

РАСПОЗНАВАНИЕ ОБРАЗОВ

Содержание

1	ВВЕДЕНИЕ	2
2	ТИПЫ ЗАДАЧ В РАСПОЗНАВАНИИ	2
2.1	Задача распознавания.....	4
2.1.1	Математическая постановка	4
2.1.2	Способы определения классов объектов.....	5
2.1.3	Обучение с учителем, обучающая выборка	7
2.1.4	Системы, решающие задачу распознавания	9
2.2	Классификация	9
2.2.1	Математическая постановка задачи классификации.....	9
2.2.2	Способы определения критериев различия	9
2.2.3	Обучение без учителя.....	10
2.2.4	Пример системы классификации	11
3	ВИДЫ ПРАВИЛ ПРИНЯТИЯ РЕШЕНИЙ.....	12
3.1	Параллельное распознавание	12
3.2	Последовательное распознавание.....	13
3.2.1	Формальное определение последовательной процедуры.....	14
3.2.2	Байесовское правило для последовательных процедур.....	14
3.2.3	Сравнение с параллельными процедурами.....	16
4	ОСНОВНЫЕ ПОДХОДЫ К РАСПОЗНАВАНИЮ	17
4.1	Евклидово пространство	17
4.1.1	Решающие (дискриминантные) функции.....	17
4.1.2	Функции расстояния.....	21
4.1.3	Функции правдоподобия.....	24
4.2	Списки признаков.....	28
4.2.1	Формализация подхода, использующего факторный анализ	28
4.2.2	Формализация подхода группирования прямых измерений	29
4.2.3	Методы выделения признаков.....	30
4.3	Структурное описание	34
4.3.1	Грамматические методы распознавания.....	39
5	НЕЙРОСЕТЕВЫЕ МЕТОДЫ	46
5.1	Обзор	46
5.2	Персептрон.....	47
5.3	Однослойные и многослойные нейронные сети	52
5.4	Сети с обратным распространением ошибки	55
5.5	Сети Кохонена и обучение без учителя.....	57
5.6	Сети Хопфилда	59
6	Список литературы	61

1 ВВЕДЕНИЕ

Распознавание — это способность живых организмов обнаруживать в потоке информации, поступающей от органов чувств, определённые объекты, закономерности, явления. Оно может осуществляться на основе зрительной, слуховой, тактильной информации. Так, человек без труда может узнать другого знакомого ему человека, взглянув на него или услышав его голос. Некоторые животные активно используют обоняние для узнавания других особей и поиска пищи.

Возможность распознавания опирается на схожесть однотипных объектов. Несмотря на то, что все предметы и ситуации уникальны в строгом смысле, между некоторыми из них всегда можно найти сходства по тому или иному признаку. Отсюда возникает понятие классификации — разбиения всего множества объектов на непересекающиеся подмножества - классы, элементы которых имеют некоторые схожие свойства, отличающие их от элементов других классов. И, таким образом, задачей распознавания является отнесение рассматриваемых объектов или явлений по их описанию к нужным классам. Т.е. понятие распознавания можно расширить, если говорить об обнаружении объектов в потоке не только чувственной, но и любой другой информации. Например, можно говорить о распознавании болезни по её симптомам у больного или о распознавании социальных явлении по статистической информации.

Развитие и распространение компьютерной обработки информации привели к возникновению в середине XX века потребностей в технологиях, позволяющих машинам осуществлять распознавание в обрабатываемой ими информации. Примерами могут служить распознавание текста, машинное зрение, распознавание речи, отпечатков пальцев. Несмотря на то, что некоторые из этих задач решаются человеком на подсознательном уровне с большой скоростью, до настоящего времени ещё не создано компьютерных программ, решающих их в столь же общем виде. Существующие системы предназначены для работы лишь в специальных случаях со строго ограниченной областью применения.

В данном разделе настоящего пособия приводится обзор существующих методов распознавания образов. За исключением некоторых примеров и специальных уточнений, речь ведётся о распознавании образов в общем, а не о распознавании в конкретных предметных областях. Рассматриваются общие принципы построения и обучения систем распознавания, описываются наиболее распространённые подходы и математические методы, применяемые в таких системах.

2 ТИПЫ ЗАДАЧ В РАСПОЗНАВАНИИ

Системы распознавания имеют следующую типичную функциональную схему (рис. 1). *Входные данные*, подлежащие распознаванию, подаются на вход системы и подвергаются *предобработке* с целью их преобразования в необходимый для следующего этапа вид или для выделения из них необходимых характерных признаков. Далее на этапе *принятия решения* над обработанным массивом данных производится ряд вычислений и на основе их результатов формируется ответ, содержащий ожидаемые от системы сведения о входных данных. Содержание *входных* и *выходных данных* определяется назначением системы. Например, если входом служит описание симптомов болезни, то в качестве выхода система может выдавать название болезни; система распознавания текста может принимать на вход

растровое изображение страницы текста в установленном формате и преобразовывать его в кодовую последовательность составляющих данный текст символов.

Кроме описанных этапов функционирования системы распознавания предусматривают свою настройку на множество возможных входных данных; этот этап называют этапом *обучения* системы. Целью обучения системы является формирование в её памяти набора сведений, необходимых для распознавания предполагаемого класса входных данных. В зависимости от специфики решаемой задачи обучение может быть выражено процедурой однократного ручного задания параметров работы системы её разработчиком, автоматической процедурой определения оптимальных значений параметров в результате проведения учебных циклов распознавания или процессом непрерывной подстройки параметров в результате анализа вырабатываемых системой ответов. Как правило, имеет место комбинация названных подходов.

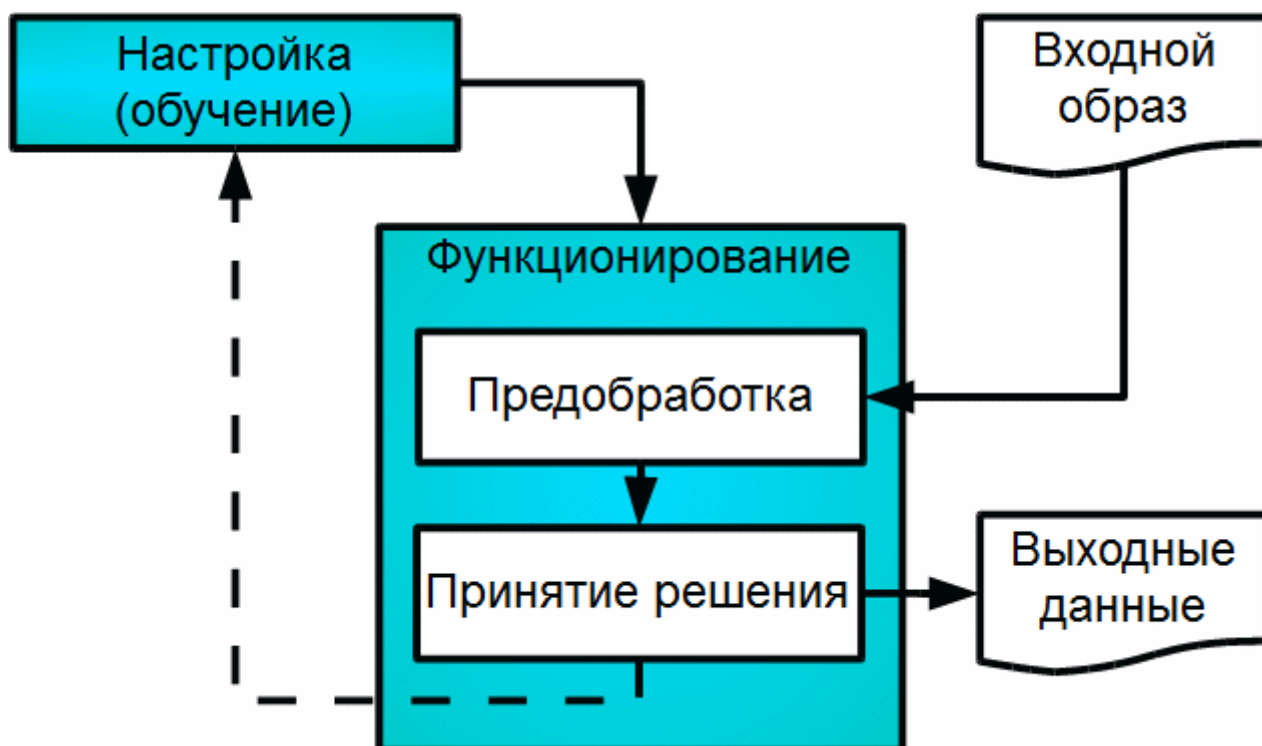


Рисунок 1: Типичная функциональная схема системы распознавания.

На этапе предобработки решается задача *создания формализованного описания объектов распознавания*, пригодного для использования алгоритмами собственно распознавания. Как правило, исходные данные о наблюдаемых объектах представлены в форме, непригодной непосредственно для распознавания. Это могут быть растровые изображения, звуковые файлы, статистические данные (числовые наборы), видео-записи и пр — т.н. «код 0». Некоторые алгоритмы распознавания требуют более высокоуровневого представления. Это приводит к необходимости произвести одно или более преобразований исходных данных, переходя от кода 0 к коду 1, 2 и т.д. В качестве примера можно рассмотреть процедуру сегментации изображения, т.е. выделения на нём одноцветных областей.

Этап принятия решения является наиболее значимым в цикле работы системы распознавания с точки зрения её характеристики в целом. Т.е. задача, решаемая на данном этапе, во многом определяет назначение системы. Кроме того, для обеспечения возможности системы выполнять качественное принятие решений выдвигается ряд требований к этапу обучения. Наконец, как уже отмечалось выше, алгоритмы этапа принятия решений требуют

необходимой предобработки входных данных.

Рассмотрим два основных класса задач, решаемых на этапе принятия решений и характеризующих назначение систем распознавания в целом [6].

1. *Распознавание* — отнесение предъявляемых объектов к определённым классам с помощью применения известных правил классификации. Это наиболее типичная задача систем распознавания. Перед тем, как система сможет выполнять данную функцию, предполагается её обучение на множестве примеров — *обучающей выборке* объектов распознавания. При этом применяется процедура т.н. *обучения с учителем*.
2. *Классификация (таксономия)* — разбиение множества объектов на непересекающиеся классы по их формализованным описаниям. Данная задача решается в тех случаях, когда от системы не требуется отнесения входных образов к каким-либо определённым классам, а требуется лишь способность различать их каким-либо способом по определённым признакам. Можно говорить об оперировании безымянными классами объектов. Для обеспечения решения задачи классификации часто используется процедура *обучение без учителя*.

Далее задачи распознавания и классификации будут рассмотрены подробно.

2.1 Задача распознавания

2.1.1 Математическая постановка

Дано множество M объектов ω . Объекты задаются значениями некоторых признаков $x_i, i=1, \dots, N$, наборы которых одинаковы для всех объектов. Совокупность признаков объекта ω определяет некоторым образом его описание $I(\omega) = (x_1(\omega), x_2(\omega), \dots, x_N(\omega))$. Признаки могут выражаться в терминах да/нет, да/нет/неизвестно, числовыми значениями, значениями из набора возможных вариантов и т.д.

На всём множестве M существует разбиение на подмножества (*классы объектов*):

$$M = \bigcup_{i=0}^m \Omega_i$$

Разбиение на *классы* может быть задано полностью (т.е. полной совокупностью информации, однозначно описывающей каждый класс) или определяться некоторой априорной информацией I_0 о классах Ω_i - например, характеристическим описанием входящих в них объектов.

Задача распознавания состоит в том, чтобы для каждого данного объекта ω по его описанию $I(\omega)$ и априорной (обучающей) информации I_0 вычислить значения предикатов

$$P_i = (\omega \in \Omega_i), \quad i = 1, \dots, m,$$

каждый из которых принимает истинное значение в том и только в том случае, когда объект ω принадлежит классу с номером i .

Для описания невозможности распознавания объектов предикаты P_i заменяются величинами $\alpha_i \in \{0 (\omega \notin \Omega_i), 1 (\omega \in \Omega_i), \Delta (\text{неизвестно})\}$.

Таким образом, для рассматриваемого объекта ω необходимо вычислить его информационный вектор

$$\alpha(\omega) = (\alpha_1(\omega), \dots, \alpha_m(\omega)),$$

описывающий принадлежность данного объекта каждому из известных классов предметной области.

Процедура, строящая информационный вектор $\alpha(\omega)$ в данном случае выражает алгоритм принятия решения об отнесении объекта ω к тому или иному классу и называется «решающей функцией».

2.1.2 Способы определения классов объектов

Разбиение рассматриваемого множества объектов на классы Ω_i может быть задано следующими способами:

1. *Перечисление.* Каждый класс задаётся путём прямого указания его членов. Такой подход используется в том случае, если доступна полная априорная информация о всех возможных объектах распознавания. Предъявляемые системе образы сравниваются с заданными описаниями представителей классов и относятся к тому классу, которому принадлежат наиболее сходные с ними образцы. Такой подход называют методом *сравнения с эталоном*. Его недостатком является слабая устойчивость к шумам и искажениям в распознаваемых образах.

Пример. Распознавание машинопечатного шрифта. Все символы имеют чётко заданное шрифтом начертание. Следовательно, необходимо обучить систему путём прямого указания изображений всех распознаваемых символов (т.е. путём задания эталонов):

А Б В а б в ... 1 2 3

Необходимо отметить, что если предполагается распознавание курсивного, полужирного или иного начертания символов шрифта, то при таком подходе будет необходимо представить каждый вариант начертания каждого символа. Это связано с характером процесса распознавания: каждый распознаваемый объект попиксельно сравнивается поочерёдно со всеми известными системе эталонами. Кроме того, способность распознавания линейных трансформаций данных эталонов требует определённых усилий на этапе предобработки.

2. *Задание общих свойств.* Класс задаётся указанием некоторых признаков, присущих всем его членам. Распознаваемый объект в таком случае не сравнивается напрямую с группой эталонных объектов. В его первичном описании выделяются значения определённого набора признаков, которые затем сравниваются с заданными признаками

классов. При этом для каждого признака может задаваться требование либо к его наличию/отсутствию, либо к нахождению его числового значения в установленных пределах. Такой подход называется *сопоставлением по признакам*. Он экономичнее метода сравнения с эталоном в вопросе количества памяти, необходимой для хранения описаний классов. Кроме того, он допускает некоторую вариативность распознаваемых образов. Однако, главной сложностью является определение полного набора признаков, точно отличающих членов одного класса от членов всех остальных.

Пример. Распознавание цифр почтовых индексов [17]. Рассматривается следующий набор распознаваемых символов (рис. 2):

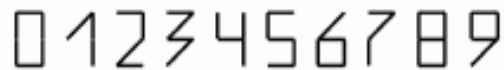


Рисунок 2: Цифры почтовых индексов

Каждый из показанных символов представляет класс распознаваемых объектов — одну из цифр. Все эти изображения построены по одному принципу — с помощью комбинирования вертикальных, горизонтальных и диагональных сегментов в определённых позициях знакомест. Для описания классов предлагаются следующие признаки:

x_1 — количество вертикальных линий минимального размера;

x_2 — количество горизонтальных линий;

x_3 — количество наклонных линий;

x_4 — количество горизонтальных линий снизу объекта.

С помощью этих признаков можно следующим образом задать классы цифр:

	x_1	x_2	x_3	x_4
0	4	2	0	1
1	2	0	1	0
2	1	2	1	1
3	0	2	2	0
4	3	1	0	0
5	2	3	0	1
6	2	2	1	1
7	1	1	1	0
8	4	3	0	1
9	2	2	1	0

Заметим, что набор выбранных признаков не является единственно возможным. Качество распознавания во многом зависит от того, насколько удачно разработчиком системы выбран набор признаков [8].

3. *Кластеризация.* В случае, когда объекты описываются векторами признаков или измерений, класс можно рассматривать как кластер. Распознавание осуществляется на

основе расчёта расстояния описания объекта до каждого из имеющихся кластеров. Если кластеры достаточно разнесены в пространстве, при распознавании хорошо работает метод оценки расстояний от рассматриваемого объекта до каждого из кластеров. Сложность распознавания возрастает, если кластеры перекрываются. Обычно это является следствием недостаточности исходной информации и может быть разрешено увеличением количества измерений объектов. Для задания исходных кластеров целесообразно использовать процедуру обучения.

Пример. Категоризация документов — задача, заключающаяся в отнесении документа к одной из нескольких (заранее известных) категорий на основании содержания документа. Класс объектов распознавания в данном случае — некоторая категория документов, обладающих определёнными содержательными характеристиками, т.е., к примеру, имеющих схожую тематику, содержащих сходные списки цитат и пр. При этом для характеристикой содержания каждого документа является определённая (многомерная) метрика. Такой метрикой может служить, например, вектор частот некоторого набора ключевых слов [9]. Метрика документа соответствует точке в многомерном пространстве описаний документов.

На этапе обучения в систему вводятся наборы эталонных документов, имеющих заведомо известные содержательные характеристики. Сходные документы образуют группы близко расположенных точек в пространстве описаний, которые группируются в кластеры. На этапе распознавания очередной документ представляется точкой в пространстве описаний по тем же правилам, что были использованы в процессе формирования кластеров. Для отнесения этого нового документа к одной из категорий необходимо вычислить отношение данной точки по выбранной метрике к каждому из имеющихся кластеров.

2.1.3 Обучение с учителем, обучающая выборка

Альтернативой ручному вводу в систему описаний классов объектов и параметров решающей функции является обучение системы. Оно представляет собой процедуру самонастройки системы распознавания на основе воспринимаемой информации и может происходить как при подготовке системы, так и в процессе её работы по мере «накопления опыта». Вид обучения, в котором системе представляется набор образцов распознаваемых объектов с указанием их принадлежности классам, называется *обучением с учителем*. Набор образцовых объектов называется *обучающей выборкой*. Необученная система производит распознавание предлагаемых объектов и сравнивает свои результаты с правильными ответами, поступающими от учителя. По результатам сравнения система корректирует параметры решающей функции (см. рис. 3). В случае использования метода перечисления для задания классов система просто сохраняет в памяти всю доступную об объектах информацию. Если классы задаются описанием общих свойств, то система определяет различия в значениях признаков объектов разных классов. При использовании кластерного подхода по анализу векторов признаков объектов обучающей выборки формируются исходные кластеры в пространстве признаков.

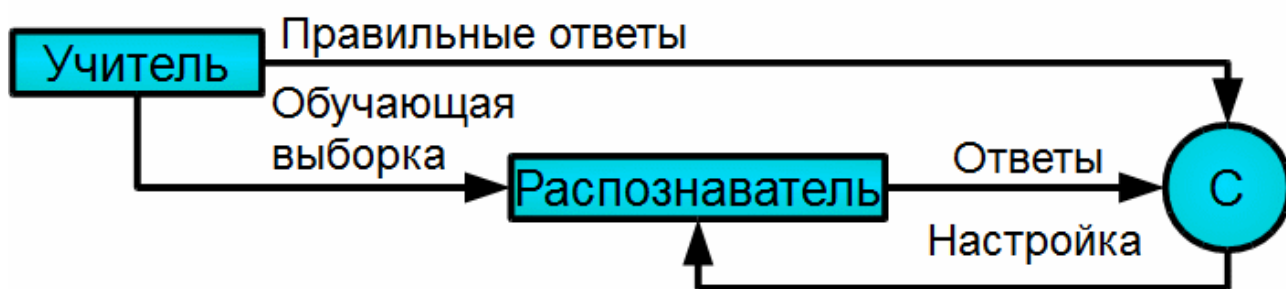


Рисунок 3: Схема обучения с учителем

Итак, обучающая выборка в задаче распознавания является априорной информацией о множестве распознаваемых объектов и представляет описание всех классов объектов:

$$T = I_0 (\Omega_1, \dots, \Omega_m).$$

Её составляют описания предложенных учителем объектов с указанием их принадлежности классам, т.е. можно определить её как совокупность описаний объектов

$$(I(\omega_1), \dots, I(\omega_{r_1}), I(\omega_{r_1+1}), \dots, I(\omega_{r_2}), \dots, I(\omega_{r_m})),$$

где объекты $(\omega_1, \dots, \omega_{r_1})$ принадлежат классу Ω_1 , объекты $(\omega_{r_1+1}, \dots, \omega_{r_2})$ - классу Ω_2 и т.д.

Таким образом, обучающая выборка представляет собой таблицу, строки которой помечены названиями объектов $\omega_i, i = 1, \dots, r_m$, а столбцы — названиями признаков $x_j, j = 1, \dots, N$. Элементами таблицы являются значения признаков объектов $x_j(\omega_i)$. Строки таблицы сгруппированы по классам $\Omega_j, j = 1, \dots, m$.

	x_1	x_2	...	x_N	Классы
ω_1	$x_1(\omega_1)$	$x_2(\omega_1)$...	$x_N(\omega_1)$	Ω_1
ω_2	$x_1(\omega_2)$	$x_2(\omega_2)$...	$x_N(\omega_2)$	
...	
ω_{r_1}	$x_1(\omega_{r_1})$	$x_2(\omega_{r_1})$...	$x_N(\omega_{r_1})$...
...
$\omega_{r_{m-1}+1}$	$x_1(\omega_{r_{m-1}+1})$	$x_2(\omega_{r_{m-1}+1})$...	$x_N(\omega_{r_{m-1}+1})$	Ω_m
...	
ω_{r_m}	$x_1(\omega_{r_m})$	$x_2(\omega_{r_m})$...	$x_N(\omega_{r_m})$	

2.1.4 Системы, решающие задачу распознавания

Задача распознавания на сегодняшний день решается целым рядом прикладных

приложений из широкого круга предметных областей. Ниже приведён неполный список областей применения методов распознавания:

- OCR (optical character recognition — распознавание символов);
- распознавание отпечатков пальцев;
- машинное зрение робототехнических систем;
- распознавание речи;
- биомедицинские приложения: анализ и автоматическая обработка рентгенограмм, электрокардиограмм, исследования хромосом, диагностика заболеваний, интерпретация опросников;
- контроль и мониторинг технических систем;
- анализ текстовых данных: фильтрация спама, автоматическое реферирование.

2.2 Классификация

2.2.1 Математическая постановка задачи классификации

Дано множество M объектов ω , имеющих свои описания $I(\omega) = (x_1(\omega), x_2(\omega), \dots, x_N(\omega))$. Задан критерий $K(I(\omega))$, позволяющий отличать объекты друг от друга в соответствии с определённым условием. Например, такой критерий может выдавать целое число, определяющее положение объекта в некоторой ранговой системе.

Требуется в соответствии с заданным критерием K построить разбиение множества M на классы $\Omega_i: M = \bigcup_{i=0}^m \Omega_i$. Действие критерия в этом случае определяется следующим выражением:

$$\forall i, j (i, j = 1, \dots, m) \quad i \neq j \Leftrightarrow K(I(\omega \in \Omega_i)) \neq K(I(\omega \in \Omega_j))$$

Следует отметить, что число классов m изначально неизвестно и определяется по результатам выполнения классификации.

2.2.2 Способы определения критериев различия

В задачах классификации для задания правил различия объектов могут быть использованы те же методы, что и при распознавании.

1. *Прямым перечислением* можно задать примеры заведомо различающихся объектов. Правило классификации формируется на основе анализа сходств и различий эталонных объектов в условиях априорной информации об их принадлежности различным классам. На основе выработанного правила принимаются решения об отнесении поступающих объектов к новому или одному из имеющихся классов.
2. *Описание общих свойств* позволяет сформировать правило классификации, принимая во внимание необходимость различия объектов по указанным характеристикам.

3. *Кластеризация* позволяет строить критерии различия объектов на основе кластерного анализа — многомерной статистической процедуры, упорядочивающей объекты в сравнительно однородные группы на основе анализа характеристик выборки объектов.

Пример. Кластеризация документов. Как и в задаче категоризации документов (см. раздел 2.1.2), имеется метрика, характеризующая каждый конкретный документ. Однако, на этапе обучения не указываются эталонные документы и не формируются исходные кластеры. Более того, не известно даже количество кластеров. На вход классификатора подаётся массив документов. Выходом классификатора будет набор кластеров, каждый из которых имеет некоторую метрическую характеристику, физически означающую определённые семантические свойства данной группы документов. Таким образом можно оценить, к примеру, какие темы затронуты в данном массиве документов.

2.2.3 Обучение без учителя

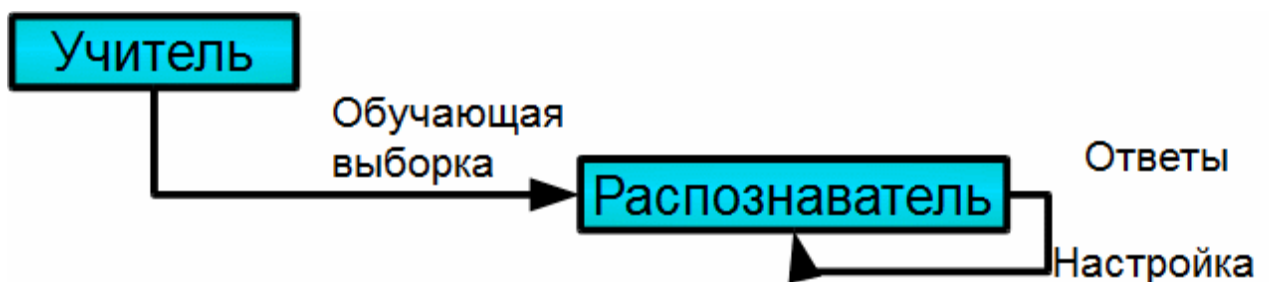


Рисунок 4: Схема обучения без учителя

Для обучения системы классификации, как и системы распознавания, можно воспользоваться обучающей выборкой. На этот раз от системы не требуется запоминания конкретных классов объектов и последующего отнесения к ним поступающих образов. Достаточно лишь определения, относятся ли представленные объекты к одному или к различным классам. Поэтому при обучении системе должна быть представлена выборка объектов с указанием, какие из них различны, а какие — сходны. Другими словами, системе нужно показать примеры отличия образов. Не важно, какие значения будет вырабатывать решающая функция, важно, чтобы эти значения были разными для разнотипных объектов и одинаковыми для однотипных. Такое обучение, в котором отсутствуют подсказки правильных ответов учителем, называется *обучением без учителя* (рис. 4). Во время обучения без учителя система строит кластеры объектов обучающей выборки, а на этапе функционирования производит определение близости входных образов к построенным кластерам.

Назначение классифицирующих систем:

- кластеризация множества объектов;
- понижение размерности описаний для упрощения дальнейшего анализа.

2.2.4 Пример системы классификации

В работе [10] описана система сегментации изображения с помощью кластеризации.

Сегментация изображения — процесс выделения на нём областей, однородных с точки зрения заданных графических свойств. Основная идея метода заключается в представлении каждой точки изображения в виде тройки (x, y, z) , где x, y — координаты точки на изображении, а z — световая интенсивность точки (рассматриваются изображения в градациях серого). Отображение множества точек исходного изображения в заданное указанным образом трёхмерное пространство подвергается кластеризации. Каждый из кластеров представляет совокупность точек на исходном изображении, обладающих свойством близости по цветовому и пространственному признаку, тем самым описывая сегмент близких по цвету точек. На рисунке 5 показана иллюстрация работы данной системы.

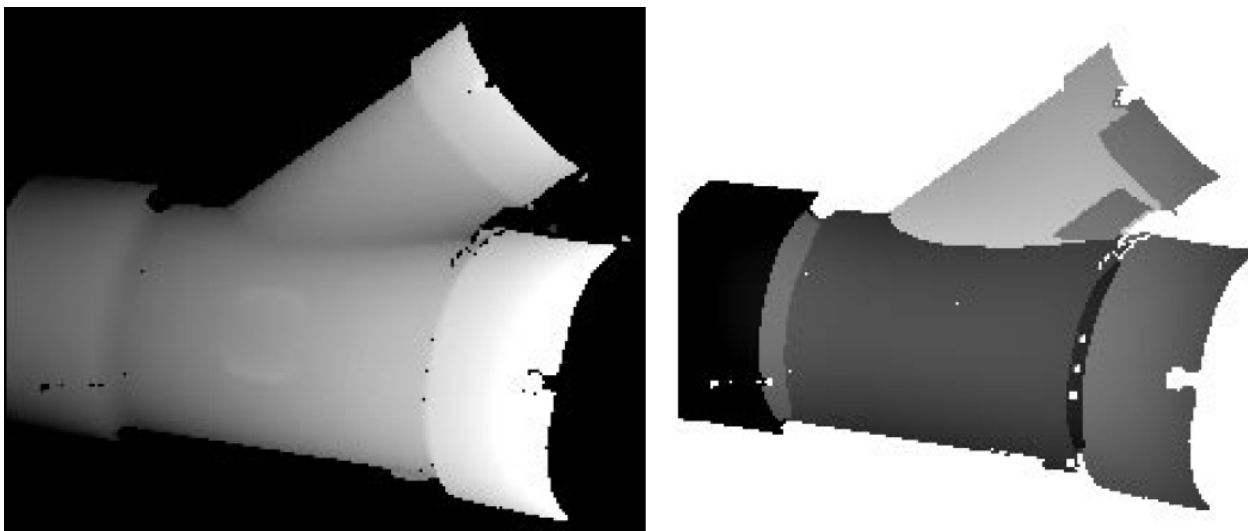


Рисунок 5: Сегментация изображения с помощью кластеризации (слева — исходное изображение; справа — сегментированное изображение).

Таким образом, действие описанной системы сегментации заключается фактически в классификации однородных областей изображения. Ведь изначально нет указания ни конкретных свойств искомых классов, ни их количества. Указана лишь метрика, позволяющая оценить однородность точек, составляющих один сегмент.

3 ВИДЫ ПРАВИЛ ПРИНЯТИЯ РЕШЕНИЙ

- *параллельные* - проведение ряда тестов над всей совокупностью выявленных данных об объекте и принятие решения на основе их результатов;
- *последовательные* - проведение последовательности тестов над подмножествами выявленных данных; выбор очередного теста определяется результатами предыдущих тестов.

3.1 Параллельное распознавание

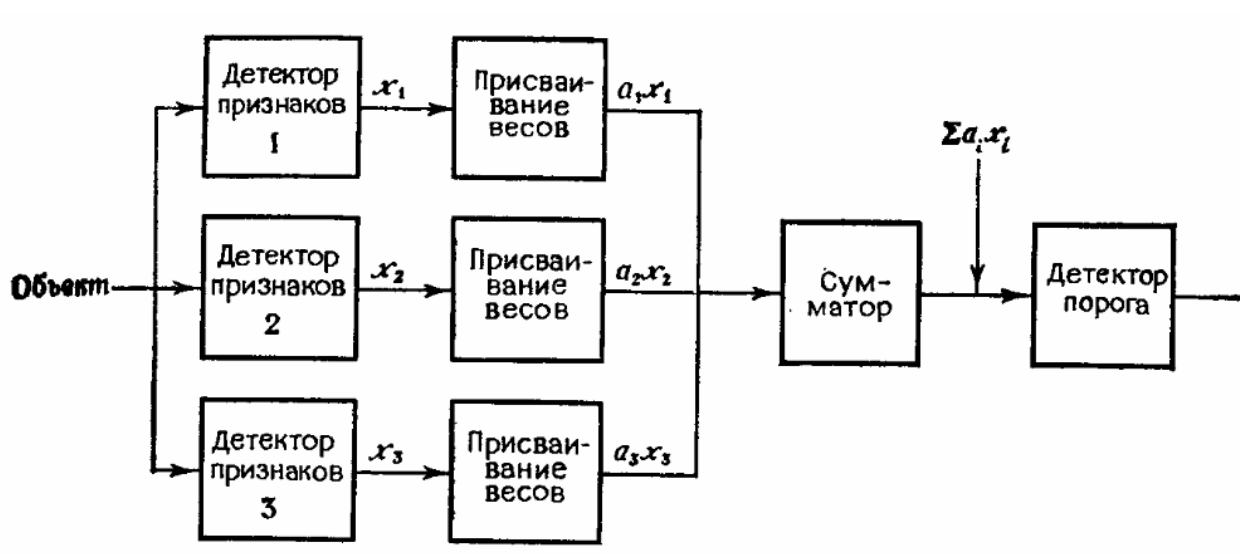


Рисунок 6: Схема действия параллельной процедуры распознавания

Систему с параллельным способом принятия решения можно проиллюстрировать следующим образом (рис. 6):

Для выполнения распознавания система производит ряд тестов над всеми компонентами $x_i, i = 1, \dots, N$ описания входного объекта одновременно. Решающая функция в этом случае представляется функцией не более, чем N переменных $g(x_1, \dots, x_N)$.

Система может быть организована в виде множества параллельных функций $F = \{f_j(x_1, \dots, x_N)\}, j = 1, \dots, m$, каждая из которых производит оценку принадлежности объекта к соответствующему ей классу. В таком случае решающая функция принимает решение на основе максимального полученного значения – $g = \max_j(f_j)$.

Параллельная процедура является достаточно надёжной и требует для распознавания постоянного времени, равного времени выполнения самой продолжительной из процедур распознавания. Однако, она не обладает гибкостью, свойственной последовательным процедурам.

Главным недостатком параллельного подхода является требование к подаче полного описания распознаваемого образа на вход системы. В случаях, когда полный образ состоит из нескольких подобразов, и требуется распознать каждый из подобразов в отдельности, для использования параллельного правила распознавания потребуется предварительное выделение каждого подобраза из общей картины. Такая ситуация характерна, например, для

распознавания изображения текста системой, обученной к распознаванию отдельных символов. В некоторых случаях предварительное выделение подобразов является выполнимой задачей, как, например, выделение символов в машинопечатном тексте. В других случаях выделение подобраза по сложности равносильно его распознаванию. Это характерно для образов, в которых подобразы не имеют чётких и формально определимых границ. Как пример — рукописный (скорописный) текст.

Параллельная процедура распознавания, как правило, реализуется методами, использующими Евклидово пространство описаний (см. раздел 0), а так же некоторыми признаковыми методами (раздел 0). Структурные же методы (раздел 0) чаще всего реализуют последовательные правила, речь о которых пойдёт в следующем разделе.

3.2 Последовательное распознавание

Последовательный способ распознавания, как и параллельный, предусматривает ряд тестов над признаками распознаваемого объекта (рис. 7). Однако, выполняются они не одновременно, а последовательно, причём порядок их выполнения может зависеть (и чаще всего зависит) от получаемых результатов.

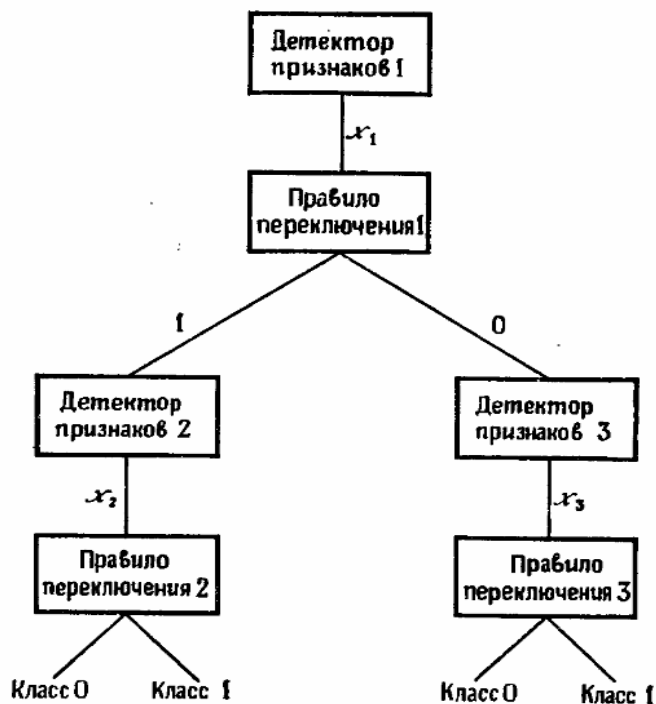


Рисунок 7: Схема действия последовательной процедуры распознавания

На каждом шаге распознавания выполняется очередной тест одним из детектирующих блоков. По результатам его работы управление может перейти к одному из нескольких возможных последующих блоков. Кроме того, выполнение очередного теста может быть параметризовано некоторым образом, зависящим от результатов предыдущих тестов. Таким образом, от характеристик образа, выявленных на ранних этапах анализа, зависит набор проверок, выполняемых на более поздних этапах. Можно говорить, что последовательная

процедура распознавания определяется деревом проверок и решений, по одной из ветви которой в каждый момент движется процесс распознавания.

Пример.

Рассмотрим, к примеру, систему считывания двумерного штрих-кода [15]. Устройство-считыватель осуществляет сканирование изображения штрих-кода. Прежде всего оно осуществляет поиск опорных элементов — ориентиров. Разные типы штрих-кодов используют различные ориентировочные элементы. В зависимости от типа найденного ориентира выбирается дальнейший алгоритм анализа изображения. В соответствии с выбранным алгоритмом и по полученным параметрам ориентиров осуществляется определение границ штрих-кода, угла его поворота относительно устройства сканирования, а также собственно расшифровка закодированных данных.

3.2.1 Формальное определение последовательной процедуры

Пусть рассматриваются образы, описание каждого из которых может быть представлено вектором $\mathbf{x} = (x_1, x_2, \dots, x_m)$, где x_1, x_2, \dots, x_m — целые числа, причём i -я компонента — значений i -го измерения в диапазоне от 1 до v_i . Множество всех возможных описаний составляет пространство описаний D . Существует конечное множество классов $\{C_j\}$, $j = 1, \dots, c$, таких, что каждый объект принадлежит точно одному из них. Класс задаёт распределение вероятностей $P(\mathbf{x} | C_j)$ на множестве D . При этом предполагается, что все классы различны, т.е. для каждой пары классов C_i и C_j найдётся хотя бы один образ $x \in D$, для которого будет выполнено

$$P(\mathbf{x} | C_i) \neq P(\mathbf{x} | C_j).$$

До начала процедуры последовательного распознавания система не знает ничего о входном образе. По мере прохождения дерева решений на каждом шаге появляется знание о значении очередной компоненты описания образа. На каждом шагу можно фиксировать получаемые факты о рассматриваемом образе в виде упорядоченной последовательности. Такую *последовательность описаний* длины v ($v < m$) определим как упорядоченное множество $S_v(y)$ пар целых чисел. i -я пара (a, b) в последовательности свидетельствует о том, что на i -м шаге анализа значение a -й компоненты описания образа было определено как b . При этом любая последовательность описаний не содержит пар с одинаковыми a -компонентами, т.е. каждое измерение образа выполняется только один раз. Можно утверждать, что каждый узел дерева решений будет иметь единственную соответствующую ему последовательность описаний образа как единственный путь к этому узлу из корня дерева.

Правилом последовательного распознавания \mathbf{R} назовём множество пар $\{(S_v(y), C_j)\}$, каждая из которых ставит в соответствие некоторой последовательности описания образа один из возможных классов. Каждая последовательность из \mathbf{R} описывает один путь из корня дерева решений к одному из листовых узлов. Заметим, что все последовательности в \mathbf{R} должны начинаться с пар, имеющих одинаковые первые элементы.

Говорят, что образ удовлетворяет последовательности, если значения измерений, описанных в последовательности, совпадают с соответствующими измерениями образа. Если объект может быть классифицирован правилом \mathbf{R} , то он удовлетворяет одной из последовательностей этого правила и должен быть отнесён к классу, приписанному данной последовательности в правиле. Это эквивалентно проходу по дереву решений по одной из ветвей.

3.2.2 Байесовское правило для последовательных процедур

При построении последовательной процедуры распознавания практически полезными оказываются Байесовские процедуры принятия решений. Для описания такой процедуры введём следующие понятия.

Стоимость ошибочного распознавания w_{ij} определяется как вещественное число, представляющее стоимость распознавания образа как элемента класса C_i , в то время как на самом деле он принадлежит классу C_j .

Стоимость измерения f_i — вещественное число, определяющее затраты на измерение i -го признака образа.

$P(C_j)$ — вероятность того, что случайным образом выбранный объект будет принадлежать классу C_j .

$P(S_v(y))$ — вероятность того, что описание случайно выбранного объекта удовлетворит последовательности длины v .

$P(S_v(y) | C_j)$ — вероятность того, что объект, случайно выбранный из класса C_j , удовлетворит последовательности $S_v(y)$.

$P(C_j | S_v(y))$ — вероятность того, что случайный объект, удовлетворяющий последовательности $S_v(y)$, будет принадлежать классу C_j .

Байесовское правило позволяет в процессе последовательного распознавания давать ответы на вопросы следующего вида: Если на определённом шаге известно, что частичное описание образа есть $S_v(y)$, к какому классу с большей вероятностью он относится? Выведем такое правило.

Совместная вероятность того, что образ удовлетворит частичному описанию $S_v(y)$, а также того, что образ принадлежит классу C_j , равна

$$P(S_v(y) \bullet C_j) = P(C_j) P(S_v(y) | C_j) \quad (5.1)$$

Эту же вероятность можно раскрыть как

$$P(S_v(y) \bullet C_j) = P(S_v(y)) P(C_j | S_v(y)) \quad (5.2)$$

Из второй записи следует выражение для искомой вероятности:

$$P(C_j | S_v(y)) = \frac{P(S_v(y) \bullet C_j)}{P(S_v(y))} \quad (5.3)$$

Знаменатель в правой части есть сумма всех возможностей появления рассматриваемой последовательности, т.е. сумма вероятностей появления последовательности в каждом классе, взвешенная вероятностями появления каждого класса:

$$P(S_v(y)) = \sum_{i=1}^c P(S_v(y) | C_i) P(C_i) \quad (5.4)$$

Подставляя в (5.3) в качестве числителя выражение (5.1), а в качестве знаменателя — (5.4), получим формулу Байеса для последовательной процедуры распознавания:

$$P(C_j | S_v(y)) = \frac{P(S_v(y) | C_j) P(C_j)}{\sum_{i=1}^c P(S_v(y) | C_i) P(C_i)}$$

Ожидаемые потери, связанные с решением отнести объект, удовлетворяющий последовательности $S_v(y)$, к классу C_j , равны

$$E(j | S_v(y)) = \sum_{i=1}^c w_{ij} P(C_i | S_v(y)).$$

Оптимальное правило распознавания строится таким образом, что любое принятие решения об отнесении образа к определённому классу минимизировало данную функцию потерь.

С точки зрения построения последовательное правило распознавания может носить эвристический характер, т.е. конструироваться вручную на основе знаний о предметной области. Существует, однако, и формальные способы построения правила на основе процедуры обучения. Один из таких способов, основанный на динамическом программировании с учётом Байесовских оценок потерь описан в [16].

3.2.3 Сравнение с параллельными процедурами

В целом последовательная процедура позволяет реализовать более сложные правила распознавания. Если не затрагивать вопрос производительности, любая параллельная процедура может быть реализована в виде эквивалентной последовательной. Обратное же неверно.

Время выполнения последовательной процедуры в сравнении с эквивалентной параллельной зависит от выполняющего её вычислительного устройства. Если устройство последовательное, то в худшем случае на выполнение последовательной процедуры потребуется ровно столько же времени, сколько и на выполнение параллельной — в случае полностью линейного (без разветвлений) алгоритма распознавания. Чаше же всего последовательный подход позволяет сократить число необходимых для распознавания тестов. Так, в предыдущем примере системе не требовалось производить несколько попыток расшифровки штрих-кода всеми известными ей способами: алгоритм декодирования выбирался однажды на начальной стадии анализа.

Если же вычислительное устройство позволяет производить параллельные вычисления, то последовательная процедура может потребовать значительно большего времени, т.к. не позволяет распараллелить процесс выполнения тестов.

Возможно комбинирование этих двух принципов с целью достижения компромисса между простотой параллельных процедур и возможностями последовательных.

Главным недостатком последовательных процедур считают их уязвимость к надёжности. Если при формировании описания (или измерении) распознаваемого образа была допущена ошибка, или образ был искажён, что в результате привело к принятию неверного решения в одной из точек разветвления алгоритма, то процесс распознавания пойдёт по ошибочному пути, и вернуться на правильную ветвь будет невозможно. Параллельные процедуры, как правило, способны справляться с такими ошибками: ошибка одного детектора может быть компенсирована действием других.

4 ОСНОВНЫЕ ПОДХОДЫ К РАСПОЗНАВАНИЮ

В настоящее время при решении задачи распознавания в различных предметных областях разработано большое количество методов. Каждый из них по-своему уникален, обладает собственными возможностями и ограничениями. В одних случаях для решения конкретной задачи разрабатывается специальный метод распознавания, в других применяется адаптация существующих решений к специфике данной задачи.

Одна из классификаций методов распознавания, приведённая в [1], различает их по способу представления объектов распознавания. Эта характеристика в значительной степени определяет содержание метода, его область применения и используемый математический аппарат. Используемый методом способ представления образов можно расценивать как принятый в нём *подход к распознаванию*, на основе которого методы можно сравнивать между собой.

Следуя данной классификации, в трёх подразделах данного раздела описываются основные особенности методов, использующих *Евклидово пространство описаний*, списки признаков образов и структурное представление образов.

4.1 Евклидово пространство

Одним из широко распространённых подходов к представлению распознаваемых объектов является их представление в виде точек Евклидова пространства, которое строится следующим образом. Над первичным представлением распознаваемого образа производится серия вычислений, определяющих необходимые для классификации характеристики. Далее в многомерном Евклидовом пространстве (параметрическом пространстве, или пространстве характеристик), каждое измерение которого соответствует одной из вычисляемых характеристик, строится точка, соответствующая совокупности полученных измерений. По совокупностям точек, Евклидово расстояние между которыми мало, выделяют в область пространства, соответствующую данному классу изображений.

Как правило, методы, работающие с Евклидовым пространством, используют параллельную процедуру подачи входного изображения (см. разд.). Распознавание основывается на проведении ряда математических вычислений над полным множеством точек изображения. Можно выделить следующие основные используемые виды правил классификации:

- решающие (дискриминантные) функции;
- функции расстояния;
- функции правдоподобия.

Далее указанные правила классификации рассматриваются подробнее.

4.1.1 Решающие (дискриминантные) функции

Построение правил классификации в данном случае заключается в поиске функций, описывающих разделение параметрического пространства на соответствующие классам объектов области. Рассмотрим в качестве примера двумерное параметрическое пространство, изображённое на рис. 8. В этом пространстве заданы точки, описывающие объекты двух классов (они обозначены кругами и квадратами). По рисунку визуально можно установить, что представленные классы удобно разделить прямой линией вида

$$d(\mathbf{x}) = w_1x_1 + w_2x_2 + w_3 = 0.$$

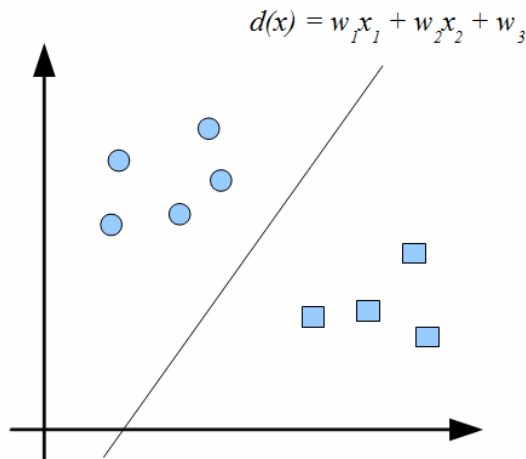


Рисунок 8: Разделение классов объектов в параметрическом пространстве прямой

Очевидно, что для всех объектов $x \in \omega_1$ (круги на рисунке) будет выполняться $d(\mathbf{x}) > 0$, а для всех $x \in \omega_2$ (квадраты) будет верно $d(\mathbf{x}) < 0$. Таким образом, функцию $d(\mathbf{x})$ можно использовать для формального определения принадлежности того или иного объекта одному из двух классов. Эта функция называется *решающей (дискриминантной)*. В описанном случае решающая функция линейна.

В общем случае n -мерного параметрического пространства линейная решающая функция задаётся выражением

$$d(\mathbf{x}) = w_1x_1 + w_2x_2 + \dots + w_nx_n + w_{n+1} = \mathbf{w}_0^T \mathbf{x} + w_{n+1}, \quad (3.1.1.1)$$

Где $\mathbf{w}_0 = (w_1, w_2, \dots, w_n)^T$ — *вектор (параметрический)* решающей функции, $\mathbf{x}_0 = (x_1, x_2, \dots, x_n)^T$ — вектор описания распознаваемого образа. Общепринято во все векторы образов после последней компоненты добавлять 1 и записывать выражение (3.1.1.1) в виде

$$d(\mathbf{x}) = \mathbf{w}^T \mathbf{x},$$

где $\mathbf{w} = (w_1, w_2, \dots, w_n, w_{n+1})^T$ и $\mathbf{x} = (x_1, x_2, \dots, x_n, x_{n+1})^T$ — *полные* векторы весов и образа соответственно. Подобное пополнение рассматриваемых векторов не затрагивает основных геометрических свойств классов в параметрическом пространстве.

Итак, в случае разбиения на два класса решающая функция имеет следующее свойство:

$$d(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \begin{cases} > 0, & \text{если } x \in \omega_1 \\ < 0, & \text{если } x \in \omega_2 \end{cases}.$$

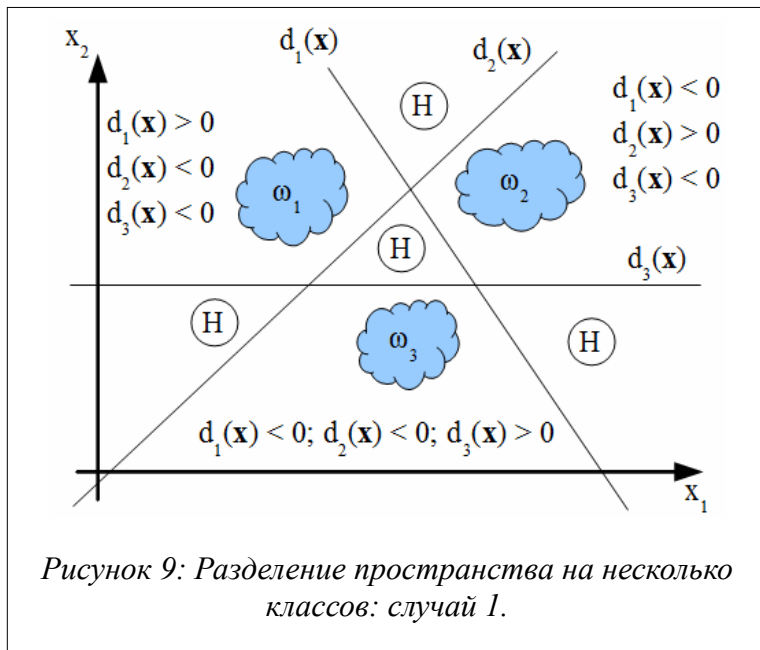
Для разбиения пространства на $M > 2$ классов образов $\omega_1, \omega_2, \dots, \omega_M$ необходимо использование более одной решающей функции. При этом возможны три варианта их задания.

1. Каждый класс отделяется от всех остальных одной разделяющей поверхностью. В этом случае необходимо задание M решающих функций, таких что

$$d_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} \begin{cases} > 0, & \text{если } x \in \omega_i \\ < 0, & \text{если } x \notin \omega_i \end{cases} \quad (3.1.1.2)$$

На рисунке 9 приведён пример разделения двумерного пространства на три класса с помощью трёх решающих функций $d_1(\mathbf{x})$, $d_2(\mathbf{x})$ и $d_3(\mathbf{x})$. Области, соответствующие решению об отнесении объекта к тому или иному классу, характеризуются выполнением одного из условий вида (3.1.1.2), т.е. выполнением $d_i(\mathbf{x}) > 0$ ровно для одного значения i .

На примере присутствуют четыре области, для которых данное условие не выполняется — они отмечены на рисунке буквами «Н». В трёх из этих областей $d_i(\mathbf{x}) > 0$ выполняется при двух значениях i , а одна (центральная на рисунке) область характерна невыполнением этого условия при любых значениях i . Эти области соответствуют непринятию какого-либо решения об отнесении объекта к представленным классам.

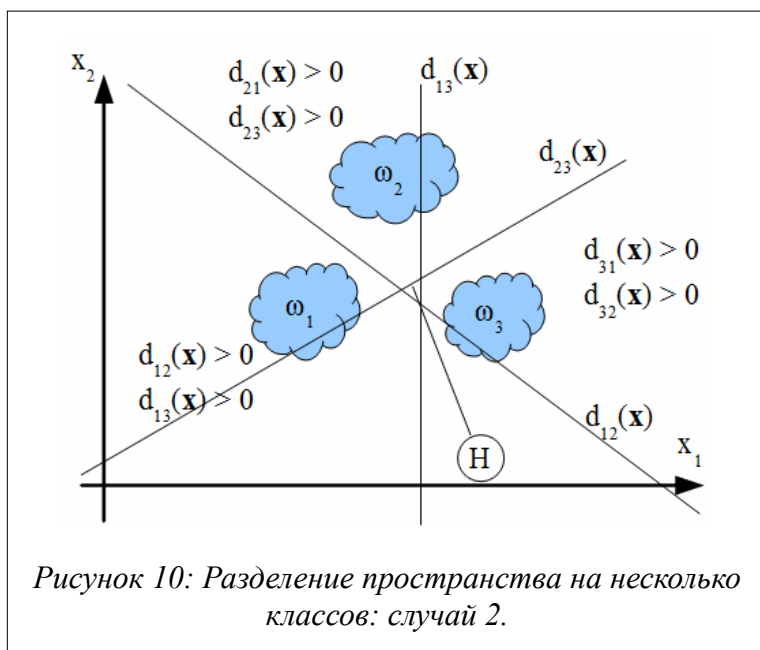


2. Каждый класс отделяется от любого другого взятого в отдельности класса «индивидуальной» поверхностью. Для M классов в таком случае потребуется $M(M-1)/2$ поверхностей (число сочетаний из M по 2). Решающие функции имеют вид $d_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^T \mathbf{x}$ и обладают тем свойством, что если образ \mathbf{x} принадлежит классу ω_i , то выполняется

$$d_{ij}(\mathbf{x}) > 0 \quad \text{для всех } j \neq i.$$

Кроме того: $d_{ij} = -d_{ji}$.

На рисунке 10 приведена иллюстрация использования данного способа разделения параметрического пространства. Каждый класс отделяется от каждого другого отдельной прямой. Так, класс ω_1 отделяется от ω_2 решающей функцией $d_{12}(\mathbf{x})$, а от ω_3 — функцией $d_{13}(\mathbf{x})$. Область принятия решения об отнесении объекта к классу ω_1 характеризуется выполнением



$$\begin{cases} d_{12}(\mathbf{x}) > 0 \\ d_{13}(\mathbf{x}) > 0 \end{cases}$$

В центральной части рисунка находится область неопределённости («Н»). Для неё

характерно отсутствие какой-либо пары чисел i и j , которых удовлетворяющих условию случая 2.

3. Существует M решающих функций $d_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x}$, $k = 1, 2, \dots, M$, таких, что если объект \mathbf{x} принадлежит классу ω_i , то выполняется

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad \text{для всех } j \neq i.$$

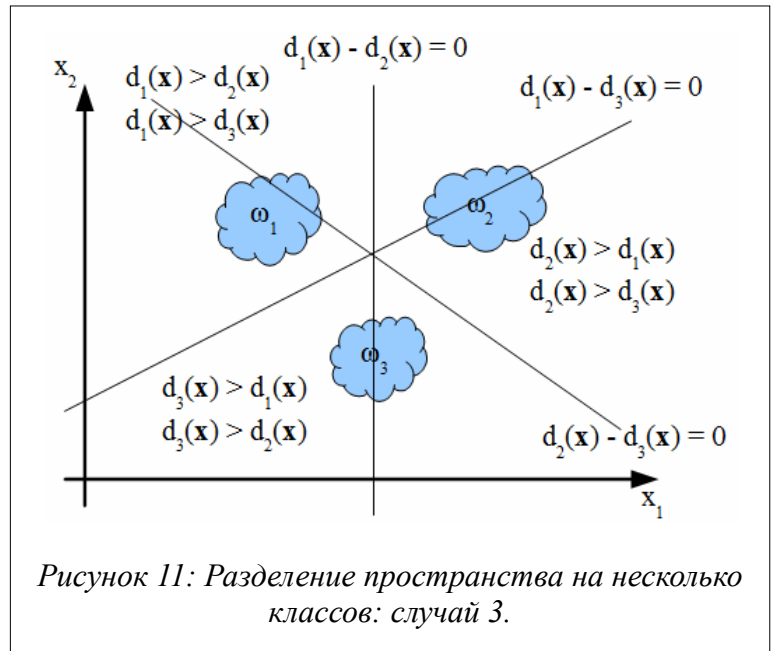
Данный случай является разновидностью случая 2, т.к. Можно положить

$$d_k(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = (\mathbf{w}_i - \mathbf{w}_j)^T \mathbf{x} = \mathbf{w}_{ij}^T \mathbf{x},$$

причём $\mathbf{w}_{ij} = \mathbf{w}_i - \mathbf{w}_j$.

Если $d_i(\mathbf{x}) > d_j(\mathbf{x})$ для всех $j \neq i$, то $d_{ij}(\mathbf{x}) > 0$ для всех $j \neq i$. Таким образом, если классы разделимы способом случая 3, то они разделимы и способом случая 2; обратное в общем случае не верно.

На рисунке 11 приведён пример разделения пространства на 3 класса описанным способом. Границы, разделяющие области классов ω_i и ω_j , обладают свойством $d_i(\mathbf{x}) = d_j(\mathbf{x})$, или, что эквивалентно, $d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$.



Если какой-либо набор классов разделяется в параметрическом пространстве набором линейных решающих функций одним из описанных способов, то классы из данного набора называются *линейно-разделимыми*. В более сложных случаях может принципиально потребоваться разделение классов линиями более сложного, чем линейные, вида.

Одним из удобных способов обобщения линейных решающих функций является введение решающих функций следующего вида:

$$d(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \dots + w_K f_K(\mathbf{x}) + w_{K+1} = \sum_{i=1}^{i=K+1} w_i f_i(\mathbf{x}), \quad (3.1.1.3)$$

где $\{f_i(\mathbf{x})\}, i = 1, 2, \dots, K$ — действительные однозначные функции образа \mathbf{x} , $f_{K+1}(\mathbf{x}) = 1$, а $K + 1$ — число членов разложения. Несмотря на то, что приведённая формула может задавать очень сложные решающие функции, применение соответствующего преобразования позволит работать с ними как с *линейными*. Определим вектор $\overset{\circ}{\mathbf{x}}$, компонентами которого являются функции $f_i(\mathbf{x})$:

$$\overset{\circ}{\mathbf{x}} = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_k(\mathbf{x}) \\ 1 \end{pmatrix}. \quad (3.1.1.4)$$

С помощью (3.1.1.4) выражение (3.1.1.3) записывается как

$$d(\mathbf{x}) = \mathbf{w}^T \overset{\circ}{\mathbf{x}}, \quad (3.1.1.5)$$

где $\mathbf{w} = (w_1, w_2, \dots, w_K, w_{K+1})^T$.

Функции $\{f_i(\mathbf{x})\}$, после того как их значения вычислены для конкретного \mathbf{x} , представляют просто набор чисел, а вектор $\overset{\circ}{\mathbf{x}}$ — обычный K -мерный числовой вектор, дополненный единицей. По отношению к новому представлению образов $\overset{\circ}{\mathbf{x}}$ выражение (3.1.1.5) представляет собой линейную функцию. Рассмотрение всех исходных образов \mathbf{x} в виде преобразованных представлений $\overset{\circ}{\mathbf{x}}$ превращает задачу в линейную.

Описанные манипуляции с векторами-описаниями образов носят исключительно математический характер. Если образы \mathbf{x} были n -мерными, то преобразованные образы $\overset{\circ}{\mathbf{x}}$ имеют размерность K , плюс приписанная единица. При этом K может оказаться гораздо больше, чем n . Таким образом, хотя в K -мерном пространстве решающие функции можно считать линейными, в n -мерном пространстве они сохраняют свой принципиально нелинейный характер.

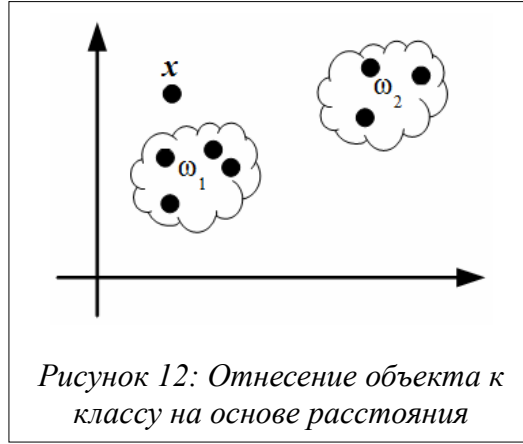
При проектировании системы распознавания после определения набора и вида решающих функций основной задачей является определение их коэффициентов. Для этого, как правило, используется некоторая обучающая выборка объектов.

4.1.2 Функции расстояния

При построении правил классификации на основе функций расстояния в качестве предпосылки используется то соображение, что естественным показателем сходства образов является степень близости точек, описывающих эти образы в Евклидовом пространстве. Рассматривая пример на рисунке 12, можно интуитивно отнести объект \mathbf{x} к классу ω_1 , исходя из тех соображений, что описывающая его точка располагается ближе к объектам именно этого класса, нежели к объектам класса ω_2 .

При использовании функций расстояния классы объектов представляются в виде кластеров в параметрическом пространстве. На этапе распознавания используется критерий минимума расстояния между точкой распознаваемого объекта и кластером класса, к которому этот объект должен быть отнесён. Построение правил классификации заключается в построении кластеров оптимальным образом.

Следует, однако, заметить, что данная методика эффективна только при наличии кластеризационных свойств у рассматриваемых классов объектов, т.е. когда расстояние между объектами внутри классов существенно меньше расстояния между группами точек, образующих классы.



Рассмотрим действие правил классификации, основанных на функциях расстояния. В некоторых случаях объекты каждого класса группируются вокруг некоторого единичного образа, являющегося типичным или репрезентативным для этого класса. В этом случае говорят, что класс представляется своим единственным *эталоном*. Например, такое явление наблюдается при распознавании печатных символов. В этой ситуации эффективным оказывается применение критерия минимального расстояния между распознаваемым объектом \mathbf{x} и каждым из эталонов имеющихся классов $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M$. Расстояние между образом \mathbf{x} и одним из эталонов \mathbf{z}_i в Евклидовом пространстве определяется как

$$D_i = \|\mathbf{x} - \mathbf{z}_i\| = \sqrt{(\mathbf{x} - \mathbf{z}_i)^T (\mathbf{x} - \mathbf{z}_i)}. \quad (3.1.1.6)$$

Образ зачисляется в класс ω_i , если выполняется условие $D_i < D_j$ для всех $j \neq i$. Возведение (3.1.1.6) в квадрат даёт

$$D_i^2 = \|\mathbf{x} - \mathbf{z}_i\|^2 = (\mathbf{x} - \mathbf{z}_i)^T (\mathbf{x} - \mathbf{z}_i) = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{z}_i + \mathbf{z}_i^T \mathbf{z}_i = \mathbf{x}^T \mathbf{x} - 2(\mathbf{x}^T \mathbf{z}_i - \frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i)$$

Поскольку все расстояния неотрицательны, выбор минимального расстояния D_i эквивалентен выбору минимального квадрата расстояния D_i^2 . Однако, поскольку член $\mathbf{x}^T \mathbf{x}$ не зависит от номера класса i , это в свою очередь эквивалентно выбору максимального значения $\mathbf{x}^T \mathbf{z}_i - \frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i$. Это обстоятельство даёт набор решающих функций

$$d_i(\mathbf{x}) = \mathbf{x}^T \mathbf{z}_i - \frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i, \quad i = 1, 2, \dots, M.$$

таких, что образ \mathbf{x} относится к классу ω_i , если выполняется условие $d_i(\mathbf{x}) > d_j(\mathbf{x})$ для всех $j \neq i$. Отметим, что полученные решающие функции линейны и могут быть записаны в виде

$$d_i(\mathbf{x}) = \mathbf{w}^T \mathbf{x},$$

$$\text{где } w_{ij} = z_{ij}, \quad j = 1, 2, \dots, n; \quad w_{i,n+1} = -\frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i; \quad \mathbf{x} = (x_1, x_2, \dots, x_n, 1)^T.$$

Таким образом, классификатор, действующим по принципу минимального расстояния, является частным случаем линейного классификатора. Специфическим свойством данного классификатора является то обстоятельство, что линейная разделяющая поверхность, обеспечивающая разбиение распознаваемых объектов на два класса, является гиперплоскостью, все точки которой равноудалены от эталонов этих классов. На рисунке 13 представлена двумерная иллюстрация разделяющей поверхности.

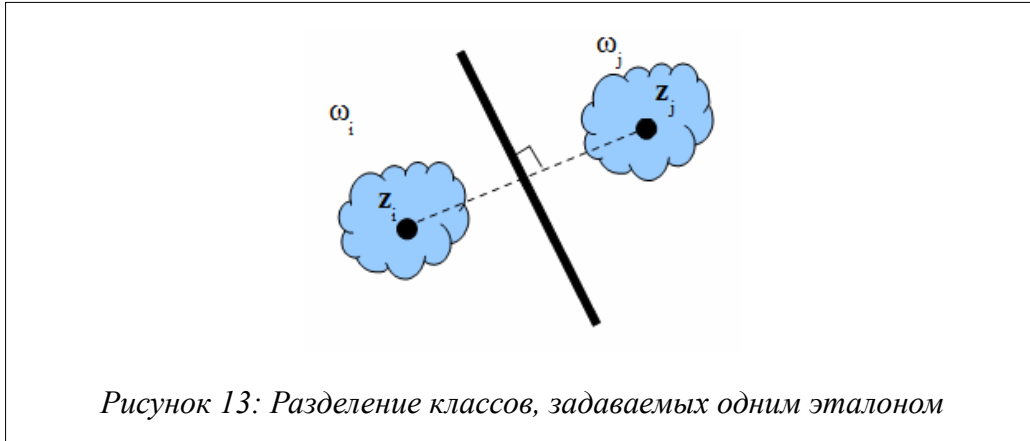


Рисунок 13: Разделение классов, задаваемых одним эталоном

Иногда класс ω_i может характеризоваться не единственным эталоном \mathbf{z}_i , а некоторой их группой $\mathbf{z}_i^1, \mathbf{z}_i^2, \dots, \mathbf{z}_i^{N_i}$, где N_i — число эталонов, задающих i -й класс. В этом случае расстояние от распознаваемого образа \mathbf{x}_i до данного класса можно задать как

$$D_i = \min_k \|x - z_i^k\|, \quad k = 1, 2, \dots, N_i,$$

т.е. использовать наименьшее расстояние от образа до одного из эталонов. Используя это выражение, в случае множественности эталонов можно следовать описанному выше критерию минимального расстояния. При этом решающие функции будут выглядеть как

$$d_i(\mathbf{x}) = \max_k \{(\mathbf{x}^T \mathbf{z}_i^k) - \frac{1}{2}(\mathbf{z}_i^k)^T \mathbf{z}_i^k\}, \quad k = 1, 2, \dots, N_i$$

При множественности эталонов границы, разделяющие два класса, будут кусочно-линейными.

Область применения подхода, основанного на использовании критерия минимального расстояния, не исчерпывается случаями задания классов с помощью эталонов. Рассмотрим случай, когда задана выборка объектов $\{\mathbf{s}_1, \mathbf{s}_1, \dots, \mathbf{s}_N\}$, для каждого из которых известна принадлежность одному из классов $\omega_1, \omega_2, \dots, \omega_M$. Определим правило классификации на основе принципа *ближайшего соседа* (БС-правило): распознаваемый объект следует отнести к тому классу, к которому принадлежит его ближайший сосед из выборки $\{\mathbf{s}_i\}$. Ближайшим соседом образа \mathbf{x} называется такой образ \mathbf{s}_i , для которого выполнено

$$D(\mathbf{s}_i, \mathbf{x}) = \min_k \{D(\mathbf{s}_k, \mathbf{x})\}, \quad k = 1, 2, \dots, N.$$

В данном случае учитывается принадлежность одному из классов только одного ближайшего соседа, вследствие чего это правило можно назвать 1-БС-правилом. Можно рассмотреть более общий случай — q -БС-правило. В этом случае для классификации образа следует определить q его ближайших соседей и отнести его к тому классу, к которому относится наибольшее число образов из получившейся группы.

Обратимся теперь к методам выявления кластеров в Евклидовом параметрическом пространстве, используемых описанными правилами классификации. Эта задача заключается в нахождении среди объектов выборки $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ некоторого поднабора $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M\}$, $M \leq N$, объекты которого представляют собой центры кластеров (их число соответствует числу классов), а также в отнесении всех прочих объектов к этим кластерам. В настоящее время разработано достаточно много методов выявления кластеров. Следует отметить, что, как правило, решение этой задачи носит в той или иной мере эвристический характер и зависит от рассматриваемых данных, выбранной меры близости и критерия

качества кластеризации. Рассмотрим несколько простых методов выявления кластеров.

Один из простейших способов выявления кластеров заключается в следующем. Имеется выборка $\{x_1, x_2, \dots, x_N\}$ и некоторая неотрицательная пороговая величина T . Пусть центр первого кластера совпадает с одной из точек выборки; для простоты положим $z_1 = x_1$. Вычисляется расстояние D_{21} между образом x_2 и первым центром z_1 . Если это расстояние больше пороговой величины T , то точка x_2 назначается центром нового, второго кластера. Иначе, эта точка включается в первый кластер. Пусть, например, это условие выполнено и $z_2 = x_2$. Далее вычисляются расстояния D_{31} и D_{32} от точки x_3 до центров z_1 и z_2 . Если обе эти величины больше порога T , x_3 назначается центром нового кластера. Если хотя бы одно из расстояний меньше порога, образ x_3 зачисляется в тот кластер, к центру которого он ближе. Далее, по тому же принципу, для каждого образа вычисляются расстояния от него до всех имеющихся центров, и если все эти величины превосходят порог, этот образ выделяется в отдельный кластер; в противном случае он зачисляется в кластер с ближайшим к нему центром.

Результаты его применения данного способа зависят от выбора первого центра, порядка обхода объектов выборки, величины порога T , а также геометрических свойств выборки. Однако, несмотря на эти недостатки, он эффективен с вычислительной точки зрения, т.к. выполняет только один проход по выборке.

Другим алгоритмом выявления кластеров является алгоритм *максиминного расстояния*. Он во многом сход с предыдущим алгоритмом, но его особенностью является стремление к выделению в первую очередь наиболее удалённых кластеров. Как и ранее, произвольный образ, например x_1 , назначается первым центром z_1 . Далее для всех остальных объектов вычисляется расстояние до этого центра и наиболее удалённый назначается новым центром. На следующем шаге вычисляются пары расстояний от каждого из оставшихся объектов до двух имеющихся центров z_1 и z_2 и из каждой пары выбирается минимальное. После этого выбирается максимальное из полученных минимальных расстояний, и, если оно составляет значительную часть расстояния между z_1 и z_2 (например, не менее половины этого расстояния), соответствующий образ назначается центром z_3 . Если максиминное расстояние недостаточно велико, алгоритм останавливается. На всех последующих шагах процедура повторяется с учётом нового набора кластеров. Для каждого объекта, не являющегося центром, вычисляются наборы расстояний до всех имеющихся центров кластеров и в каждой группе выбирается минимальное. Максимальное среди минимальных расстояний сравнивается с «типичным» расстоянием между центрами кластеров (например, «типичным» расстоянием может служить среднее). Если это расстояние достаточно велико, образуется новый кластер, в противном случае процедура завершается. Каждый их объектов, не назначенных в качестве центров, относится к тому кластеру, центр которого к нему ближе.

Ещё один распространённый алгоритм носит название алгоритма « K -средних». Этот подход имеет формальный критерий качества, который минимизируется в процессе выполнения алгоритма: сумма квадратов расстояний от всех объектов каждого кластера до центра кластера. Ещё одним его отличием является необходимость знания требуемого числа кластеров до начала кластеризации. Алгоритм состоит из следующих шагов.

1. В исходной выборке произвольно назначаются K исходных кластеров: $z_1(1), z_2(1), \dots, z_K(1)$.

2. На k -м шаге заданная выборка образов $\{x_i\}$ распределяется по K имеющимся кластерам по следующему правилу:

$$x \in S_j(k) \Leftrightarrow \|x - z_j(k)\| < \|x - z_i(k)\| \quad \forall i = 1, 2, \dots, K, i \neq j,$$

где $S_j(k)$ — множество образов, входящих в кластер с центром в точке $\mathbf{z}_j(k)$.

3. Определяются новые центры кластеров $\mathbf{z}_j(k+1)$, $j=1,2,\dots,K$, исходя из минимизации показателя качества

$$J_j = \sum_{x \in S_j(k)} \|\mathbf{x} - \mathbf{z}_j(k+1)\|^2, \quad j=1,2,\dots,K,$$

представляющего сумму квадратов расстояний от объектов кластера до его центра. Объект $\mathbf{z}_j(k+1)$, минимизирующий показатель качества, представляет собой, по сути, выборочное среднее по множеству $S_j(K)$. Следовательно, новые центры кластеров можно вычислить как

где N_j — число образов в множестве $S_j(k)$.

4. При выполнении условия $z_j(k) \neq z_j(k)$ для всех $j=1,2,\dots,K$ выполнение алгоритма прекращается. В противном случае выполнение переносится к шагу 2.

С другими методами кластеризации можно ознакомиться в [10].

4.1.3 Функции правдоподобия

Существует класс областей, где процесс порождения распознаваемых образов находится под влиянием ряда случайных факторов. К таким областям можно в первую очередь отнести задачи интерпретации физических явлений, данные о которых снимаются приборами, подверженными действию шумов и искажений. Например, речь может идти об анализе ЭКГ, аэрофотоснимков, сканированных изображений. В этих случаях важную роль играет аппарат статистического анализа.

В условиях влияния случайных факторов процесс распознавания может представляться как игра распознающего устройства с реальным миром, в которой машина пытается угадать образ, задуманный природой. Этот процесс аналогичен игре двух лиц с нулевой суммой. Это означает, что в каждый момент времени выигрыш одного игрока в точности равен проигрышу второго. В играх такого типа используют различные стратегии поведения, такие как байесовская стратегия, минимаксная стратегия и стратегия Неймана-Пирсона. Задача классификатора состоит в поиске такой стратегии, которая обеспечивала бы минимальность среднего проигрыша.

Представим игру формально в виде тройки $G=(Y,Z,L)$, где Y и Z — множества произвольной природы, L — ограниченная числовая функция, определённая на множестве прямых произведений $(Y \times Z)$. Элементы $y \in Y$ и $z \in Z$ представляют стратегии первого и второго игроков соответственно, а функция L интерпретируется как *функция выигрыша* или *функция потерь*. На каждом шаге игры первый игрок выбирает элемент своей стратегии $y \in Y$, а второй — элемент $z \in Z$ своей. Если побеждает первый игрок, он получает выигрыш, равный значению $L(y,z)$, который выплачивается вторым игроком. Если побеждает второй игрок, то сумму $L(y,z)$ выплачивает ему первый игрок. Игра G называется конечной, если множества Y и Z содержат конечные количества элементов:

$$Y = \{y_1, y_2, \dots, y_M\}$$

$$Z = \{z_1, z_2, \dots, z_N\}$$

В этом случае функцию L можно задать в виде матрицы размером $M \times N$ с элементами $L_{ij} = L(y_i, z_j)$, которую называют *матрицей выигрышей* или *матрицей потерь*. Элемент L_{ij} матрицы потерь определяет проигрыш первого игрока, выбравшего ход y_i , при условии, что

второй игрок выбрал ход z_j . Принято считать положительные значения L_{ij} истинными потерями, а отрицательные — выигрышами.

В контексте задачи распознавания будем считать первым игроком природу, элементами стратегии которой являются классы ω_i предъявляемых ею образов \mathbf{x} , а вторым игроком — распознаватель, оперирующий решениями относительно возможных принадлежностей объектов \mathbf{x} классам ω_i . Будем считать, что число элементов в обоих множествах одинаково, хотя в общем случае это может быть и не так.

В каждой игре, т.е. на каждом сеансе распознавания, природа предъявляет объект \mathbf{x} , принадлежащий классу ω_i в соответствии с априорной вероятностью $p(\omega_i)$. Эта вероятность просто определяет вероятность встретить произвольный объект класса ω_i . Таким образом, в распоряжении второго игрока оказывается описание объекта \mathbf{x} с неизвестной принадлежностью к одному из классов. Его задача как распознавателя заключается в выборе элемента своей стратегии, соответствующего задуманного первым игроком классу. Следует отметить, что игрок-природа не является «разумным противником», который осуществляет свои ходы с целью максимизировать свой выигрыш и принести наибольшие убытки своему сопернику. Можно допустить, что стратегия этого игрока полностью определяется набором априорных вероятностей $p(\omega_1), p(\omega_2), \dots, p(\omega_M)$. Следовательно, это знание может быть использовано игроком-распознавателем при построении своей стратегии.

Итак, игрок-природа предъявляет образ \mathbf{x} . Вероятность его принадлежности классу ω_i определяется величиной $p(\omega_i | \mathbf{x})$. Если распознаватель относит объект к классу ω_j , а на самом деле он принадлежит классу ω_i , то распознаватель терпит убытки в размере L_{ij} . Т.к. объект может принадлежать любому из классов, то математическое ожидание потерь, возможных в результате отнесения его классу ω_j , будет равно величине

$$r_j(\mathbf{x}) = \sum_{i=1}^M L_{ij} p(\omega_i | \mathbf{x}), \quad (3.1.1.7)$$

называемой *условным средним риском* или *условными средними потерями*. Если для каждого предъявленного объекта \mathbf{x} вычисляются значения условных средних потерь $r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_M(\mathbf{x})$ и классификатор причисляет объект к классу, которому соответствует наименьшая из этих величин, то и математическое ожидание полных потерь на множестве всех решений будет минимизировано. Классификатор, минимизирующий математическое ожидание общих потерь, называется *байесовским классификатором*. Со статистической точки зрения такой классификатор соответствует оптимальному качеству классификации (расознавания).

Используя формулу Байеса

$$p(\omega_i | \mathbf{x}) = \frac{p(\omega_i)p(\mathbf{x} | \omega_i)}{p(\mathbf{x})},$$

представим (3.1.1.7) в следующем виде:

$$r_j(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{i=1}^M L_{ij} p(\mathbf{x} | \omega_i) p(\omega_i)$$

Выражение $1/p(\mathbf{x})$ присутствует во всех выражениях $r_i(\mathbf{x}), i = 1, \dots, M$, следовательно его можно исключить:

$$r_j(\mathbf{x}) = \sum_{i=1}^M L_{ij} p(\mathbf{x} | \omega_i) p(\omega_i) \quad (3.1.1.8)$$

$p(\mathbf{x} | \omega_i)$ называется функцией правдоподобия для класса ω_i .

При рассмотрении задачи разделения на два класса, объект \mathbf{x} будет отнесён к классу ω_1 при выполнении условия $r_1(\mathbf{x}) < r_2(\mathbf{x})$. Раскрывая это условие при помощи (3.1.1.8) с подстановкой $M = 2$, получим следующую его запись:

$$L_{11}p(\mathbf{x} | \omega_1)p(\omega_1) + L_{21}p(\mathbf{x} | \omega_2)p(\omega_2) < L_{12}p(\mathbf{x} | \omega_1)p(\omega_1) + L_{22}p(\mathbf{x} | \omega_2)p(\omega_2)$$

или

$$(L_{21} - L_{22})p(\mathbf{x} | \omega_2)p(\omega_2) < (L_{12} - L_{11})p(\mathbf{x} | \omega_1)p(\omega_1)$$

Обычно считается, что $L_{ij} > L_{ji}$ при $i \neq j$. При этом допущении получаем

$$\frac{p(\mathbf{x} | \omega_1)}{p(\mathbf{x} | \omega_2)} > \frac{p(\omega_2)(L_{21} - L_{22})}{p(\omega_1)(L_{12} - L_{11})}. \quad (3.1.1.9)$$

Выполнение этих условий определяет отнесение объекта \mathbf{x} к классу ω_1 . При этом левую часть неравенства (3.1.1.9) называют *отношением правдоподобия*:

$$l_{12} = \frac{p(\mathbf{x} | \omega_1)}{p(\mathbf{x} | \omega_2)}$$

Итак, для случая двух классов, имеем следующее байесовское решающее правило:

- если выполнено $l_{12}(\mathbf{x}) > \theta_{12}$, то образ \mathbf{x} зачисляется в класс ω_1 ;
- 1. если выполнено $l_{12}(\mathbf{x}) < \theta_{12}$, то образ \mathbf{x} зачисляется в класс ω_2 ;
- 2. если выполнено $l_{12}(\mathbf{x}) = \theta_{12}$, то решение принимается произвольным образом.

Величину θ_{12} называют *пороговым значением*:

$$\theta_{12} = \frac{p(\omega_2)(L_{21} - L_{22})}{p(\omega_1)(L_{12} - L_{11})}$$

Рассмотрим случай разделения пространства образов на $M > 2$ классов. В этом случае байесовское правило относит объект \mathbf{x} к классу ω_i при выполнении условия $r_i(\mathbf{x}) < r_j(\mathbf{x})$, $j = 1, 2, \dots, M$; $j \neq i$, которое в расширенной форме выглядит как

$$\sum_{k=1}^M L_{ij} p(\mathbf{x} | \omega_k) p(\omega_k) < \sum_{q=1}^M L_{qj} p(\mathbf{x} | \omega_q) p(\omega_q)$$

$$j = 1, 2, \dots, M; \quad j \neq i$$

Это неравенство также может быть представлено с использованием отношений правдоподобия и пороговых величин, если принять

$$l_{ij} = \frac{p(\mathbf{x} | \omega_i)}{p(\mathbf{x} | \omega_j)},$$

$$\theta_{ij} = \frac{p(\omega_j)(L_{ji} - L_{jj})}{p(\omega_i)(L_{ij} - L_{ii})}$$

Для лучшего представления общего правила разделения на несколько классов целесообразно воспользоваться функцией потерь специального вида. Во многих задачах распознавания при правильном принятии решения потери считаются нулевыми, а потери при любом неправильном решении считаются одинаковыми и равными, например, 1:

$$L_{ij} = 0, \quad i = j$$

$$L_{ij} = 1, \quad i \neq j$$

При использовании такой функции величину средних потерь можно записать как

$$r_j(\mathbf{x}) = \sum_{i=1}^M L_{ij} p(\mathbf{x} | \omega_i) p(\omega_i) = p(\mathbf{x}) - p(\mathbf{x} | \omega_j) p(\omega_j)$$

Тогда отнесение объекта \mathbf{x} к классу ω_i выполняется байесовским правилом при истинности условия

$$p(\mathbf{x}) - p(\mathbf{x} | \omega_i) p(\omega_i) < p(\mathbf{x}) - p(\mathbf{x} | \omega_j) p(\omega_j),$$

или

$$p(\mathbf{x} | \omega_i) p(\omega_i) > p(\mathbf{x} | \omega_j) p(\omega_j), \quad j = 1, 2, \dots, M; j \neq i.$$

Заметим, что это выражение байесовского решающего правила эквивалентно решающей функции (см. выше) следующего вида

$$d_i(\mathbf{x}) = p(\mathbf{x} | \omega_i) p(\omega_i) \quad (3.1.1.10).$$

Образ \mathbf{x} зачисляется в класс ω_i при выполнении условия $d_i(\mathbf{x}) > d_j(\mathbf{x}), \forall j \neq i$, что соответствует случаю 3 разбиения пространства на несколько классов решающими функциями.

Если преобразовать полученное выражение решающей функции с помощью формулы Байеса, получим формулу

$$d_i(\mathbf{x}) = p(\omega_i | \mathbf{x}) p(\mathbf{x}),$$

в которой множитель $p(\mathbf{x})$ может быть исключён, что даёт

$$d_i(\mathbf{x}) = p(\omega_i | \mathbf{x}) \quad (3.1.1.11).$$

Формула (3.1.1.11) по сравнению с (3.1.1.10) избавляет решающее правило от необходимости знания в явном виде вероятностей $p(\mathbf{x} | \omega_i)$ и $p(\omega_i)$. Заметим, что хотя оба выражения эквивалентны, они представляют разные правила классификации. Поскольку оценка априорной вероятности классов $p(\omega_i), i = 1, \dots, M$ обычно не вызывает затруднений, основное различие состоит в том, что в первом случае используются функции правдоподобия $p(\mathbf{x} | \omega_i)$, а во втором — вероятности $p(\omega_i | \mathbf{x})$ принадлежности объекта \mathbf{x} к классу ω_i .

К классу методов, использующих Евклидово пространство описаний, можно также отнести нейросетевые методы. Нейронные сети способны решать как задачи распознавания, так и автоматической классификации, допуская обучение как с учителем, так и без него. Рассмотрение этих методов ведётся в разделе 0.

4.2 Списки признаков

Методы данной категории опираются на возможность классификации распознаваемых объектов на основе наличия в них некоторых характерных признаков.

Существуют два основных подхода. Первый основывается на предположении, что непосредственные измерения, проводимые над изображением, есть результат действия совокупности небольшого числа порождающих признаков. m -мерное пространство прямых измерений сводится к k -мерному пространству признаков, где $k < m$, в котором и производится распознавание образа. При построении решающих правил задача состоит в определении пространства признаков. Эта задача является по сути задачей понижения размерности описания образов, для решения которой используется аппарат факторного анализа.

Второй подход определяет признаки как подмножества множеств прямых измерений.

При распознавании бинарных изображений такими признаками могут служить наличие чёрных точек в определённых областях изображения (например, диагональ или горизонтальная черта в середине изображения), число чёрных точек на характеристической линии, проводимой через изображение и т.д. Распознаваемые объекты представляются как различные совокупности выделяемых признаков.

4.2.1 Формализация подхода, использующего факторный анализ

Пусть распознаваемый образ \mathbf{x} описывается вектором прямых измерений длины m :

$$\mathbf{x} = (x_1, x_2, \dots, x_m).$$

Пусть, в то же время, он может быть описан не только с помощью прямых измерений, но и с помощью некоторого набора признаков. Эти признаки имеют собственные количественные значения и их количество k меньше количества прямых измерений объекта ($k < m$). Запишем это представление объекта в виде вектора измерений признаков образа:

$$\mathbf{f} = (f_1, f_2, \dots, f_k),$$

где f_i отражает «содержание» в описании образа i -го признака (или *фактора* по терминологии факторного анализа). Связь между двумя описаниями образа задаётся следующим образом. Как было сказано в формулировке основополагающего предположения данного подхода, прямые измерения образа есть результат действия его признаков. При этом каждый признак оказывает на каждое измерение в общем случае разное влияние. Тогда можно рассматривать j -е прямое измерение образа в виде взвешенной суммы значений признаков данного образа:

$$x_j = (a_{1j}f_1 + a_{2j}f_2 + \dots + a_{kj}f_k),$$

где коэффициент a_{ij} указывает вклад i -го признака в j -е измерение образа. Коэффициенты вкладов (нагрузок) признаков в описание образа образуют матрицу \mathbf{A} размером $(k \times m)$. Тогда связь между прямыми измерениями образа и измерениями его признаков можно записать в виде

$$\mathbf{x} = \mathbf{f} \cdot \mathbf{A}.$$

Если рассматривать описание не единственного образа, а целой выборки, состоящей из n образов, вектора \mathbf{x} и \mathbf{f} заменяются матрицами $\mathbf{X}(n \times m)$ и $\mathbf{F}(n \times k)$, строками которых являются, соответственно, вектора прямых измерений и измерений признаков отдельных образов выборки:

$$\mathbf{X} = \mathbf{F} \cdot \mathbf{A}$$

Задача при построении метода распознавания заключается в определении \mathbf{F} и \mathbf{A} , обеспечивающих адекватное и удобное для распознавания описание распознаваемых объектов, по имеющимся описаниям образов обучающей выборки \mathbf{X} . Для её решения широко используется метод *главных компонент*. Пусть \mathbf{R} — матрица корреляций между основными измерениями образа. Её диагональные элементы r_{ij} представляются множественными корреляциями r_i с остальными $m-1$ измерениями. Пусть \mathbf{Q} — матрица, строки которой являются собственными векторами матрицы \mathbf{R} , соответствующими k собственным значениям матрицы \mathbf{R} в порядке убывания от наибольшего к наименьшему. И пусть λ_j — j -е собственное значение \mathbf{R} . Тогда элементы матрицы \mathbf{A} могут быть вычислены по следующей формуле:

$$a_{ij} = q_{ij} (\lambda_j)^{\frac{1}{2}} \left(\sum_{s=1}^m q_{sj}^2 \right)^{-\frac{1}{2}}$$

И, далее, матрица факторов \mathbf{F} вычисляется как

$$\mathbf{F} = \mathbf{Z} \cdot \mathbf{A}^{-1},$$

или, в случае описания единственного образа:

$$\mathbf{f} = \mathbf{x} \cdot \mathbf{A}^{-1}$$

Последняя формула позволяет «переводить» непосредственное описание распознаваемого образа в термины принятых к рассмотрению признаков и осуществлять распознавание в пространстве описаний пониженной размерности.

Открытым вопросом остаётся количество признаков, необходимых для адекватного описания образов. В общем смысле выбор количества признаков достаточно произволен и обуславливается особенностями предметной области. Но часто в его решении пользуются некоторыми формальными критериями и рекомендациями. Например, можно воспользоваться следующим статистическим критерием. Пусть V^2 — сумма квадратов расстояний от каждой точки \mathbf{x}_i до центра тяжести n точек в исходном m -мерном пространстве прямых измерений, и пусть W_k^2 — та же мера для точек \mathbf{f}_i в k -мерном пространстве признаков. Начиная с $k=1$, постоянно увеличивают размер пространства признаков до тех пор, пока разность $V^2 - W_k^2$ или $1 - (W_k^2 / V^2)$ не станет достаточно малой.

Более подробно метод главных компонент, а также другие методы определения матрицы вкладов рассмотрены в [13].

4.2.2 Формализация подхода группирования прямых измерений

Как было указано выше, суть этого подхода заключается в трактовке некоторых характерных подмножеств множества прямых измерений образа в качестве его признаков. Опишем этот подход формально. Будем рассматривать класс предметных областей, в которых описания образов строятся с помощью наборов бинарных измерений. Примером таких образов могут служить чёрно-белые изображения, каждая точка которых может быть описана с помощью 0, если её цвет — белый, или с помощью 1, если цвет — чёрный. Более формально, j -й элемент вектора описания образа \mathbf{x} принимает значение 1, если j -е измерение образа принимает значение *есть* при возможных значениях *есть* и *нет*.

Рассматриваемая модель признаков предполагает, что при наличии у образа определённого признака некоторый поднабор его измерений будет иметь значение 1 независимо от присутствия или отсутствия у образа других признаков. Это также означает, что если образ имеет несколько признаков, их влияния на прямые измерения будут комбинироваться по правилу булевого сложения:

$$1 +_2 1 = 1,$$

$$1 +_2 0 = 1,$$

$$0 +_2 0 = 0,$$

j -й признак из k рассматриваемых может задаваться как двоичный вектор \mathbf{f}_i длины m , элемент f_{js} которого имеет значение 1 в том случае, если s -е измерение образа, имеющего данный признак, даёт значение 1. Тогда вектор описания образа может быть записан как

$$\mathbf{x} = \sum_{j=1}^k {}_2 \mathbf{f}_j,$$

где \sum_2 обозначает булево суммирование нескольких переменных (векторов). Если рассматривать выборку из n образов, то последнее выражение можно записать в матричной форме. Пусть $*$ обозначает булево умножение матриц, которое ведётся по обычным

матричным правилам, но с заменой суммирования на булево суммирование. И пусть \mathbf{V} — матрица размером $n \times k$, элемент b_{ij} которой равен 1 тогда и только тогда, когда объект i имеет признак j . Тогда матричная запись описания выборки образов будет иметь следующий вид:

$$\mathbf{X} = \mathbf{V} * \mathbf{F},$$

где матрица \mathbf{F} образована векторами описания признаков \mathbf{f}_j и имеет размер $k \times m$.

4.2.3 Методы выделения признаков

При рассмотрении подхода к признаковому описанию образов с использованием факторного анализа в разделе 0 был описан один из формальных способов определения количества признаков и их описания. Нередко, однако, в особенности при использовании второго подхода к заданию признаков, формальное решение этой задачи неочевидно либо требует значительных трудозатрат. В этом случае со стороны разработчика метода распознавания требуется применение творческого подхода, основанного на знаниях о предметной области. В данном разделе приводятся примеры использования формального и эвристического подхода к выделению наборов признаков.

Имитационный метод Ура и Восслера. Серия экспериментов, описанных в [14], показала, что хорошее выделение признаков может дать простая процедура имитации частей входных образов. В работе рассматривается классификация двумерных образов, таких как нарисованные лица, рукописные буквы и пр. На вход программы подавались бинарные изображения размером 25×25 . Их входных образов программа строила набор «локальных операторов» размером 5×5 . Для этого фрагмент размером 5×5 одного из входных изображений, расположенный в определённом его месте, сравнивался с другими входными изображениями. Если несколько входных изображений имели фрагменты 5×5 с почти одинаковой конфигурацией, независимо от местоположения этих фрагментов, то такое подыображение принималось в качестве признака. Имея набор признаков, выделенных таким способом, каждое изображение 25×25 могло быть описано с помощью перечисления типов содержащихся в нём признаков с указанием их местоположений. Например, изображение могло описываться с помощью высказываний типа «горизонталь вверху» или «круг слева внизу».

Набор признаков для каждого изображения собирался в двоичный вектор \mathbf{z} . Для этих векторов затем в процессе классификации применялся простой алгоритм группирования. В каждый момент времени использовался фиксированный набор признаков. При группировании каждой из компонент вектора \mathbf{z} приписывался вес, отражающий её полезность для распознавания. Если какой-либо из весов приближался к 0, то соответствующий признак отбрасывался и производился поиск нового признака. Такой процесс генерации, группирования и новой генерации признаков позволял построить эффективную систему распознавания изображений данного вида. Она решала задачи на зрительное восприятие изображений быстрее людей в случаях, когда речь шла о распознавании изображений, которые не были хорошо известны людям, например, абстрактных математических фигур.

Формальный имитационный метод Блока, Нильсона и Дуды. Рассмотрим первичное бинарное описание образа в несколько модифицированной форме. Преобразуем двоичный вектор описания \mathbf{x}_i i -го объекта выборки в множество чисел

$$P(i) \subseteq \{j\}, \quad 1 \leq j \leq m,$$

где $j \in P(i) \Leftrightarrow x_{ij} = 1$. Будем также пользоваться функцией $\eta(P) = |P|$, вычисляющей число элементов в описании объекта.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Рисунок 14: Представление бинарного объекта на сетке

Объект, показанный на рисунке 14, имеющий вектор описания $\mathbf{x} = (0,0,1,0,0,0,0,1,0,0,1,1,1,1,0,0,1,0,0,0,0,1,0,0)$, в модифицированной форме будет описан в виде множества $P = \{3,8,11,12,13,14,15,18,23\}$.

Будем называть любое множество целых чисел $\sigma \subseteq \{1, \dots, t\}$ *маской* и будем говорить, что маска σ присутствует в объекте $P(i)$, если $\sigma \subseteq P(i)$. При такой постановке понятие маски соответствует некоторому признаку образа. Задача определения множества признаков по выборке образов $S = \{P(i)\}, i = 1, \dots, n$ заключается в определении такого наименьшего множества масок $\Phi = \{\sigma\}$, что для каждого $P(i)$ существует множество масок $M(i) \subseteq \Phi$, такое что

$$P(i) = \bigcup_{\sigma \in M(i)} \sigma.$$

При этом для заданного множества масок \square и множества описаний $P(i)$ набор множеств $M(i)$ может быть неединственным.

Пусть $F(\sigma)$ — множество описаний объектов, содержащих маску σ . Тогда, по утверждению Блока и др., маска σ будет определяться как пересечение таких описаний:

$$\sigma = \bigcap_{P(i) \in F(\sigma)} P(i)$$

Таким образом, если $F(\sigma)$ известно, т.е. если априори известны объекты, обладающие одинаковым признаком, то можно найти формальное описание этого признака σ . Однако в большинстве случаев $F(\sigma)$ неизвестно. Блок и др. предлагают процедуру, позволяющую решить задачу нахождения набора неизвестных признаков среди множества объектов. Необходимо решить, когда два объекта $P(i_1)$ и $P(i_2)$ содержат общую маску. Решив этот вопрос для всех объектов выборки, можно определить множество $F(\sigma)$ и, следовательно, найти саму маску σ . Решение предлагается осуществлять введением порога \square для общих ненулевых элементов двух образов. Считается, что два образа $P(i_1)$ и $P(i_2)$ содержат общую маску тогда и только тогда, когда

$$\eta(P(i_1) \cap P(i_2)) > \theta.$$

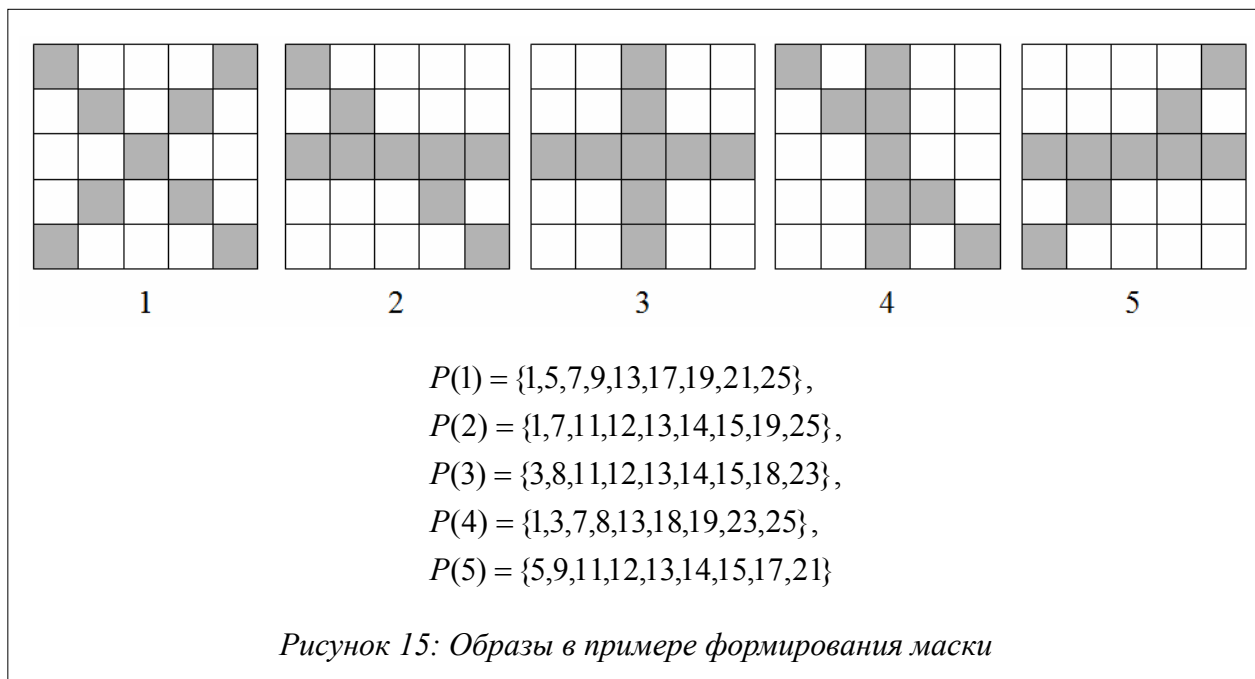
Опишем процедуру получения в точности одной маски, содержащейся в объектах выборки S . Пусть эта выборка как-то упорядочена: $S = (P(1), P(2), \dots, P(n))$. Будем рассматривать объекты выборки один за другим в этом порядке. Пусть $\sigma(i)$ есть определение маски σ , построенным после рассмотрения i -го описания из S , так что $\sigma = \sigma(n)$. В этом

случае алгоритм имеет вид:

$$\sigma(0) = \{j\}, \quad j = 1, \dots, m;$$

$$\sigma(i+1) = \begin{cases} \sigma(i) \cap P(i+1) \text{ тогда и только тогда,} \\ \text{когда } \eta(\sigma(i) \cap P(i+1)) > \theta, \\ \sigma(i) \text{ в противном случае;} \end{cases}$$

$$\sigma = \sigma(n).$$



Проиллюстрируем работу данного алгоритма. Пусть имеется выборка образов, показанная на рис. 15. Анализ данных изображений приводит к предположению о существовании нескольких возможных масок: диагоналей, горизонталей, вертикалей (в отличие от метода Ура и Восслера, без привязки к местоположению на изображении). Положим $\theta = 2$, ограничивая размер искомых масок минимальной величиной в 3 точки. Последовательность действий алгоритма такова:

	$\sigma(0) = \{1,2,3,\dots,25\}$
$\eta(\sigma(0) \cap P(1)) = 9 > 2$	$\sigma(1) = \{1,5,7,9,13,17,19,21,25\}$
$\eta(\sigma(1) \cap P(2)) = 5 > 2$	$\sigma(2) = \{1,7,13,19,25\}$
$\eta(\sigma(2) \cap P(3)) = 1 \leq 2$	$\sigma(3) = \sigma(2)$
$\eta(\sigma(3) \cap P(4)) = 5 > 2$	$\sigma(4) = \{1,7,13,19,25\}$
$\eta(\sigma(4) \cap P(5)) = 1 \leq 2$	$\sigma(5) = \sigma(4)$
	$\sigma = \sigma(5)$

Окончательная маска представляет собой диагональ, идущую от левого верхнего края в левый нижний. Эта маска действительно используется при построении некоторых приведённых на рисунке 15 объектов.

Другие маски данной выборки объектов можно получить, упорядочивая её другими

различными способами. Для выборки из n объектов всего существует $n!$ упорядочений. Множество полученных таким образом масок не будет являться наименьшим. Более эффективный алгоритм проходит выборку объектов k раз и формирует k масок. Пусть σ_j — маска, полученная после j -й итерации, а $\sigma_j(i)$ — определение маски σ_j после рассмотрения i -го описания в ходе j -й итерации.

Определим K_{ij} как множество целых чисел $\{k\}$, для которых

$$k < j, \quad \eta(P(i) \cap \sigma_k) > \theta, \quad \sigma_k \subseteq \sigma_j(i).$$

Таким образом, в K_{ij} перечислены номера тех определённых к итерации j масок, которые образуют подмножества маски, вырабатываемой в настоящий момент, и содержатся в описании, рассматриваемом в настоящий момент. Тогда обсуждаемый алгоритм можно записать в виде

$$\begin{aligned} \sigma_j(0) &= \{j\}, \quad j = 1, \dots, m; \\ \sigma_j(i+1) &= \begin{cases} \sigma_j(i) \cap P(i+1) \text{ тогда и только тогда,} \\ \text{когда } \eta(\sigma_j(i) \cap P(i+1)) > \theta + \eta\left(\bigcup_{k \in K_{i+1,j}} \sigma_k\right), \\ \sigma_j(i) \text{ в противном случае;} \end{cases} \\ \sigma_j &= \sigma_j(n). \end{aligned}$$

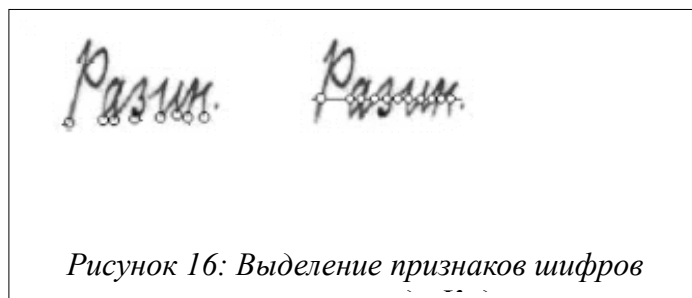
Из определения $\sigma_j(i+1)$ вытекает, что решение о модификации σ_j принимается в результате сравнения с переменным порогом, изменяемым так, чтобы учесть степень, в которой описание $P(i+1)$ уже было обусловлено другими признаками. Любые два признака σ_{j_1} и σ_{j_2} должны отличаться по крайней мере порогом θ .

Блок и др. показали, что если признаков действительно k и этого достаточно для восстановления всех описаний из выборки S , характеризуемого N_{\min} компонентами в наименьшей маске и не более N_{\max} общими компонентами для любых двух масок, то, если θ удовлетворяет соотношениям

$$\frac{1}{2}k(k-1)N_{\max} \leq \theta \leq N_{\min} - (k-1)N_{\max},$$

алгоритм построит соответствующие маски.

Пример из работ Кадакина. В работе [7] описан метод идентификации рукописных шифров источников на карточках архивной картотеки. На каждой карточке в специальном выделенном поле находится нанесённый от руки шифр источника, к которому относится информация на данной карточке. Для каждой строки шифра осуществлялось определение значений нескольких признаков. В роли этих признаков выступали количество чёрных точек на средней и нижней линии строки с учётом её возможного наклона (рис. 16), а также число верхних и нижних выступающих элементов букв. По полученным данным производилась оценка числа букв в словах. Число букв рассматривалось в качестве сложного признака более высокого уровня. Его значение, вместе со значениями ещё нескольких признаков, сопоставлялось со сведениями о имеющихся шифрах. На основе результатов сопоставления принималось решение об идентификации рассматриваемого изображения шифра с одним из шифров в базе данных.



Следует отметить принципиальное различие подходов, использованных авторами приведённых примеров. В первых двух примерах данного раздела рассматривались различные способы (эвристический и формальный) получения набора признаков, достаточного для решения задачи распознавания образов определённого класса, на основе анализа обучающей выборки объектов. В последнем же примере набор признаков предложен автором метода распознавания на основе эмпирических данных о рассматриваемых изображениях, полученных автором в процессе анализа предметной области.

4.3 Структурное описание

В рассмотренных ранее подходах объекты представлялись наборами чисел – результатами некоторого множества измерений, характеризующих объекты, и наборами выявленных признаков из некоторого заранее установленного множества. Распознавание образов проводилось при помощи разбиения параметрического пространства, либо пространства признаков на области. Развитие исследований по распознаванию образов в этом классе методов было большей частью связано с дискриминантным подходом и его применениями. Структурный подход применяется к задачам распознавания образов, в которых важна информация, описывающая структуру каждого объекта. А от процедуры распознавания требуется, чтобы она давала возможность не только отнести объект к определённому классу (классифицировать его), но и описать те стороны объекта, которые исключают его отнесение к другому классу. Типичным примером таких задач служит распознавание изображений, а если говорить шире, то в качестве примера можно привести задачу распознавания сцен, описанных не только одним из графических способов, но и вербальным или текстовым. Рассматриваемые в этом классе задач объекты обычно сложны, и число требуемых признаков часто велико. Это делает привлекательной идею описания сложного объекта в виде иерархической структуры более простых подобразов.

Для того чтобы представить иерархическую (древовидную) структурную информацию, содержащуюся в каждом образе, т.е. описывать образ при помощи более простых подобразов, а каждый подобраз снова описывать ещё более простыми подобразами и т.д., был предложен синтаксический, или структурный, подход. В одном из своих проявлений этот подход основан на аналогии между структурой образов (иерархической или древовидной) и синтаксисом языков. В рамках данной теории считается, что образы строятся из соединённых различными способами подобразов, так же как фразы и предложения строятся путём соединения слов, а слова соединяются из букв. В результате проведения данной аналогии получили развитие методы т.н. грамматического распознавания, т.е. методы, использующие аппарат теории формальных языков для описания и анализа образов. Другим проявлением структурного подхода к распознаванию является класс методов, основанных на структурных описаниях более общего типа, нежели иерархические — сетевых графах. В этом классе

методов используются такие структуры, как семантические и фреймовые сети, а рассуждения строятся на основе аппарата декларативной логики и теории графов.

Очевидно, что структурный подход полезен только в том случае, когда распознать выбранные простейшие подобразы, называемые производными элементами, легче, чем сами образы. Под производными элементами понимаются фрагменты распознаваемых образов, которые, с одной стороны, формируют эти образы, а с другой — просты в смысле собственной структуры, т.е. не содержат других производных элементов, сколь-нибудь значимых для описания образа. Аналогами производных элементов в распознавании образов являются терминальные символы грамматик формальных языков.

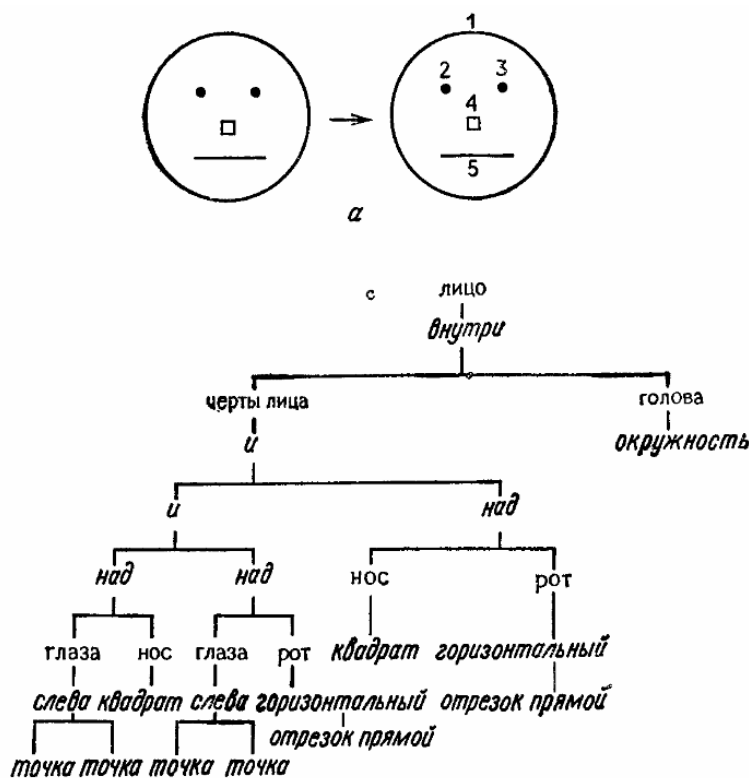


Рисунок 17: Пример структурного иерархического описания образа (изображения)

Как правило, системы, использующие структурное описание объектов, реализуют последовательную процедуру распознавания. В процессе решения информационной задачи они обрабатывают входной образ, обходя его структуру элемент за элементом. Систему структурного распознавания образов можно считать состоящей из трёх основных частей, а именно: из блока предобработки, блока формирования структурного описания или представления объекта и блока структурного анализа. На рисунке 18 показана функциональная схема системы структурного распознавания, аналогичная схеме общей системы распознавания из раздела (рис. 1). Блок предобработки осуществляет, во-первых, функции кодирования и аппроксимации и, во-вторых, функции фильтрации, восстановления и улучшения объекта. Исходный объект сначала кодируется или аппроксимируется так, чтобы это было удобно для дальнейшей обработки. Черно-белое изображение, например, можно кодировать при помощи решётки (или матрицы), состоящей из нулей и единиц, а кривую можно аппроксимировать отсчётами в дискретные моменты времени или конечным набором коэффициентов Фурье. Затем используют методы фильтрации, восстановления и (или) улучшения для ликвидации шума, восстановления искажений и (или) улучшения

качества закодированных (или аппроксимированных) объектов. Обычно предполагают, что на выходе препроцессора имеются объекты относительно «хорошего качества».

Каждый подвергнутый предобработке объект затем представляют в виде структуры языкового типа. Этот процесс представления объекта состоит, во-первых, из сегментации и, во-вторых, из выделения производных элементов. Чтобы представить объект через его подобразы, нужно этот объект сегментировать и в то же время идентифицировать (или выделить) в нем производные элементы. Другими словами, каждый подвергнутый предобработке объект разделяется на части и производные элементы на основе заранее заданных синтаксических операций. Каждая выделенная часть в свою очередь идентифицируется относительно заданного множества производных элементов. На этой стадии каждый объект получает своё представление через множество производных элементов и определённые синтаксические операции.

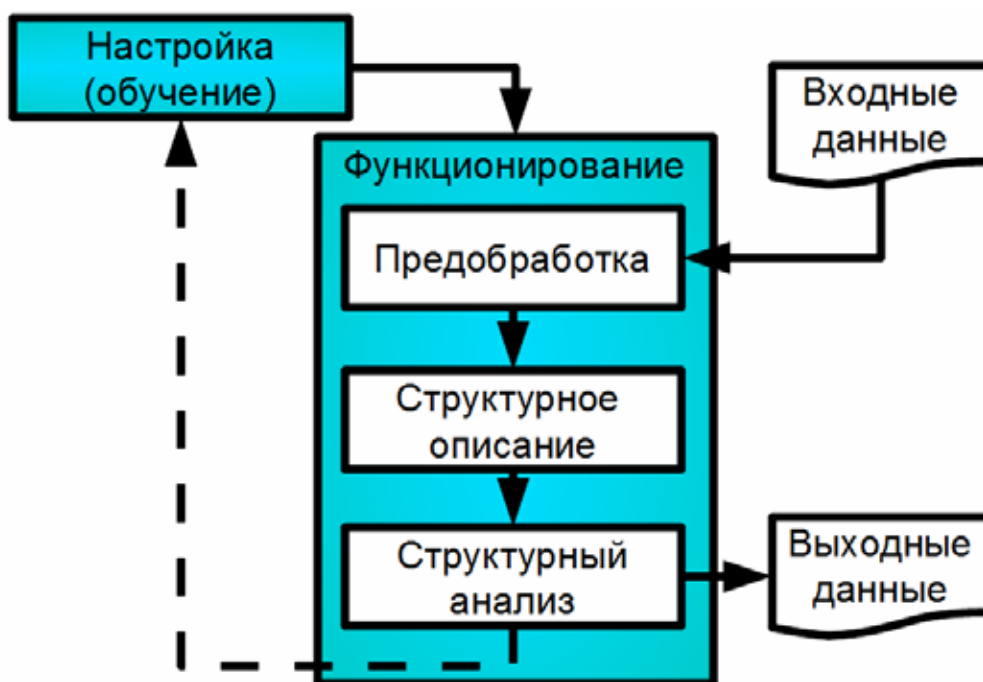


Рисунок 18: Функциональная схема системы структурного распознавания

Решение о том, является ли представление объекта синтаксически правильным (т.е. принадлежит ли он классу образов, описываемых данным синтаксисом или данной грамматикой), принимается «блоком синтаксического анализа» или «блоком грамматического разбора». По ходу синтаксического анализа или грамматического разбора этот блок обычно может давать полное синтаксическое описание объекта в терминах грамматических единиц или дерева грамматического разбора, если представление объекта синтаксически правильно. В противном случае объект либо исключают из рассмотрения, либо анализируют на основе других заданных грамматик, которые, быть может, описывают другие возможные классы рассматриваемых образов.

Простейшей формой распознавания является, по-видимому, «сравнение с эталоном». Цепочка производных элементов, представляющая исходный объект, сравнивается с цепочками производных элементов, представляющих каждый эталонный образ. Из этих образов выбирается тот, который наиболее «согласован» или «близок» к исходному объекту в соответствии с некоторым заданным критерием. При этом иерархическая, структурная

информация игнорируется. С другой стороны, при полном грамматическом разборе цепочки, представляющей исходный объект, исследуется полное структурное описание. Между этими крайностями существует ряд промежуточных подходов. Например, могут быть сконструированы последовательности тестов для проверки наличия определённых подобразов либо определённых комбинаций подобразов (или непроеизводных элементов). Результаты тестов, полученные, например, с помощью просмотра таблицы, дерева решений или логических операций, используются для классификации. Каждый тест может быть схемой сравнения с эталоном или грамматическим разбором поддерева, представляющего подобраз. Выбор подхода зависит от задачи. Если для распознавания требуется полное описание объекта, то необходим грамматический разбор. В противном случае можно обойтись без полного грамматического разбора и для увеличения эффективности использовать более простые подходы.

Для того чтобы получить грамматику, описывающую структурную информацию класса образов, необходимо устройство вывода грамматики, позволяющее восстановить её по заданному множеству объектов в виде, подобном языковому. Функции этого устройства соответствуют блоку «обучения» в приведённой схеме системы распознавания образов. В результате обучения на примерах объектов рассматриваемого класса формируется структурное описание этого класса. Затем полученное в грамматическом или ином виде описание используется для описания образов и синтаксического анализа. По-видимому, для нахождения наилучшего множества непроеизводных элементов и соответствующего структурного описания класса рассматриваемых образов необходимо обучение более общего вида.

Первый этап построения структурной модели образов состоит в определении множества непроеизводных элементов, при помощи которых можно эти образы описать. Это определение существенно зависит от природы образов, от специфики рассматриваемой области применения и от характера доступных технологических средств. Общего решения проблемы выбора непроеизводных элементов пока не найдено. Обычно при этом выборе стараются выполнить следующие требования:

1. Нпроеизводные элементы должны служить основными элементами образов и обеспечивать адекватное и сжатое описание исходных данных в терминах заданных структурных отношений (например, отношение конкатенации).

2. Поскольку структурная информация в непроеизводных элементах не важна, и они считаются простыми и компактными образами, их выделение и распознавание должны легко осуществляться существующими несинтаксическими методами.

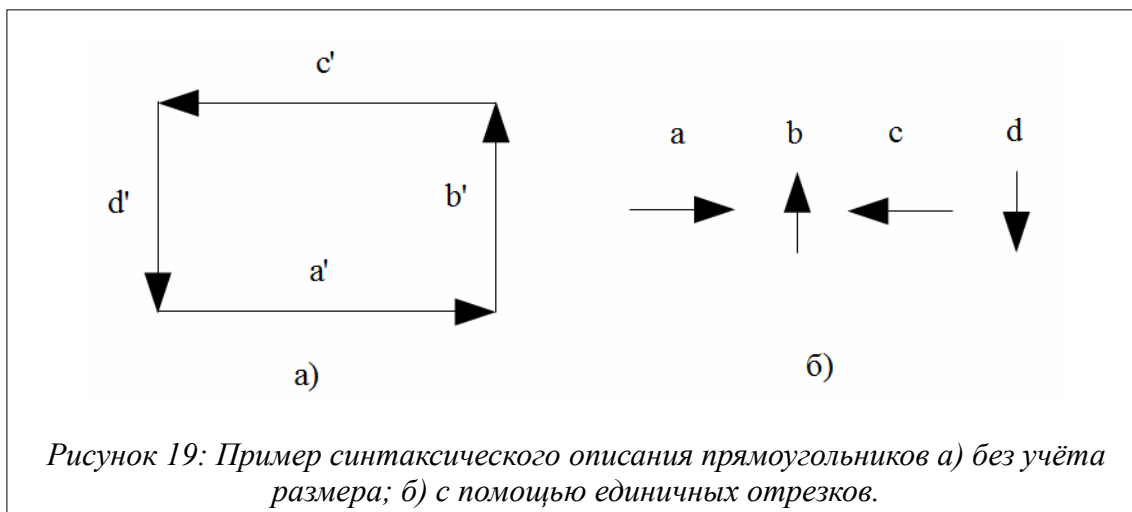
Для речевых образов и для отношения конкатенации «хорошим» множеством непроеизводных элементов считают совокупность фонем. Для описания рукописного текста в качестве непроеизводных элементов было предложено использовать штрихи. Однако вообще для изображений не найдено «универсальных» элементов, аналогичных штрихам в рукописном тексте и фонемам в речи. В ряде случаев для обеспечения адекватного описания образов требуют, чтобы непроеизводные элементы содержали информацию, важную для данного конкретного приложения. Например, если в задаче распознавания важен размер, или форма, или положение, то в непроеизводных элементах должна быть информация о размере (форме или положении), чтобы классы объектов были различимы при любом методе анализа описаний. Разные постановки задачи, даже при одних и тех же исходных данных, обуславливают выбор разных непроеизводных элементов.

Пример. Пусть необходимо отличать прямоугольники (разного размера) от непрямоугольников. Тогда выбирают следующие непроеизводные элементы:

- a' – горизонтальный отрезок;
- b' – вертикальный отрезок;
- c' – горизонтальный отрезок;

d' – вертикальный отрезок.

Множество всех прямоугольников (разного размера) можно представить одним предложением или цепочкой $a'b'c'd'$ (рис. 19а).

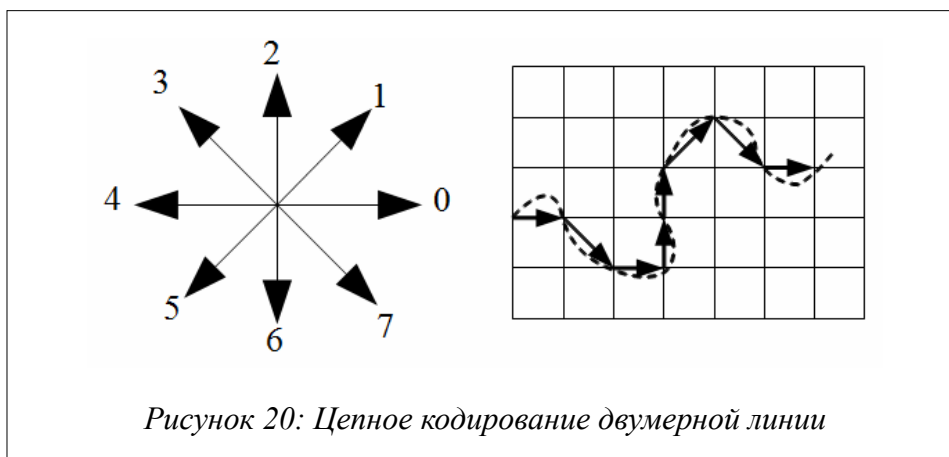


Если задача состоит в различении прямоугольников разного размера, то это описание не годится. В этом случае в качестве производных элементов нужно выбрать отрезки единичной длины (рис. 19б). Тогда множество прямоугольников разного размера можно описать при помощи языка: $L = \{a^n b^m c^n d^m, n, m = 1, 2, \dots\}$.

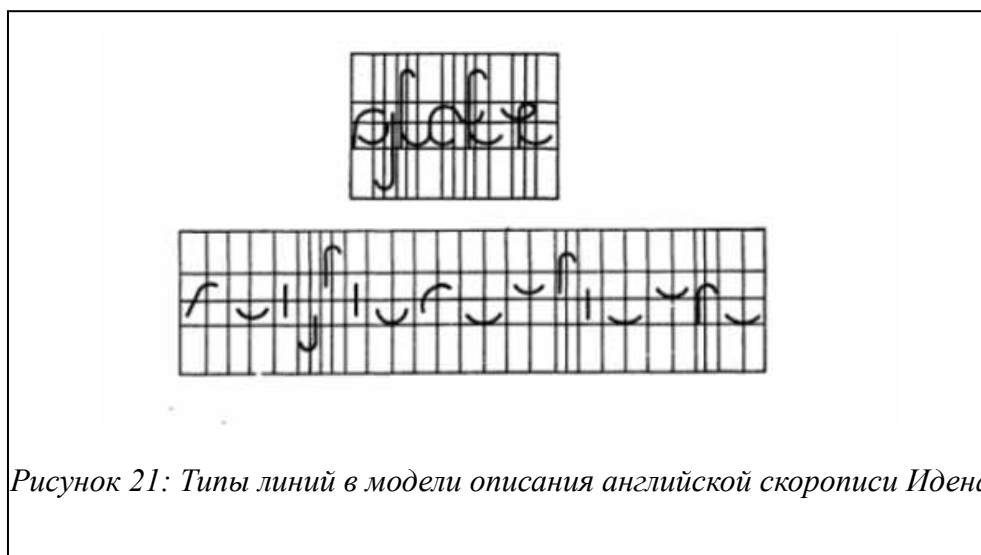
Требование 2 часто вступает в противоречие с требованием 1, поскольку выбранные в соответствии с требованием 1 производные элементы не всегда легко распознать при помощи соответствующих методов. С другой стороны, в соответствии с требованием 2 можно выбирать довольно сложные производные элементы, лишь бы они были доступны распознаванию. Чем сложнее производные элементы, тем проще могут быть структурные описания, т.е. они могут укладываться в рамки простых грамматик. В реализации распознающей системы достижение компромисса между указанными требованиями может стать весьма важной задачей.

Пример. Множество производных элементов, которые обычно используют для описания границ и остовов изображений, получают по схеме цепного кодирования, предложенной Фрименом [19]. На двумерное изображение накладывают прямоугольную сетку, и узлы сетки, которые наиболее близки к точкам изображения, соединяют отрезками прямых. Каждому такому отрезку в соответствии с наклоном присваивают восьмеричное число. Таким образом, изображение представляется цепью (последовательностью) или цепями восьмеричных чисел.

На рис. 20 показано множество начальных элементов и кодовая цепочка, описывающая кривую кодом 07022170. Эта схема обладает рядом полезных свойств. Например, поворот изображения на угол, кратный 45° , сводится к прибавлению восьмеричного числа (сложение по модулю 8) к каждому числу цепочки. Конечно, при этом изображение может исказиться. Только поворот на угол, кратный 90° , никогда не приводит к искажениям изображения. Изменяя зернистость сетки, накладываемой на изображение, можно получить любое желаемое разрешение. Этот метод не ограничен изображениями с односвязными замкнутыми границами. Его можно применять для описания произвольных двумерных фигур, составленных из прямых и кривых линий и отрезков.



В работе Идена [22] приводится модель английской скорописи. В результате анализа английских скорописных текстов был выявлен ограниченный набор типов линий, с помощью комбинации которых можно изобразить любую скорописную букву (рис. 21). В качестве структурных элементов изображений могут использоваться траектории движения пишущего или рисующего инструмента, описываемые им при начертании того или иного элемента символов. В таком случае эти траектории могут задаваться описанными выше цепными кодами.



4.3.1 Грамматические методы распознавания

Одним из видов структурных методов являются методы грамматического распознавания. Они используют представление образов в виде предложений специального языка. «Язык», который обеспечивает структурное описание образов в терминах множества производных элементов и операций композиции этих элементов, называют «языком описания образов». Правила композиции производных элементов обычно задают при помощи грамматики языка описания образов. Процесс распознавания осуществляется после идентификации в объекте производных элементов и составлении описания объекта. Распознавание состоит в синтаксическом анализе, или грамматическом разборе, «предложения», описывающего данный объект. Эта процедура устанавливает, является ли это предложение синтаксически (или грамматически) правильным по отношению к заданной

грамматике. Параллельно синтаксический анализ даёт некоторое структурное описание предложения (обычно в виде древовидной структуры). Синтаксический подход к распознаванию образов даёт возможность описывать большое множество сложных объектов путём использования небольшого множества непроеизводных элементов и грамматических правил. Грамматическое правило (правило подстановки) может быть применено любое число раз, так что оказывается возможным очень компактно выразить некоторые основные, структурные характеристики бесконечного множества предложений. Одним из наиболее привлекательных аспектов этой возможности является использование рекурсивной природы грамматик. Практическая полезность такого подхода зависит, конечно, от способности распознавать непроеизводные элементы образов и их взаимные отношения, выраженные операциями композиции. Различные отношения, определённые между подобразами, или операции композиции, обычно могут быть выражены логическими и (или) математическими операциями. Если, например, в качестве единственного отношения (операции композиции) для описания образов выбрать «конкатенацию», то при непроеизводных элементах, показанных на рисунке 19б, некоторый прямоугольник будет представлен цепочкой $aaabbbccdd$. Точнее, если использовать знак плюс для обозначения операции «конкатенация начала одного элемента к концу другого», то прямоугольник будет представлен предложением $a+a+a+b+b+c+c+c+d+d$. Использование древовидных структур позволяет непосредственно приспособить методы теории формальных языков к задаче компактного представления и анализа образов, имеющих существенное структурное содержание.

Допустим, что рассматриваемая задача выделения непроеизводных элементов может быть удовлетворительно решена. Следующий шаг состоит в построении грамматики, порождающей язык для описания исследуемых образов. При построении структурного описания объекта, т.е. при порождении соответствующего языка в рамках выбранной или заданной грамматики, исходным элементом служит начальный символ I , входящий в множество вспомогательных символов W . Правило, принадлежащее набору правил R данной грамматики, позволяет преобразовать начальный символ I в цепочку символов, например, $I \rightarrow aA$, где $a \in V$, $A \in W$, что соответствует замене символа I символами aA . Некоторые символы этой цепочки могут входить в множество вспомогательных символов W , а другие – принадлежать множеству непроеизводных элементов V . Вспомогательные символы, входящие в преобразованную цепочку, могут снова подвергнуться преобразованиям в соответствии с правилами грамматики, а вспомогательные символы, появляющиеся в цепочках, полученных в результате очередного преобразования, подвергаться преобразованиям и т.д. Предложение (язык) считается сформированным, если представляющая его цепочка содержит только непроеизводные элементы.

Более строго, порождающая грамматика G есть четвёрка

$$G = (V_N, V_T, P, S),$$

где V_N – множество вспомогательных символов;

V_T – множество непроеизводных элементов;

P – конечное множество правил вывода или правил подстановки;

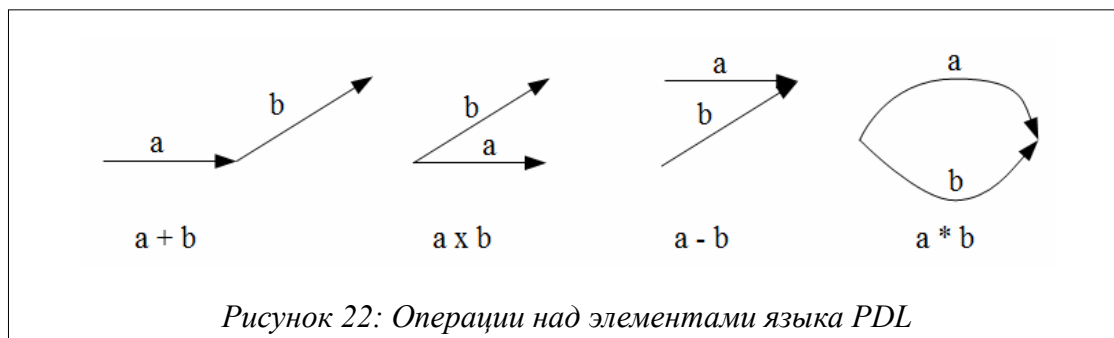
S – начальный символ ($I \in V_N$).

Язык, порождаемый грамматикой G , есть

$$L(G) = \{x \mid x \in V_T^*, S \Rightarrow^* x\}$$

Пример. Рассмотрим в качестве примера простую грамматику, построенную в терминах языка описания изображений PDL (Picture Definition Language), разработанного Шоу [21]. Нпроеизводными элементами данного языка служат двумерные структуры произвольного вида, имеющие выделенные хвостовую и головную точки. Обобщённо можно представлять любую двумерную структуру с заданными точками в виде направленного

отрезка. Элементы могут прилегать друг к другу только своими хвостовыми или головными точками. В соответствии с этим правилом в языке PDL вводятся 4 операции над элементами, проиллюстрированные на рис. 22.



Рассмотрим грамматику G , где:

$$V_N = \{S, A_1, A_2, A_3, A_4, A_5\}$$

$$V_T = \{a, b, c, d\}$$

P :

$$S \rightarrow d + A_1$$

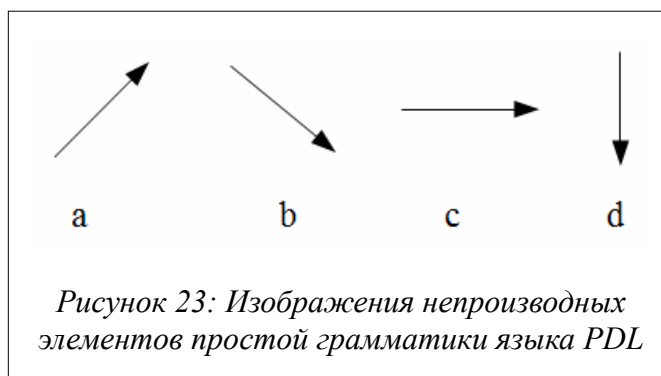
$$A_1 \rightarrow c + A_2$$

$$A_2 \rightarrow -d * A_3$$

$$A_3 \rightarrow a + A_4$$

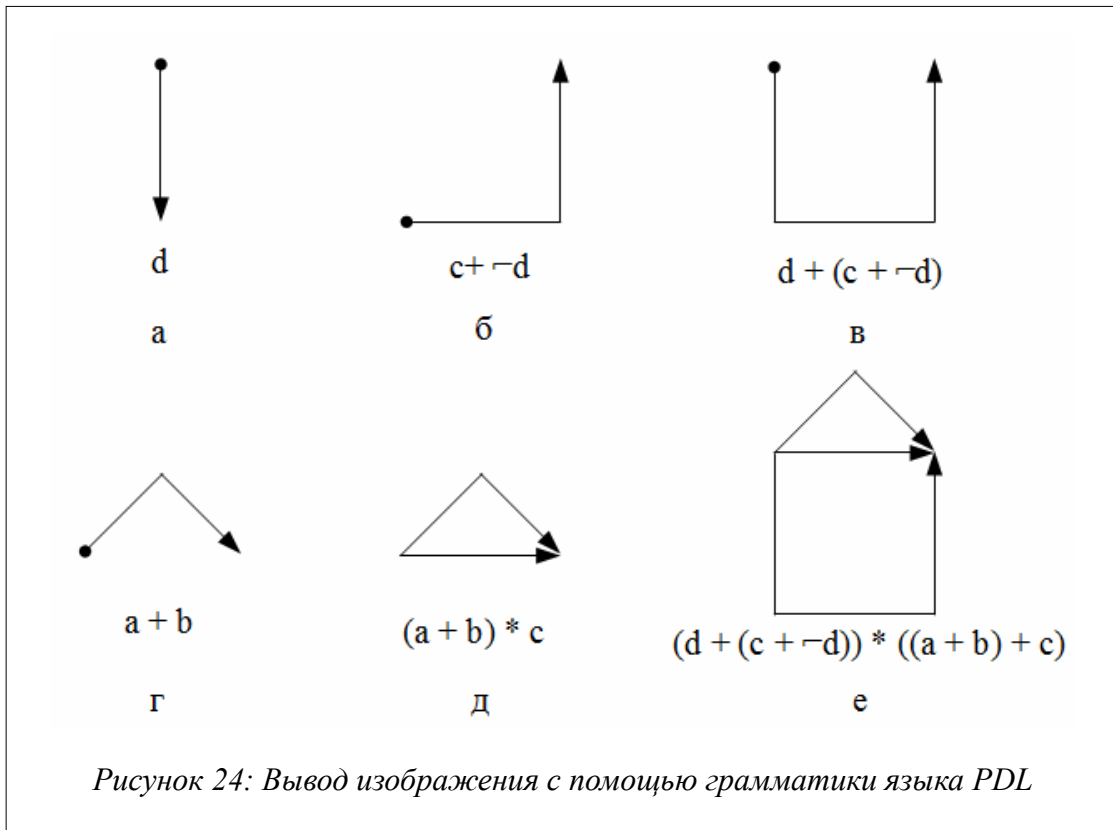
$$A_4 \rightarrow b * A_5$$

$$A_5 \rightarrow c$$



Причём неприводимые элементы a , b , c и d внешне выглядят, как показано на рисунке 23, а выражение $-d$ означает перемену мест хвостовой и головной точек элемента d .

Применение первого правила приводит к появлению неприводимого элемента d , сопровождаемого неопределённой ещё фигурой A_1 . Известно только то, что полученный элемент d будет связан с неизвестной фигурой своей головной точкой, т.к. применённое правило содержит оператор «+». Переменная A_1 разлагается на $c + A_2$, причём A_2 пока не определена. Аналогично из A_2 получаем $-d * A_3$. Результаты применения первых трёх правил показаны на рисунке 24абв. По определению оператора «*» при разложении элемента A_3 происходит его присоединение к структуре, показанной на рисунке 24в, путём соединения головных точек с головными, а хвостовых — с хвостовыми. Оставшиеся шаги вывода показаны на рисунке 24где.



Описанная выше грамматика PDL способна порождать только одну структуру. Можно, однако, расширить число порождаемых структур введением в правила подстановки рекурсивности — способности переменной замещаться этой же переменной. Например:

$$\begin{aligned}
 S &\rightarrow d + A_1 \\
 A_1 &\rightarrow c + A_1 \\
 A_1 &\rightarrow \neg d * A_2 \\
 A_2 &\rightarrow a + A_2 \\
 A_2 &\rightarrow b * A_2 \\
 A_2 &\rightarrow c
 \end{aligned}$$

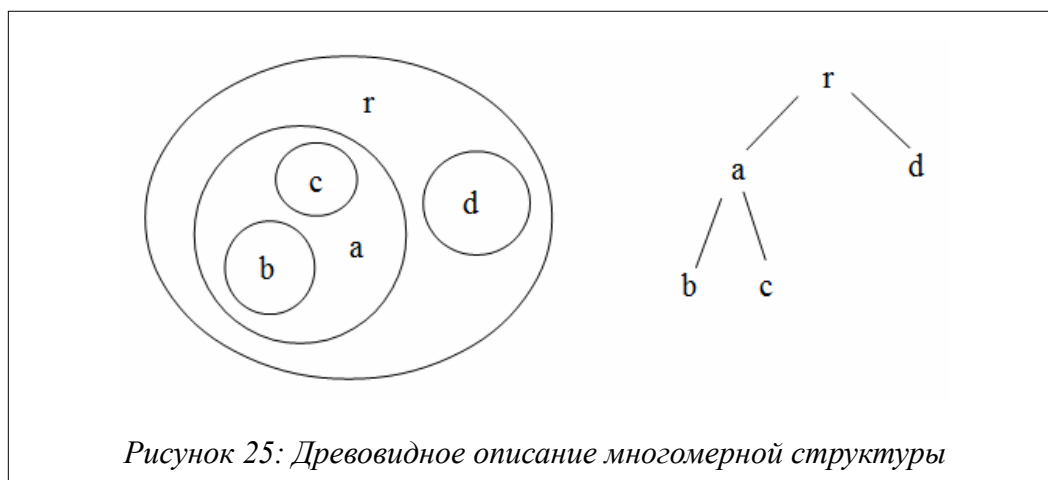
Эти правила позволяют порождать бесконечные структуры путём многократного замещения переменной этой же переменной. Многообразие порождаемых данной грамматикой структур может быть ещё более расширено, если положить, что A_1 и A_2 равны S . При этом возможности грамматики возрастут до максимума. Нужно, однако, заметить, что подобное увеличение порождающей способности грамматики может иногда оказаться нежелательным. Чрезмерное многообразие приводит к уменьшению различающей мощности грамматик при описании распознаваемых объектов.

При описании образов цепочками используется лишь одно отношение – конкатенация. Это означает, что каждый подобраз или производный элемент можно сочленять с остальными только один раз слева. Для описания двухмерных или трёхмерных образов это одномерное отношение недостаточно эффективно. Естественно применить более общий формализм, включающий другие полезные отношения.

Пусть R – множество n -арных отношений ($n > 1$). Тот факт, что производные элементы x_1, x_2, \dots, x_n удовлетворяют отношению $r \in R$, обозначают через $r(x_1, x_2, \dots, x_n)$.

Например, запись *ТРЕУГОЛЬНИК*(a, b, c) означает, что три отрезка a, b и c , удовлетворяют тернарному отношению *ТРЕУГОЛЬНИК*, а запись *НАД* (X, Y) означает, что элемент X находится над элементом Y . Например, математическое выражение $\frac{a+b}{c}$ может быть описано предложением : *НАД*(*СЛЕВА*($a, +, b$), $-$, c), где запись *СЛЕВА* (X, Y, Z) означает, что элемент X находится слева от элемента Y , который, в свою очередь, находится левее элемента Z .

Использование деревьев для описания многомерных структур довольно простая процедура. В сущности, любая иерархическая упорядоченная схема ведёт к представлению объекта в виде дерева. Например, пусть упорядочение состоит в группировке областей, показанных на рис. 25. Обозначив корень дерева символом r , получаем древовидную структуру:

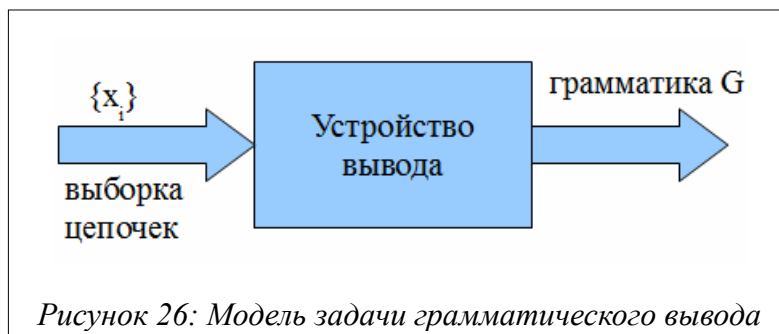


Описав методику грамматического описания распознаваемых образом, сформулируем постановку задачи грамматического распознавания образов. Пусть имеются два класса распознаваемых объектов ω_1 и ω_2 и образы этих классов могут быть построены из неприводимых элементов некоторого конечного множества V_T . Каждый образ может рассматриваться как цепочка или предложение некоторого языка описания образов над алфавитом V_T . Предположим, что существует грамматика G , такая, что предложения порождаемого ей языка принадлежат исключительно одному из классов, скажем, ω_1 . Очевидно, что эта грамматика может быть использована в целях распознавания образов, т.к. заданный образ неизвестной природы может быть отнесён к классу ω_1 , если он является предложением языка $L(G)$. В противном случае образ приписывается классу ω_2 . Процедура, используемая для определения, является или не является цепочка предложением, грамматически правильным для данного языка, называется *грамматическим разбором*.

В случае, если в результате грамматического разбора объект не отнесён к классу ω_1 , он может не принадлежать при этом и классу ω_2 , а представлять собой искажённую или зашумлённую цепочку, которую, возможно, лучше всего изъять из рассмотрения. При такой постановке задачи требуются две грамматики — G_1 и G_2 . Расознаваемый образ зачисляется в тот класс, грамматика которого признаёт этот образ предложением соответствующего класса языка. Если образ не является предложением ни одного из языков или, напротив, принадлежит обоим языкам, он исключается из рассмотрения или относится произвольно к любому классу. Аналогично, для случая $M > 2$ классов требуется M грамматик,

описывающих языки $L(G_i), i = 1, 2, \dots, M$ соответствующих классов.

В описанных ранее методах распознавания рассматривались процедуры построения решающих функций с помощью обучающих выборок. В контексте грамматического подхода к распознаванию данная задача может быть сформулирована как задача получения грамматики из множества выборочных предложений. Эта процедура, называемая обычно *грамматическим выводом* или *восстановлением грамматики*, составляет суть обучения системы грамматического распознавания. Модель задачи грамматического вывода показана на рисунке 26. На вход некоторого устройства вывода подаётся обучающая выборка цепочек $\{x_i\}$ некоторого языка. Осуществляя обработку этой выборки, устройство формирует и подаёт на выход грамматику, позволяющую строить все предъявленные устройству цепочки. Универсальные методы решения данной задачи, позволяющие решить её в общем виде, до настоящего времени являются объектом исследований. Предложен ряд методов для восстановления различных грамматик специального вида. Ниже будет рассмотрен один из методов построения автоматных грамматик по заданной выборке цепочек, основанный на процедуре Фельдмана [20].



Основная идея рассматриваемого метода состоит в том, чтобы сначала построить нерекурсивную грамматику, порождающую в точности предъявленные цепочки, а затем преобразовать её в рекурсивную, порождающую бесконечное количество цепочек. Решение задачи, таким образом, можно разделить на три шага: формирование нерекурсивной грамматики, преобразование её в рекурсивную, упрощение рекурсивной грамматики. Действие данного метода рассмотрим на примере.

Пусть имеется следующая выборка цепочек: $\{caaab, bbaab, caab, bbab, cab, bbb, cb\}$. Требуется получить автоматную грамматику, порождающую данные цепочки. Алгоритм вывода состоит из следующих шагов.

Шаг 1. Построение нерекурсивной грамматики. Выборочные цепочки обрабатываются в порядке уменьшения их длины. Автоматные правила подстановки вида $X \rightarrow zY, X, Y \in V_N, z \in V_T$ строятся и прибавляются к грамматике по мере рассмотрения очередного символа текущей цепочки в случае необходимости.

Первой цепочкой максимальной длины является цепочка $caaab$. Для её порождения строятся следующие правила подстановки:

$$S \rightarrow cA_1$$

$$A_1 \rightarrow aA_2$$

$$A_2 \rightarrow aA_3$$

$$A_3 \rightarrow ab$$

Для порождения второй цепочки $bbaab$ к имеющимся правилам добавляются следующие:

$$\begin{aligned}
S &\rightarrow bA_4 \\
A_4 &\rightarrow bA_5 \\
A_5 &\rightarrow aA_6 \\
A_6 &\rightarrow ab
\end{aligned}$$

Заметим, что в работе первого шага алгоритма в грамматику вносится некоторая избыточность. Например, вторая цепочка могла быть также получена добавлением только двух правил $S \rightarrow bA_4$, $A_4 \rightarrow bA_2$. Но на данном этапе важно получения лишь набора правил, порождающих в точности имеющиеся цепочки, а не оптимизация грамматики. В сложных случаях трудно соблюдать это требования, попутно минимизируя число правил грамматики. Устранение избыточности производится в третьей части алгоритма.

Для порождения третьей цепочки требуется добавление только одного правила:

$$A_3 \rightarrow b$$

Рассматривая все остальные цепочки приходим к следующему набору правил, требуемых для получения полного набора цепочек:

$$\begin{array}{ll}
S \rightarrow cA_1 & A_3 \rightarrow ab \\
S \rightarrow bA_4 & A_4 \rightarrow bA_5 \\
A_1 \rightarrow b & A_5 \rightarrow b \\
A_1 \rightarrow aA_2 & A_5 \rightarrow aA_6 \\
A_2 \rightarrow b & A_6 \rightarrow b \\
A_2 \rightarrow aA_3 & A_6 \rightarrow ab \\
A_3 \rightarrow b &
\end{array}$$

Шаг 2. Получение рекурсивной грамматики. Каждое остаточное правило длины 2 (правило вида $X \rightarrow yz$, $X \in V_N$, $y, z \in V_T$) соединяется с другим, неостаточным правилом. Это достигается путём слияния каждого нетерминального символа остаточного правила с нетерминальным символом неостаточного правила, который может порождать остаточную цепочку. Например, если A_r — остаточный нетерминал вида $A_r \rightarrow a_1a_2$ и A_n — неостаточный нетерминал вида $A_n \rightarrow a_1A_m$, где $A_m \rightarrow a_2$, все встречающиеся A_r заменяются на A_n , а правило $A_r \rightarrow a_1a_2$ отбрасывается. Таким образом создаётся рекурсивная автоматная грамматика, порождающая все цепочки обучающей выборки, а также обладающая общностью, достаточной для порождения бесконечного множества других цепочек.

В полученных правилах A_6 может сливаться с A_5 , а A_3 — с A_2 , образуя следующий набор правил:

$$\begin{array}{ll}
S \rightarrow cA_1 & A_2 \rightarrow b \\
S \rightarrow bA_4 & A_4 \rightarrow bA_5 \\
A_1 \rightarrow b & A_5 \rightarrow b \\
A_1 \rightarrow aA_2 & A_5 \rightarrow aA_5 \\
A_2 \rightarrow b & A_5 \rightarrow b \\
A_2 \rightarrow aA_2 &
\end{array}$$

Здесь рекурсивными являются правила $A_2 \rightarrow aA_2$ и $A_5 \rightarrow aA_5$.

Шаг 3. Упрощение грамматики. Упрощение производится путём объединения

эквивалентных правил подстановки. Два правила с левыми частями A_i и A_j считаются эквивалентными, если соблюдаются следующие условия. Предположим, что начиная с символа A_i можно породить множество цепочек $\{x\}_i$. Аналогично предположим, что из A_j выводится множество цепочек $\{x\}_j$. Если $\{x\}_i \equiv \{x\}_j$, то правила подстановки, левые части которых содержат символы A_i и A_j , считаются эквивалентными, и каждый символ A_j может быть заменён на A_i без ущерба для языка, порождаемого этой грамматикой. Формально, два правила эквивалентны, если $\{x | A_i \Rightarrow^* x\} \equiv \{x | A_j \Rightarrow^* x\}$, где $A_i \Rightarrow^* x$ обозначает, что цепочка x порождается из нетерминала A_i соответствующим применением правил подстановки за конечное число шагов.

В приведённом примере эквивалентны правила с левыми частями A_1 и A_2 . После их слияния получаем

$$\begin{array}{ll} S \rightarrow cA_1 & A_4 \rightarrow bA_5 \\ S \rightarrow bA_4 & A_5 \rightarrow b \\ A_1 \rightarrow b & A_5 \rightarrow aA_5 \\ A_1 \rightarrow aA_1 & \end{array}$$

Аналогично заметив, что эквивалентны A_1 и A_5 , получаем

$$\begin{array}{ll} S \rightarrow cA_1 & A_1 \rightarrow aA_1 \\ S \rightarrow bA_4 & A_4 \rightarrow bA_1 \\ A_1 \rightarrow b & \end{array}$$

Дальнейшее слияние правил невозможно, поэтому результирующей грамматикой, строящей заданную выборку цепочек, будет грамматика

$$G = \{V_N, V_T, P, S\},$$

где $V_N = \{S, A, B\}$, $V_T = \{a, b, c\}$

$$P: S \rightarrow cA$$

$$S \rightarrow bB$$

$$A \rightarrow aA$$

$$A \rightarrow bA$$

$$A \rightarrow b$$

Легко проверить, что данная грамматика может порождать цепочки обучающей выборки, использованные в процессе построения этой грамматики.

5 НЕЙРОСЕТЕВЫЕ МЕТОДЫ

Среди параллельных методов распознавания, использующих Евклидово пространство описаний образов, можно выделить широкий класс методов, заслуживающий отдельного рассмотрения — нейросетевые методы. В их основе лежат нейронные сети — вычислительные модели, принцип функционирования которых сходен с сетями биологических нейронов головного мозга. Благодаря заимствованию принципов организации

у биологических структур мозга, нейросети демонстрируют многие их свойства, такие как обучение на основе предыдущего опыта, извлечение существенных свойств из поступающей информации, обобщение имеющихся прецедентов на новые случаи. Возможности, предоставляемые нейронными сетями, были использованы для решения задач распознавания и классификации образов во множестве исследований и прикладных разработок. В данном разделе рассматриваются некоторые основные разновидности нейронных сетей, их возможности и способы обучения.

5.1 Обзор

В пятидесятые и шестидесятые годы XX века на основе биологических и физиологических исследований головного мозга были созданы первые искусственные нейронные сети. Выполненные первоначально как электронные сети, они были позднее перенесены в более гибкую среду компьютерного моделирования, сохранившуюся и в настоящее время. Первые успехи вызвали взрыв активности и оптимизма. Минский, Розенблатт, Уидроу и другие разработали сети, состоящие из одного слоя искусственных нейронов. Часто называемые перцептронами, они были использованы для такого широкого класса задач, как предсказание погоды, анализ электрокардиограмм и искусственное зрение. В течение некоторого времени казалось, что ключ к интеллекту найден и воспроизведение человеческого мозга является лишь вопросом конструирования достаточно большой сети.

Но эта иллюзия скоро рассеялась. Сети не могли решать задачи, внешне весьма сходные с теми, которые они успешно решали. В 1969 г. Минский показал [18], что используемые в то время однослойные сети теоретически неспособны решить многие простые задачи, в том числе реализовать функцию «Исключающее ИЛИ». Минский также не был оптимистичен относительно потенциально возможного здесь прогресса. Исследования нейронных сетей были оставлены ради более обещающих областей.

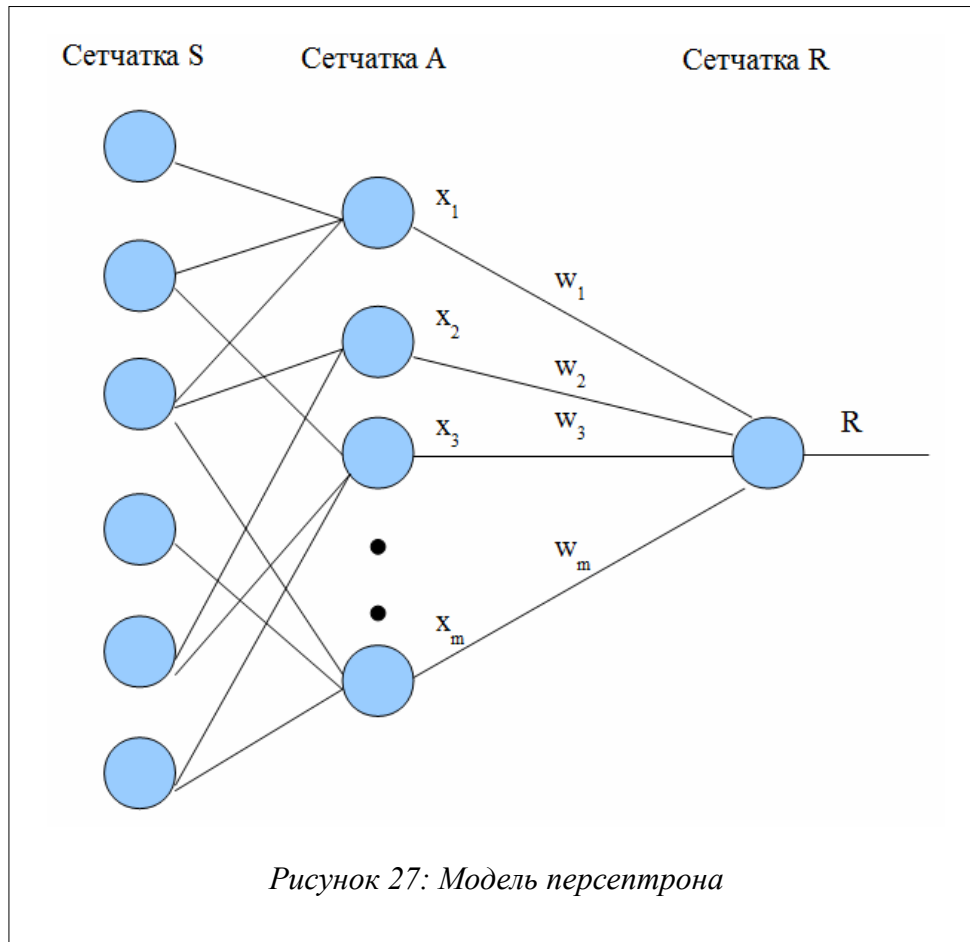
Тем не менее несколько наиболее настойчивых ученых, таких как Кохонен, Гроссберг, Андерсон продолжили исследования. Постепенно появился теоретический фундамент, на основе которого сегодня конструируются наиболее мощные многослойные сети. Оценка Минского оказалась излишне пессимистичной, многие из поставленных в его книге задач решаются сейчас сетями с помощью стандартных процедур.

5.2 Перцептрон

Перцептрон является простейшей нейронной сетью, состоящей из одного нейрона. Рассмотрим следующую простую задачу. Пусть имеется параметрическое пространство и набор объектов, описываемых в этом пространстве векторами $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Пусть необходимо разделить эти объекты на два класса ω_1 и ω_2 . Поставим задачу в виде отыскания решающей функции (см. раздел 0), разделяющей эти классы. Эта функция определяется весовым вектором $\mathbf{w} = (w_1, w_2, \dots, w_n)$, таким что

$$\begin{aligned} \forall \mathbf{x} \in \omega_1 \quad \mathbf{w}^T \mathbf{x} > 0 \\ \forall \mathbf{x} \in \omega_2 \quad \mathbf{w}^T \mathbf{x} < 0 \end{aligned}$$

Рассмотрим схему работы персептрона (рис. 27). Он состоит из трёх сетчаток сенсорных элементов: сенсорной (S), ассоциативной (A) и реакционной (R). Элементы сенсорной сетчатки случайным образом соединены с ассоциативными элементами. Каждый из ассоциативных элементов вырабатывает выходной сигнал x_i лишь в том случае, когда достаточное количество соединённых с ним сенсорных элементов находятся в возбуждённом состоянии. Таким образом, сенсорная сетчатка представляет собой некоторое устройство, воспринимающее сигналы из внешнего мира, а ассоциативная сетчатка — входное описание



воспринимаемого объекта.

Собственно нейрон представляется элементом в сетчатке R. Он соединён со всеми элементами ассоциативной сетчатки связями, каждая из которых обладает собственным весом w_i . Реакция элемента R на воспринимаемый объект представляется взвешенной суммой выходных сигналов ассоциативных элементов:

$$R = \sum_{i=1}^m w_i x_i = \mathbf{w}^T \mathbf{x}$$

Если $R > 0$, воспринимаемый объект считается отнесённым к классу ω_1 , если же $R < 0$ — к классу ω_2 .

Описанная схема работы персептрона соответствует принципу распознавания на основе решающей функции. Таким образом, процесс обучения системы состоит в определении весового вектора \mathbf{w} таким образом, чтобы система выдавала соответствующий результат R для объектов разных классов.

При рассмотрении нейронных сетей сенсорную сетчатку, как правило, не берут во внимание, т.к. суть процесса функционирования сети выражена в вычислении взвешенных сумм сигналов. По этой же причине сетчатка A обычно служит только для обозначения входов и называется *входным слоем* сети, или тоже опускается.

Алгоритм обучения персептрона проводится по схеме, называемой алгоритмом *персептрона*, и руководствующейся принципом *подкрепления-наказания*. Эта процедура относится к классу процедур *обучения с учителем*. Пусть $\mathbf{w}(1)$ — начальный весовой вектор. Системе поочерёдно предъявляются объекты из обучающей выборки с известными принадлежностями к классам. В зависимости от реакции системы производится коррекция весового вектора. Правило коррекции следующее: пусть на k -м шаге обучения имеем весовой вектор $\mathbf{w}(k)$ и объект $\mathbf{x}(k)$, принадлежащий классу ω_1 . Тогда если $\mathbf{w}^T(k)\mathbf{x}(k) \leq 0$, вектор $\mathbf{w}(k)$ на шаге $k+1$ заменяется вектором

$$\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{x}(k),$$

где c — корректирующее приращение или скорость обучения персептрона. Теперь предположим, что $\mathbf{x}(k) \in \omega_2$. Тогда если $\mathbf{w}^T(k)\mathbf{x}(k) \geq 0$, то весовой вектор на шаге $k+1$ заменяется вектором

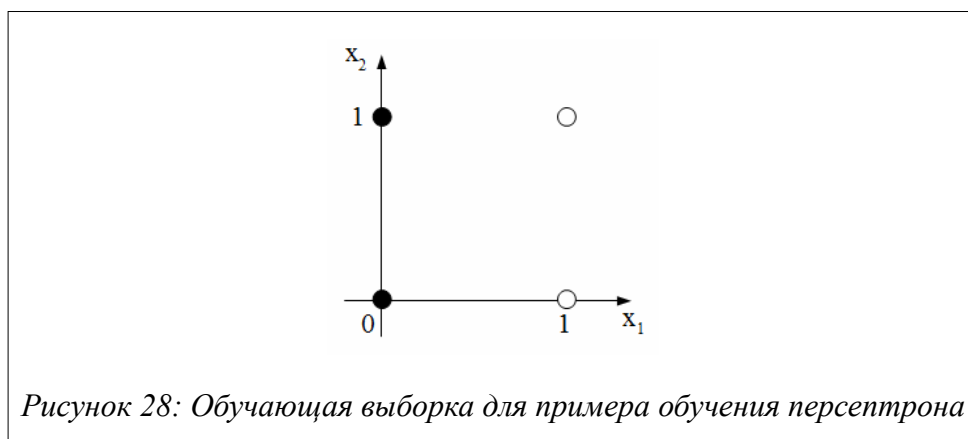
$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{x}(k)$$

В случае невыполнения указанных условий весовой вектор не изменяется. Таким образом, в случае неправильного распознавания класса предъявляемого объекта используемый вектор «наказывается», т.е. корректируется. В случае правильной реакции вектор «подкрепляется», что в данном алгоритме выражается отсутствием изменений. Выполнение алгоритма заканчивается при правильном распознавании всех объектов обучающей выборки. В [3] приводится доказательство того, что если распознаваемые классы линейно разделимы в параметрическом пространстве, то данный алгоритм сходится.

Приведём пример обучения персептрона по описанному алгоритму. Пусть имеется обучающая выборка объектов, имеющих двумерные описания:

$$\omega_1 = \{(0,0)^T, (0,1)^T\}$$

$$\omega_2 = \{(1,0)^T, (1,1)^T\}$$



На рисунке 28 обучающая выборка представлена графически. По рисунку видно, что классы линейно разделимы, следовательно, описанный алгоритм применим для обучения.

Прежде всего пополним описания объектов следующим образом:

$$\omega_1 = \{(0,0,1)^T, (0,1,1)^T\}$$

$$\omega_2 = \{(1,0,1)^T, (1,1,1)^T\}$$

Зададим $c=1$ и $\mathbf{w}(1)=(0,0,0)$ и проведём первую серию тестов над объектами обучающей выборки (в порядке их описания выше). Проверка первого объекта $(0,0,1)^T$:

$$\mathbf{w}^T(1)\mathbf{x}(1) = (0,0,0) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0,$$

$$\mathbf{w}(2) = \mathbf{w}(1) + c\mathbf{x}(1) = \mathbf{w}(1) + \mathbf{x}(1) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Так как проверка распознавания первого объекта дала ошибку (результат не больше 0), весовой вектор был скорректирован на величину $c\mathbf{x}(1)$, и далее будет использоваться скорректированное значение. (Т.к. $c=1$, будем опускать этот множитель). Проверим второй объект:

$$\mathbf{w}^T(2)\mathbf{x}(2) = (0,0,1) \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = 1,$$

$$\mathbf{w}(3) = \mathbf{w}(2) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Умножение весового вектора на вектор описания объекта дало положительный результат, следовательно, весовой вектор не изменяется. Аналогично для двух оставшихся объектов:

$$\mathbf{w}^T(3)\mathbf{x}(3) = (0,0,1) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = 1, \quad \mathbf{w}(4) = \mathbf{w}(3) - \mathbf{x}(3) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix};$$

$$\mathbf{w}^T(4)\mathbf{x}(4) = (-1,0,0) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1, \quad \mathbf{w}(5) = \mathbf{w}(4) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix};$$

Корректировка весового вектора происходила на шагах 1 и 3. Для завершения алгоритма необходимо, чтобы весовой вектор не изменялся ни на одном из шагов серии тестов. Это означает, что обучающую выборку следует предъявить ещё раз:

$$\mathbf{w}^T(5)\mathbf{x}(1) = (-1,0,0) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0, \quad \mathbf{w}(6) = \mathbf{w}(5) + \mathbf{x}(1) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix};$$

$$\mathbf{w}^T(6)\mathbf{x}(2) = (-1,0,1) \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = 1, \quad \mathbf{w}(7) = \mathbf{w}(6) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix};$$

$$\mathbf{w}^T(7)\mathbf{x}(3) = (-1,0,1) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = 0, \quad \mathbf{w}(8) = \mathbf{w}(7) - \mathbf{x}(3) = \begin{pmatrix} -2 \\ 0 \\ 0 \end{pmatrix};$$

$$\mathbf{w}^T(8)\mathbf{x}(4) = (-2,0,0) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -2, \quad \mathbf{w}(9) = \mathbf{w}(8) = \begin{pmatrix} -2 \\ 0 \\ 0 \end{pmatrix}.$$

При втором предъявлении обучающей выборки также имели место корректировки весов, следовательно, выполняется повторное предъявление:

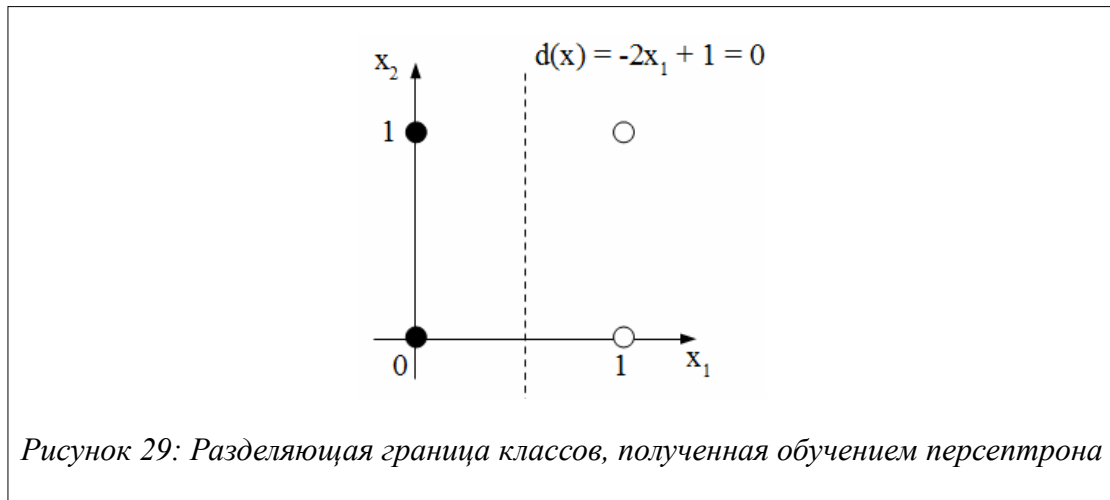
$$\mathbf{w}^T(9)\mathbf{x}(1) = (-2,0,0) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0, \quad \mathbf{w}(10) = \mathbf{w}(9) + \mathbf{x}(1) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix};$$

$$\mathbf{w}^T(10)\mathbf{x}(2) = (-2,0,1) \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = 1, \quad \mathbf{w}(11) = \mathbf{w}(10) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix};$$

$$\mathbf{w}^T(11)\mathbf{x}(3) = (-2,0,1) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1, \quad \mathbf{w}(12) = \mathbf{w}(11) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix};$$

$$\mathbf{w}^T(12)\mathbf{x}(4) = (-2,0,1) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1, \quad \mathbf{w}(13) = \mathbf{w}(12) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}.$$

Совершена одна ошибка распознавания и весовой вектор скорректирован. Нетрудно убедиться, что при следующем предъявлении обучающей выборки ошибок допущено не будет. Таким образом, весовым вектором обученного персептрона является вектор $(-2,0,1)^T$. Этому вектору будет соответствовать решающая функция $d(\mathbf{x}) = -2x_1 + 1$, приравнивание нулю которой даст разделяющую границу классов, показанную на рисунке 29:



В приведённом примере использовалась процедура т.н. *фиксированного приращения* весового вектора. Для его коррекции использовалась фиксированная константа c . В качестве альтернативной дисциплины можно рассмотреть алгоритм *коррекции абсолютной величины*. В этом случае величина коррекции c выбирается на каждом шаге так, чтобы гарантировать правильность распознавания текущего объекта скорректированным весовым вектором, т.е. чтобы выполнялось

$$\mathbf{w}^T(k+1)\mathbf{x}(k) = [\mathbf{w}(k) + c\mathbf{x}(k)]^T \mathbf{x}(k) > 0$$

Одним из способов выбора величины c , удовлетворяющей данному условию, является выбор наименьшего целого числа, превышающего $\mathbf{w}^T(k)\mathbf{x}(k) / \mathbf{x}^T(k)\mathbf{x}(k)$.

Другой подход к выбору величины коррекции является процедура *дробной коррекции*. Значение c выбирается так, чтобы величина приращения $|\mathbf{w}^T(k)\mathbf{x}(k) - \mathbf{w}^T(k+1)\mathbf{x}(k)|$ составляла некоторую долю λ от величины $|\mathbf{w}^T(k)\mathbf{x}(k)|$, т.е.

$$|\mathbf{w}^T(k)\mathbf{x}(k) - \mathbf{w}^T(k+1)\mathbf{x}(k)| = \lambda |\mathbf{w}^T(k)\mathbf{x}(k)|$$

Подставляя в это уравнение $\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{x}(k)$, получим

$$c = \lambda \frac{|\mathbf{w}^T(k)\mathbf{x}(k)|}{\mathbf{x}^T(k)\mathbf{x}(k)}$$

Приведённые алгоритмы коррекции являются по сути вариациями более общего метода градиентного спуска. Пусть имеется некоторая функция критерия $J(\mathbf{w}, \mathbf{x})$, имеющая единственный минимум при таком значении \mathbf{w} , что выполняется $\mathbf{w}^T \mathbf{x}_i > 0$ для всех i . Например, такой функцией может являться функция

$$J(\mathbf{w}, \mathbf{x}) = |\mathbf{w}^T \mathbf{x}| - \mathbf{w}^T \mathbf{x}$$

Тогда коррекция весового вектора должна осуществляться в направлении отрицательного градиента этой функции:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c \left\{ \frac{\delta J(\mathbf{w}, \mathbf{x})}{\delta \mathbf{w}} \right\}_{\mathbf{w}=\mathbf{w}(k)},$$

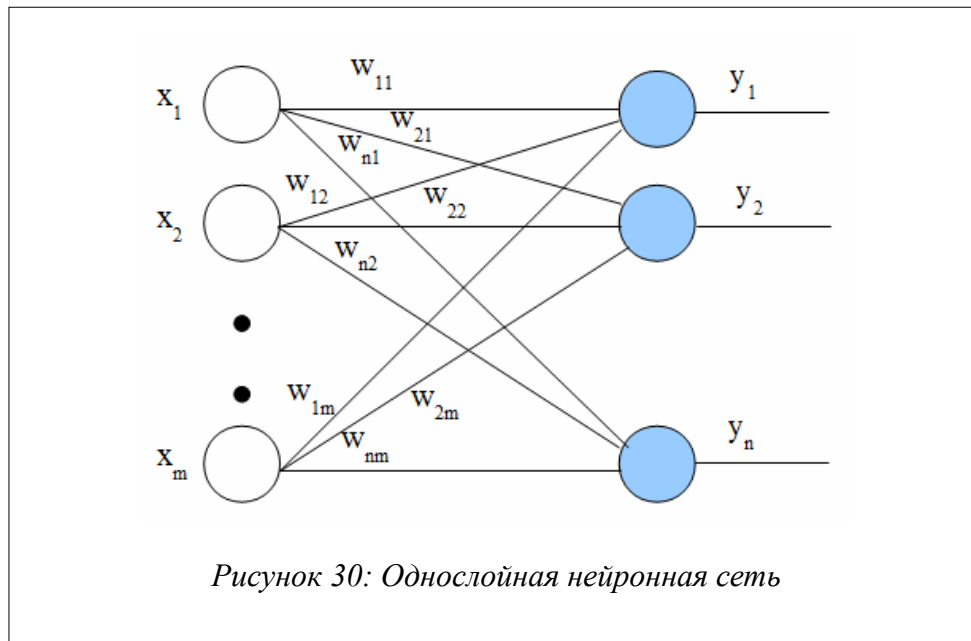
что ведёт к стремлению функции к её минимуму при значении константы $c > 0$. При достижении минимума величина градиента приравняется нулю и коррекция прекращается.

Рассмотренный пример представляет класс процедур обучения, называемых *детерминистскими*. Их характеризует определённая величины и направления коррективы на каждой итерации процесса. В противовес детерминистским процедурам используются *стохастические* процедуры. Их основной идеей является применение случайных изменений к параметрам правил классификации и фиксации тех из них, которые приводят к улучшению показателей распознавания. Случайный характер изменений позволяет преодолевать трудности, вызываемые наличием локальных минимумов в функции стоимости ошибок распознавания. Примеры подобных процедур в отношении нейронных сетей описаны в [4].

5.3 Однослойные и многослойные нейронные сети

Один нейрон способен выполнять простейшие процедуры распознавания. Как было показано в предыдущем разделе, он позволяет выполнять линейное разделение параметрического пространства на два класса. Для достижения возможности разделения пространства на большее количество классов необходимо одновременное функционирование нескольких нейронов, соединённых в сеть. Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рис. 30. Напомним, что вершины-круги слева служат лишь для распределения входных сигналов и не выполняют каких-либо вычислений, а потому не считаются слоем. Каждый элемент из множества входов \mathbf{x} отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. В искусственных и биологических сетях многие соединения могут отсутствовать, все соединения показаны в целях общности. Могут иметь место также соединения между выходами и входами элементов в слое. Такие конфигурации рассматриваются в разделе 0.

Удобно считать веса элементами матрицы \mathbf{W} . Матрица имеет n строк и m столбцов, где m – число входов, а n – число нейронов. Например, w_{23} – это вес, связывающий третий вход со вторым нейроном. Таким образом, вычисление выходного вектора \mathbf{y} , компонентами которого являются выходы нейронов, сводится к матричному умножению $\mathbf{y} = \mathbf{W}\mathbf{x}$.



Значение выхода каждого нейрона, вычисленное по приведённой выше формуле, обозначим через NET . Как правило, значение NET перед подачей на выход сети подвергается преобразованию *активационной функцией* F , так что окончательное значение выхода нейрона OUT оказывается равным

$$OUT = F(NET).$$

F может быть простой линейной функцией или пороговой функцией вида:

$$OUT = \begin{cases} 1, & NET \geq T \\ 0, & NET < T \end{cases},$$

где T — некоторая постоянная величина порога.

Если F сужает диапазон изменения величины NET так, что при любых значениях NET значения OUT принадлежат некоторому конечному интервалу, то F называется «сжимающей» функцией. В качестве «сжимающей» функции часто используется логистическая или «сигмоидальная» (S-образная) функция. Эта функция математически выражается как

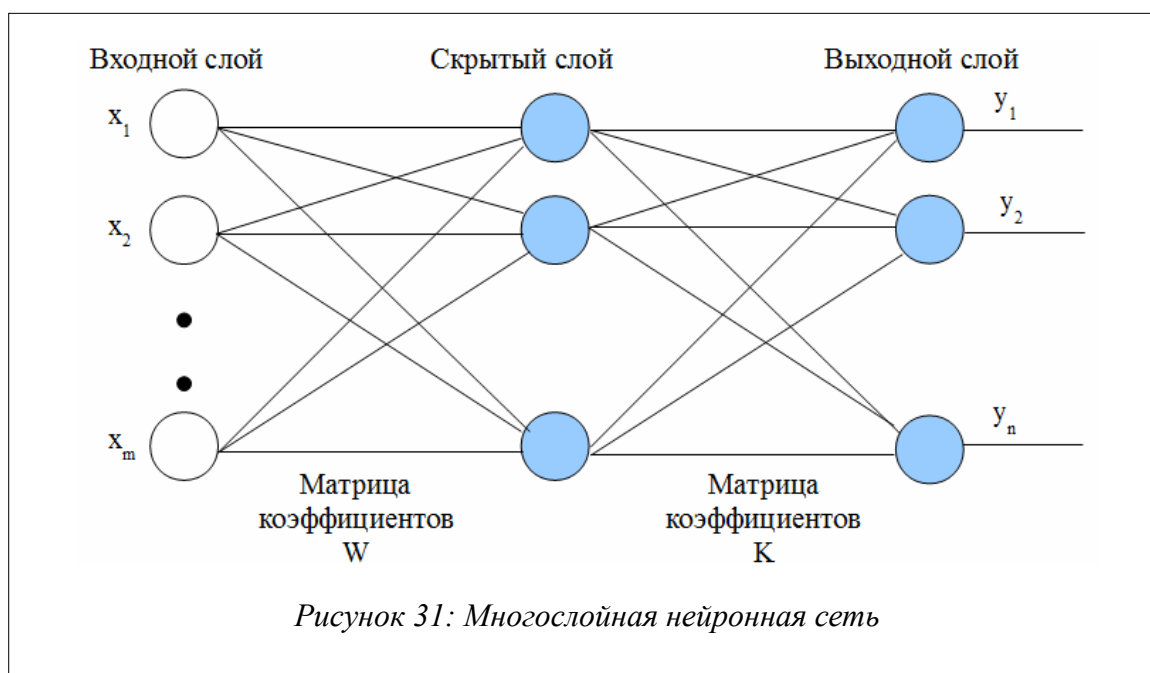
$$F(x) = \frac{1}{1 + e^{-x}}.$$

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона. Другой широко используемой активационной функцией является гиперболический тангенс $F(x) = th(x)$. По форме она сходна с логистической функцией и часто используется биологами в качестве математической модели активации нервной клетки.

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями. Хотя созданы сети всех конфигураций, какие только можно себе представить, послойная организация нейронов копирует слоистые структуры определенных отделов мозга. Оказалось, что такие многослойные сети обладают большими возможностями, чем однослойные; были разработаны алгоритмы для их обучения.

В многослойной нейронной сети, по сравнению с однослойной, присутствует один или

более дополнительных слоёв нейронов. Причём входы каждого нейрона в следующем слое соединены взвешенными связями с выходами всех нейронов в предыдущем слое (см. рис. 31). Последний слой, как правило, называют *выходным слоем*, а все слои между входным и выходным — *скрытыми*.



Вычисления в многослойных сетях выполняются слой за слоем, начиная с ближайшего к входу слоя. Величина *NET* каждого нейрона первого слоя вычисляется как взвешенная сумма входов нейрона. Затем активационная функция F «сжимает» *NET* и дает величину *OUT* для каждого нейрона в этом слое. Когда множество выходов слоя получено, оно является входным множеством для следующего слоя. Процесс повторяется слой за слоем, пока не будет получено заключительное множество выходов сети.

Этот процесс может быть выражен в сжатой форме с помощью векторной нотации. Веса между нейронами могут рассматриваться как матрица W . Тогда *NET*-вектор слоя N может быть выражен как $\mathbf{n} = W\mathbf{x}$. Покомпонентным применением функции F к *NET*-вектору \mathbf{n} получается выходной вектор \mathbf{o} . Таким образом, для данного слоя вычислительный процесс описывается следующим выражением:

$$\mathbf{o} = F(W\mathbf{x})$$

Выходной вектор одного слоя является входным вектором для следующего, поэтому вычисление выходов последнего слоя требует применения последнего уравнения к каждому слою от входа сети к ее выходу.

Многослойные сети могут привести к увеличению вычислительной мощности по сравнению с однослойной сетью лишь в том случае, если активационная функция между слоями будет нелинейной. Вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением (если отсутствует нелинейная активационная функция) результирующего вектора на вторую весовую матрицу:

$$W_2(W_1X)$$

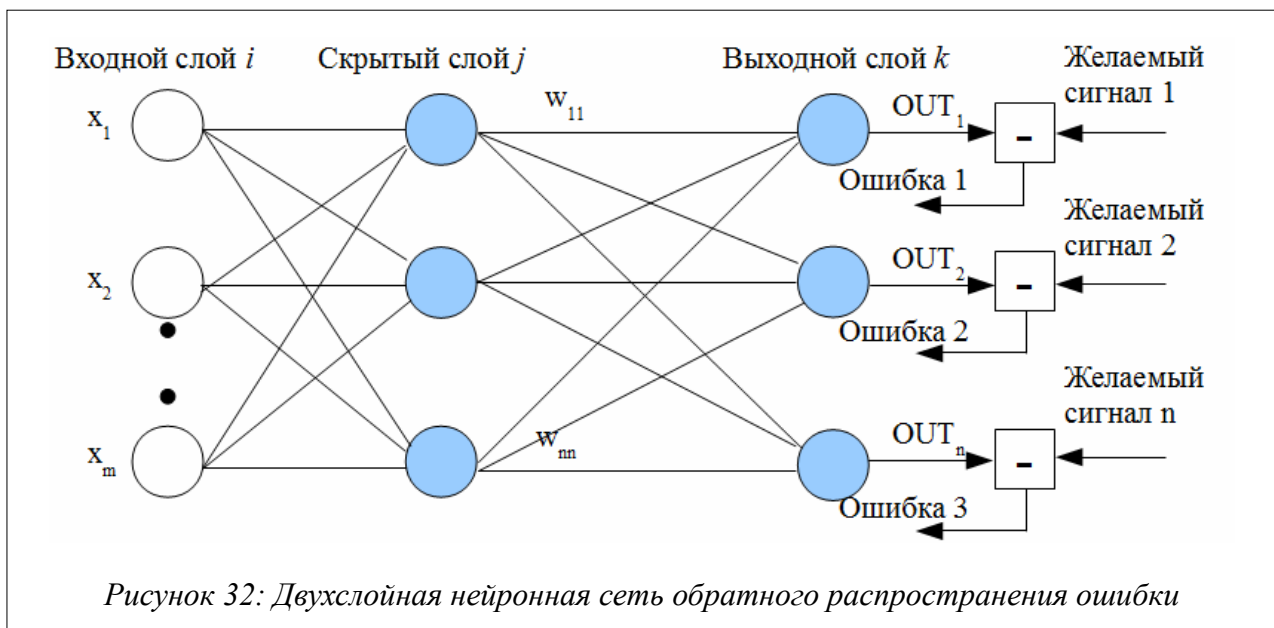
Так как умножение матриц ассоциативно, то

$$(W_2 W_1)X$$

Это показывает, что двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью. Для расширения возможностей сетей по сравнению с однослойной сетью необходима нелинейная активационная функция.

5.4 Сети с обратным распространением ошибки

Долгое время не было теоретически обоснованного алгоритма для обучения многослойных искусственных нейронных сетей. Разработка алгоритма обратного распространения сыграла важную роль в возрождении интереса к искусственным нейронным сетям. Обратное распространение — это систематический метод для обучения многослойных искусственных нейронных сетей. Он имеет солидное математическое обоснование. Несмотря на некоторые ограничения, процедура обратного распространения сильно расширила область проблем, в которых могут быть использованы искусственные нейронные сети.



На рисунке 32 показана структура двухслойной нейронной сети с обратным распространением ошибки. В качестве активационной функции в таких сетях используется сигмоидальная функция

$$F(x) = \frac{1}{1 + e^{-x}}.$$

Она имеет очень простую производную

$$\frac{dF}{dx} = x(1 - x),$$

что используется при обучении сети.

Целью обучения сети является такая подстройка ее весов, чтобы приложение некоторого множества входных векторов приводило к требуемому множеству выходных векторов. Данная процедура обучения относится к классу обучения с учителем, поэтому в обучающей выборке для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход.

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов, и предотвращает ряд других патологических случаев. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то сеть не сможет обучиться.

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 1 и 2, сходны с теми, которые выполняются при функционировании уже обученной сети, т.е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются послойно. На рис. 32 сначала вычисляются выходы нейронов слоя j , затем они используются в качестве входов слоя k , вычисляются выходы нейронов слоя k , которые и образуют выходной вектор сети.

На шаге 3 каждый из выходов сети, которые на рисунке 32 обозначены как *OUT*, вычитается из соответствующей компоненты целевого вектора, чтобы получить ошибку. Эта ошибка используется на шаге 4 для коррекции весов сети, причем знак и величина изменений весов определяются алгоритмом обучения (см. ниже).

После достаточного числа повторений этих четырех шагов разность между действительными выходами и целевыми выходами должна уменьшиться до приемлемой величины. При этом говорят, что сеть обучилась и её можно использовать для распознавания, причём веса более не изменяются.

На шаги 1 и 2 можно смотреть как на «проход вперед», так как сигнал распространяется по сети от входа к выходу. Шаги 3, 4 составляют «обратный проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов.

Обратный проход — подстройка весов выходного слоя. Так как для каждого нейрона выходного слоя задано целевое значение, то подстройка весов легко осуществляется с использованием модифицированного правила, аналогичного описанному в разделе 0. Внутренние слои называют «скрытыми слоями», для их выходов не имеется целевых значений для сравнения. Поэтому обучение усложняется.

Рассмотрим процесс обучения для одного веса от нейрона p в скрытом слое j к нейрону q в выходном слое k . Выход нейрона слоя k , вычитаясь из целевого значения (*Target*), дает сигнал ошибки. Он умножается на производную сжимающей функции [$OUT(1 - OUT)$],

вычисленную для этого нейрона слоя k , давая, таким образом, величину коррекции \square_q .

$$\delta_q = OUT_q(1 - OUT_q)(Target - OUT_q) \quad (5.4.1)$$

Затем δ_q умножается на величину OUT_p нейрона p слоя j , из которого выходит рассматриваемый вес. Это произведение в свою очередь умножается на коэффициент скорости обучения η (обычно от 0,01 до 1,0), и результат прибавляется к весу:

$$\begin{aligned} \Delta w_{qp} &= \eta \delta_q OUT_p \\ w_{qp}(n+1) &= w_{qp}(n) + \Delta w_{qp} \end{aligned} \quad (5.4.2)$$

где $w_{qp}(n)$ — величина веса от нейрона p в скрытом слое к нейрону q в выходном слое на шаге n (до коррекции); $w_{qp}(n+1)$ — величина веса на шаге $n+1$ (после коррекции); δ_q — величина δ для нейрона q , в выходном слое k ; OUT_p — величина OUT для нейрона p в скрытом слое j . Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.

Обратимся теперь к процедуре обучения нейронов скрытого слоя. Рассмотрим один нейрон в скрытом слое, предшествующем выходному слою. При проходе вперед этот нейрон передает свой выходной сигнал нейронам в выходном слое через соединяющие их веса. Во время обучения эти веса функционируют в обратном порядке, пропуская величину δ от выходного слоя назад к скрытому слою. Каждый из этих весов умножается на величину δ нейрона, к которому он присоединен в выходном слое. Величина δ_p , необходимая для нейрона скрытого слоя, получается суммированием всех таких произведений и умножением на производную сжимающей функции:

$$\delta_p = OUT_p(1 - OUT_p) \sum_q \delta_q w_{qp}$$

С помощью данной величины веса скрытого слоя могут быть скорректированы посредством уравнений (5.4.2).

Обратное распространение было использовано в широкой сфере прикладных исследований, среди которых распознавание букв, в том числе рукописных, преобразование печатного текста в речь, сжатие изображений.

5.5 Сети Кохонена и обучение без учителя

Сети Кохонена являются однослойными нейронными сетями, которые позволяют определять группы схожих входных векторов. Это достигается с помощью такой подстройки весов, что близкие входные векторы активируют один и тот же нейрон. Таким образом, задачей, решаемой сетями данного класса, является задача классификации (см. раздел 0).

Схема работы сети Кохонена построена на принципе «победитель получает всё». На вход подаётся вектор и вычисляются его скалярные произведения с векторами весов, связанными со всеми нейронами. Нейрон с максимальным значением скалярного произведения объявляется «победителем». Выход данного нейрона устанавливается равным полученному максимальному значению либо некоторому другому, заранее определённом (например, 1). Выходные значения всех прочих нейронов сети устанавливаются равными 0.

Перед предъявлением сети входных векторов рекомендуется их нормализовать. Это

выполняется с помощью деления каждой компоненты входного вектора на длину вектора:

$$x_i^{norm} = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}}$$

Это превращает входной вектор в единичный вектор с тем же самым направлением. Нормализация входных векторов позволяет интерпретировать их как единичные вектора, исходящие из начала координат n -мерного параметрического пространства, концы которых располагаются на сфере с единичным радиусом и центром в начале координат.

Обучение Кохонена является самообучением, протекающим без учителя. Поэтому трудно (и не нужно) предсказывать, какой именно нейрон Кохонена будет активироваться для заданного входного вектора. Необходимо лишь гарантировать, чтобы в результате обучения разделялись несхожие входные векторы.

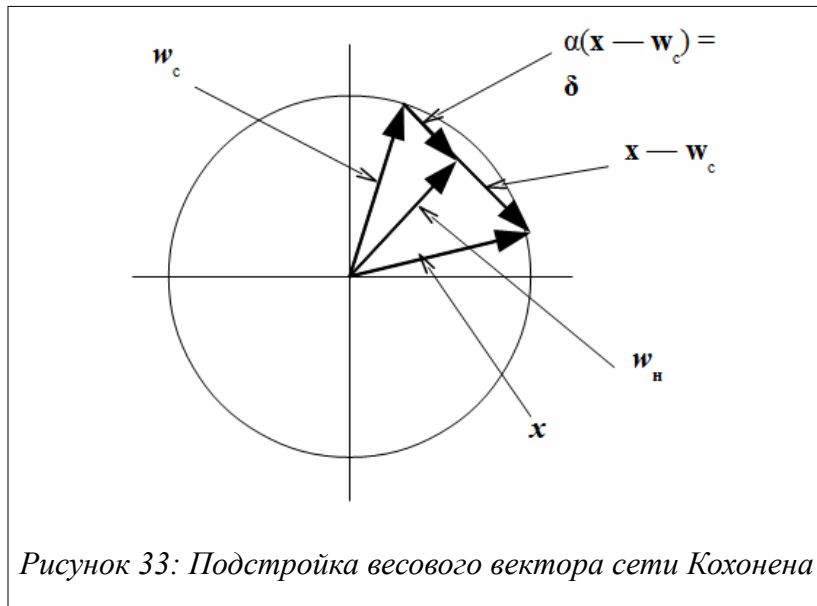
При предъявлении сети обучающего входного вектора сеть выполняет те же действия, что и при нормальном функционировании. После выявления нейрона-«победителя» происходит подстройка его весов. Так как скалярное произведение, используемое для вычисления величин NET , является мерой сходства между входным вектором и вектором весов, то процесс обучения состоит в выборе нейрона Кохонена с весовым вектором, наиболее близким к входному вектору, и дальнейшем приближении весового вектора к входному. Снова отметим, что процесс является самообучением, выполняемым без учителя. Сеть самоорганизуется таким образом, что данный нейрон Кохонена имеет максимальный выход для данного входного вектора. Уравнение, описывающее процесс обучения имеет следующий вид:

$$\mathbf{w}_n = \mathbf{w}_c + \alpha(\mathbf{x} - \mathbf{w}_c)$$

где \mathbf{w}_n — новое значение веса, соединяющего входную компоненту \mathbf{x} с выигравшим нейроном; \mathbf{w}_c — предыдущее значение этого веса; α — коэффициент скорости обучения, который может варьироваться в процессе обучения.

Каждый вес, связанный с выигравшим нейроном Кохонена, изменяется пропорционально разности между его величиной и величиной входа, к которому он присоединен. Направление изменения минимизирует разность между весом и его входом.

На рисунке 33 этот процесс показан геометрически в двумерном виде для нормализованных векторов. Сначала находится вектор $\mathbf{x} - \mathbf{w}_c$, для этого проводится отрезок из конца \mathbf{w}_c в конец \mathbf{x} . Затем этот вектор укорачивается умножением его на скалярную величину α , меньшую единицы, в результате чего получается вектор изменения δ . Окончательно новый весовой вектор \mathbf{w}_n является отрезком, направленным из начала координат в конец вектора δ . Отсюда можно видеть, что эффект обучения состоит во вращении весового вектора в направлении входного вектора без существенного изменения его длины.



5.6 Сети Хопфилда

На рис. 34 показана сеть с обратными связями, состоящая из двух слоев. Нулевой слой не выполняет вычислительной функции, а лишь распределяет выходы сети обратно на входы. В качестве активационной функции F используется простая пороговая функция. Выход нейрона равен единице, если взвешенная сумма выходов с других нейронов больше порога T_j , в противном случае она равна нулю. Он вычисляется следующим образом:

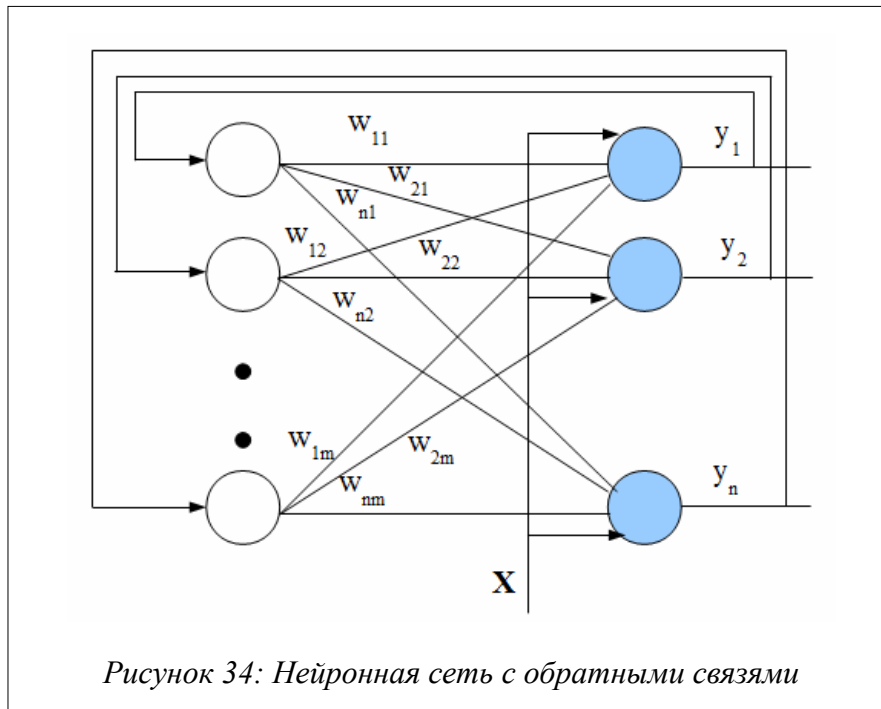
$$NET_j = \sum_{i \neq j} w_{ij} OUT_i + IN_j,$$

$$OUT_j = \begin{cases} 1, & NET_j > T_j, \\ 0, & NET_j < T_j, \\ \text{не изменяется,} & \text{если } NET_j = T_j \end{cases}.$$

Состояние сети – это множество текущих значений сигналов OUT от всех нейронов. В первоначальной сети Хопфилда состояние каждого нейрона менялось в дискретные случайные моменты времени, в последующей работе состояния нейронов могли меняться одновременно. Так как выходом бинарного нейрона может быть только ноль или единица (промежуточных уровней нет), то текущее состояние сети является двоичным числом, каждый бит которого является сигналом OUT некоторого нейрона.

Функционирование сети легко визуализируется геометрически. Сеть из двух нейронов в выходном слое может быть представлена квадратом, каждой вершине которого соответствует одно из четырех состояний системы (00, 01, 10, 11). Сеть с тремя выходными нейронами представляется кубом (в трехмерном пространстве), имеющим восемь вершин, каждая из которых помечена трёхбитовым бинарным числом. В общем случае система с n нейронами имеет 2^n различных состояний и представляется n -мерным гиперкубом.

Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой.



Человеческая память ассоциативна, т. е. некоторое воспоминание может породить большую связанную с ним область. Например, несколько музыкальных тактов могут вызвать целую гамму чувственных воспоминаний, включая пейзажи, звуки и запахи. Напротив, обычная компьютерная память является локально адресуемой, предъявляется адрес и извлекается информация по этому адресу.

Сеть с обратной связью формирует ассоциативную память. Подобно человеческой памяти по заданной части нужной информации вся информация извлекается из «памяти». Чтобы организовать ассоциативную память с помощью сети с обратными связями, веса должны выбираться так, чтобы образовывать энергетические минимумы в нужных вершинах единичного гиперкуба.

Хопфилд разработал ассоциативную память с непрерывными выходами, изменяющимися в пределах от +1 до -1, соответствующих двоичным значениям 0 и 1. Запоминаемая информация кодируется двоичными векторами и хранится в весах согласно следующей формуле:

$$w_{ij} = \sum_{d=1..m} (OUT_{i,d} OUT_{j,d})$$

где m – число запоминаемых выходных векторов; d – номер запоминаемого выходного вектора; $OUT_{i,j}$ – i -компонента запоминаемого выходного вектора.

Это выражение может стать более ясным, если заметить, что весовой массив \mathbf{W} может быть найден вычислением внешнего произведения каждого запоминаемого вектора с самим собой (если требуемый вектор имеет n компонент, то эта операция образует матрицу размером $n \times n$) и суммированием матриц, полученных таким образом. Это может быть записано в виде

$$\mathbf{W} = \sum_i \mathbf{D}_i^t \mathbf{D}_i$$

где \mathbf{D}_i – i -й запоминаемый вектор-строка.

Как только веса заданы, сеть может быть использована для получения запомненного выходного вектора по данному входному вектору, который может быть частично неправильным или неполным. Для этого выходам сети сначала придают значения этого входного вектора. Затем входной вектор убирается и сети предоставляется возможность

«расслабиться», опустившись в ближайший глубокий минимум. Сеть идущая по локальному наклону функции энергии, может быть захвачена локальным минимумом, не достигнув наилучшего в глобальном смысле решения.

6 Список литературы

1. Хант Э. Искусственный интеллект. М.: Мир, 1978.
2. Фу К. Структурные методы в распознавании образов. М.: Мир, 1977
3. Ту Дж, Гонсалес Р. Принципы распознавания образов. М.: Мир, 1978.
4. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика. Пер. с англ. Ю.А.Зуев, В.А. Точенов, 1992.
5. Мински М. Фреймы для представления знаний. Пер. с англ. 1979.
6. Журавлёв Ю.И. Распознавание. Классификация. Прогноз. Математические методы и их применение. Вып.2. М.: Наука, 1989.
7. Кадакин М.Ю. Распознавание шифров источников Рукописной древнерусской картотеки XI–XVII вв. // Интеллектуальные технологии и системы: сборник статей аспирантов и студентов / Под ред. Ю.Н.Филипповича — М.:МГУП, 2001. — Вып. 3.
8. Шамис А. Л. Принципы интеллектуализации машинного распознавания изображений и их реализация в системах оптического чтения текстов - ABBYY FineReader и FormReader. Новости Искусственного интеллекта, вып. 1, 2002.
9. [F. Sebastiani. Machine Learning in Automated Text Categorization](#)
10. [A.K. Jain, M.N. Murty, P.J. Flynn. Data Clustering: A Review, 1999.](#)
11. Местецкий Л.М. Математические методы распознавания образов. Курс лекций. МГУ, 2004.
12. Новикова Н.М. Структурное распознавание образов. Учебно-методическое пособие для вузов. Издательско-полиграфический центр Воронежского государственного университета, 2008.
13. Харман Г. Современный факторный анализ. М.:Статистика, 1972.
14. Uhr L., Vossler C. A pattern recognition program that generates, evaluates and adjusts its own operators, в сб. «Computers and thought» под ред. Feigenbaum E., Feldman J., New York. (Русский перевод в сб. «Вычислительные машины и мышление» под ред. Фейгенбаума Э. И Фельдмана Дж. М.: Мир, 1967).
15. Кручинин, А.Ю. Алгоритм распознавания двумерных графических кодов с произвольным углом поворота и наклона камеры : материалы седьмой всероссийской научно-практической конференции (с международным участием) «Современные информационные технологии в науке, образовании и практике» / А.Ю. Кручинин. – Оренбург: ИПК ГОУ ОГУ, 2008. – С. 193-196.
16. Fu S. Sequential methods in pattern recognition and machine learning. New York, 1969.
17. [В.Л. Афонин, В.А. Макушкин.](#) Интеллектуальные робототехнические системы. Эл.ресурс: <http://www.intuit.ru/department/human/isrob/>.
18. Minsky M., and Papert S., 1969. Perceptrons. Cambridge, MA: MIT Press. (Русский перевод: Минский М. Л., Пейперт С. Перцептроны. –М. Мир. – 1971.
19. H. Freeman. On the encoding of arbitrary geometric configurations, IRE Transactions on Electronic Computers EC- 10(1961) 260-268.
20. Feldman, J. A., First thoughts on grammatical inference, Stanford A.I. Memo No. 55, August, 1967.
21. Shaw, A.C., A formal picture description scheme as a basis for picture processing systems, Information and Control – 14 (1969) 9-52.
22. Kolers P., Eden M., Recognizing patterns. Studies in living and automatic systems. Cambridge, MA: MIT Press, 1968. (Русский перевод: Распознавание образов.

Исследование живых и автоматических распознающих систем. -М. Мир. - 1970).