

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

**Л. В. Серебряная, Ф. И. Третьяков**

## ***МЕТОДЫ И АЛГОРИТМЫ ПРИНЯТИЯ РЕШЕНИЙ***

*Рекомендовано УМО по образованию в области  
информатики и радиоэлектроники в качестве учебно-методического  
пособия для специальности 1-40 01 01  
«Программное обеспечение информационных технологий»*

Минск БГУИР 2016

УДК 519.81(075.8)  
ББК 22.18я73  
С32

**Р е ц е н з е н т ы:**

кафедра интеллектуальных систем Белорусского национального технического университета (протокол №4 от 08.12.2014);

старший научный сотрудник государственного научного учреждения  
«Объединенный институт проблем информатики  
Национальной академии наук Беларуси»,  
кандидат физико-математических наук С. В. Чебаков

**Серебряная, Л. В.**

С32      Методы и алгоритмы принятия решений : учеб.-метод. пособие /  
Л. В. Серебряная, Ф. И. Третьяков. – Минск : БГУИР, 2016. – 64 с. : ил  
ISBN 978-985-543-232-7.

Изложены основы теории распознавания образов и искусственных нейронных сетей. Рассмотрены две группы методов распознавания образов, приведены алгоритмы, относящиеся к каждому из методов. Предложены девять лабораторных работ, реализация которых позволит практически закрепить знания по курсу «Методы и алгоритмы принятия решений».

**УДК 519.81(075.8)  
ББК 22.18я73**

**ISBN 978-985-543-232-7**

© Серебряная Л. В., Третьяков Ф. И., 2016  
© УО «Белорусский государственный  
университет информатики  
и радиоэлектроники», 2016

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>1. ОСНОВЫ ТЕОРИИ РАСПОЗНАВАНИЯ ОБРАЗОВ .....</b>	<b>5</b>
1.1. Две группы методов распознавания .....	5
1.2. Обучение в решении задачи классификации.....	5
1.3. Расстояние между классами .....	6
1.4. Решающее правило и разделяющая функция .....	7
1.5. Особенности синтаксических методов.....	11
1.6. Системы распознавания образов .....	12
1.7. Области применения систем распознавания .....	19
1.8. Критерии развития систем распознавания.....	23
1.9. Распознавание образов с помощью искусственных нейронных сетей.....	28
<b>2. ЛАБОРАТОРНЫЙ ПРАКТИКУМ .....</b>	<b>31</b>
2.1. Лабораторная работа №1 .....	31
2.2. Лабораторная работа №2 .....	33
2.3. Лабораторная работа №3 .....	35
2.4. Лабораторная работа №4 .....	38
2.5. Лабораторная работа №5 .....	42
2.6. Лабораторная работа №6 .....	45
2.7. Лабораторная работа №7 .....	48
2.8. Лабораторная работа №8 .....	53
2.9. Лабораторная работа №9 .....	56
<b>ЛИТЕРАТУРА.....</b>	<b>64</b>

## ВВЕДЕНИЕ

Развитие науки и техники привело к тому, что в последние десятилетия во всех сферах нашей жизни появились автоматизированные информационные системы. Одна из функций, возложенных на них, – это автоматическое распознавание образов. Его реализация техническими средствами может быть осуществлена путем моделирования операций, выполняемых живыми организмами в процессе взаимодействия и восприятия окружающего мира. Наиболее естественно положить в основу модели распознавания способности человека и его реакции на окружающую действительность. Дополнительным аргументом в пользу такого подхода явилось стремление возложить функции человека на автоматические устройства в тех областях, где условия работы однообразны, утомительны или опасны для жизнедеятельности людей.

Распознавание как научное направление включает в себя большое число различных дисциплин и использует методы, характерные для каждой из них. Для того чтобы свести воедино отдельные составные части этого направления, можно дать следующее общее определение понятию распознавания образов. Будем полагать, что это есть совокупность методов и средств, позволяющих, как минимум, достигнуть, а если удастся, то и превзойти естественные средства восприятия и анализа окружающего мира живыми организмами.

Процедура восприятия образов предшествует процессу распознавания, после чего следует процесс идентификации. К прикладным задачам, связанным с данным научным направлением, относятся такие, как автоматическое чтение текстов и их синхронный перевод на разные языки, восприятие техническим устройством слитной речи, автоматизация медицинской диагностики, дистанционная идентификация объектов, криминалистика и др.

Если рассматривать проблему распознавания в общем виде, то ее можно сформулировать как задачу разработки процедуры, позволяющей разбивать множество объектов на классы, считая, что разбиение существует. Хотя это не всегда выполнимо, так как не любая задача может быть формализована. Поэтому проблема распознавания в самом общем виде является неразрешимой.

В соответствии с характером распознаваемых образов соответствующие им процедуры можно разделить на два основных типа: *распознавание конкретных объектов* и *распознавание абстрактных объектов*. В данном учебно-методическом пособии рассматриваются вопросы, относящиеся только к *конкретным объектам*.

В процессе распознавания участвует не сам объект, а некоторое его приближение, называемое *образом*. В алгоритмах распознавания образ представляется набором характеристик (признаков). Специфика задач распознавания образов заключается в следующем: с одной стороны, приходится принимать решение в условиях неполноты информации, а с другой – число признаков, характеризующих один объект, может быть очень большим. Поэтому требуется выбрать ограниченное количество признаков, с помощью которых удастся принять максимально верное решение.

# 1. ОСНОВЫ ТЕОРИИ РАСПОЗНАВАНИЯ ОБРАЗОВ

## 1.1. Две группы методов распознавания

Несмотря на многообразие методов распознавания образов, их можно разделить на две группы. Первая основана на понятии пространства признаков и их обработки в этом пространстве. Вторая – на исследовании конструкции рассматриваемых образов (синтаксическое распознавание).

Для первой группы методов в качестве основополагающей принята гипотеза о возможности представления образа в виде вектора, принадлежащего множеству  $V$ . Множество векторов, состоит из  $N$  таких подмножеств, что каждый вектор, отнесенный в результате классификации к  $j$ -му классу, принадлежит подмножеству  $E_j$ . Свойства множества  $V$  могут быть записаны в виде

$$\bigcup_{i=1}^N E_i = V, E_i \cap E_j = \emptyset (\forall i \neq j).$$

Синтаксический метод распознавания основан на восприятии основных элементов языка – *примитивов*. Они делятся на еще более мелкие составляющие – *символы*, являющиеся наименьшими элементами языка. Множество используемых символов называется *алфавитом* или *словарем*. Язык создается не только с помощью алфавита символов. Правила построения, преобразования и взаимодействия слов определяются *грамматикой*. Она представляет собой множество правил, по которым строятся фразы, а следовательно, и сам язык.

Формально грамматика может быть задана следующей записью:

$$G = \langle V_n, V_t, P, S \rangle,$$

где  $V_n$  – нетерминальный словарь;  $V_t$  – терминальный словарь;  $P$  – множество правил подстановки;  $S$  – начальная аксиома ( $S \in V_n$ ).

Для грамматики характерны следующие соотношения:  $V = V_n \cup V_t$  – словарь,  $V_n \cap V_t = \emptyset$ ,  $P = \{\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \dots, \alpha_m \rightarrow \beta_m\}$ , где  $\alpha_i \in V^* - \{\lambda\}$ ,  $\beta_i \in V^*$ ;  $\{\lambda\}$  – пустая строка;  $V^*$  – множество всех возможных последовательностей, которые удастся построить с помощью итерационных процедур на основе данного словаря.

## 1.2. Обучение в решении задачи классификации

Методы распознавания, входящие в обе группы, связаны с процедурой *обучения*, в задачу которой входит постепенное усовершенствование алгоритма разделения предъявляемых объектов на классы. Массив исходных данных в

обучаемой системе состоит из двух частей: *обучающей выборки* и *тестовой выборки*, используемой в процессе испытаний.

Если совокупность классов известна заранее, то обучение называют *контролируемым* (обучение «с учителем»). Если классы, составляющие обучающую выборку, неизвестны заранее до начала процедуры классификации, то обучение называют *неконтролируемым*, или «без учителя».

Главная особенность контролируемого метода классификации заключается в неперенном наличии «справочных» сведений о принадлежности к определенному классу каждого вектора измерений, входящего в обучающую выборку. Роль обучающего состоит в том, чтобы создать такую систему, которая позволила бы каждый вектор измерений, источник которого неизвестен, отнести к одному из уже известных классов. Задача заключается в уточнении и оптимизации процедуры принятия решений. В основу процедуры положено понятие расстояния от рассматриваемой точки до границы, отделяющей пространство, характеризуемое определенными признаками.

В обучении «без учителя» автоматическое устройство самостоятельно устанавливает классы, на которые делится исходное множество, и одновременно определяет присущие им признаки. Для разделения данных определяются критерии. При таком разделении не известны ни классы, ни их количество. Поэтому процесс организуется так, чтобы среди всех возможных вариантов найти такой, при котором группы обладают наибольшей компактностью.

Если классификация, полученная на каком-то этапе, неудовлетворительная, то следует вернуться к одному из предыдущих этапов с учетом имеющейся, но ранее не использованной информации. Каждое возвращение на очередной виток цикла характеризуется определенной ценой, в качестве которой можно принять расстояние, отделяющее конечный этап от того, с которого начинается новый цикл. Процедура классификации не всегда сходится. На рис. 1.1 приведена общая структурная схема алгоритма обучения.

### **1.3. Расстояние между классами**

Введем понятие расстояния между классами, исходя из гипотезы о том, что оно должно обеспечивать выполнение следующих двух свойств.

1. *Компактность* выражается в том, что точки, представляющие объекты одного класса, расположены друг к другу ближе, чем к точкам, относящимся к другим классам.

2. *Сепарабельность* означает, что классы ограничены и не пересекаются между собой.

На практике оба эти свойства выполняются не всегда, так как зависят от того, насколько удачно выбраны признаки объектов. Понятие расстояния позволяет оценить степень сходства как между отдельными реализациями, так и между целыми классами. С точки зрения распознавания образов можно полагать, что чем меньше расстояние между образами, тем больше у них общего.

Расстояние используется как средство оценки того, насколько близки между собой две реализации или два образа. Широкий диапазон решаемых задач потребовал использования различных способов нахождения расстояния между объектами. Ниже приводятся три из них, которые используются наиболее часто при распознавании образов.

$$1. d_1 - \text{евклидово расстояние: } d_1(\bar{X}_i, \bar{X}_j) = \left\{ \sum_{k=1}^n |x_{ik} - x_{jk}|^2 \right\}^{1/2};$$

$$2. d_2 - \text{расстояние по Манхэттену: } d_2(\bar{X}_i, \bar{X}_j) = \sum_{k=1}^n |x_{ik} - x_{jk}|;$$

$$3. d_3 - \text{чебышевское расстояние: } d_3(\bar{X}_i, \bar{X}_j) = \max |x_{ik} - x_{jk}|.$$

Здесь  $\bar{X}_i$  и  $\bar{X}_j$  – векторы, между которыми оценивается расстояние, а  $x_{ik}$  и  $x_{jk}$  –  $k$ -е составляющие векторов  $\bar{X}_i$  и  $\bar{X}_j$  соответственно.

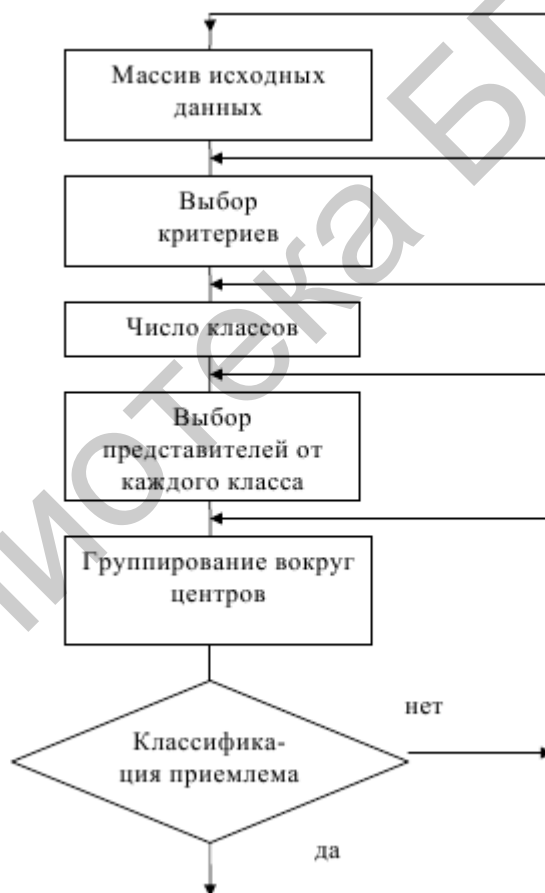


Рис. 1.1. Схема процесса обучения

#### 1.4. Решающее правило и разделяющая функция

Задача классификации связана с нахождением функции  $f$ , обеспечивающей разделение пространства  $V$  на классы, отвечающие заданным требованиям вида  $f: V \rightarrow \Pi(V)$ .

Процедура классификации состоит в том, чтобы для каждой области  $R_i$  найти решающую функцию  $g_i(x)$ , такую, что если

$$g_i(\bar{x}) > g_j(\bar{x}), \text{ то } \bar{x} \in R_i \quad \forall j = 1, 2, \dots, N,$$

где  $N$  – общее количество областей.

Разделяющую функцию часто представляют в виде линейной суммы:

$$g(\bar{x}) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n,$$

где  $\omega_i$  – весовые коэффициенты, каждый из которых относится к определенной составляющей разделяющей функции. Для удобства записи вводится весовой коэффициент с нулевым индексом  $\omega_0$ . Это позволяет записать решающую функцию в более компактной форме:

$$g(\bar{x}_a) = \bar{\omega} \bar{x}_a,$$

где  $\bar{x}_a = \{1, x_1, x_2, \dots, x_n\}$  – вектор, в число составляющих которого входит дополнительно одна вещественная константа. Ее величину обычно принимают равной единице. Решающее правило  $d$  для их разделения можно записать в виде

$$d = \begin{cases} c_1, & \text{если } g(\bar{x}) \geq 0, \\ c_2 & \text{в противном случае.} \end{cases}$$

Для случая  $N$  сепарабельных классов ( $N > 2$ ) решение о принадлежности объекта к определенному классу будет

$$d = \begin{cases} c_i, & \text{если } g_i(\bar{x}) = \bar{\omega}_i \bar{x}_a \geq 0, \\ \bar{c}_i, & \text{если } g_i(\bar{x}) < 0, \end{cases}$$

где  $C$  – множество, состоящее из  $N$  классов:

$$C = \{c_1, c_2, \dots, c_N\}, \quad c_i + \bar{c}_i = C.$$

В процессе построения разделяющей функции основная задача заключается в том, чтобы найти весовые коэффициенты вида  $\bar{\omega}_i = \{\omega_{0i}, \omega_{1i}, \dots\}$  для каждого конкретного применения.

Рассмотрим случай применения линейных разделяющих функций для разбиения объектов более чем на два класса.



Пусть существует  $M$  решающих функций  $d_k(x) = w_k x$ ,  $k = 1, 2, \dots, M$ , таких, что если образ  $x$  принадлежит классу  $\omega_i$ , то  $d_i(x) > d_j(x)$  для всех  $j \neq i$ .

Граница между классами  $\omega_i$  и  $\omega_j$  определяется теми значениями вектора  $x$ , при которых имеет место равенство  $d_i(x) = d_j(x)$ . Поэтому при выводе уравнения разделяющей границы для классов  $\omega_i$  и  $\omega_j$  значения решающих функций  $d_i(x)$  и  $d_j(x)$  используются совместно.

Пример этого случая приведен на рис. 1.2. Для образов, принадлежащих классу  $\omega_1$ , должны выполняться условия  $d_1(x) > d_2(x)$ ,  $d_1(x) > d_3(x)$ . В общем случае требуется, чтобы входящие в класс  $\omega_i$  образы располагались в положительных зонах поверхностей  $d_i(x) - d_j(x) = 0$ ,  $j = 1, 2, \dots, M$ ,  $i \neq j$ . Положительная зона границы  $d_i(x) - d_j(x) = 0$  совпадает с отрицательной зоной границы  $d_j(x) - d_i(x) = 0$ .

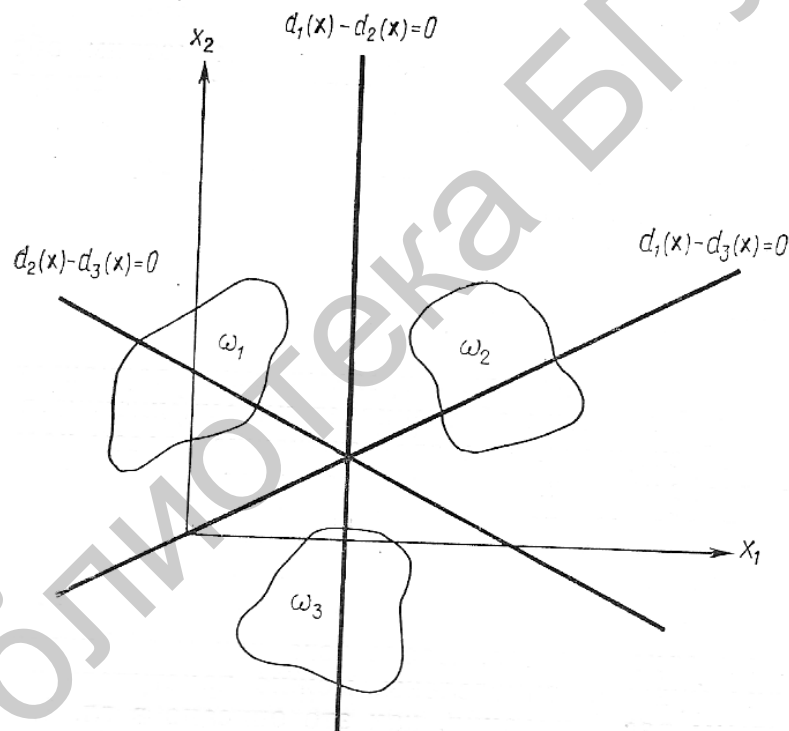


Рис. 1.2. Разделение образов на три класса

Пусть в качестве решающих правил выбраны следующие:

$$\begin{aligned} d_1(x) &= -x_1 + x_2, \\ d_2(x) &= x_1 + x_2 - 1, \\ d_3(x) &= -x_2. \end{aligned}$$

Разделяющие границы для трех классов выглядят при этом так:

$$\begin{aligned}d_1(x) - d_2(x) &= -2x_1 + 1 = 0, \\d_1(x) - d_3(x) &= -x_1 + 2x_2 = 0, \\d_2(x) - d_3(x) &= x_1 + 2x_2 - 1 = 0.\end{aligned}$$

Для того чтобы определить область решений, соответствующую классу  $\omega_1$ , необходимо выделить часть плоскости, в которой выполняются неравенства  $d_1(x) > d_2(x)$ ,  $d_1(x) > d_3(x)$ . Она совпадает с положительными зонами для прямых  $-2x_1 + 1 = 0$  и  $-x_1 + 2x_2 = 0$ . Область принятия решения о принадлежности образа классу  $\omega_2$  совпадает с положительными зонами для прямых  $2x_1 - 1 = 0$  и  $x_1 + 2x_2 - 1 = 0$ . Область, отвечающая классу  $\omega_3$ , определяется положительными зонами для прямых  $x_1 - 2x_2 = 0$  и  $-x_1 - 2x_2 + 1 = 0$ . Эти области представлены на рис. 1.3.

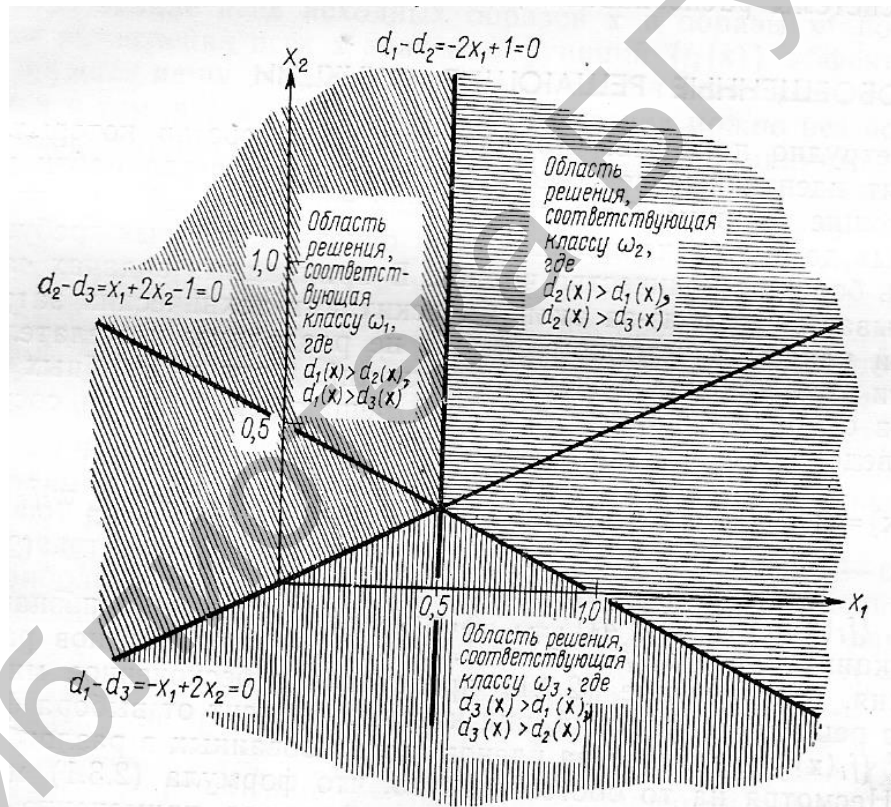


Рис. 1.3. Области трех классов, построенные с помощью разделяющих функций

В качестве примера классификации рассмотрим обработку образа  $x = (1, 1)$ . Подстановка признаков образа в выбранные решающие функции дает следующие значения:  $d_1(x) = 0$ ,  $d_2(x) = 1$ ,  $d_3(x) = -1$ . Поскольку  $d_2(x) > d_j(x)$ ,  $j = 1, 3$ , образ относится к классу  $\omega_2$ .

Если какой-либо из рассмотренных вариантов линейной решающей функции обеспечивает классификацию в некоторой заданной ситуации, то соответствующие классы называются *линейно разделимыми*.

### 1.5. Особенности синтаксических методов

Процесс создания языка начинается с аксиомы  $S$ , к которой применяются одно за другим правила подстановок. Основным вопросом после определения грамматики является разработка процедуры грамматического разбора, устанавливающей, является или нет рассматриваемый объект предложением языка, созданного на основе грамматики. Разборы удобно выполнять с помощью деревьев, поскольку любая иерархически упорядоченная схема ведет к представлению объекта в виде дерева. Наиболее популярны два типа грамматических разборов: сверху вниз и снизу вверх, каждый из которых применяется для выполнения определенной процедуры. Если необходимо установить, принадлежит ли некоторая структура классу объектов, порождаемых заданной грамматикой, выполняется разбор снизу вверх. Если же требуется построить объекты по правилам определенной грамматики, выполняется разбор сверху вниз.

При построении дерева его корень ассоциируется с начальной аксиомой  $S$ . Терминальные предложения (образы) представляют нижнюю часть или листья дерева. Процедура разбора сверху вниз начинается с корневого символа  $S$  и заключается в попытках посредством повторяющегося применения грамматических правил получить заданное терминальное предложение. И наоборот, процедура разбора снизу вверх начинается с конкретного предложения и заключается в попытках дойти до символа  $S$  с помощью инверсии правил подстановки. В каждом из этих случаев при неудачном исходе грамматического разбора заданный образ отклоняется как представляющий «неправильное» предложение.

Интересным приложением лингвистических понятий в распознавании образов является язык PDL (Picture Description Language) – язык описания изображений. Терминальным элементом PDL служит любая  $n$ -мерная структура с двумя выделенными точками: хвостовой и головной.

По правилам языка PDL практически любая структура может обобщенно рассматриваться как ориентированный отрезок прямой, так как по определению для нее существует только две точки. Терминальные элементы связываются между собой только в хвостовых и (или) головных точках. Следовательно, структуры языка PDL представляют собой ориентированные графы и для их обработки можно использовать грамматики.

Кроме использования языка PDL, грамматику можно расширить путем введения в ее правила подстановки рекурсивности, когда переменная способна замещаться этой же переменной. Однако увеличение порождающей способности грамматики не всегда желательно. Особенно это касается тех исследований, где используется более одной грамматики. В этом случае чрезмерное их многообразие приводит к уменьшению различающей мощности каждой из грамматик.

## 1.6. Системы распознавания образов

Если процесс распознавания объектов связывать с техническими средствами, то его возлагают на системы распознавания. Абстрагируясь от типа системы и прикладной области поставленной перед ней проблемы, можно определить последовательность действий, составляющих ход решения задачи.

1. Выделяются характерные признаки, по которым будет выполняться распознавание объектов, и создается словарь признаков.

2. Назначается алфавит классов, т. е. в соответствии с выбранным принципом совокупность объектов или явлений подразделяется на классы.

3. Каждый класс описывается на языке словаря признаков.

4. Выбираются и (или) создаются средства для определения признаков.

5. На вычислительных средствах реализуется алгоритм сопоставления апостериорных и априорных данных и принимается решение о результатах распознавания.

Рассмотрим более подробно этапы, составляющие процедуру распознавания. Необходимо определить все признаки, характеризующие сущность распознаваемых объектов или явлений, так как неполнота информации приводит в дальнейшем к ошибкам или невозможности правильной классификации. В современных условиях не существует способов автоматической генерации признаков. Это эвристическая операция, выполняемая человеком. Эффективнее выбирать признаки, имея представление об их общих свойствах. С этих позиций признаки делятся на *детерминированные, вероятностные, логические и структурные*.

*Детерминированные признаки* – это характеристики объектов или явлений, которые имеют конкретные и постоянные числовые значения. В задачах распознавания с детерминированными признаками ошибки их вычисления часто не играют существенной роли. Например, если точность измерений такого признака, как размах крыльев самолета, значительно выше (например 1 см), чем различие этого признака у разных классов самолетов (например 10 м).

*Вероятностные признаки* – это характеристики объектов или явлений, которые носят случайный характер. В основном они встречаются в природе и технике. В силу случайности соответствующей величины признак одного класса может принимать значения из области значений других классов, каждый из которых подлежит распознаванию в системе. Если признак принимает значения из области значений одного класса, то он переходит в разряд детерминированных признаков. Для распознавания объектов в условиях неопределенности следует потребовать, чтобы вероятности наблюдения значений признака в классе, которому он принадлежит, были как можно больше, чем в остальных классах. В противном случае данный признак не позволит построить систему, использующую описание классов на его основе. Поскольку разделительная способность признака окажется недостаточной для принятия достоверного решения на его основе.

*Логические признаки распознавания* – это характеристики объекта или явления, представленные в виде элементарных высказываний об истинности. Такие признаки не имеют количественного выражения, а являются качественными суждениями о наличии или отсутствии некоторых свойств у объектов или явлений. Кроме того, к логическим можно отнести признаки, для которых несущественно конкретное значение, а важен факт его попадания или непопадания в заданный интервал.

*Структурные признаки* – это элементы изображений, являющихся объектами распознавания. Появление структурных признаков связано с проблемой распознавания изображений и ее специфическими особенностями.

Когда все существенные признаки объектов определены, на их основе выделяются классы объектов распознавания. Чаще всего решение этой задачи осуществляется эвристически. Для одной задачи могут быть предложены различные варианты составления алфавита классов. Выбор определенного из них отразится на качестве распознавания в дальнейшем.

Располагая перечнем признаков и априорным алфавитом классов, необходимо провести анализ возможностей измерения признаков или расчета их по данным измерений, выбрать те, которые обеспечиваются измерениями, а в случае необходимости создать новые средства для достижения требуемого качества распознавания. В результате первоначальный набор признаков может подвергнуться корректировке. Однако оценить качество словаря признаков удастся, только выполнив испытания системы распознавания в целом. Поскольку система еще не существует, создается ее математическая модель, работа с которой методом последовательных приближений, можно добиться выбора словаря признаков, обеспечивающего желаемое качество решений.

Описание классов априорного алфавита на языке априорного словаря признаков – творческая и наиболее сложная из задач в процессе распознавания объектов, требующая глубокого изучения их свойств. В рамках этой задачи необходимо каждому классу поставить в соответствие числовые параметры детерминированных и вероятностных признаков, значения логических признаков и предложения, составленные из структурных признаков-примитивов. Значения перечисленных признаков получают как результат следующих действий:

- специально поставленных экспериментов и наблюдений за ними;
- обработки экспериментальных данных;
- математических расчетов;
- математического моделирования;
- обработки литературных источников.

Следующий шаг – это реализация алгоритма распознавания, когда на основе словаря признаков и алфавита классов объектов или явлений происходит разбиение пространства значений признаков на области, соответствующие классам. Разбиение должно быть выполнено таким образом, чтобы обеспечивались минимальные значения ошибок отнесения классифицируемых объектов или явлений к «чужим» классам.

В алгоритмах, использующих детерминированные признаки в качестве меры близости, используется среднеквадратичное расстояние между интересующим объектом и совокупностью объектов, представляющих каждый класс.

В алгоритмах, работающих с вероятностными признаками, в качестве меры близости используется риск, связанный с решением о принадлежности объекта к классу  $W_i$ , где  $i$  – номер класса ( $i = 1, 2, \dots, m$ ).

Для алгоритмов, основанных на логических признаках, понятие «мера близости» не имеет смысла. Поскольку достаточно подставить значения признаков в соответствующие булевы соотношения, чтобы сразу получить результат как истину или ложь булевой функции описания класса.

Для алгоритмов, основанных на структурных признаках, понятие «меры близости» более специфично. Каждый класс описывается совокупностью предложений, характеризующих структурные особенности принадлежащих ему объектов. Распознавание неизвестного объекта выполняется путем идентификации предложения, описывающего объект, с одним из предложений, определяющих класс. Под идентификацией подразумевается наибольшее сходство предложения-объекта с предложениями из наборов описаний каждого класса.

В процессе работы с системой распознавания меняются ее признаки и алфавит классов. Поэтому становится необходимым переход от априорных словаря признаков и алфавита классов к рабочим словарю и алфавиту. В них выбираются признаки и классы, позволяющие при всех имеющихся ограничениях на их получение добиться максимума вероятности правильной классификации объектов и минимума вероятности ошибочных классификаций в системе.

Классификации предшествует процедура обучения, в задачу которого входит постепенное усовершенствование алгоритма разделения предъявляемых объектов на классы. Этот процесс стремятся автоматизировать, для чего отбирают часть предъявляемых объектов и используют их для «тренировки» системы. По количеству первоначальной информации системы распознавания делят на системы без обучения, обучающиеся системы и самообучающиеся системы.

Системы первого типа являются самыми простыми по своей структуре и алгоритмам работы, для их построения необходимо располагать полной первоначальной информацией. Массив исходных данных в обучаемой системе состоит из двух частей: обучающей выборки и тестовой выборки, используемой в ходе испытаний. Если совокупность классов известна заранее, то обучение называют контролируемым, или «с учителем». Роль разработчика заключается в определении наилучших критериев классификации, учитывающих различия между признаками, характерными для отдельных классов. Главной задачей в этом случае становится поиск оптимальных методов разделения. Если классы, составляющие обучающую выборку, не известны до начала процедуры классификации, то речь идет о самообучающейся системе, а обучение называют неконтролируемым, или «без учителя». Решение задачи при таких условиях значительно сложнее, чем в предыдущих случаях.

Для того чтобы облегчить решение всех вышеперечисленных задач, связанных с созданием систем распознавания, качественнее выбирать признаки объектов распознавания, планировать использование как априорной, так и апостериорной информации, необходимо уметь классифицировать системы распознавания (СР) и знать связи между ними. Однако взаимосвязи могут принимать различные формы и иметь свои особенности. Начнем с классификации СР, что позволит разобраться в их структуре, понять взаимосвязи и решать поставленные задачи.

В основе любой классификации лежат определенные принципы, для классификации СР будем использовать следующие из них:

1. Однородность информации для описания распознаваемых объектов или явлений.
2. Способ получения апостериорной информации.
3. Количество первоначальной априорной информации.
4. Характер информации о признаках распознавания.

Первый принцип подразумевает различную или единую физическую природу информации. В связи с ним СР делятся на простые и сложные. Простые СР характеризуются единой физической природой признаков. Для простых систем распознавания необязательно использовать вычислительную технику. Их можно реализовать в виде механических или электромеханических устройств. Однако компьютерные реализации оказываются предпочтительнее, если наряду с задачей распознавания в этой системе решаются и другие более сложные задачи. Сложные СР характеризуются физической неоднородностью признаков.

По второму принципу классификации сложные СР делятся на одноуровневые и многоуровневые. Одноуровневая СР включает в себя: разнородные по физической природе измерители; априорное описание классов распознаваемых объектов; алгоритм классификации; систему автоматического управления алгоритмом распознавания. Многоуровневые СР отличаются от одноуровневых тем, что в них не все признаки, полученные от разнородных физических измерителей, используются непосредственно для решения задачи распознавания. Здесь на основе объединения и обработки признаков нескольких измерителей могут быть получены вторичные признаки, которые или используются в алгоритме классификации, или сами служат основой для дальнейшего объединения. В результате чего образуются новые уровни, определяющие многоуровневость СР. При этом подсистемы, которые осуществляют объединение признаков, также могут представлять собой устройства распознавания в виде локальных СР. Таким образом, в одноуровневых СР апостериорная информация о признаках распознаваемого объекта формируется непосредственно путем обработки прямых измерений, а в многоуровневых СР аналогичная информация формируется на основе косвенных измерений как результат функционирования вспомогательных распознающих устройств. Например, измерение дальности радиолокатором по времени задержки излученного импульса.

По третьему принципу классификации решается вопрос о том, достаточно или нет априорной информации для определения априорного алфавита

классов, построения априорного словаря признаков и описания каждого класса на языке этих признаков в результате непосредственной обработки исходных данных. В соответствии с этим СР делятся на системы без обучения, обучающиеся системы и самообучающиеся системы. Заметим, что многоуровневые сложные СР однозначно нельзя разделить на указанные классы, так как каждая из локальных СР, входящих в их состав, сама может представлять как систему без обучения, так и систему обучающуюся или самообучающуюся.

Для построения систем без обучения необходимо располагать полной первоначальной априорной информацией. Для обучающихся систем характерна ситуация, когда априорной информации не хватает для описания распознаваемых классов на языке признаков. Возможны случаи, когда информации хватает, однако делать упомянутое описание нецелесообразно или трудно. Исходная информация для таких систем представляется в виде набора объектов  $w_1, w_2, \dots, w_l$ , распределенных по  $m$  классам:

$$\begin{array}{ll} (w_1, w_2, \dots, w_l) & W_1, \\ (w_{r+1}, w_{r+2}, \dots, w_q) & W_2, \\ \dots\dots\dots & \dots\dots\dots \\ (w_{g+1}, w_{g+2}, \dots, w_l) & W_m. \end{array}$$

Цель обучения и ее достижение для обучающихся систем заключаются в определении разделяющих функций:

$$F_i(X_1, X_2, \dots, X_n), \text{ где } i = 1, 2, \dots, m \text{ (номер класса).}$$

Определение разделяющей функции осуществляется путем многократного предъявления системе объектов из набора  $w_1, w_2, \dots, w_l$  с указанием, какому классу они принадлежат. На стадии создания системы работают с «учителем», осуществляющим тренировку СР на основе обучающей выборки, и прежде чем система будет применяться, должен пройти этап обучения. На рис. 1.4 изображена схема обучающейся системы распознавания. Где  $W$  – неизвестные распознаваемые объекты;  $Y$  – учитель;  $OO$  – обучающие объекты;  $ТС$  – технические средства, включающие в себя измерители признаков распознавания;  $АПРФ$  – алгоритм построения разделяющих функций;  $АО$  – априорное описание классов распознаваемых объектов;  $САУ$  – система автоматического управления (алгоритм) распознавания;  $АК$  – алгоритм классификации.

В отличие от систем без обучения и систем, обучающихся «с учителем», для самообучающихся систем характерна недостаточность информации для формирования не только описаний классов, но даже алфавита классов. Определен только словарь признаков распознавания. Однако для организации процесса обучения задается все-таки некоторый набор правил, в соответствии с которым система сама вырабатывает классификацию. Для этих систем так же, как и для предыдущих, существует период обучения. Характерно наличие периода самообучения, когда системе предъявляются объекты обучающей последова-



тельности, только при этом не указывается принадлежность их к каким-либо классам.

Функциональная схема самообучающейся системы приведена на рис. 1.5. В дополнение к обозначениям, введенным на рис. 1.4, здесь ОС – объекты самообучения; ПК – правила классификации; АФК – алгоритм формирования классов.

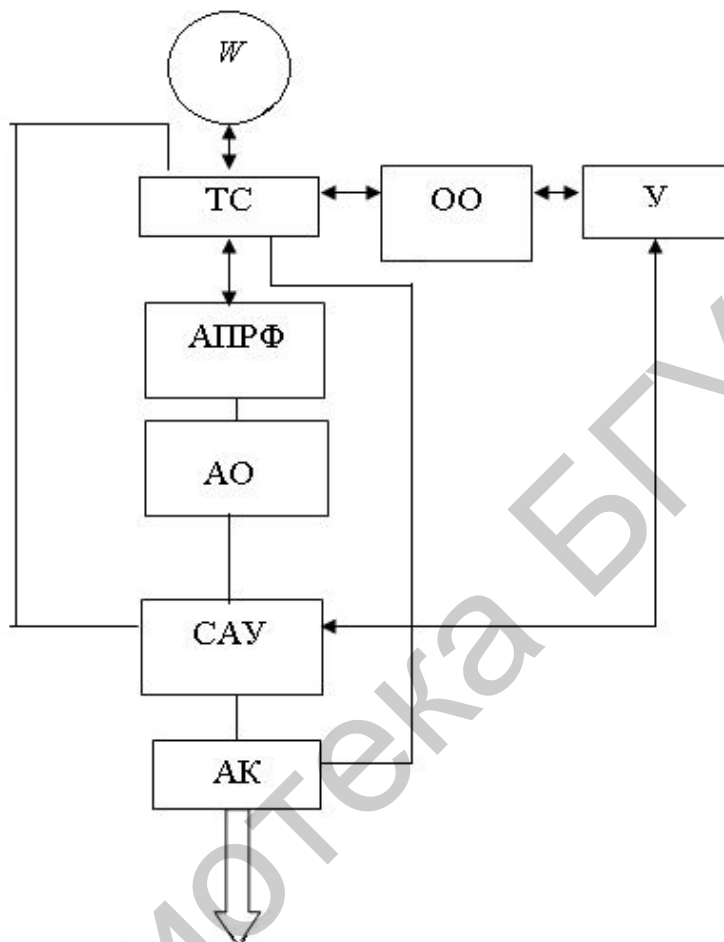


Рис. 1.4. Система распознавания «с учителем»

Примером самообучающейся системы может быть система разделения на классы промышленных предприятий для сравнительного анализа эффективности их функционирования. При этом в качестве правил классификации могут быть указания либо о равенстве объемов выпускаемой продукции, либо о равенстве численности рабочих и т. п.

Завершая рассмотрение классификации СР по количеству первоначальной априорной информации, заметим, что СР, в которых недостаточно информации для назначения словаря признаков, не существует. Без этого не создается никакая система.

В основе четвертого принципа классификации СР лежит их разделение по характеру информации о признаках распознавания на детерминированные, вероятностные, логические, структурные (лингвистические), комбинированные.

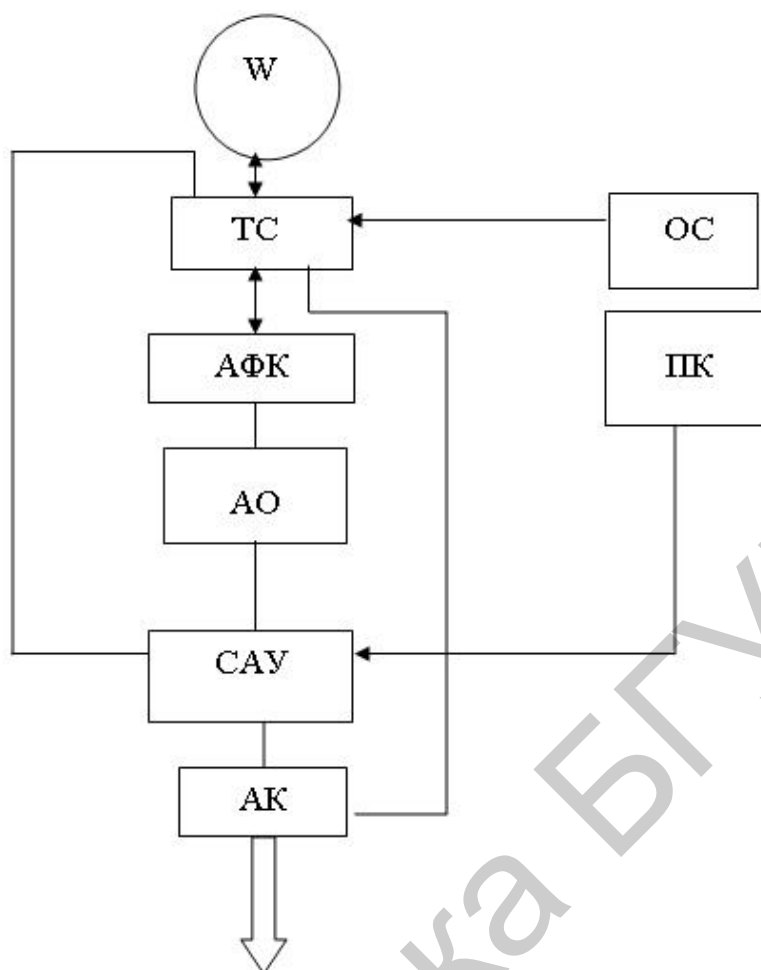


Рис. 1.5. Самообучающаяся система распознавания

*Детерминированные системы:*

а) метод решения задачи распознавания – использование геометрических мер близости;

б) метод априорного описания классов – координаты векторов-эталонов по каждому из классов или координаты всех объектов, принадлежащих классам (наборы эталонов по каждому классу).

*Вероятностные системы:*

а) метод решения задачи распознавания – вероятностный, основанный на вероятностной мере близости (средний риск);

б) метод априорного описания классов – вероятностные зависимости между признаками и классами.

*Логические системы:*

а) метод решения задачи распознавания – логический, основанный на дискретном анализе и исчислении высказываний;

б) метод априорного описания классов – логические связи, выражаемые через систему булевых уравнений, где признаки – переменные, классы – неизвестные величины.

*Структурные (лингвистические) системы:*

а) метод решения задачи распознавания — грамматический разбор предложения, описывающего объект на языке производных структурных элементов с целью определения его правильности;

б) метод априорного описания классов — подмножества предложений, описывающих объекты каждого класса.

*Комбинированные системы:*

а) метод решения задачи распознавания — специальные методы вычисления оценок;

б) метод априорного описания классов — табличный, предполагающий использование таблиц, содержащих классифицированные объекты и их признаки (детерминированные, вероятностные, логические).

После проведенной классификации еще раз обратимся к понятиям «достаточное» или «недостаточное» количество информации. С точки зрения этих понятий все СР делятся на два больших класса:

1) СР без обучения;

2) обучающиеся и самообучающиеся СР, т. е. требующие обучения.

СР без обучения работают в условиях полной информации, для обучающихся систем характерна неполнота информации, когда нет описания классов на языке признаков, а для самообучающихся отсутствует даже алфавит классов. Однако следует понимать, что понятие «неполнота информации» является относительным. Для СР без обучения при прочих равных условиях этой информации просто больше. Это означает, что результативность систем первого класса при имеющемся объеме априорной информации значительно выше, чем систем второго класса. О результативности СР, для которой невозможно априорно назначить алфавит классов, говорить вообще нельзя. Отсюда следует, что не нужно пренебрегать информацией. Поэтому при построении любой СР необходимо всегда использовать принцип обратной связи для расширения объема информации. То есть результаты решения задачи распознавания неизвестных объектов после апостериорного подтверждения правильности их классификации необходимо использовать для уточнения описания классов в простых СР без обучения и для дополнительного обучения в системах с обучением.

## **1.7. Области применения систем распознавания**

*Осязание роботов. Общие положения.* Многие задачи робототехники, в частности автоматическая сборка изделий, требуют непосредственного физического контакта между объектом и захватывающим узлом робота. Поскольку наличие такого контакта является необходимым условием успешного решения задачи, возникает желание использовать его одновременно и для распознавания объекта, с которым взаимодействует робот, для оценки его параметров: геометрических размеров, ориентации в пространстве, степени шероховатости поверхности.

К настоящему времени приборов и методов для искусственного осязания разработано значительно меньше, чем для искусственного зрения. Однако скорость обработки информации у тактильных датчиков может быть даже более высокой, чем у оптических. Это связано с тем, что в первом случае обрабатывается значительно меньшее количество сигнальных отсчетов, чем при обработке видеосигналов. Тактильные датчики могут решать различные задачи: определение состояния поверхности, распознавание формы контуров и исследование других характеристик объекта путем «ощупывания» его чувствительным элементом. Достоинством тактильных датчиков являются их механическая гибкость, дешевизна, удобство использования, а также линейная зависимость электрического сопротивления от деформации, а следовательно, и от локальной нагрузки. К областям применения тактильных датчиков можно отнести промышленную робототехнику, телеуправление, в том числе на устройствах, предназначенных для работы в неблагоприятных средах.

*Дистанционное обнаружение в геофизике.* Здесь подразумеваются наблюдения за поверхностью Земли и других планет – это различные спутники, корабли многоразового использования, орбитальные станции и многое другое. Каждая из таких систем выдает огромные потоки информации. Поскольку число потребителей этой информации быстро возрастает, представляется необходимым выполнять автоматическую классификацию наблюдений за минимальное время, согласованное со срочностью запросов. Идеальным решением считается обработка изображений в реальном масштабе времени и выдача результатов пользователям.

*Сейсмология.* Сейсмические волны можно наблюдать и записывать в любой точке земной поверхности. Для этого используются сейсмографы – приборы, обладающие чрезвычайно высокой чувствительностью к механическим колебаниям земли. Автоматическое дешифрирование этих записей представляет огромный интерес для понимания явлений, происходящих в толще земной коры. Для описания сейсмических волн была разработана грамматика, реализуемая на детерминированном конечном автомате. С учетом того что отрезки сигнала имеют одинаковую длительность, среди возможных помех следует учитывать только ошибки подстановки. Для этого каждому грамматическому правилу вида  $A \rightarrow aB$  или  $A \rightarrow a$  (правила «грамматики без помех») добавляются правила вида  $A \rightarrow bB$  и  $A \rightarrow b$  с классическими условиями  $A, B \in V_n$  и  $a, b \in V_t$ . Каждое из этих дополнительных правил можно брать с определенным весом, зависящим от ошибки.

Решение состоит в том, чтобы перестановкам между двумя словами придавать веса в соответствии с расстоянием, разделяющим эти классы в выбранном пространстве признаков. При этом операция распознавания оказывается идентичной операции отыскания оптимального пути в древовидном графе, поскольку такое представление может быть связано с любой грамматикой конечных состояний. Эксперименты проводились на массиве, в который входила 321 реализация. Из них 50 использовались в качестве обучающей выборки и как

основа для разработки грамматики. В зависимости от числа классов общие результаты показали две величины: процент правильного распознавания и время его выполнения. При числе классов около 10 время выполнения резко возрастает. Если учесть получающийся при этом достаточно высокий процент правильного распознавания, то оказывается, что интуитивный выбор именно этого количества классов вполне оправдан. Результаты можно еще несколько улучшить за счет использования длины отрезка в качестве дополнительного признака. Кроме того, интерес представляет рассмотрение сочетания этой процедуры с системой экспертных оценок.

*Электрокардиография* представляет собой один из методов исследования работы сердца, основанный на записи разности электрических потенциалов, возникающей в процессе сердечной деятельности. Схематически сердце может быть представлено в виде электрического диполя переменной длины, зависящей от сердечного ритма. Форма электрического сигнала, изменяющегося во времени, и его амплитуда зависят от точки съема. Типичная осциллограмма ЭКГ-сигнала показана на рис. 1.6.

Буквы PQRST, предложенные В. Эйнтховеном, позволяют в удобной форме описывать отдельные особенности этой непрерывной кривой. Периодический сигнал ЭКГ имеет сравнительно простую структуру, поэтому для его автоматического распознавания была предложена процедура на основе грамматического описания.

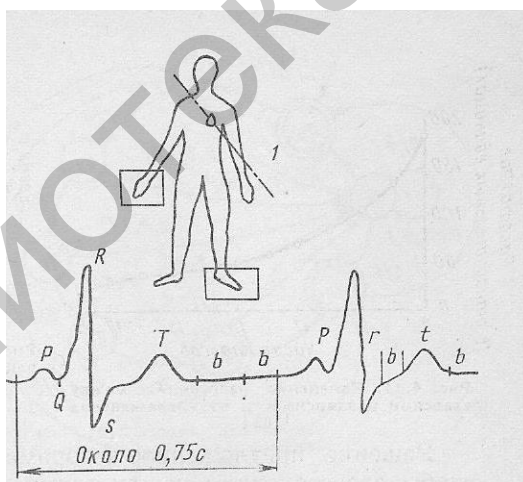


Рис. 1.6. Типичная электрокардиограмма, где 1 – электрическая ось сердца

При этом описание ЭКГ составляется из четырех символов –  $p$ ,  $r$ ,  $b$ ,  $t$ , каждый из которых соответствует определенному участку кривой на рис. 1.6. Символ  $p$  соответствует волне  $P$ ;  $r$  –  $RS$ -переходу;  $b$  – относительно плоской части, разделяющей экстремумы  $S$  и  $T$  (около 0,1 с);  $t$  – волне  $T$ . Если за начало отсчета принять волну  $P$ , то в таких обозначениях нормальная ЭКГ может быть описана последовательностями символов:  $prbtb$ ,  $prbtbb$ ,  $prbtbbb$  и т. д.

Синтаксические описания такого вида могут быть получены с использованием грамматики  $G$ :

$$G = \{ V_b, V_n, P, S \}, V_t = \{ p, r, t, b \}, V_n = \{ S, A, B, C, D, E, H \},$$

$$P = \{ S \rightarrow pA, A \rightarrow rB, B \rightarrow bc, C \rightarrow tD, D \rightarrow b, D \rightarrow bE, E \rightarrow b, E \rightarrow bH, \\ E \rightarrow pA, H \rightarrow b, H \rightarrow bS, H \rightarrow pA \}.$$

Языку, порождаемому этой грамматикой, можно привести в соответствие конечный автомат, схема которого показана на рис. 1.7. Для того чтобы обнаружить аномальную ЭКГ и отличить ее от нормальной, используют выход «0», если исследуемая ЭКГ соответствует «нормальному» стандарту, и выход «1» – в противном случае. Этот выход на рис. 1.7 соединен с линиями переходов. Такой тип распознающего автомата весьма примитивен: он способен обнаруживать лишь грубые отклонения от нормы. В действительности анализ аномальной ЭКГ представляет собой серьезную задачу, которая выполняется квалифицированными специалистами.

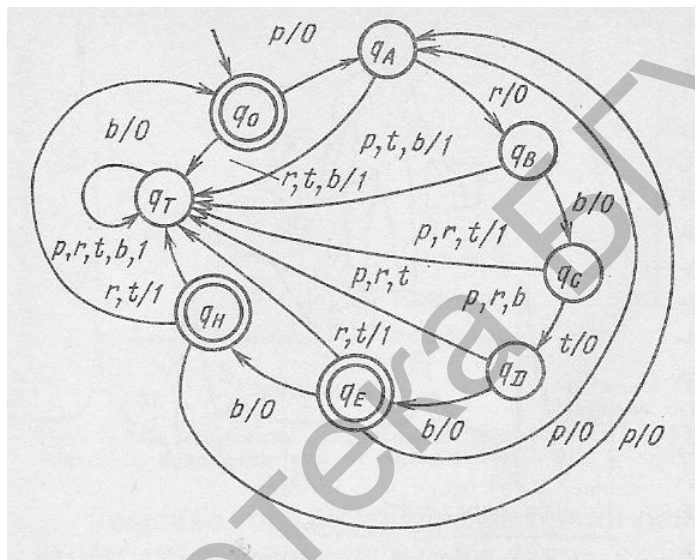


Рис. 1.7. Схема автомата для распознавания электрокардиограммы

*Промышленное применение.* Автоматический контроль деталей в процессе их изготовления – это та задача, для решения которой используются и разрабатываются различные средства обработки информации. Однако внедрение устройств, автоматически измеряющих размеры, состояние поверхности и другие характеристики предметов в процессе производства, часто сопряжено с немалыми трудностями. Одной из главных задач, возникающих при автоматическом контроле, является задача обучения распознающего устройства.

В процессе изготовления деталей или узлов разнообразие возможных дефектов может быть настолько велико, что составить для машины обучающую выборку, включающую все варианты подлежащих контролю ситуаций, часто представляет собой невыполнимую задачу. Поэтому следует либо создать обучающую выборку на базе ограниченного набора типовых дефектов, требующих идентификации, либо ввести процедуру автоматического обучения, что представляется априори несовместимым со многими промышленными применениями. Однако потребность в таких системах становится все более острой в тех об-

лостях, где приходится сталкиваться с многократно повторяющимися операциями или где требуется большая скорость выполнения.

Задачи сборки узлов и механизмов также постепенно передаются роботам. При этом в большинстве случаев собираемые элементы должны быть поданы и ориентированы строго определенным образом. Проверка правильности этого выполняется автоматически с помощью различных датчиков, чаще всего оптических.

### 1.8. Критерии развития систем распознавания

Среди параметров и показателей, характеризующих любой технический объект (ТО), всегда имеется один или несколько таких, которые на протяжении длительного времени имеют тенденцию монотонного изменения или поддержания на определенном уровне при достижении своего предела. Эти показатели осознаются как мера совершенства и прогрессивности и оказывают сильное влияние на развитие отдельных классов технических объектов и техники в целом. Такие параметры называются *критериями развития* ТО.

Наборы критериев развития для различных классов ТО в большой степени совпадают, поэтому в целом развитие техники подчинено единому набору критериев. Этот набор включает в себя четыре группы:

- функциональные, характеризующие показатели реализации функции технического объекта;
- технологические, связанные с возможностью и простотой изготовления технического объекта;
- экономические, определяющие экономическую целесообразность реализации функции с помощью ТО;
- антропологические, связанные с вопросами человеческого фактора, т. е. с воздействием на человека ТО.

Рассмотрим три наиболее распространенных критерия, входящих в группу функциональных критериев.

Критерий *производительности* представляет собой интегральный показатель уровня развития техники, который непосредственно зависит от ряда параметров, влияющих на производительность ТО. Эти параметры представляют собой частные функциональные критерии:

- 1) скорость обработки объекта;
- 2) физические и химические параметры, влияющие на интенсивность обработки;
- 3) степень механизации труда;
- 4) степень автоматизации труда;
- 5) непрерывность процесса обработки.

Критерий механизации определяется как отношение механизированных операций к общему числу операций. Критерий автоматизации определяется как отношение автоматизированных операций к общему числу операций. В основе

критерия непрерывности лежит один из главных способов повышения производительности труда.

Критерий *точности* включает в себя следующие частные критерии:

- 1) точность измерений;
- 2) точность попадания в цель;
- 3) точность обработки материала или вещества;
- 4) точность обработки потока энергии;
- 5) точность обработки информации.

Критерий *надежности* содержит следующие частные критерии:

- 1) безотказность;
- 2) долговечность;
- 3) сохраняемость;
- 4) ремонтпригодность.

Под надежностью ТО обычно подразумевают способность без отказов выполнять свою функцию с заданной вероятностью в течение определенного интервала времени. Критерий надежности возрастает с увеличением времени и вероятности безотказной работы. Критерии производительности, точности и надежности представляют собой монотонно возрастающие функции. Актуальность и вес этих критериев всегда были выше других.

*Технологические* критерии призваны обеспечивать всестороннюю экономию живого труда при создании ТО и подготовке их к эксплуатации. Выделяют четыре частных критерия. Критерий *трудоемкости изготовления* ТО равен отношению суммарной трудоемкости  $T_c$  проектирования, изготовления и подготовки к эксплуатации изделия к его показателю эффективности  $Q$ , т. е. представляет собой удельную трудоемкость изготовления на единицу получаемой эффективности:  $K_T = T_c/Q$ . Главный показатель эффективности  $Q$  выбирают так, чтобы  $K_T$  объективно отражал прогрессивное развитие рассматриваемых ТО. Критерий трудоемкости представляет собой монотонно убывающую функцию при условии, что сопоставление различных поколений ведется по одному и тому же показателю эффективности  $Q$ . Этот критерий считается одним из самых древних.

Критерий *технологических возможностей* связан с выделением в ТО пяти типов элементов:

- 1)  $A_c$  – стандартные;
- 2)  $A_y$  – унифицированные, заимствованные у существующих ТО;
- 3)  $A_{k1}$  – оригинальные, но не сложные;
- 4)  $A_{k2}$  – оригинальные и сложные;
- 5)  $A_{k3}$  – оригинальные, вызывающие принципиальные трудности.

Критерий технологических возможностей, который должен отражать простоту и принципиальность возможности изготовления ТО, можно определить по формуле



$$K_{Т.В} = \varepsilon \frac{K_c A_c + K_y A_y + K_{k1} A_{k1} + K_{k2} A_{k2}}{A_c + A_y + A_{k1} + A_{k2}},$$

где  $\varepsilon = \begin{cases} 1, & \text{если } A_{k3} = 0, \\ 0, & \text{если } A_{k3} > 0; \end{cases}$   $K_c, K_y, K_{k1}, K_{k2}$  – весовые коэффициенты, причем

$K_c = 1, K_c > K_y > K_{k1} > K_{k2}$ ;  $A_c, A_y, A_{k1}, A_{k2}, A_{k3}$  – число наименований соответствующих элементов.

На практике часто используются частные случаи этого критерия: стандартизации, унификации. Однако в любой форме представления  $K_{Т.В}$  находится в диапазоне значений от 0 до 1. Его нельзя отнести к монотонно возрастающей функции, так как часто для улучшения более важных критериев приходится ухудшать  $K_{Т.В}$ . Основная форма представления этого критерия стимулирует исключение абсолютно нетехнологичных элементов  $A_{k3}$  и минимизацию  $A_y, A_{k1}, A_{k2}$  в соответствии с их весовыми коэффициентами.  $K_{Т.В}$  отражает фактор наследственности в технике. Он заставляет в наибольшей мере сохранять и использовать проверенные практикой функциональные элементы, отработанную технологию изготовления и внедрения.

Критерий *использования материалов* равен отношению массы изделия  $G$  к массе израсходованных материалов  $P$ , при этом покупные комплектующие элементы не учитываются:

$$K_{И.М} = G/P.$$

В случае, когда в ТО используются материалы, существенно различающиеся по стоимости, при вычислении данного критерия рекомендуется применять следующие формулы:

$$G_n = \sum_{i=0}^m k_i q_i, P_n = \sum_{i=0}^m k_i p_i,$$

где  $i$  – номера материалов;  $q_i$  – масса  $i$ -го материала;  $k_i$  – весовой коэффициент  $i$ -го материала;  $p_i$  – масса израсходованного  $i$ -го материала.

Критерий использования материалов представляет собой монотонно убывающую функцию в диапазоне от 0 до 1. Иногда случаются ступенчатые возрастания данного критерия, связанные с переходом на новые технологические процессы с большей производительностью или на новые более дешевые материалы.

*Критерий разбиения ТО на элементы.* Всегда существует оптимальное разбиение ТО на узлы и детали, которое значительно упрощает технологию разработки, доводки, изготовления, модернизации, являясь основой для унификации и стандартизации. Данный критерий обеспечивает в каждом новом поко-

лении изделий приближение к оптимальному с точки зрения декомпозиции представлению технической системы.

Рассмотрим группу *экономических* критериев, первый из которых критерий *расхода материалов*. Он равен отношению массы технической системы  $G$  к ее главному показателю эффективности  $Q$ :

$$K_{p.m} = G/Q,$$

т. е. представляет собой удельную массу материалов на единицу получаемой эффективности. В случае применения в ТО материалов со значительно различающимися стоимостями параметр  $G$  следует вычислять, как для критерия использования материалов. Критерий расхода материалов обычно является монотонно убывающей функцией при условии, что сопоставление различных поколений ТО ведется по одному показателю эффективности  $Q$ .

*Критерий расхода энергии*. При разработке технической системы стараются свести к минимуму энергетические затраты. В связи с этим существует критерий расхода энергии:

$$K_9 = \frac{W_n + E}{TQ}, \quad (1.1)$$

где  $W_n$  – полная затрата энергии за время эксплуатации ТО;  $E$  – затраты энергии при изготовлении ТО;  $T$  – время эксплуатации ТО;  $Q$  – показатель эффективности.

Формулу (1.1) рекомендуют использовать, когда  $W_n$  и  $E$  соизмеримы. Если  $W_n$  намного больше  $E$ , то используется следующая формула:

$$K_9 = W/Q, \quad (1.2)$$

где  $W$  – затраты энергии при эксплуатации ТО в единицу времени.

В инженерной практике широко используется еще одна модификация этого критерия, которую называют коэффициентом полезного действия:

$$K_9 = W_0/W, \quad (1.3)$$

где  $W_0$  – полезная работа;  $W$  – вся выполненная работа.

Критерии, представленные формулами (1.1) и (1.2), – монотонно убывающие функции, а критерии, представленные формулой (1.3), – монотонно возрастающая функция.

*Критерий затрат на информационное обеспечение* выражается формулой

$$K_{и.о} = S/Q,$$

где  $S$  – затраты на подготовку и обработку информации, включающие стоимость или эксплуатацию вычислительной техники, разработку программного и информационного обеспечения;  $Q$  – показатель эффективности.

Критерий представляет собой монотонно убывающую функцию, однако возможны скачки, связанные с переходом на новую вычислительную технику, которая сразу не дает повышения эффективности ТО.

*Критерий габаритных размеров* ТО выражается следующей формулой:

$$K_{\Gamma} = V / Q,$$

где  $V$  – основные габаритные размеры ТО;  $Q$  – показатель эффективности.

Обычно этот критерий является монотонно убывающей функцией и влияет на развитие всех объектов, кроме тех, у которых уменьшение габаритов функционально ограничено.

*Антропологические критерии.* Группа этих критериев обеспечивает по возможности наибольшее соответствие и приспособление ТО к человеку.

*Критерий эргономичности.* Свойство системы «человек – машина» менять свою эффективность в зависимости от степени использования возможностей человека называют эргономичностью. Эффективность ТО при этом в первую очередь выражается через функциональные критерии развития системы (производительность, надежность, точность). Критерий эргономичности для конкретного ТО равен отношению реализуемой эффективности системы «человек – машина» к максимально возможной эффективности этой же системы. Он представляет монотонно возрастающую функцию, стремящуюся к 1. Критерий эргономичности можно трактовать как КПД человека в системе. Роль этого критерия возрастает с усложнением ТО. Эргономика стремится создавать наиболее удобные для человека системы.

*Критерий безопасности ТО.* Многие технические системы могут оказывать на человека вредные воздействия разных степеней тяжести. В связи с этим введен критерий безопасности, учитывая который, ТО в своем развитии имеет тенденцию понижать или исключать вредные и опасные воздействия на людей:

$$K_6 = \sum_{i=1}^n \beta_i \gamma_i \frac{S_i}{S_i^n},$$

где  $n$  – число вредных и опасных факторов;  $\beta_i$  – весовой коэффициент  $i$ -го фактора при условии, что  $\sum_{i=1}^n \beta_i = 1$ ;  $\gamma_i$  – весовой коэффициент, который равен 1,

если  $S_i = S_i^n$ , а иначе  $1/\min(\beta_i)$ ;  $S_i$  – величина  $i$ -го вредного фактора;  $S_i^n$  – нормативная (предельно допустимая) величина  $i$ -го вредного фактора.

При отсутствии нарушений всех условий данный критерий принимает значение в диапазоне от 0 до 1, при нарушениях он становится больше 1, а в

случаях больших нарушений критерий безопасности может намного превышать 1. Данный критерий относится как к новым, так и к существующим ТО.

*Критерий экологичности* регулирует взаимоотношения между ТО и окружающей средой. Он выражается формулой

$$K_{\text{эк}} = (S_n + S_k) / S_o,$$

где  $S_n$  – площадь, на которой по одному или нескольким факторам имеются недопустимые загрязнения или изменения выше нормы, но ниже критических;  $S_k$  – площадь, на которой по одному или нескольким факторам имеются критические загрязнения и изменения, при которых жизнь человека становится невозможной;  $S_o$  – вся площадь.

К факторам загрязнения относятся инородные примеси, изменения в неживой и живой природе в виде отклонений от нормы. Вредное влияние на природу необходимо свести по возможности к минимуму, так как природа очень «напряжена» и любой перегиб может привести к экологической катастрофе.

### **1.9. Распознавание образов с помощью искусственных нейронных сетей**

Тематика искусственных нейронных сетей относится к междисциплинарной сфере знаний, связанных с биокибернетикой, электроникой, прикладной математикой, статистикой, автоматикой, медициной и др. Искусственные нейронные сети (ИНС) возникли на основе знаний о функционировании нервной системы живых существ. Они представляют собой попытку использования процессов, происходящих в нервных системах, для выработки новых технологических решений.

Постоянно расширяется круг задач, для решения которых применяются ИНС. В этот круг входит и распознавание разнообразных образов, отличающихся по природе, сложности и другим признакам.

Современные ИНС по сложности и «интеллекту» постоянно растут и развиваются, демонстрируя ряд ценных свойств.

1. *Обучение.* ИНС могут менять свое поведение в зависимости от внешней среды. После предъявления входных сигналов, возможно вместе с требуемыми выходами, они самонастраиваются, чтобы обеспечить ожидаемую реакцию.

2. *Обобщение.* Отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов. Важно, что ИНС делает обобщение автоматически благодаря своей структуре.

3. *Абстрагирование.* Если, например, предъявить сети несколько искаженных вариантов входного образа, то сеть сама сможет создать на выходе «идеальный» образ, с которым она никогда не встречалась.

Решение проблемы распознавания образов с помощью ИНС, как и в ранее рассмотренных случаях, состоит из двух процедур: обучения и непосредственно самого распознавания незнакомых образов.

Процедура поиска решения задачи с помощью сети, прошедшей обучение, оказывается более гибкой, чем использование других вычислительных средств, поскольку ИНС может повышать точность результатов по мере накопления ею опыта и адаптироваться к происходящим изменениям.

Одной из наиболее популярных моделей ИНС с контролируемым обучением считается ИНС в виде многослойного персептрона (рис. 1.8).

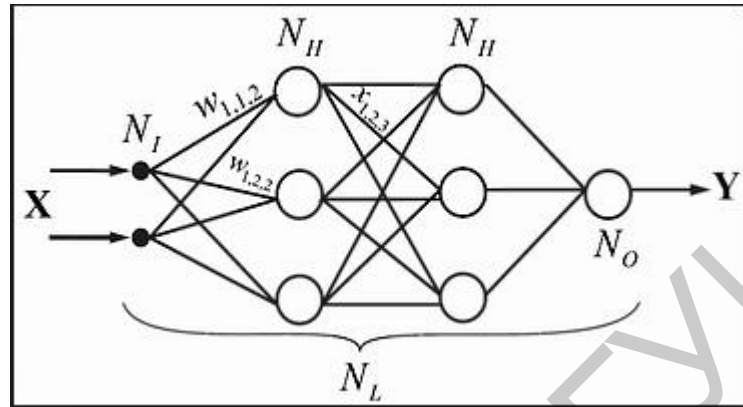


Рис. 1.8. ИНС в виде многослойного персептрона

Нейроны могут объединяться в сети различным образом. Сеть состоит из произвольного количества слоев нейронов, в простейшем случае – однослойная сеть. Первый слой называется сенсорным, или входным, внутренние слои называются скрытыми, или ассоциативными, последний – выходным, или результативным. Количество нейронов в слоях может быть произвольным. Обычно во всех скрытых слоях одинаковое количество нейронов. В каждом слое выполняется нелинейное преобразование линейной комбинации сигналов предыдущего слоя. Следовательно, в тех сетях, где требуется последовательное соединение слоев нейронов друг за другом, необходима нелинейная функция активации. В противном случае многослойность оказывается ненужной, так как ее можно заменить эквивалентной однослойной сетью с соответствующими весовыми коэффициентами. Многослойная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов.

Введем обозначения согласно рис. 1.8. Входной слой состоит из  $N_I$  нейронов; каждый скрытый слой содержит по  $N_H$  нейронов;  $N_O$  – количество выходных нейронов;  $x$  – вектор входных сигналов сети;  $y$  – вектор выходных сигналов.

Входной слой не выполняет никаких вычислений, а лишь распределяет входные сигналы, поэтому иногда его не учитывают, считая количество слоев в сети. Обозначим через  $N_L$  полное количество слоев в сети, считая и входной.

Работа многослойного персептрона описывается формулами:

$$NET_{jl} = \sum_i w_{ijl} x_{ijl},$$

$$OUT_{jl} = F(NE_{Tjl} - \theta_{jl}),$$

$$x_{ij(l+1)} = OUT_{il},$$

где индексом  $i$  обозначен номер входа;  $j$  – номер нейрона в слое;  $l$  – номер слоя.

Кроме того,  $x_{ijk}$  –  $i$ -й входной сигнал  $j$ -го нейрона в слое  $l$ ;  $w_{ijk}$  – весовой коэффициент  $i$ -го входа  $j$ -го нейрона в слое  $l$ ;  $NET_{jl}$  – сигнал NET  $j$ -го нейрона в слое  $l$ ;  $OUT_{jl}$  – выходной сигнал нейрона;  $\theta_{jl}$  – пороговый уровень  $j$ -го нейрона в слое  $l$ .

Введем еще некоторые обозначения:  $w_{jl}$  – вектор-столбец весов для всех входов нейрона  $j$  в слое  $l$ ;  $W_l$  – матрица весов всех нейронов слоя  $l$  (в столбцах матрицы расположены векторы  $w_{jl}$ ;  $x_{jl}$  – входной вектор-столбец слоя  $l$ ).

В каждом слое рассчитывается нелинейное преобразование от линейной комбинации сигналов предыдущего слоя.

## 2. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

### 2.1. Лабораторная работа №1

#### РАСПОЗНАВАНИЕ ОБРАЗОВ НА ОСНОВЕ КОНТРОЛИРУЕМОГО ОБУЧЕНИЯ

**Цель работы:** изучить особенности методов распознавания образов, использующих контролируемое обучение, и научиться классифицировать объекты с помощью алгоритма *K-средних*.

#### Порядок выполнения работы

1. Изучение теоретической части лабораторной работы.
2. Реализация алгоритма *K-средних*.
3. Защита лабораторной работы.

Процесс распознавания образов напрямую связан с процедурой обучения. Главная особенность контролируемого обучения заключается в обязательном наличии априорных сведений о принадлежности к определенному классу каждого вектора измерений, входящего в обучающую выборку. Роль обучающего состоит в том, чтобы помочь отнести каждый вектор из тестовой выборки к одному из имеющихся классов. И хотя классы известны заранее, необходимо уточнить и оптимизировать процедуры принятия решений. В основу всех алгоритмов распознавания образов положено понятие «расстояние», выступающее критерием в ходе принятия решений.

В качестве примера метода распознавания образов, использующего процедуру контролируемого обучения, рассмотрим алгоритм *K-средних*.

**Исходные данные** — число образов и число классов ( $K$ ), на которое нужно разделить все образы. Количество образов предлагается брать в диапазоне от 1000 до 100 000, число классов — от 2 до 20. Признаки объектов задаются случайным образом, это координаты векторов. Обычно  $K$  элементов из набора векторов случайным образом назначают центрами классов.

**Цель и результат работы алгоритма** — определить ядрами классов  $K$  типичных представителей классов и максимально компактно распределить вокруг них остальные объекты выборки.

*Примечание.* Результат работы представить графически.

На рис. 2.1 и 2.2 показаны примеры реализации алгоритма *K-средних* в случае распределения 20 000 объектов на 6 классов. На рис. 2.1 показана первая итерация алгоритма, на рис. 2.2 — завершающая итерация.

### Алгоритм $K$ -средних

1. Фиксируются  $K$  ядер (центров областей). Затем вокруг них формируются области по правилу минимального расстояния. На  $r$ -м этапе вектор  $\bar{X}_p$  связывается с ядром  $\bar{N}_i(r)$ , если удовлетворяется следующее неравенство:

$$\|\bar{X}_p - \bar{N}_i(r)\| < \|\bar{X}_p - \bar{N}_j(r)\| \forall i \neq j, \text{ тогда } \bar{X}_p \in \bar{N}_i(r).$$

2. На  $(r+1)$ -м этапе определяются новые элементы, характеризующие новые ядра  $\bar{N}_i(r+1)$ . За их значения принимают векторы  $\bar{X}$ , обеспечивающие минимум среднеквадратичного отклонения:

$$J_i = \sum_{\bar{X}_p \in \bar{N}_i(r)} \|\bar{X}_p - \bar{N}_i(r+1)\|^2, i=1,2,\dots,K,$$

$J_i$  принимает минимальное значение лишь при одном  $\bar{X}$ , равном среднему арифметическому векторов, принадлежащих одной области  $N_i$ .

3. Если хотя бы в одной из областей поменялось положение ядра, то пересчитываются области принадлежащих им векторов, т. е. определяются расстояния от объектов (не ядер) до новых ядер. В результате этого может произойти перераспределение областей. Затем повторяется шаг 2. Процедура заканчивается, если на  $(r+1)$ -м шаге ее выполнения положения центров областей не меняются по сравнению с  $r$ -м шагом.

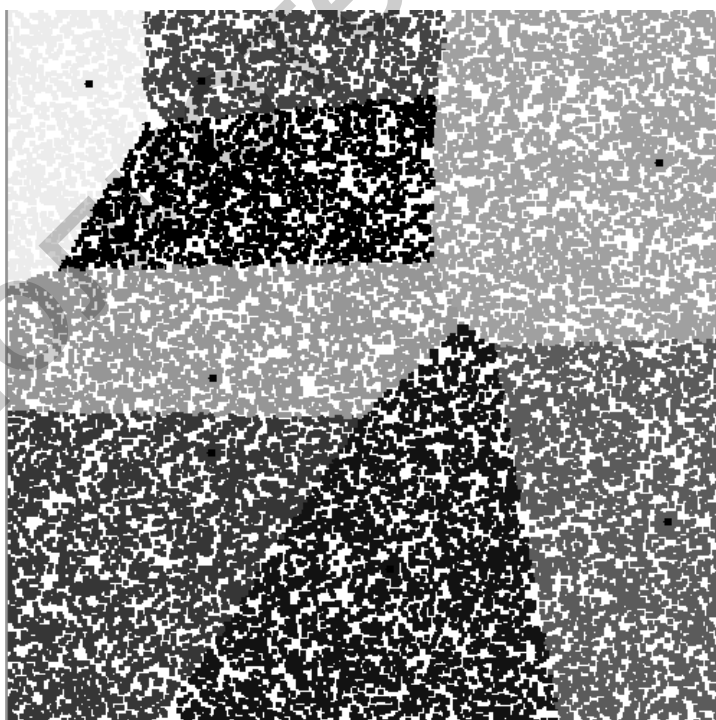


Рис. 2.1. Начальное распределение объектов в алгоритме  $K$ -средних



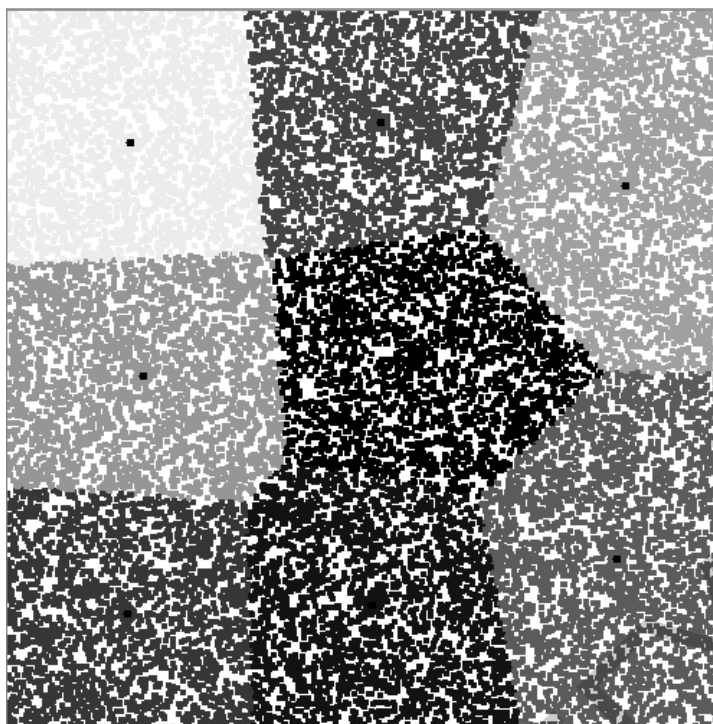


Рис. 2.2. Результат работы алгоритма  $K$ -средних

## 2.2. Лабораторная работа №2

### РАСПОЗНАВАНИЕ ОБРАЗОВ НА ОСНОВЕ САМООБУЧЕНИЯ

**Цель работы:** изучить особенности распознавания образов в самообучающихся системах и научиться классифицировать объекты с помощью алгоритма *максимина*.

#### Порядок выполнения работы

1. Изучение теоретической части лабораторной работы.
2. Реализация алгоритма *максимина*.
3. Защита лабораторной работы.

По сравнению с методами контролируемого обучения алгоритмы самообучения отличаются большей неполнотой информации. В этих алгоритмах не известны ни классы, ни их количество, ни признаки. Необходимым минимумом информации для классификации объектов являются сами образы и их признаки, без этого не выполняется ни один алгоритм. В обучении «без учителя» алгоритм самостоятельно определяет классы, на которые делится исходное множество данных, и одновременно определяет присущие им признаки. Для разделения данных используется следующий универсальный критерий. Процесс организуется так, чтобы среди всех возможных вариантов группировок найти такой, когда группы обладают наибольшей компактностью.

В качестве примера метода распознавания образов, использующего процедуру самообучения, рассмотрим алгоритм *максимина*.

**Исходные данные** – число образов, которые нужно разделить на классы. Количество образов предлагается брать в диапазоне от 1000 до 100 000. Признаки объектов задаются случайным образом, это координаты векторов.

**Цель и результат работы алгоритма** – исходя из произвольного выбора максимально компактно разделить объекты на классы, определив ядро каждого класса.

*Примечание.* Результат работы представить графически.

На рис. 2.3 показан пример реализации алгоритма *максимина* в случае распределения по классам 20 000 объектов. В результате было определено 8 классов образов.

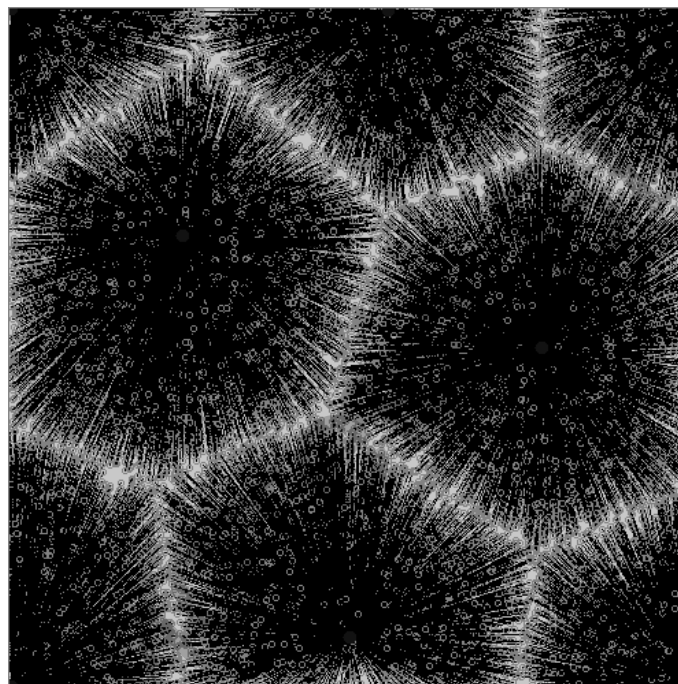


Рис. 2.3. Результат работы алгоритма максимина

#### *Алгоритм максимина*

1. Из множества векторов  $X = \{X(1), X(2), X(3), \dots, X(V)\}$  произвольно выбирается один и назначается ядром первого класса. Пусть  $N_1 = X(1)$ . Затем будут определяться другие ядра  $N_2, N_3, \dots, N_m$ , число которых заранее неизвестно.

2. Вычисляются расстояния  $d_{1i}(\bar{N}_1, \bar{X}(i)) \forall i \neq 1$ . Ядро  $N_2$  выбирается следующим образом:  $\bar{N}_2 = \bar{X}(l)$ , где  $d_{1l} = \max d_{1i}(\bar{N}_1, \bar{N}(i))$ .

3. Выполняется распределение оставшихся объектов по классам по критерию минимального расстояния.

4. В каждом классе вычисляются расстояния от ядра до каждого объекта данного класса:  $d_{ki} = d(\bar{N}_k, \bar{X}(i))$ ,  $k = 1, 2$ ;  $i = 1, 2, \dots, \nu - k$ , среди которых находятся наибольшие  $\delta_{ki} = \max(d_{ki})$ ,  $k = 1, 2$  (пока имеется два максимума).

5. Выбирается максимальное среди всех максимальных расстояний, которое становится претендентом на очередное ядро. Это значение  $\delta_{kp}$ . Если  $\delta_{kp}$  больше половины среднего арифметического расстояния между всеми ядрами, то создается очередное ядро  $\bar{N}_3 = \delta_{kp} = X(p)$  и выполняется переход к шагу 3, иначе алгоритм останавливается.

*Комментарий.* Новое ядро вводится по следующим соображениям:  $N_1$  и  $N_2$  – ядра двух классов, а один из векторов  $X$  удален от одного из этих ядер на расстояние, превышающее половину расстояния между ядрами. Следовательно,  $\bar{X}$  не относится ни к одному из существующих классов и становится ядром очередного класса. Алгоритм останавливается, когда ни в одном из классов не будет найден объект, для которого выполнится условие из шага 5. К этому моменту найдено  $m$  классов и их ядра  $N_1, N_2, \dots, N_m$ .

### 2.3. Лабораторная работа №3

#### РАЗДЕЛЕНИЕ ОБЪЕКТОВ НА ДВА КЛАССА ПРИ ВЕРОЯТНОСТНОМ ПОДХОДЕ

**Цель работы:** изучить особенности классификации объектов при вероятностном подходе и научиться находить ошибку классификации.

#### Порядок выполнения работы

1. Изучение теоретической части лабораторной работы.
2. Выполнение классификации случайной величины и определение ошибки классификации.
3. Защита лабораторной работы.

#### Исходные данные:

1. Две случайные величины, распределенные по закону Гаусса.
2. Априорные вероятности отнесения каждой из случайных величин к первому из двух классов, в зависимости от того, для какого из них определяется ошибка классификации.

**Выходные данные:** вероятность ложной тревоги, вероятность пропуска обнаружения ошибки, вероятность суммарной ошибки классификации. Результаты работы программы должны представляться в графическом виде.

*Примечание.* Результат работы представить графически.

На основе апостериорных вероятностей можно разработать метод автоматической классификации. Примером апостериорной плотности вероятности является случай одномерного гауссова распределения, выражаемого формулой (2.1).

$$p(x/j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{x - \mu_j}{\sigma_j}\right)^2\right]. \quad (2.1)$$

Плотность распределения является функцией двух параметров:  $\mu_j$  – математическое ожидание и  $\sigma_j$  – среднеквадратичное отклонение. Эти параметры могут быть вычислены по  $N$  опытам, в каждом из которых измеряется величина  $x_k$  ( $k = 1, 2, \dots, N$ ), а затем вычисляются  $\hat{\mu}_j = \frac{1}{N} \sum_{k=1}^N x_k$ ;  $\hat{\sigma}^2_j = \frac{1}{N} \sum_{k=1}^N (x_k - \hat{\mu}_j)^2$ .

Пусть задано сепарабельное пространство признаков, которое по определению может быть разделено на классы.  $X$  – вектор, представляющий  $k$ -й класс, сепарабельного пространства. Априорная вероятность того, что  $X$  относится к классу с номером  $k$ , есть  $P(X_k)$ . Она считается заданной самой постановкой задачи.

Задача заключается в том, чтобы отнести неизвестный предъявляемый объект  $X$  к одному из известных классов  $C_k$  с минимальной ошибкой. Для этого выполняют  $n$  измерений в соответствии с признаками, выбранными надлежащим образом. В результате получают вектор измерений  $X_m$ , для которого можно найти условную вероятность или ее плотность:  $p(X_m/C_k)$ .

Решение об отнесении неизвестного объекта к классу с номером  $k$  можно считать оправданным, если для любого  $j$  выполняется условие

$$p(C_k / \vec{X}_m) \geq p(C_j / \vec{X}_m) \quad \forall j.$$

Эти вероятности могут быть вычислены согласно теореме Байеса по тем условным вероятностям  $p(\vec{X}_m/C_k)$ , которые получаются непосредственно в процессе измерений:

$$P(C_k / \vec{X}_m) = \frac{P(C_k)p(\vec{X}_m/C_k)}{p(X_m)}, \quad P(C_j / \vec{X}_m) = \frac{P(C_j)p(\vec{X}_m/C_j)}{p(X_m)}.$$

Откуда следует решающее правило:

$$P(C_k)p(\vec{X}_m/C_k) \geq P(C_j)p(\vec{X}_m/C_j).$$

Рассмотрим случай, когда весь набор возможных решений сводится к двум, т. е. предъявленный объект может быть отнесен к одному из двух имею-

щихся классов. На рис. 2.4 показаны плотности распределения случайной величины  $X_m$  в случае ее отнесения к классам  $C_1$  и  $C_2$ .  $P(C_1)$  – это вероятность отнесения  $X_m$  к классу  $C_1$ , а  $P(C_2)$  – вероятность отнесения случайной величины к классу  $C_2$ . Рассмотрим вероятности ошибок, которые могут возникать при такой процедуре. Очевидно, что на прямой  $AB$  неравенство Бейеса выполняется, и можно заключить, что  $X_m$  принадлежит классу  $C_1$ .

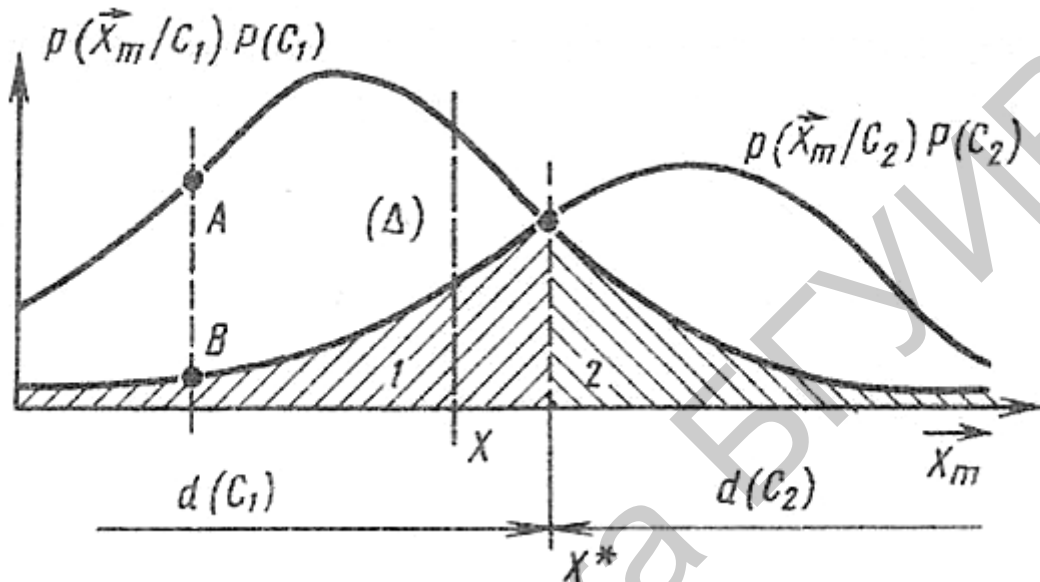


Рис. 2.4. Плотности распределения случайной величины

Рассмотрим линию раздела, обозначенную  $\Delta$ . Любая точка, для которой  $X_m < X$ , считается принадлежащей классу  $C_1$ , в то время как все точки, для которых  $X_m > X$ , относятся к классу  $C_2$ . Однако вероятность того, что в первом случае точка может принадлежать классу  $C_2$ , отлична от нуля (область 1), так же как и то, что во втором случае точка  $X$  принадлежит классу  $C_1$  (область 2). Для класса  $C_1$  зона 1 является зоной ложной тревоги, а зона 2 является зоной пропуска обнаружения. Они определяются соответственно выражениями:

$$P_{\text{л.т}} = \int_{-\infty}^x P(C_2) p(\vec{X}_m / C_2) d\vec{X}_m; \quad P_{\text{п.о}} = \int_x^{\infty} P(C_1) p(\vec{X}_m / C_1) d\vec{X}_m.$$

Суммарная ошибка классификации представляется суммой этих двух вероятностей. Если перемещать линию  $\Delta$ , разделяющую два решения, вдоль оси  $X$ , то она должна достичь точки  $X^*$ , в которой имеет место равенство  $P(C_1) p(\vec{X}_m / C_1) = P(C_2) p(\vec{X}_m / C_2)$ , показывающее, что при бинарных ценах правило максимума правдоподобия обеспечивает оптимальную классификацию по отношению к возможности ошибочного решения.

## 2.4. Лабораторная работа №4

### КЛАССИФИКАЦИЯ ОБЪЕКТОВ НА $N$ КЛАССОВ МЕТОДОМ ПЕРСЕПТРОНА

**Цель работы:** изучить особенности классификации объектов методом персептрона, а также научиться применять этот метод на практике.

#### Порядок выполнения работы

1. Ознакомление с теоретической частью лабораторной работы.
2. Реализация метода персептрона.
3. Оформление отчета по лабораторной работе.

#### Исходные данные:

1.  $N$  – количество классов, на которые требуется разделить объекты.
2. Обучающая выборка, представленная векторами с наборами признаков.

**Выходные данные:**  $N$  решающих функций.

После того как получено  $N$  решающих функций, предъявляются объекты тестовой выборки, которые необходимо классифицировать, отнеся к одному из классов. Тестовый объект подставляется в каждую из решающих функций и относится к тому из классов, где было получено максимальное значение.

Количество классов, обучающих объектов и их признаков может быть произвольным.

Допустим существование  $M$  решающих функций, характеризующихся тем свойством, что при  $x \in \omega_i$ , где  $x$  – объект,  $\omega_i$  – класс  $d_i(x) > d_j(x)$  для всех  $i \neq j$ .

Рассмотрим  $M$  классов  $\omega_1, \omega_2, \dots, \omega_M$ . Пусть на  $k$ -м шаге процедуры обучения системе предъявляется образ  $x(k)$ , принадлежащий классу  $\omega_i$ . Вычисляются значения  $M$  решающих функций  $d_j[x(k)] = w_j(k)x(k)$ ,  $j = 1, 2, \dots, M$ . Затем если выполняются условия  $d_i[x(k)] > d_j[x(k)]$ ,  $j = 1, 2, \dots, M; j \neq i$ , то векторы весов не изменяются, т. е.  $w_j(k+1) = w_j(k)$ ,  $j = 1, 2, \dots, M$ .

С другой стороны, допустим, что для некоторого  $l$   $d_i[x(k)] \leq d_l[x(k)]$ . В этом случае выполняются следующие коррекции весов:

$$\begin{aligned}w_i(k+1) &= w_i(k) + cx(k), \\w_l(k+1) &= w_l(k) - cx(k), \\w_j(k+1) &= w_j(k), j = 1, 2, \dots, M; j \neq i, j \neq l,\end{aligned}\tag{2.2}$$

где  $c$  – положительная константа.

Если при рассмотрении случая 3 классы разделимы, то доказано, что этот алгоритм сходится за конечное число итераций при произвольных начальных векторах. Рассмотрим это на примере.

Даны классы, причем каждый из них содержит один образ:  $\omega_1: \{(0, 0)\}$ ,  $\omega_2: \{(1,1)\}$ ,  $\omega_3: \{(-1, 1)\}$ . Дополним заданные образы:  $(0, 0, 1)$ ,  $(1, 1, 1)$ ,  $(-1, 1, 1)$ .

Выберем в качестве начальных векторов весов  $w_1(1) = w_2(1) = w_3(1) = (0, 0, 0)$ , положим  $c = 1$  и, предъявляя образы в указанном порядке, получим следующее:

$$\begin{aligned}d_1[x(1)] &= w_1(1)x(1) = 0, \\d_2[x(1)] &= w_2(1)x(1) = 0, \\d_3[x(1)] &= w_3(1)x(1) = 0.\end{aligned}$$

Поскольку  $x(1) \in \omega_1$  и  $d_2[x(1)] = d_3[x(1)] = d_1[x(1)]$ , первый весовой вектор увеличивается, а два других уменьшаются в соответствии с соотношениями (2.2), т. е.

$$\begin{aligned}w_1(2) &= w_1(1) + x(1) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \\w_2(2) &= w_2(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \\w_3(2) &= w_3(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.\end{aligned}$$

Следующий предъявляемый образ  $x(2) = (1, 1, 1)$  принадлежит классу  $\omega_2$ . Для него получаем

$$w_1(2)x(2) = 1, w_2(2)x(2) = -1, w_3(2)x(2) = -1.$$

Поскольку все произведения больше либо равны  $w_2(2)x(2)$ , вводятся корректировки векторов коэффициентов:

$$w_1(3) = w_1(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix},$$

$$w_2(3) = w_2(2) + x(2) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

$$w_3(3) = w_3(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ  $x(3) = (-1, 1, 1)$  принадлежит классу  $\omega_3$ . Для него получаем  $w_1(3)x(3) = 0$ ,  $w_2(3)x(3) = 0$ ,  $w_3(3)x(3) = -2$ . Все эти произведения опять требуют корректировки:

$$w_1(4) = w_1(3) - x(3) = \begin{pmatrix} 0 \\ -2 \\ -1 \end{pmatrix},$$

$$w_2(4) = w_2(3) - x(3) = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(4) = w_3(3) + x(3) = \begin{pmatrix} -2 \\ 0 \\ -1 \end{pmatrix}.$$

Поскольку в данном цикле итерации присутствовали ошибки, следует провести новый цикл. Положив  $x(4) = x(1)$ ,  $x(5) = x(2)$ ,  $x(6) = x(3)$ , получим  $w_1(4)x(4) = -1$ ,  $w_2(4)x(4) = -1$ ,  $w_3(4)x(4) = -1$ . Так как образ  $x(4)$  принадлежит классу  $\omega_1$ , то все произведения «неверны». Поэтому

$$w_1(5) = w_1(4) + x(4) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(5) = w_2(4) - x(4) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(5) = w_3(4) - x(4) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$



Следующий предъявленный образ  $x(5) = (1, 1, 1)$  принадлежит классу  $\omega_2$ . Соответствующие скалярные произведения равны  $w_1(5)x(5) = -2$ ,  $w_2(5)x(5) = 0$ ,  $w_3(5)x(5) = -4$ . Образ  $x(5)$  классифицирован правильно. Поэтому

$$w_1(6) = w_1(5) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(6) = w_2(5) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(6) = w_3(5) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий образ  $x(6) = (-1, 1, 1)$  принадлежит классу  $\omega_3$ , для него получаем  $w_1(6)x(6) = -2$ ,  $w_2(6)x(6) = -4$ ,  $w_3(6)x(6) = -0$ . Этот образ также классифицирован правильно, так что коррекции не нужны, т. е.

$$w_1(7) = w_1(6) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(7) = w_2(6) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(7) = w_3(6) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Если продолжить процедуру обучения, рассматривая образы  $x(7)$ ,  $x(8)$ ,  $x(9)$ , можно убедиться, что в следующем полном цикле никакие коррекции не производятся. Поэтому искомые решающие функции имеют следующий вид:

$$d_1(x) = 0 \cdot x_1 - 2x_2 + 0 = -2x_2,$$

$$d_2(x) = 2x_1 - 0 \cdot x_2 - 2 = 2x_1 - 2,$$

$$d_3(x) = -2x_1 + 0 \cdot x_2 - 2 = -2x_1 - 2.$$

## 2.5. Лабораторная работа №5

### РАСПОЗНАВАНИЕ ОБЪЕКТОВ МЕТОДОМ ПОТЕНЦИАЛОВ

**Цель работы:** изучить особенности классификации объектов методом потенциалов, а также научиться применять этот метод на практике.

#### Порядок выполнения работы

1. Ознакомление с теоретической частью лабораторной работы.
2. Реализация метода потенциалов.
3. Оформление отчета по лабораторной работе.

**Исходные данные:** обучающая выборка из 4–6 объектов, представленных векторами с наборами признаков.

**Выходные данные:** разделяющая функция и решающее правило для классификации тестовых объектов.

*Примечание.* Результат работы представить графически.

После того как получено решающее правило и построена разделяющая функция, предъявляются объекты тестовой выборки, которые необходимо классифицировать, отнеся к одному из двух классов. Тестовая выборка также задается векторами с наборами признаков. Результаты работы программы должны представляться в графическом виде.

Метод потенциалов относится к группе алгоритмов контролируемого обучения, где все объекты делятся на обучающую и тестовую выборки. Алгоритм состоит из двух этапов.

На *первом этапе* задача состоит в поиске разделяющей функции, позволяющей, исходя из обучающей выборки, определить границу между двумя классами. Эту процедуру называют обучением системы. На *втором этапе* разделяющая функция используется для классификации заданных объектов.

Разделяющая функция находится с помощью суммарного потенциала  $K(\vec{x})$ , вычисляемого как сумма частных потенциалов  $K(\vec{x}, \vec{x}_i)$ , связанных с каждым отдельным предъявляемым источником  $i$ . Суммарный потенциал вычисляется по следующему алгоритму  $K_{i+1}(\vec{x}) = K_i(\vec{x}) + \rho_{i+1}K(\vec{x}, \vec{x}_{i+1})$ , в котором через  $i$  обозначен номер этапа, соответствующий номеру предъявляемого для распознавания объекта. Корректирующий член  $\rho_{i+1}$  удовлетворяет следующим условиям:

$$\rho_{i+1} = \begin{cases} 1, & \text{если } x_{i+1} \in C_1 \text{ и } K_i(\vec{x}, \vec{x}_{i+1}) \leq 0, \\ -1, & \text{если } x_{i+1} \in C_2 \text{ и } K_i(\vec{x}, \vec{x}_{i+1}) > 0, \\ 0 & \text{при правильной классификации.} \end{cases} \quad (2.3)$$

Правильная классификация соответствует случаям, когда  $K(x) > 0$  при  $\bar{x} \in C_1$  и  $K(x) < 0$  при  $\bar{x} \in C_2$ . Поэтому можно использовать  $K_i(\bar{x})$  как разделяющую функцию и определить ее итеративным путем:  $d_{i+1}(\bar{x}) = d_i(\bar{x}) + \rho_{i+1}K(\bar{x}, \bar{x}_{i+1})$ .

Поскольку интервал изменения аргументов  $x_1$  и  $x_2$  может простирается от  $-\infty$  до  $\infty$ , воспользуемся полиномами Эрмита, ограничиваясь первыми четырьмя слагаемыми и двумя переменными  $x_1$  и  $x_2$ . Полиномы связаны следующим рекуррентным соотношением:

$$H_{n+1} = 2xH_n - 2nH_{n-1}, \text{ где } H_0 = 1, H_1 = 2x.$$

Тогда определим значения первых четырех  $\phi_i(\bar{x})$ :

$$\begin{aligned}\phi_1(\bar{x}) &= H_0(x_1)H_0(x_2) = 1 \cdot 1 = 1; \\ \phi_2(\bar{x}) &= H_1(x_1)H_0(x_2) = 2x_1 \cdot 1 = 2x_1; \\ \phi_3(\bar{x}) &= H_0(x_1)H_1(x_2) = 1 \cdot 2x_2 = 2x_2; \\ \phi_4(\bar{x}) &= H_1(x_1)H_1(x_2) = 2x_1 \cdot 2x_2 = 4x_1x_2,\end{aligned}$$

при этом потенциальная функция  $K(\bar{x}, \bar{x}_i) = \sum_{n=1}^4 \phi_n(\bar{x})\phi_n(\bar{x}_i)$  для элемента  $x_i$  будет иметь вид

$$K(\bar{x}, \bar{x}_i) = 1 + 4x_1x_1^{(i)} + 4x_2x_2^{(i)} + 16x_1x_2x_1^{(i)}x_2^{(i)}, \quad (2.4)$$

где  $x_1^{(i)}$  – составляющая  $x_1$  от  $i$ -го элемента,  $x_2^{(i)}$  – составляющая  $x_2$  от  $i$ -го элемента.

Рассмотрим пример, в котором требуется построить разделяющую функцию между двумя классами  $C_1$  и  $C_2$ , для которых имеются представители: объекты  $X_1(-1, 0), X_2(1, 1) \in C_1$  и объекты  $X_3(2, 0), X_4(1, -2) \in C_2$ . В качестве начального значения разделяющей функции примем  $K_0(\bar{x}) = 0$ .

#### Метод потенциалов

1. Суммарный потенциал на первом шаге вычисляется через суммарный потенциал на нулевом шаге и частный потенциал в первом объекте-образце следующим образом:  $K_1(\bar{x}) = K_0(\bar{x}) + K(\bar{x}, \bar{x}_1)$ . Частный потенциал  $K(\bar{x}, \bar{x}_1)$  определяется с помощью выражения (2.4) путем подстановки в него координат первого объекта. В результате  $K_1(\bar{x}) = 1 - 4x_1$ . Определим значение разделяющей функции в точке  $X_2$ , подставив ее координаты в полученное выражение:  $K_1(\bar{x}_2) = 1 - 4 = -3 < 0$ . При такой классификации разделяющая функция требует корректировки в соответствии с равенством (2.3).

2.  $K_2(\vec{x}) = K_1(\vec{x}) + K(\vec{x}, \vec{x}_2)$ , где в результате подстановки координат объекта  $X_2$  в выражение (2.4) получаем  $K(\vec{x}, \vec{x}_2) = 1 + 4x_1 + 4x_2 + 16x_1x_2$ . Тогда  $K_2(\vec{x}) = 2 + 4x_2 + 16x_1x_2$ . Определим значение разделяющей функции в точке  $X_3$ , подставив ее координаты в полученное выражение:  $K_2(\vec{x}_3) = 2 > 0$ . При такой классификации разделяющая функция требует корректировки в соответствии с равенством (2.3).

3.  $K_3(\vec{x}) = K_2(\vec{x}) - K(\vec{x}, \vec{x}_3)$ , где в результате подстановки координат объекта  $X_3$  в выражение (2.4) получаем  $K(\vec{x}, \vec{x}_3) = 1 + 8x_1$ . Тогда  $K_3(\vec{x}) = 1 - 8x_1 + 4x_2 + 16x_1x_2$ . Определим значение разделяющей функции в точке  $X_4$ , подставив ее координаты в полученное выражение:  $K_3(\vec{x}_4) = -47 < 0$ . Классификация верна, и разделяющая функция не требует корректировки. Поэтому  $K_3(\vec{x}) = K_4(\vec{x})$ .

4. Поскольку в начале алгоритма было сделано предположение для первого объекта, проверяем, как классифицируется точка  $X_1$ :  $K_4(\vec{x}_1) = 9 > 0$ . Классификация верна, и разделяющая функция не требует корректировки.

Таким образом, все четыре объекта-образца классифицированы правильно, и разделяющая функция описывается уравнением

$d(\vec{x}) = 1 - 8x_1 + 4x_2 + 16x_1x_2$ , откуда  $x_2 = \frac{8x_1 - 1}{16x_1 + 4}$ . График этой функции при-

веден на рис. 2.5. На нем видно, что объекты  $X_1, X_2$ , принадлежащие первому классу, помечены значком  $\times$ , объекты  $X_3, X_4$ , принадлежащие второму классу, помечены значком  $\circ$ , и разделяющая функция является границей между областями двух классов.

На рис. 2.5 показана разделяющая функция, построенная по указанной обучающей выборке.

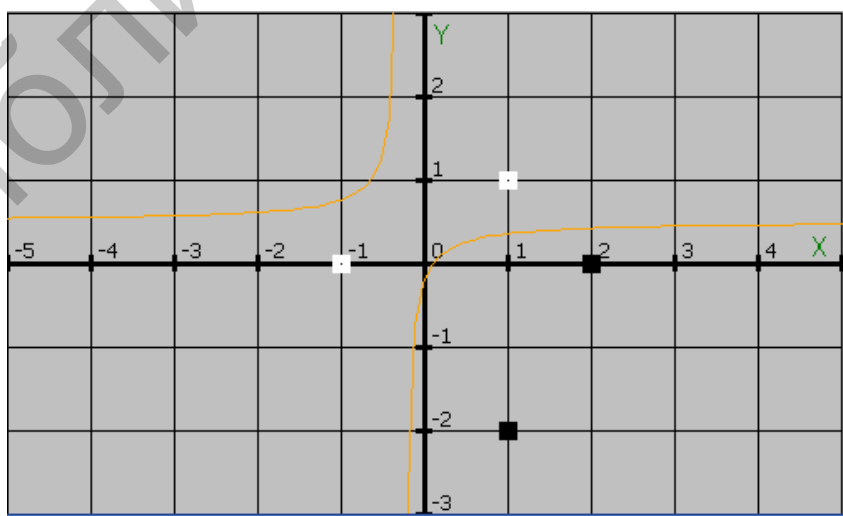


Рис. 2.5. Разделяющая функция и обучающие точки для двух классов

На рис. 2.6 показано распределение 250 точек на два класса с помощью ранее построенной разделяющей функции.

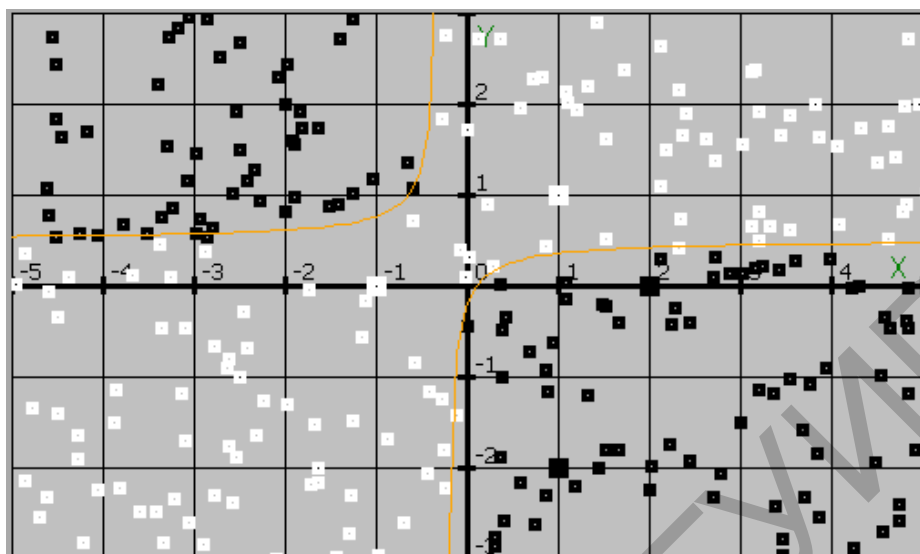


Рис. 2.6. Разделяющая функция и классификация 250 точек

Если в полученное уравнение разделяющей функции подставить координаты объектов-образцов, то для  $X_1$  и  $X_2$  ее значения будут положительными, а для  $X_3$  и  $X_4$  – отрицательными. Для классификации других объектов необходимо выполнить те же действия. Если значение разделяющей функции больше нуля, объект принадлежит первому классу, если ее значение меньше нуля, объект принадлежит второму классу. В случае нулевого значения разделяющей функции предъявляемый объект находится на границе классов.

## 2.6. Лабораторная работа №6

### КЛАССИФИКАЦИЯ ОБЪЕКТОВ МЕТОДОМ ИЕРАРХИЧЕСКОГО ГРУППИРОВАНИЯ

**Цель работы:** изучить правила построения иерархических группировок, а также метод классификации объектов на основе иерархических группировок.

#### Порядок выполнения работы

1. Ознакомление с теоретической частью лабораторной работы.
2. Реализация классификации объектов с помощью иерархий.
3. Оформление отчета по лабораторной работе.

**Исходные данные:**

1.  $n$  – количество объектов группирования.
2. Таблица расстояний между объектами. Таблица заполняется автоматически случайными значениями.

**Выходные данные:** иерархии, построенные по критериям минимума и максимума. Результаты работы программы должны представляться в графическом виде.

*Примечание.* Результат работы представить графически.

Методы распознавания, где классы известны заранее и разделяющие функции вырабатывались в процессе обучения, сильно влияют на выбор признаков и критериев разделения, от которых зависит получаемый результат.

Для того чтобы уменьшить влияние первоначальных сведений, их обогащают дополнительной информацией. Например, уточняют пространственные или временные отношения (общепринятое пространственное отношение: глаза на лице находятся выше носа); находят существующие отношения между исследуемыми объектами (в частности с помощью графов). Такие действия называются символическим описанием, которое получается в результате процедуры группирования, выполняющей роль и процедуры классификации.

Искомое символическое представление может иметь вид иерархической структуры, дерева минимальной длины или символического описания классов. Иерархия строится на основе понятия расстояния. Метод состоит в том, чтобы разработать последовательность разделений рассматриваемого множества на подгруппы, одна из которых обладает некоторым свойством, неприсущим другим. Искомая иерархия основывается на предъявляемых выборах. Поскольку их число весьма велико, иногда на одном и том же множестве исходных данных могут быть получены различные иерархии.

Рассмотрим правила построения иерархических группировок. Пусть  $X$  – множество, состоящее из  $m$  реализаций  $\{X_1, X_2, \dots, X_m\}$ , а  $P(X)$  – множество всех его частей:  $P(X) = \{0, X_1, \{X_1, X_2\}, \{X_1, X_3\}, \dots, X_m\}$ . Иерархией  $H$  называется подмножество, удовлетворяющее следующим условиям:

1.  $X \in H$ ;
2.  $\forall x_i \in X, x_i \in H$ ;
3.  $\forall h, h' \in H$ , если  $h \cap h' \neq 0$ , то либо  $h \subset h'$ , либо  $h' \subset h$ .

На практике чаще всего используется иерархия, обозначаемая вещественной функцией, откладываемой вдоль оси ординат. Эта функция называется расстоянием в широком смысле слова, поскольку она не связана с евклидовым расстоянием между двумя точками. Выбор расстояния обуславливает построение иерархии.

Существует ряд алгоритмов для построения иерархических группировок и иерархий на их основе. Рассмотрим пример построения иерархии по критерию минимума. В этом случае иерархические группы  $A$  и  $B$  объединяются, если  $d(A, B) = \min\{d(A, p), d(B, q)\}$ .

Даны четыре атома ( $x_1, x_2, x_3, x_4$ ), расстояния между ними приведены в табл. 2.1.

Таблица 2.1

Признаки	$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	0	5	0,5	2
$x_2$	5	0	1	0,6
$x_3$	0,5	1	0	2,5
$x_4$	2	0,6	2,5	0

$d(x_1, x_3) = 0,5$  – минимальное расстояние, содержащееся в таблице, следовательно, оно становится первым иерархическим объединением и обозначается  $d(x_1, x_3) = \{a\}$ , после чего элементы  $x_1$  и  $x_3$  в явном виде больше не участвуют в дальнейшем построении иерархии. Вместо них используется группировка  $a$ . Расстояния от нее до остальных элементов определяются следующим образом:

$$d\{a, x_2\} = \min\{d(x_1, x_2), d(x_3, x_2)\} = 1;$$

$$d\{a, x_4\} = \min\{d(x_1, x_4), d(x_3, x_4)\} = 2.$$

Продолжая процесс сокращения, выделяем новую группировку  $(x_2, x_4) = b$ , в результате остаются две группы  $a$  и  $b$ , объединяемые окончательно в  $c = \{a, b\} = 1$ . На рис. 2.7 показано дерево, соответствующее исходным данным.

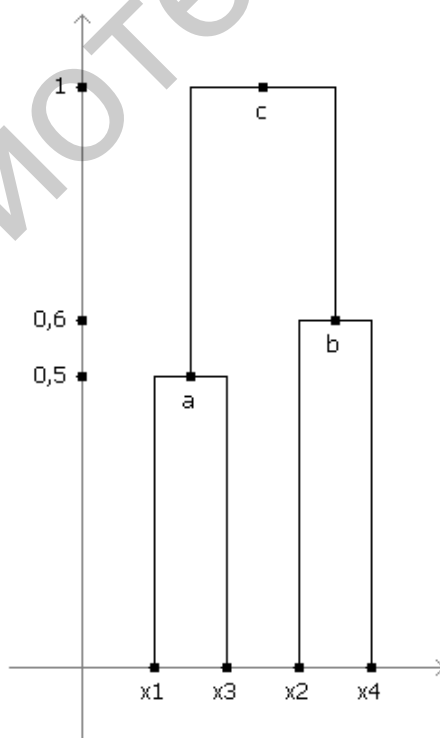


Рис. 2.7. Результирующее иерархическое дерево, построенное по критерию минимума

Для тех же исходных данных на рис. 2.8 приведена иерархия, построенная по критерию максимума. В этом случае иерархические группы  $A$  и  $B$  объединяются, если  $d(A, B) = \max\{d(A, p), d(B, q)\}$ .

Рассмотрим один из возможных вариантов построения такой иерархии. Заменяем числа в таблице расстояний их обратными значениями, т. е. числами вида  $1/5$ ,  $1/0,5$ ,  $1/2$  и т. д. После чего можно опять построить иерархию по критерию минимума, однако в этом случае самому маленькому числу будет соответствовать максимальное число из исходных данных и т. д. по убыванию всех значений.

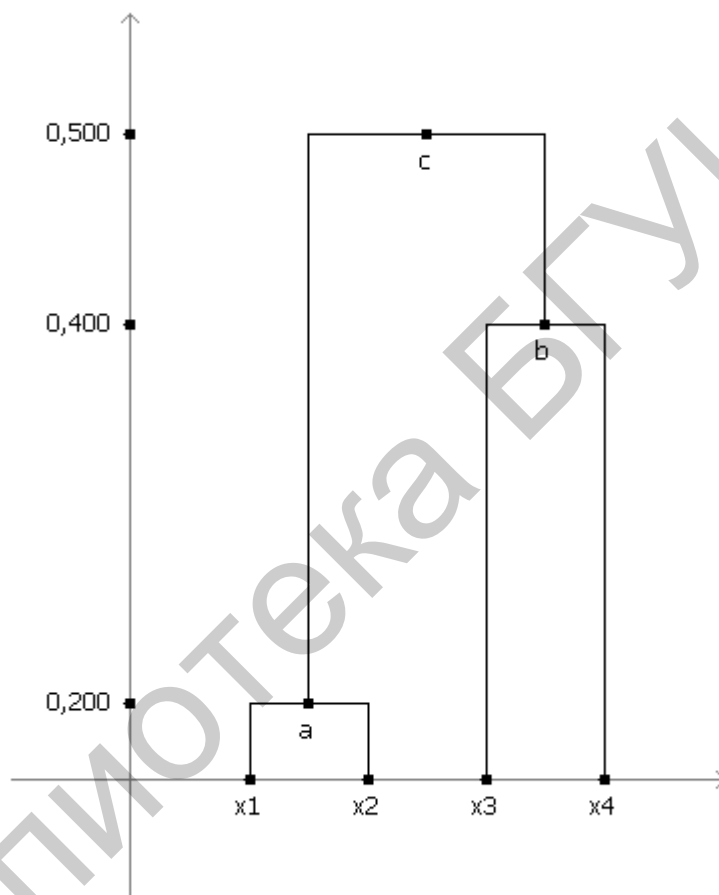


Рис. 2.8. Результирующее иерархическое дерево, построенное по критерию максимума

## 2.7. Лабораторная работа №7

### РАСПОЗНАВАНИЕ ОБЪЕКТОВ С ПОМОЩЬЮ СИНТАКСИЧЕСКИХ МЕТОДОВ

**Цель работы:** изучить особенности синтаксических методов распознавания объектов, методы распознавания объектов на основе деревьев и графов, а также типы грамматических разборов сверху вниз и снизу вверх.



## Порядок выполнения работы

1. Ознакомление с теоретической частью лабораторной работы.
2. Реализация распознавания объектов синтаксическими методами.
3. Оформление отчета по выполненному заданию.

**Исходные данные:** грамматика, задающая описание терминальных и нетерминальных объектов, а также правила построения и распознавания образов.

**Выходные данные:** изображения, построенные и классифицированные с помощью заданной грамматики.

*Примечание.* Результат работы представить графически.

После того как грамматика задана, она применяется для автоматического синтеза и распознавания требуемых образов. В качестве примеров рекомендуется использовать произвольное изображение, состоящее из 10–15 элементов, т. е. терминальных объектов.

Синтаксический метод распознавания основан на восприятии и обработке различных конструкций языка. Они могут делиться на более мелкие элементы, наименьшими из которых являются символы языка. Множество используемых символов называется алфавитом или словарем. Язык создается с помощью грамматики, которая определяется на основе правил построения, преобразования и взаимодействия слов. Следовательно, грамматика представляет собой множество правил, по которым строятся фразы, а значит, и сам язык.

Формально грамматика может быть задана следующей записью:

$$G = \langle V_n, V_t, P, S \rangle,$$

где  $V_n$  – нетерминальный словарь;  $V_t$  – терминальный словарь;  $P$  – множество правил подстановки;  $S$  – начальная аксиома ( $S \in V_n$ ).

Для грамматики характерны следующие соотношения:

$$V = V_n \cup V_t \text{ – словарь, } V_n \cap V_t = \emptyset, \quad P = \{\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \dots, \alpha_m \rightarrow \beta_m\},$$

где  $\alpha_i \in V^* - \{\lambda\}$ ,  $\beta_i \in V^*$ . Здесь  $\{\lambda\}$  – пустая строка,  $V^*$  – множество всех возможных последовательностей, которые удастся построить с помощью итерационных процедур на основе данного словаря.

Рассмотрим решение задач анализа и синтеза образов путем выполнения грамматических разборов. Пусть задана грамматика, порождающая геометрические фигуры типа квадратов.

$$G = \langle V_n, V_t, P, S \rangle$$

$$V_t = \{a_1, a_2\}, \quad V_n = \{S, O_1, O_2\},$$

$$P: S \rightarrow A(a_1, O_2), O_2 \rightarrow A(O_1, a_1), O_1 \rightarrow L(a_2, a_2),$$

где терминальными элементами служат горизонтальный и вертикальный отрезки определенной длины, обозначенные  $a_1$  и  $a_2$ , а высказывания  $A(x, y)$  и  $L(x, y)$  читаются соответственно « $x$  расположен над  $y$ » и « $x$  расположен слева от  $y$ ». Заданные структуры порождаются следующим набором грамматических правил:

1.  $S \rightarrow A(a_1, O_2)$ . Это правило заменяет начальный символ терминальным элементом  $a_1$ , расположенным над некоторым пока еще не определенным объектом  $a_2$ .

2. Правило  $O_2 \rightarrow A(O_1, a_1)$  заменяет нетерминальный объект  $O_2$  другим неопределенным объектом  $O_1$ , расположенным над горизонтальным отрезком  $a_1$ .

3.  $O_1 \rightarrow L(a_2, a_2)$ , где  $O_1$  заменяется на два вертикальных терминальных элемента.

Приведенный пример поясняет рис. 2.9.

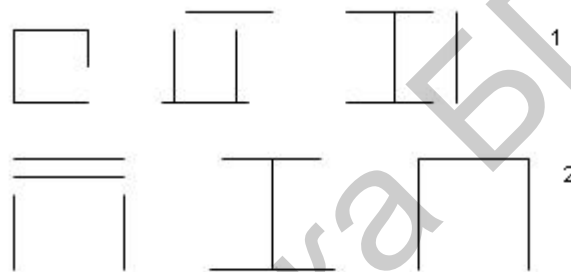


Рис. 2.9. Образы, поддающиеся разбору с помощью данной грамматики (1); образы, не поддающиеся разбору (2)

Изложенный грамматический разбор представляет собой тривиальную процедуру, так как в ней используется только одна последовательность правил подстановки. Данный недостаток алгоритмов, реализованных на основе деревьев, исправляется с помощью объектов, сводимых к структурам типа графов.

Интересным приложением лингвистических понятий в распознавании образов является язык PDL (Picture Description Language) – язык описания изображений. Терминальным элементом PDL служит любая  $n$ -мерная структура с двумя выделенными точками: хвостовой и головной.

По правилам языка PDL практически любая структура может обобщенно рассматриваться как ориентированный отрезок прямой, так как определение вводит для нее только две точки. Терминальные элементы связываются между собой только в хвостовых и (или) головных точках. Следовательно, структуры языка PDL представляют собой ориентированные графы и для их обработки можно использовать грамматики. На рис. 2.10 показаны типичные правила соединения терминалов языка PDL.

Кроме использования языка PDL грамматику можно расширить путем введения в ее правила подстановки рекурсивности, когда переменная способна

замещаться этой же переменной. Однако увеличение порождающей способности грамматики не всегда желательно. Особенно это касается тех исследований, где используется более одной грамматики. В этом случае чрезмерное их многообразие приводит к уменьшению различающей мощности каждой из грамматик.

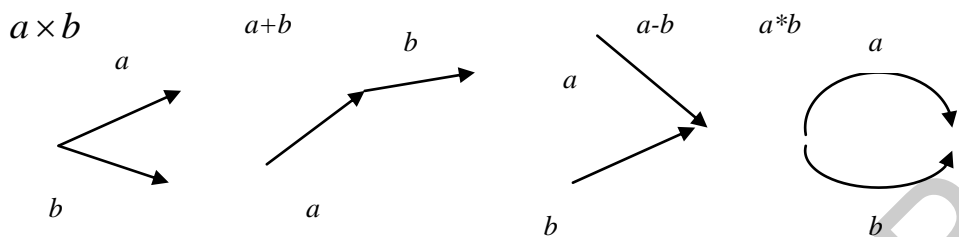


Рис. 2.10. Правила соединения терминалов

Рассмотрим практический пример синтаксического распознавания образов, в котором выполняется автоматическая классификация телоцентрических и V-образных хромосом. Классифицирующая грамматика имеет следующий вид:  
 $V_t = \{a, b, c, d, e\}$ ;

$V_n = \{S, T, \text{Основание}, \text{Сторона}, \text{Пара плеч}, \text{Правая часть}, \text{Левая часть}, \text{Плечо}\}$ .

На рис. 2.11 изображены терминальные элементы хромосом  $\{a, b, c, d, e\}$ , на рис. 2.12 – типичный вид телоцентрической и V-образной хромосом.



Рис. 2.11. Терминальные элементы хромосом

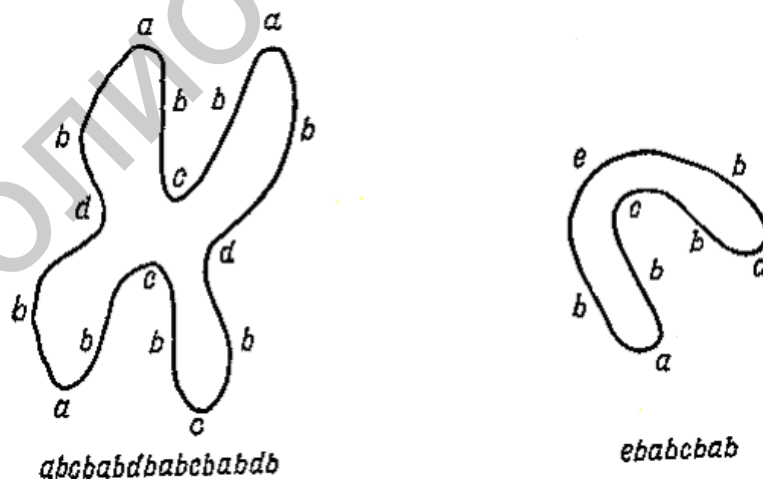


Рис. 2.12. Телоцентрическая и V-образная хромосомы

Оператор «•», используемый при построении правил грамматики, означает связность отдельных частей хромосомы, фиксируемую при продвижении вдоль ее границы по направлению часовой стрелки.

$P: S \rightarrow \text{Пара плеч} \bullet \text{Пара плеч},$   
 $T \rightarrow \text{Основание} \bullet \text{Пара плеч},$   
 $\text{Пара плеч} \rightarrow \text{Сторона} \bullet \text{Пара плеч},$   
 $\text{Пара плеч} \rightarrow \text{Пара плеч} \bullet \text{Сторона},$   
 $\text{Пара плеч} \rightarrow \text{Плечо} \bullet \text{Правая часть},$   
 $\text{Пара плеч} \rightarrow \text{Левая часть} \bullet \text{Плечо},$   
 $\text{Левая часть} \rightarrow \text{Плечо} \bullet c,$   
 $\text{Правая часть} \rightarrow c \bullet \text{Плечо},$   
 $\text{Основание} \rightarrow b \bullet \text{Основание},$   
 $\text{Основание} \rightarrow \text{Основание} \bullet b,$   
 $\text{Основание} \rightarrow e,$   
 $\text{Сторона} \rightarrow b \bullet \text{Сторона},$   
 $\text{Сторона} \rightarrow \text{Сторона} \bullet b,$   
 $\text{Сторона} \rightarrow b,$   
 $\text{Сторона} \rightarrow d,$   
 $\text{Плечо} \rightarrow b \bullet \text{Плечо},$   
 $\text{Плечо} \rightarrow \text{Плечо} \bullet b,$   
 $\text{Плечо} \rightarrow a.$

Начальные символы  $S$  и  $T$  представляют телоцентрические и  $V$ -образные хромосомы. Для разделения на два класса используется одна грамматика с двумя начальными символами. Если грамматический разбор снизу приводит к начальному символу  $T$ , хромосому относят в класс  $V$ -образных. Если же разбор приводит к  $S$ , хромосома классифицируется как телоцентрическая. В силу схожести поставленных задач целесообразно их решать в рамках одной грамматики. На рис. 2.13 приведено дерево, отражающее порядок разбора предложения.

В качестве первого шага на пути распознавания заданного цифрового изображения хромосомы необходимо найти точку на границе хромосомы и затем осуществлять продвижение вдоль границы по направлению часовой стрелки. По мере продвижения система процедур распознавания обеспечит обнаружение терминальных элементов  $\{a, b, c, d, e\}$ . В результате такого отслеживания границы хромосома оказывается эффективно сведенной к цепочке терминальных элементов и образует терминальное предложение, как показано на рис. 2.13. После сведения хромосомы к терминальному предложению начинается его синтаксическое распознавание. Рассмотрим предложение для телоцентрической хромосомы и применим к нему разбор снизу вверх. Будет происходить обратный порядок применения правил подстановки, начиная с правила  $\text{Плечо} \rightarrow a$ .

Рассмотрим алгоритм проверки классификации хромосомы.

1. Анализатор находит  $a$  и выдает нетерминал *Плечо*. Символ  $a$  находится 4 раза, что приводит к появлению четырех нетерминалов *Плечо* на первом уровне поиска, считая снизу.
2. Сочетание *Плеча* с терминалом  $b$ .
3. Порождение *Плеч*.
4. Порождение нетерминала *Сторона* при помощи символов  $d$  и  $b$ .
5. Комбинация *Плеча* и  $c$  порождает *Правую часть*.

- ## 8. Объединение двух Пар Плеч в $S$ .

Поскольку за конечное число шагов алгоритм закончился на символе  $S$ , хромосома была правильно классифицирована как телоцентрическая.

Предложенный грамматический разбор привел к искомому результату при первой реализации. Так получается далеко не всегда, поскольку обычно приходится выполнять частые возвраты. Однако их число можно минимизировать введением в процесс поиска эвристических правил, указывающих грамматическому анализатору способ действия в ситуациях, когда возможны несколько вариантов продолжения.

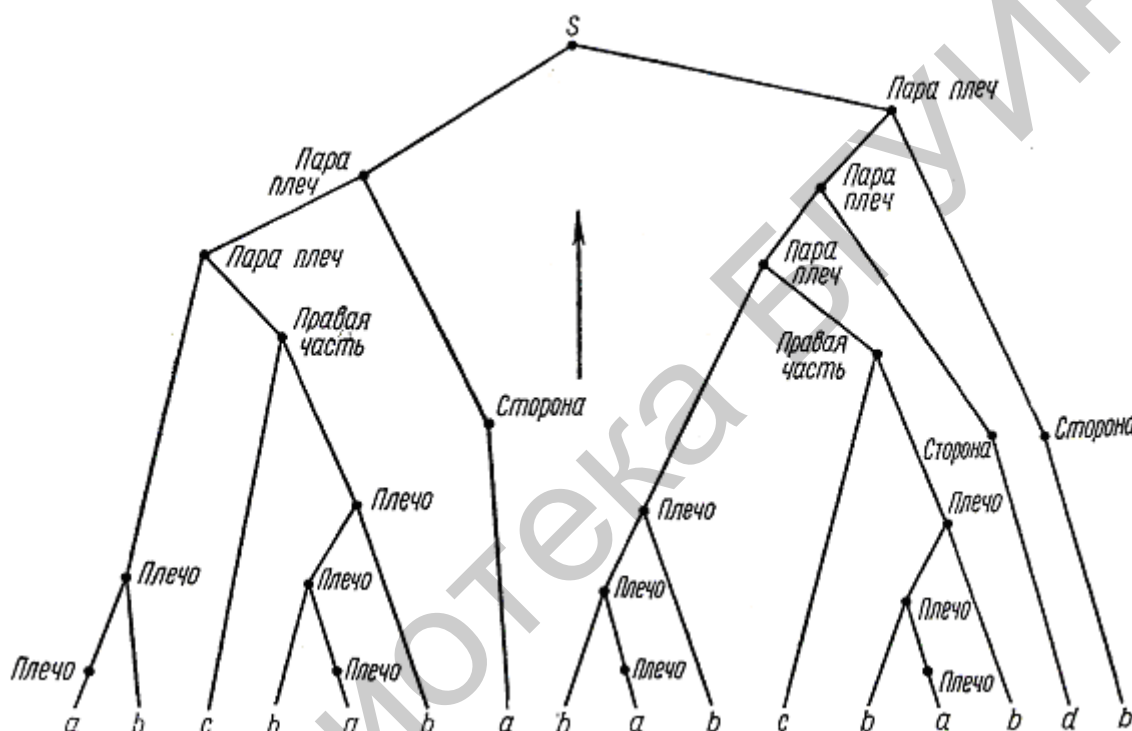


Рис. 2.13. Восходящий грамматический разбор хромосомы

## 2.8. Лабораторная работа №8

# МЕТОД АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ТЕКСТОВОЙ ИНФОРМАЦИИ

**Цель работы:** научиться строить автоматические грамматики на основе обучающей выборки.

## Порядок выполнения работы

- ## 1. Изучение теоретической части работы.

2. Реализация алгоритма автоматического синтеза текстовой информации с помощью синтаксического метода.

3. Оформление отчета по лабораторной работе.

**Исходные данные:** заданное множество терминальных цепочек.

**Выходные данные:** грамматика, способная автоматически порождать заданные цепочки, а также бесконечное множество цепочек, схожих по структуре с исходными.

Используя лингвистическую терминологию, процедуру получения решений с помощью обучающей выборки легко интерпретировать как задачу получения грамматики из множества выборочных предложений. Эта процедура называется грамматическим выводом и играет важную роль в изучении синтаксического распознавания образов в связи с ее значением для реализации автоматического обучения. На рис. 2.14 представлена модель вывода цепочечных грамматик.



Рис. 2.14. Модель вывода цепочечных грамматик

Задача, показанная на рис. 2.14, заключается в том, что множество выборочных цепочек подвергается обработке с помощью адаптивного обучающего алгоритма, представленного блоком. На выходе этого блока в конечном счете воспроизводится грамматика  $G$ , согласованная с данными цепочками, т. е. множество цепочек  $\{x_i\}$  является подмножеством языка  $L(G)$ . Пока ни одна из известных схем не в состоянии решить эту задачу в общем виде. Вместо этого предлагаются многочисленные алгоритмы для вывода ограниченных грамматик. Рассмотрим один из алгоритмов, в котором сначала строят нерекурсивную грамматику, порождающую в точности заданные цепочки, а затем, срачивая нетерминальные элементы, получают более простую рекурсивную грамматику, порождающую бесконечное число цепочек. Алгоритм можно разделить на три части. Первая часть формирует нерекурсивную грамматику. Вторая часть преобразует ее в рекурсивную грамматику. В третьей части происходит упрощение этой грамматики.

Рассмотрим выборочное множество терминальных цепочек (*caaab*, *bbaab*, *caab*, *bbab*, *cab*, *bbb*, *cb*). Требуется получить грамматику, способную автоматически порождать эти цепочки. Алгоритм построения грамматики состоит из следующих этапов.

1 этап. Строится нерекурсивная грамматика, порождающая в точности заданное множество выборочных цепочек. Они обрабатываются в порядке уменьшения длины. Правила подстановки строятся и прибавляются к грамматике по мере того, как они становятся нужны для построения соответствующей цепочки из выборки. Заключительное правило подстановки, используемое для порождения самой длинной выборочной цепочки, называется *остаточным правилом*, а длина его правой части равна 2 (это значение выбрано для удобства алгоритма). Остаточное правило длины  $n$  имеет вид  $A \rightarrow a_1 a_2, \dots, a_n$ , где  $A$  – нетерминальный символ, а  $a_1 a_2, \dots, a_n$  – терминальные элементы. Предполагается, что остаток каждой цепочки максимальной длины является суффиксом (хвостом) некоторой более короткой цепочки. Если какой-либо остаток не отвечает этому условию, цепочка, равная остатку, добавляется к обучающей выборке.

В нашем примере первой цепочкой максимальной длины в обучающей выборке является *caaab*. Для ее порождения строятся следующие правила подстановки:  $S \rightarrow cA_1, A_1 \rightarrow aA_2, A_2 \rightarrow aA_3, A_3 \rightarrow ab$ , где  $A_3$  – правило остатка.

Вторая цепочка – *bbaab*. Для ее порождения к грамматике добавляются следующие правила:  $S \rightarrow bA_4, A_4 \rightarrow bA_5, A_5 \rightarrow aA_6, A_6 \rightarrow ab$ . Поскольку цепочки *bbaab* и *caaab* имеют одинаковую длину, требуется остаточное правило длины 2. Работа первого этапа алгоритма приводит к некоторой избыточности правил подстановки. Например, вторая цепочка может быть также получена введением следующих правил подстановки:  $S \rightarrow bA_4, A_4 \rightarrow bA_2$ . Но первый этап алгоритма занимается лишь определением множества правил постановки, которое способно в точности порождать обучающую выборку, и не касается вопроса избыточности. Устранение избыточности выполняется на третьем этапе алгоритма. Для порождения третьей цепочки *saab* требуется добавление к грамматике только одного правила  $A_3 \rightarrow b$ . Рассмотрев остальные цепочки из обучающей выборки, устанавливаем, что окончательно множество правил подстановки для порождения выборки выглядит так:

$$S \rightarrow cA_1, S \rightarrow bA_4, A_1 \rightarrow aA_2, A_1 \rightarrow b, A_2 \rightarrow aA_3, A_2 \rightarrow b, A_3 \rightarrow ab, A_3 \rightarrow b, \\ A_4 \rightarrow bA_5, A_5 \rightarrow aA_6, A_5 \rightarrow b, A_6 \rightarrow ab, A_6 \rightarrow b.$$

2 этап. Здесь, соединяя каждое правило остатка длиной 2 с другим (неостаточным) правилом грамматики, получаем рекурсивную автоматную грамматику. Это происходит в результате слияния каждого нетерминального элемента правила остатка с нетерминальным элементом неостаточного правила, который может порождать остаток. Например, если  $A_r$  – остаточный нетерминал вида  $A_r \rightarrow a_1 a_2$  и  $A_n$  – неостаточный нетерминал вида  $A_n \rightarrow a_1 A_m$ , где  $A_m \rightarrow a_2$ , все встречающиеся  $A_r$  заменяются на  $A_n$ , а правило подстановки  $A_r \rightarrow a_1 a_2$  отбрасывается. Так создается автоматная грамматика, способная порождать данную обучающую выборку, а также обладающая общностью, достаточной для порождения бесконечного множества других цепочек. В рассматри-

ваемом примере  $A_6$  может сливаться с  $A_5$ , а  $A_3$  может сливаться с  $A_2$ , образуя следующие правила подстановки:

$$S \rightarrow cA_1, S \rightarrow bA_4, A_1 \rightarrow aA_2, A_1 \rightarrow b, A_2 \rightarrow aA_2, A_2 \rightarrow b, A_4 \rightarrow bA_5, \\ A_5 \rightarrow aA_5, A_5 \rightarrow b.$$

Рекурсивными правилами являются  $A_2 \rightarrow aA_2$  и  $A_5 \rightarrow aA_5$ .

3 этап. Грамматика, полученная на втором этапе, упрощается объединением эквивалентных правил подстановки. Два правила с левыми частями  $A_i$  и  $A_j$  эквивалентны, если выполняются следующие условия. Предположим, что, начиная с  $A_i$ , можно породить множество цепочек  $\{x\}_i$  и, начиная с  $A_j$ , можно породить множество цепочек  $\{x\}_j$ . Если  $\{x\}_i = \{x\}_j$ , то два правила подстановки считаются эквивалентными и каждый символ  $A_j$  может быть заменен на  $A_i$  без ущерба для языка, порождаемого этой грамматикой, т. е. два правила эквивалентны, если они порождают тождественные цепочки языка.

В рассматриваемом примере эквивалентны правила с левыми частями  $A_1$  и  $A_2$ . После слияния  $A_1$  и  $A_2$  получаем

$$S \rightarrow cA_1, S \rightarrow bA_4, A_1 \rightarrow aA_1, A_1 \rightarrow b, A_4 \rightarrow bA_5, A_5 \rightarrow aA_5, A_5 \rightarrow b,$$

где исключены многократные повторения одного и того же правила. Следующая пара эквивалентных правил –  $A_1$  и  $A_5$ . Выполнив преобразования для них, получим

$$S \rightarrow cA_1, S \rightarrow bA_4, A_1 \rightarrow aA_1, A_1 \rightarrow b, A_4 \rightarrow bA_4.$$

Дальнейшее слияние правил невозможно, поэтому алгоритм в процессе обучения строит следующую автоматную грамматику:

$$G = (V_N, V_T, P, S), V_N = (S, A, B), V_T = (a, b, c), \\ P: S \rightarrow cA, S \rightarrow bB, A \rightarrow aA, B \rightarrow bA, A \rightarrow b.$$

Можно проверить, что данная грамматика порождает обучающую выборку, использованную в процессе построения грамматики.

## 2.9. Лабораторная работа №9

### РАСПОЗНАВАНИЕ ОБРАЗОВ НА ОСНОВЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ

**Цель работы:** научиться классифицировать объекты с помощью искусственной нейронной сети (ИНС).

#### Порядок выполнения работы

1. Ознакомление с теоретической частью работы.
2. Реализация алгоритма автоматической классификации объектов с помощью ИНС в виде многослойного персептрона.



### 3. Оформление отчета по лабораторной работе.

**Исходные данные:** обучающая выборка, количество классов.

**Выходные данные:** классы и отнесенные к ним объекты.

Одной из наиболее популярных моделей ИНС с контролируемым обучением считается модель в виде многослойного персептрона (рис. 2.15).

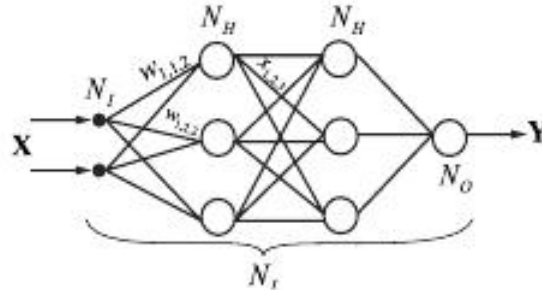


Рис. 2.15. ИНС в виде многослойного персептрона

Формальные нейроны могут объединяться в сети различным образом. Сеть состоит из произвольного количества слоев нейронов, в простейшем случае – однослойная сеть. Первый слой называется сенсорным, или входным, внутренние слои называются скрытыми, или ассоциативными, последний – выходным, или результативным. Количество нейронов в слоях может быть произвольным. Обычно во всех скрытых слоях одинаковое количество нейронов. В каждом слое выполняется нелинейное преобразование линейной комбинации сигналов предыдущего слоя. Следовательно, в тех сетях, где требуется последовательное соединение слоев нейронов друг за другом, необходима нелинейная функция активации. В противном случае многослойность оказывается ненужной, так как ее можно заменить эквивалентной однослойной сетью с соответствующими весовыми коэффициентами. Многослойная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов.

Сеть состоит из произвольного количества слоев нейронов. Нейроны каждого слоя соединяются с нейронами предыдущего и последующего слоев по принципу «каждый с каждым». Первый слой (слева) называется сенсорным или входным, внутренние слои называются скрытыми или ассоциативными, последний (самый правый, на рис. 2.15 состоит из одного нейрона) – выходным или результативным. Количество нейронов в слоях может быть произвольным. Обычно во всех скрытых слоях одинаковое количество нейронов.

Введем обозначение количества слоев и нейронов в слое. Входной слой:  $N_I$  нейронов;  $N_H$  нейронов в каждом скрытом слое;  $N_O$  выходных нейронов;  $x$  – вектор входных сигналов сети;  $y$  – вектор выходных сигналов.

Входной слой не выполняет никаких вычислений, а лишь распределяет входные сигналы, поэтому иногда его не учитывают, считая количество слоев в

сети. Обозначим через  $N_L$  полное количество слоев в сети, считая и входной. Работа многослойного персептрона (МСП) описывается формулами:

$$\begin{aligned} NET_{jl} &= \sum_i w_{ijl} x_{ijl}, \\ OUT_{jl} &= F(NE_{jl} - \theta_{jl}), \\ x_{ij(l+1)} &= OUT_{il}, \end{aligned}$$

где индексом  $i$  обозначен номер входа;  $j$  – номер нейрона в слое;  $l$  – номер слоя.

Кроме того,  $x_{ijk}$  –  $i$ -й входной сигнал  $j$ -го нейрона в слое  $l$ ;  $w_{ijk}$  – весовой коэффициент  $i$ -го входа  $j$ -го нейрона в слое  $l$ ;  $NET_{jl}$  – сигнал NET  $j$ -го нейрона в слое  $l$ ;  $OUT_{jl}$  – выходной сигнал нейрона;  $\theta_{jl}$  – пороговый уровень  $j$ -го нейрона в слое  $l$ .

Введем еще некоторые обозначения:  $w_{jl}$  – вектор-столбец весов для всех входов нейрона  $j$  в слое  $l$ ;  $W_l$  – матрица весов всех нейронов в слое  $l$ , в столбцах матрицы расположены векторы  $w_{jl}$ ;  $x_{jl}$  – входной вектор-столбец слоя  $l$ .

В каждом слое рассчитывается нелинейное преобразование от линейной комбинации сигналов предыдущего слоя. Отсюда видно, что линейная функция активации может применяться только для тех моделей сетей, где не требуется последовательное соединение слоев нейронов друг за другом. Для многослойных сетей функция активации должна быть нелинейной, иначе можно построить эквивалентную однослойную сеть и многослойность оказывается ненужной. Если применена линейная функция активации, то каждый слой будет давать на выходе линейную комбинацию входов. Следующий слой даст линейную комбинацию выходов предыдущего, а это эквивалентно одной линейной комбинации с другими коэффициентами и может быть реализовано в виде одного слоя нейронов. Многослойная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов. Как и ряды, многослойные сети оказываются универсальным инструментом аппроксимации функций.

В нейронной сети за счет поочередного расчета линейных комбинаций и нелинейных преобразований достигается аппроксимация произвольной многомерной функции при соответствующем выборе параметров сети. В многослойном персептроне нет обратных связей. Такие модели называются сетями прямого распространения.

Для построения персептрона необходимо выбрать его параметры. Чаще всего выбор пороговых значений и весовых коэффициентов происходит в процессе обучения, т. е. веса и коэффициенты пошагово изменяются в ходе обучения сети. Рассмотрим по шагам общий алгоритм решения задачи с помощью персептрона.

1. Определить смысл, вкладываемый в компоненты входного вектора  $x$ . Вектор  $x$  должен содержать формализованное условие задачи, т. е. всю информацию, необходимую для получения требуемого результата.

2. Выбрать выходной вектор  $y$  таким образом, чтобы его компоненты содержали ответ поставленной задачи.

3. Выбрать вид нелинейности в нейронах (функцию активации). При этом желательно учесть специфику задачи, так как удачный выбор сократит время обучения.

4. Выбрать число слоев и нейронов в слое.

5. Задать диапазон изменения входов, выходов, весов и пороговых уровней, учитывая множество значений выбранной функции активации.

6. Присвоить начальные значения весовым коэффициентам и пороговым уровням и дополнительным параметрам (например, крутизне функции активации, если она будет настраиваться при обучении). Начальные значения не должны быть большими, чтобы нейроны не оказались в насыщении (на горизонтальном участке функции активации), иначе обучение окажется фиктивным: пороговые значения и веса будут достигнуты, а сеть не обучится. Начальные значения не должны быть и слишком малыми, чтобы выходы большей части нейронов не были равны нулю, иначе обучение замедлится.

7. Провести обучение, т. е. подобрать параметры сети так, чтобы задача решалась наилучшим образом. По окончании обучения сеть готова решать задачи того типа, которым она обучена.

8. Подать на вход сети условия задачи в виде вектора  $x$ . Рассчитать выходной вектор  $y$ , который и предоставит искомое решение задачи.

Существенное значение на решение поставленной задачи оказывает выбор количества слоев в сети и нейронов в слоях. Нет строгого правила для выбора этих параметров сети. Однако, чем больше количество нейронов и слоев, тем шире возможности сети, тем медленнее она обучается и работает и тем более сложной может быть зависимость вход-выход. Количество нейронов и слоев связано:

- со сложностью задачи;
- с количеством данных для обучения;
- с требуемым количеством входов и выходов сети;
- с имеющимися ресурсами: памятью и быстродействием машины, на которой моделируется сеть.

Поскольку персептрон обучается с учителем, должно быть задано множество пар векторов  $\{x, d\}$ , где  $x$  — условие задачи,  $d$  — известное решение для этого условия. Количество элементов в обучающем множестве должно быть достаточным для обучения сети, чтобы под управлением алгоритма сформировать набор параметров сети, дающий нужное отображение  $X \rightarrow Y$ . При этом количество элементов в обучающей выборке не регламентируется. Выберем один из векторов  $x^s$  ( $s$  указывает на то, что  $x$  принадлежит обучающей выборке) и подадим его на вход сети. На выходе получится некоторый вектор  $y^s$ . Тогда

ошибкой сети можно считать  $E = |d - y|$  для каждой пары  $(x, d)$ . Для оценки качества обучения выбирают суммарную квадратическую ошибку или среднюю относительную ошибку.

Приведем общую схему обучения персептрона:

1. Инициализировать веса и параметры функции активации в малые ненулевые значения;
2. Подать на вход один образ и рассчитать выход;
3. Посчитать ошибку  $E^s$ , сравнив  $d^s$  и  $y^s$ .
4. Изменить веса и параметры функции активации так, чтобы ошибка  $E^s$  уменьшилась.
5. Повторить шаги 2–4 до тех пор, пока ошибка не перестанет убывать или не станет достаточно малой.

Рассмотрим пример классификации объектов с помощью ИНС персептронного типа (рис. 2.16).

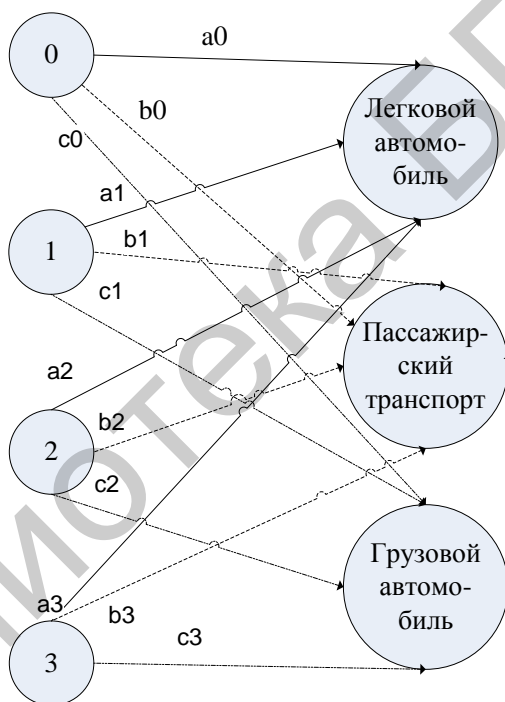


Рис. 2.16. ИНС персептронного типа

Сеть предназначена для разделения объектов на три класса: *легковой автомобиль*, *пассажирский транспорт*, *грузовой автомобиль*. Для обучения используется выборка, каждый объект которой имеет четыре признака.

В качестве функции активации выбрана кусочно-заданная функция «линейный порог», изображенная на рис. 2.17. Функция активации определяет уровень возбуждения нейрона в зависимости от суммарного уровня входного сигнала  $S$ .

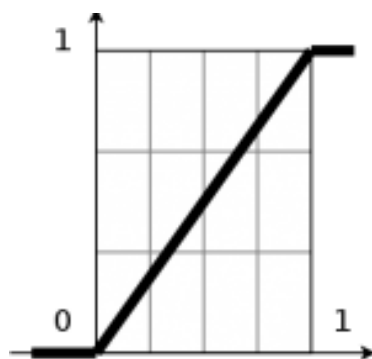


Рис. 2.17. Функция «линейный порог»

В зависимости от  $S$  функция принимает следующие значения:

$$Y = \begin{cases} 0, & S < 0; \\ S, & 0 \leq S \leq 1; \\ 1, & S > 1. \end{cases}$$

Начальные значения коэффициентов сети приведены в табл. 2.2.

Таблица 2.2

Класс	Признак			
	0	1	2	3
Легковой автомобиль (класс 1, коэффициенты $a_i$ )	– 0,3	0,1	0,4	– 0,5
Пассажирский транспорт (класс 2, коэффициенты $b_i$ )	0,1	0,3	– 0,3	0,2
Грузовой автомобиль (класс 3, коэффициенты $c_i$ )	0,4	– 0,2	0,1	0,2

Обучающая выборка представлена тремя объектами (табл. 2.3), каждый из которых принадлежит одному из трех классов.

Таблица 2.3

Объект	Признак			
	Масса	Двигатель	Пассажировместимость	Грузоподъемность
Ока (класс 1)	0,25	0,25	0,25	0
Газель пассажирская (класс 2)	0,75	0,75	0,75	0,25
КамаЗ (класс 3)	1	1	0,25	1

Рассмотрим процесс обучения ИНС с помощью обучающей выборки. Поскольку в основе ИНС лежит модель «Персептрон», для ИНС применяются те же условия и правила корректировки коэффициентов, что и для персептрона.

Результаты обучения для объекта «КамАЗ».

$$\text{класс 1: } -0,3 \cdot 1 + 0,1 \cdot 1 + 0,4 \cdot 0,25 + (-0,5) \cdot 1 = -0,6;$$

$$\text{класс 2: } 0,1 \cdot 1 + 0,3 \cdot 1 + (-0,3) \cdot 0,25 + 0,2 \cdot 1 = 0,525;$$

$$\text{класс 3: } 0,4 \cdot 1 + (-0,2) \cdot 1 + 0,1 \cdot 0,25 + 0,2 \cdot 1 = 0,65.$$

Обучающий объект принадлежит к 3-му классу, на котором достигается максимальное значение, поэтому не требуется коррекция коэффициентов сети. Все коэффициенты остаются теми же, что и на предыдущем шаге.

Результаты обучения для объекта «Ока».

$$\text{класс 1: } -0,3 \cdot 0,25 + 0,1 \cdot 0,25 + 0,4 \cdot 0,25 + 0 \cdot (-0,5) = 0,05;$$

$$\text{класс 2: } 0,1 \cdot 0,25 + 0,3 \cdot 0,25 + (-0,3) \cdot 0,25 + 0,2 \cdot 0 = 0,025;$$

$$\text{класс 3: } 0,4 \cdot 0,25 + (-0,2) \cdot 0,25 + 0,1 \cdot 0,25 + 0,2 \cdot 0 = 0,075;$$

Поскольку обучающий объект принадлежит классу 1, а максимум достигается для класса 3, эти два класса требуют коррекции коэффициентов. Она выполняется методом обратного распространения ошибки. Коэффициенты класса 2 не требуют исправления. Новые коэффициенты ИНС приведены в табл. 2.4.

$$\text{класс 1: } 1 - 0,05 = 0,95;$$

$$\text{класс 2: } 0 - 0,025 = -0,025;$$

$$\text{класс 3: } 0 - 0,075 = -0,075.$$

Таблица 2.4

Класс	Признак			
	0	1	2	3
Легковой автомобиль (1 класс)	$-0,3 + 0,95 =$ <b>0,65</b>	$0,1 + 0,95 =$ <b>1,05</b>	$0,4 + 0,95 =$ <b>1,35</b>	$-0,5 + 0,95 =$ <b>0,45</b>
Пассажирский транспорт (2 класс)	<b>0,1</b>	<b>0,3</b>	<b>-0,3</b>	<b>0,2</b>
Грузовой автомобиль (3 класс)	$0,4 - 0,075 =$ <b>0,325</b>	$-0,2 - 0,075 =$ <b>-0,275</b>	$0,1 - 0,075 =$ <b>0,025</b>	$0,2 - 0,075 =$ <b>0,125</b>

Результат обучения для объекта «Газель пассажирская»:

$$\text{класс 1: } 0,65 \cdot 0,75 + 1,05 \cdot 0,75 + 1,35 \cdot 0,75 + 0,45 \cdot 0,25 = 2,475;$$

класс 2:  $0,1 \cdot 0,75 + 0,3 \cdot 0,75 - 0,3 \cdot 0,75 + 0,2 \cdot 0,25 = 0,125$ ;  
класс 3:  $0,325 \cdot 0,75 - 0,275 \cdot 0,75 + 0,025 \cdot 0,75 + 0,125 \cdot 0,25 = 0,0875$ .

Поскольку обучающий объект принадлежит классу 2, а максимум достигается для класса 1, эти два класса требуют коррекции коэффициентов. Она выполняется методом обратного распространения ошибки. Коэффициенты класса 3 не требуют исправления. Новые коэффициенты ИНС приведены в табл. 2.5.

класс 1:  $0 - 2,475 = -2,475$ ;  
класс 2:  $1 - 0,125 = 0,875$ ;  
класс 3:  $0 - 0,0875 = -0,0875$ .

Таблица 2.5

Класс	Признак			
	0	1	2	3
Легковой ав- томобиль (1 класс)	$0,65 - 2,475 =$ <b>1,825</b>	$1,05 - 2,475 =$ <b>- 1,425</b>	$1,35 - 2,475 =$ <b>- 1,125</b>	$0,45 - 2,475 =$ <b>- 2,025</b>
Пассажирский транспорт (2 класс)	$0,1 + 0,875 =$ <b>0,975</b>	$0,3 + 0,875 =$ <b>1,175</b>	$-0,3 + 0,875 =$ <b>0,575</b>	$0,2 + 0,875 =$ <b>1,075</b>
Грузовой ав- томобиль (3 класс)	<b>0,325</b>	<b>- 0,275</b>	<b>0,025</b>	<b>0,125</b>

На следующей итерации опять выполняется классификация объектов обучающей выборки. Если не потребовалось ни одной коррекции коэффициентов сети, то считается, что сеть обучена и может использоваться для классификации тестовых объектов. В противном случае вся обучающая выборка повторно подается на вход ИНС для выполнения следующей итерации обучения.

## ЛИТЕРАТУРА

1. Фор, А. Восприятие и распознавание образов / А. Фор ; пер. с франц. – М. : Машиностроение, 1989.
2. Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес ; пер. с англ. – М. : Мир, 1978.
3. Дуца, Р. Распознавание образов и анализ сцен / Р. Дуца, П. Харт ; пер. с англ. – М. : Мир, 1979.
4. Бочкарева, Л. В. Методы и алгоритмы принятия решений : учеб.-метод. пособие для студ. спец. ПОИТ / Л. В. Бочкарева, М. В. Кирейцев. – Минск : БГУИР, 2006.
5. Грешилов, А. А. Математические методы принятия решений : учеб. пособие для студ. вузов, обучающихся по машиностроительным специальностям / А. А. Грешилов. – М. : МГТУ им. Н. Э. Баумана, 2006.
6. Горелик, А. Л. Методы распознавания / А. Л. Горелик. – М. : Высш. шк., 1986.
7. Аркадьев, А. Г. Обучение машины распознаванию образов / А. Г. Аркадьев, Э. М. Браверман. – М. : Наука, 1964.
8. Фор, Р. Современная математика / Р. Фор, А. Кафман, М. Дени-Папен ; пер. с франц. – М. : Мир, 1995.
9. Шестаков, К. М. Теория принятия решений и распознавание образов: курс лекций / К. М. Шестаков. – Минск : БГУ, 2005.
10. Шестаков, К. М. Математические основы теории принятия решений: курс лекций / К. М. Шестаков. – Минск : Акад. упр. при Президенте Республики Беларусь, 2004.



*Учебное издание*

**Серебряная Лия Валентиновна  
Третьяков Федор Игоревич**

## ***МЕТОДЫ И АЛГОРИТМЫ ПРИНЯТИЯ РЕШЕНИЙ***

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ**

Редактор *Е. И. Герман*  
Корректор *Е. Н. Батурчик*  
Компьютерная правка, оригинал-макет *А. А. Лущикова*

Подписано в печать 01.11.2016. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».  
Отпечатано на ризографе. Усл. печ. л. 3,95. Уч.-изд. л. 4,0. Тираж 150 экз. Заказ 21.

Издатель и полиграфическое исполнение: учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники».  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий №1/238 от 24.03.2014,  
№2/113 от 07.04.2014, №3/615 от 07.04.2014.  
ЛП №02330/264 от 14.04.2014.  
220013, Минск, П. Бровки, 6