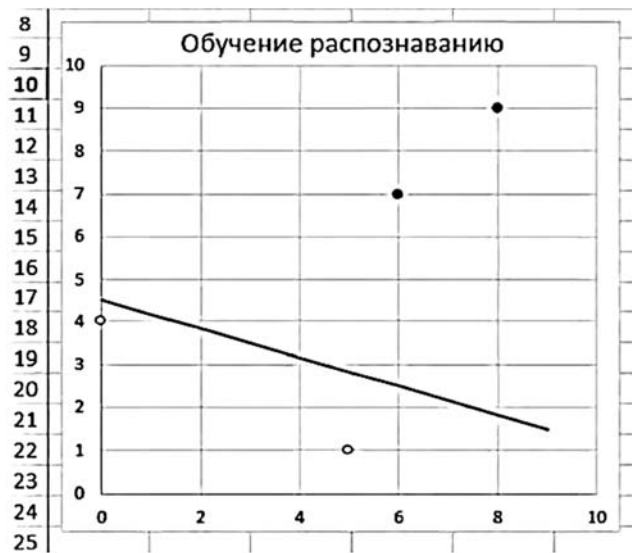


Рисунок 6.1 – Графическое изображение разделяющей функции

Алгоритм включает несколько шагов.

Шаг 1. Начальные значения весов w_0 , w_1 , w_2 дискриминантной функции

$$d = w_0 + w_1x_1 + w_2x_2$$

полагаются случайными.

x_1 и x_2 – признаки.

Шаг 2. Вычисление функции принадлежности ведется по формуле

$$z = \text{sign}(d) = \text{sign}(w_0 + w_1x_1 + w_2x_2),$$

которая равна +1 или -1.

Корректировка весовых коэффициентов ведется по выражению

$$w_i^{n+1} = w_i^n + c \cdot (z_t - z_r)x_i,$$

где c – мера обучения, $c = 0 \dots 1$;

z_t , z_r – требуемое и действительное значения функции принадлежности

$$w_0^{n+1} = w_0^n + c \cdot (z_t - z_r) \cdot 1;$$

$$w_1^{n+1} = w_1^n + c \cdot (z_t - z_r)x_1;$$

$$w_2^{n+1} = w_2^n + c \cdot (z_t - z_r)x_2.$$

Шаг 3. Для $x = x_1$, а $y = x_2$ построить в интервале изменения переменных x_1 в пределах обоих классов соответствующую дискриминантной функции линию

$$y = -\frac{w_0}{w_2} - \frac{w_1}{w_2}x.$$

Шаг 4. Шаги 1–3 повторяются для всех обучающих векторов (признаков). Обучение завершается:

- а) когда итерации сойдутся, то есть вектор весов перестает изменяться;
- б) когда полная просуммированная по всем векторам абсолютная ошибка станет меньше некоторого малого значения.

Задание

По вариантам, выданным преподавателем, обучить систему распознаванию двух классов состояний технических объектов (работоспособное и неработоспособное).

Порядок выполнения работы

1 Рассчитать центроиды классов и найти выражение для дискриминантной функции. Определить принадлежность технических объектов $P_1 \dots P_n$ классам состояний.

2 Используя алгоритм обучения, найти выражение для дискриминантной функции.

3 Сравнить результаты, полученные в пп. 1 и 2.

Содержание отчета

1 Название и цель работы.

2 Исходные данные.

3 Графическое изображение исследуемых объектов и обучающих множеств.

4 Результаты расчетов.

5 Выводы.

Контрольные вопросы

1 В чем заключается смысл обучения распознаванию образов?

2 Что такое дискриминантная функция?

3 В чем заключается проблема линейной делимости образов?

7 Лабораторная работа № 7. Построение нейронных сетей

Цель работы: приобретение практических навыков построения нейронных сетей.

Теоретические сведения

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 рисунке 7.1.

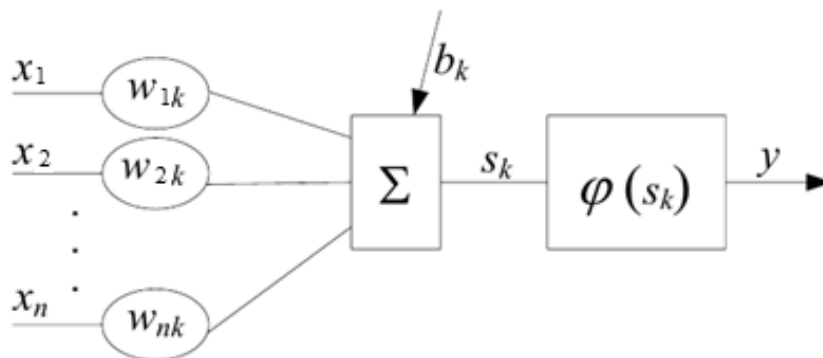


Рисунок 7.1 – Модель искусственного нейрона

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* $\varphi(s_k)$, 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:

$$s_k = \sum_{i=1}^n w_{ik} x_i;$$

$$y_k = \varphi(s_k + b_k).$$

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 $\varphi(s_k)$, 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* (рисунок 7.2), используется в классическом искусственном нейроне для обработки двоичных сигналов. К сожалению, такой вид функции активации не предоставляет ИНС достаточную гибкость при обучении и настройке на заданную задачу, т. к. функция является недифференцируемой:

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

– *сигмоидальная функция* (рисунок 7.3) 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:

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

где a – коэффициент крутизны функции активации (параметр наклона).

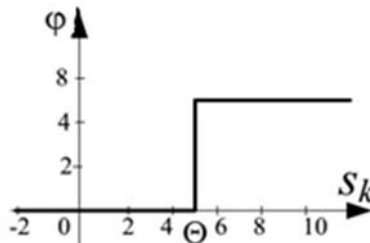


Рисунок 7.2 – Пороговая функция активации

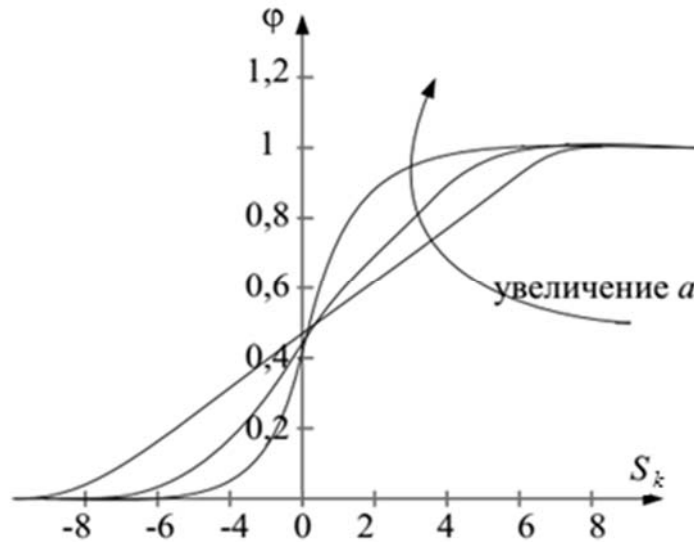


Рисунок 7.3 – Сигмоидальная функция активации

При $a \rightarrow \infty$ сигмоидальная функция вырождается в пороговую, а при $a \rightarrow 0$ становится полой. Область значений $(0;1)$.

Суммарная среднеквадратичная ошибка E между желаемым выходом d_j и фактической реакцией нейронной сети y_j^N ($j = \overline{1, n_N}$) по всем нейронам выходного слоя определяется по формуле

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

Задание

Решить задачу прямого распространения входного примера по искусственной нейронной сети. Рассчитать среднеквадратичную ошибку. В качестве функции активации принять сигмоиду с единичным коэффициентом крутизны.

Имеется двухвходовая однослойная ИНС прямого распространения (однослойный персептрон) (рисунок 7.4), которая состоит из двух нейронов.

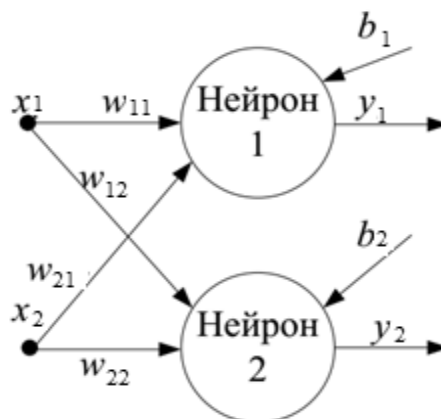


Рисунок 7.4 – Однослойный персептрон

Входом нейронной сети является пример X : $x_1 = 1$, $x_2 = -1$. Целевые (желаемые) значения выходов ИНС $d_1^0 = 0,4$, $d_2^0 = 0,25$. Пороговые значения (смещения) для нейронов $b_1 = b_2 = 0,2$.

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

Порядок выполнения работы

1 Определить взвешенную сумму входных сигналов и выходной сигнал первого нейрона s_1 , y_1 . В качестве функции активации принять сигмоиду с единичным коэффициентом крутизны.

2 Определить взвешенную сумму входных сигналов и выходной сигнал второго нейрона s_2 , y_2 .

3 Определить эталонные (целевые) значения выходов нейронной сети, преобразованные нелинейной функции возбуждения.

4 Определить среднеквадратичную ошибку ИНС.

Содержание отчета

1 Название и цель работы.

2 Исходные данные.

3 Расчет взвешенной суммы входных сигналов и выходной сигнал первого нейрона s_1 , y_1 .

4 Расчет взвешенной суммы входных сигналов и выходной сигнал второго нейрона s_2 , y_2 .

5 Расчет выходов нейронной сети.

6 Расчет среднеквадратичной ошибки ИНС.

7 Выводы.

Контрольные вопросы

1 Какова структура искусственного нейрона?

2 Чем отличаются пороговая и сигмоидальная функции активации?

3 Запишите выражение для коррекции весовых коэффициентов нейронной сети.

8 Лабораторная работа № 8. Методы обучения сетей

Цель работы: приобретение навыков обучения нейронных сетей на основе алгоритма обратного распространения ошибки.

Теоретические сведения

Основным преимуществом нейронных сетей над обычными алгоритмами вычисления является их возможность обучения. В общем смысле слова обучение заключается в нахождении верных коэффициентов связи между нейронами, а также в обобщении данных и выявлении сложных зависимостей между входными и выходными сигналами. Фактически, удачное обучение нейронной сети означает, что система будет способна выявить верный результат на основании данных, отсутствующих в обучающей выборке.

Схема процесса обучения нейронной сети представлена на рисунке 8.1.

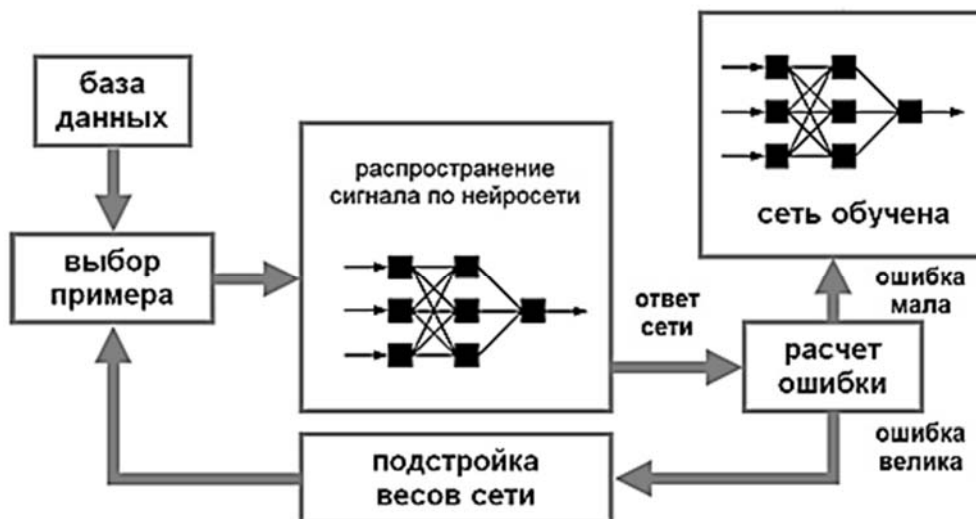


Рисунок 8.1 – Процесс обучения нейронной сети

Искусственные нейронные сети – 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 нейронной сети 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.

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 рисунке 8.2 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_j^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) .

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* (ОРО) (*error back propagation algorithm*) – 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.

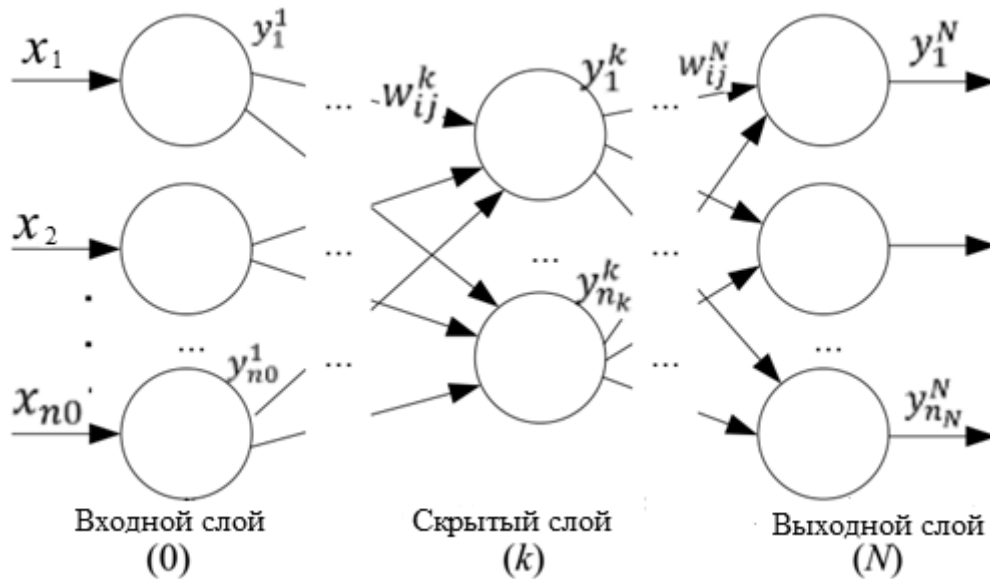


Рисунок 8.2 – Структура многослойного персептрона

Обучение ИНС с использованием алгоритма ОРО проводится в несколько этапов.

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

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:

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

где s_j^k – взвешенная сумма входных сигналов j -го нейрона в слое k ;

y_i^{k-1} – выход i -го нейрона в предыдущем слое $(k-1)$.

Выходной сигнал j -го нейрона k -го слоя

$$y_j^k = f(s_j^k).$$

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

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

Таким образом осуществляется последовательное *прямое распространение* входного обучающего примера X .

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

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

Шаг 5. Обучение ИНС основано на адаптивной коррекции весовых коэффициентов $w_{ij}^k(t)$ таким образом, чтобы минимизировать значение среднеквадратичной ошибки (t – номер итерации обучения).

Коррекция весовых коэффициентов, начиная с выходного слоя сети и по направлению ко входу, проводится по формуле

$$\Delta w_{ij}^k = -\eta \cdot \delta_j^k \cdot y_i^{k-1},$$

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

y_i^{k-1} – выход i -го нейрона в предыдущем слое ($k - 1$).

$$\delta_j^k = \frac{\partial E}{\partial s_j^k} = -(y_j^k - d_j) \cdot \frac{\partial y_j^k}{\partial s_j^k}.$$

Шаг 6. Проверка критерия остановки 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 остановки 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.

Задание

По вариантам, выданным преподавателем, решить задачу обучения искусственной нейронной сети, используя алгоритм обратного распространения ошибки. В качестве функции активации принять сигмоиду с единичным коэффициентом крутизны. Скорость обучения равна 1. При решении задачи выполнить две итерации.

Порядок выполнения работы

1 На первой итерации определить выход нейронной сети путем распространения входного примера по ИНС.

Определить среднеквадратичную ошибку нейронной сети. Затем провести корректировку коэффициентов синаптических связей в направлении, обратном прямому распространению входных сигналов.

2 На второй итерации вновь рассчитать выход нейронной сети путем прямого распространения входного примера по ИНС и определить среднеквадратичную ошибку нейронной сети.

3 Сделать вывод о качестве обучения ИНС по уменьшению ошибки.

Содержание отчета

1 Название и цель работы.

2 Исходные данные.

3 Ход решения задачи обучения нейронной сети методом обратного распространения ошибки.

4 Выполненные две итерации обучения нейронной сети; по каждой итерации представить результаты распространения входного примера по ИНС.

5 Рассчитанные среднеквадратичные ошибки.

6 Результаты корректировки коэффициентов синаптических связей.

7 Выводы.

Контрольные вопросы

1 В чем сущность обучения нейронных сетей?

2 Какие существуют парадигмы обучения нейронных сетей?

3 Что представляет собой модель искусственного нейрона?

4 В чем заключается идея обратного распространения ошибки?

5 Чем определяется скорость обучения в алгоритме обратного распространения ошибки?

Список литературы

1 **Андрейчиков, А. В.** Интеллектуальные информационные системы и методы искусственного интеллекта: учебник / А. В. Андрейчиков, О. Н. Андрейчикова. – Москва : ИНФРА-М, 2023. – 530 с.

2 **Пенькова, Т. Г.** Модели и методы искусственного интеллекта: учебное пособие / Т. Г. Пенькова, Ю. В. Вайнштейн. – Красноярск: Сиб. федер. ун-т, 2019. – 116 с.

3 **Пятаева, А. В.** Интеллектуальные системы и технологии: учебное пособие / А. В. Пятаева, К. В. Раевич. – Красноярск : Сиб. федер. ун-т, 2018. – 144 с.

4 **Сидоркина, И. Г.** Системы искусственного интеллекта: учебное пособие / И. Г. Сидоркина. – Москва : Кнорус, 2017. – 246 с.

5 **Перфильев, Д. А.** Интеллектуальные системы поддержки принятия решений: учебное пособие / Д. А. Перфильев, К. В. Раевич, А. В. Пятаева. – Красноярск : Сиб. федер. ун-т, 2018. – 136 с.