

## Лекция 6. Системы распознавания образов (идентификации). Нейронные сети

### *Понятие образа*

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

Образное восприятие мира — одно из загадочных свойств живого мозга, позволяющее разобраться в бесконечном потоке воспринимаемой информации и сохранять ориентацию в океане разрозненных данных о внешнем мире. Воспринимая внешний мир, мы всегда производим классификацию воспринимаемых ощущений, т. е. разбиваем их на группы похожих, но не тождественных явлений. Например, несмотря на существенное различие, к одной группе относятся все буквы А, написанные различными почерками, или все звуки, соответствующие одной и той же ноте, взятой в любой октаве и на любом инструменте, а оператор, управляющий техническим объектом, на целое множество состояний объекта реагирует одной и той же реакцией. Характерно, что для составления понятия о группе восприятий определенного класса достаточно ознакомиться с незначительным количеством ее представителей. Ребенку можно показать всего один раз какую-либо букву, чтобы он смог найти эту букву в тексте, написанном различными шрифтами, или узнать ее, даже если она написана в умышленно искаженном виде. Это свойство мозга позволяет сформулировать такое понятие, как образ.

Образы обладают характерным свойством, проявляющимся в том, что ознакомление с конечным числом явлений из одного и того же множества дает возможность узнавать сколь угодно большое число его представителей. Примерами образов могут быть: река, море, жидкость, музыка Чайковского, стихи Маяковского и т. д. В качестве образа можно рассматривать и некоторую совокупность состояний объекта управления, причем вся эта совокупность состояний характеризуется тем, что для достижения заданной цели требуется одинаковое воздействие на объект. Образы обладают характерными объективными свойствами в том смысле, что разные люди, обучающиеся на различном материале наблюдений, большей частью одинаково и независимо друг от друга классифицируют одни и те же объекты. Именно эта объективность образов позволяет людям всего мира понимать друг друга.

Способность восприятия внешнего мира в форме образов позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов позволяет моделировать процесс их распознавания. Будучи отражением объективной реальности, понятие образа столь же объективно, как и сама реальность, а поэтому это понятие может быть само по себе объектом специального исследования.

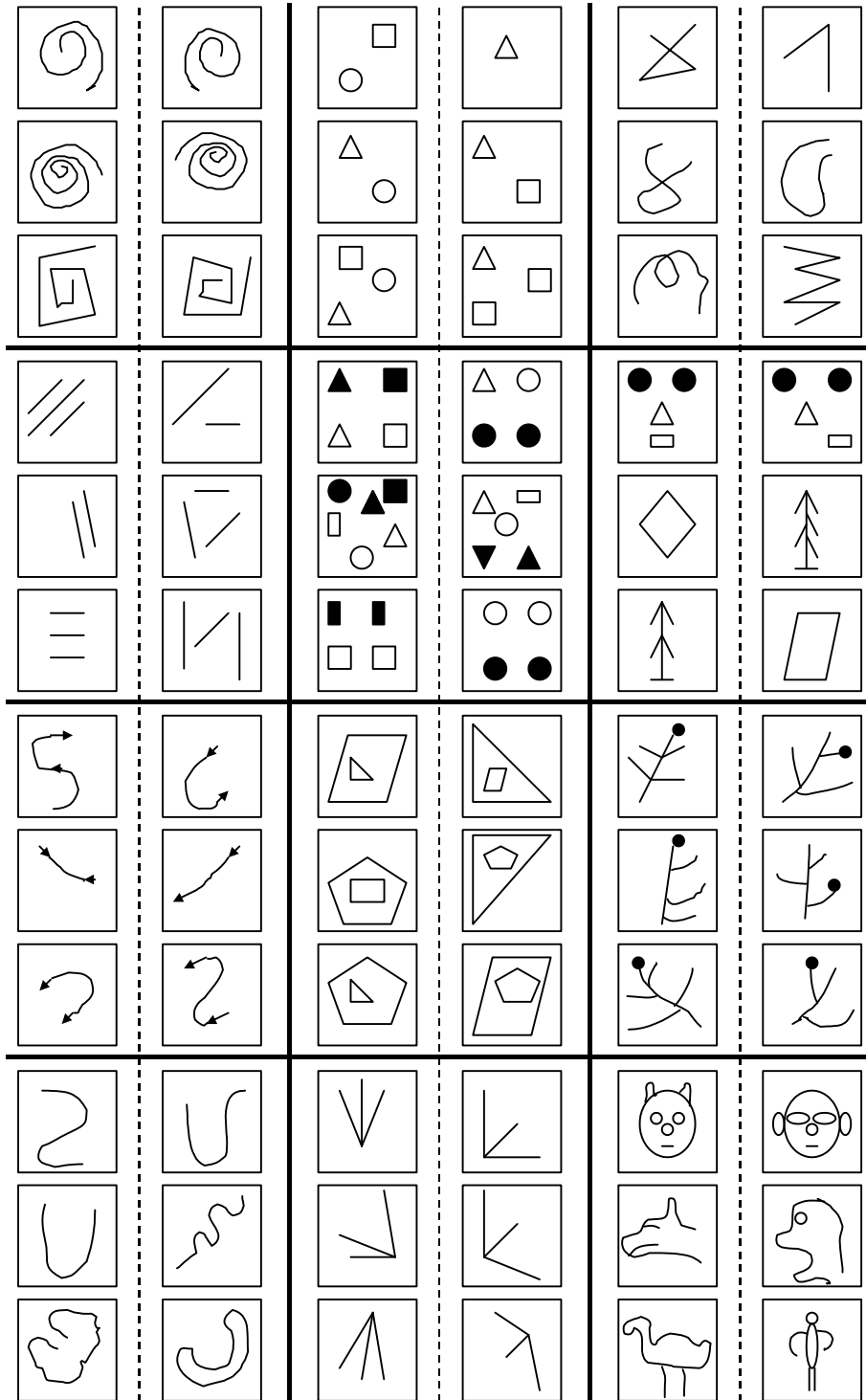
В литературе, посвященной проблеме обучения распознавания образов (ОРО), часто вместо понятия образа вводится понятие класса.

### ***Проблема обучения распознаванию образов (ОРО)***

Одним из самых интересных свойств человеческого мозга является способность отвечать на бесконечное множество состояний внешней среды конечным числом реакций. Может быть, именно это свойство позволило человеку достигнуть высшей формы существования живой материи, выражающейся в способности к мышлению, т. е. активному отражению объективного мира в виде образов, понятий, суждений и т. д. Поэтому проблема ОРО возникла при изучении физиологических свойств мозга.

Рассмотрим пример задач из области ОРО.

Рис. 1



Здесь представлены 12 задач, в которых следует отобрать признаки, при помощи которых можно отличить левую триаду картинок от правой. Решение данных задач требует моделирования логического мышления в полном объеме.

В целом проблема распознавания образов состоит из двух частей: обучения и распознавания. Обучение осуществляется путем показа отдельных объектов с указанием их принадлежности тому или другому образу. В результате обучения распознающая система должна приобрести способность реагировать

одинаковыми реакциями на все объекты одного образа и различными — на все объекты различных образов. Очень важно, что процесс обучения должен завершиться только путем показов конечного числа объектов без каких-либо других подсказок. В качестве объектов обучения могут быть либо картинки, либо другие визуальные изображения (буквы), либо различные явления внешнего мира, например звуки, состояния организма при медицинском диагнозе, состояние технического объекта в системах управления и др. Важно, что в процессе обучения указываются только сами объекты и их принадлежность образу. За обучением следует процесс распознавания новых объектов, который характеризует действия уже обученной системы. Автоматизация этих процедур и составляет проблему обучения распознаванию образов. В том случае, когда человек сам разгадывает или придумывает, а затем навязывает машине правило классификации, проблема распознавания решается частично, так как основную и главную часть проблемы (обучение) человек берет на себя.

Проблема обучения распознаванию образов интересна как с прикладной, так и с принципиальной точки зрения. С прикладной точки зрения решение этой проблемы важно прежде всего потому, что оно открывает возможность автоматизировать многие процессы, которые до сих пор связывали лишь с деятельностью живого мозга. Принципиальное значение проблемы тесно связано с вопросом, который все чаще возникает в связи с развитием идей кибернетики: что может и что принципиально не может делать машина? В какой мере возможности машины могут быть приближены к возможностям живого мозга? В частности, может ли машина развить в себе способность перенять у человека умение производить определенные действия в зависимости от ситуаций, возникающих в окружающей среде? Пока стало ясно только то, что если человек может сначала сам осознать свое умение, а потом его описать, т. е. указать, почему он производит действия в ответ на каждое состояние внешней среды или как (по какому правилу) он объединяет отдельные объекты в образы, то такое умение без принципиальных трудностей может быть передано машине. Если же человек обладает умением, но не может объяснить его, то остается только один путь передачи умения машине — обучение примерами.

Круг задач, которые могут решаться с помощью распознающих систем, чрезвычайно широк. Сюда относятся не только задачи распознавания зрительных и слуховых образов, но и задачи распознавания сложных процессов и явлений, возникающих, например, при выборе целесообразных действий руководителем предприятия или выборе оптимального управления технологическими, экономическими, транспортными или военными операциями. В каждой из таких задач анализируются некоторые явления, процессы, состояния внешнего мира, всюду далее называемые объектами наблюдения. Прежде чем начать анализ какого-либо объекта, нужно получить о нем определенную, каким-либо способом упорядоченную информацию. Такая информация представляет собой характеристику объектов, их отображение на множестве воспринимающих органов распознающей системы.

Но каждый объект наблюдения может воздействовать по-разному, в зависимости от условий восприятия. Например, какая-либо буква, даже одинаково написанная, может в принципе как угодно смещаться относительно

воспринимающих органов. Кроме того, объекты одного и того же образа могут достаточно сильно отличаться друг от друга и, естественно, по-разному воздействовать на воспринимающие органы.

Каждое отображение какого-либо объекта на воспринимающие органы распознающей системы, независимо от его положения относительно этих органов, принято называть изображением объекта, а множества таких изображений, объединенные какими-либо общими свойствами, представляют собой образы.

При решении задач управления методами распознавания образов вместо термина "изображение" применяют термин "состояние". Состояние — это определенной формы отображение измеряемых текущих (или мгновенных) характеристик наблюдаемого объекта. Совокупность состояний определяет ситуацию. Понятие "ситуация" является аналогом понятия "образ". Но эта аналогия не полная, так как не всякий образ можно назвать ситуацией, хотя всякую ситуацию можно назвать образом.

Ситуацией принято называть некоторую совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же или схожими характеристиками объекта. Например, если в качестве объекта наблюдения рассматривается некоторый объект управления, то ситуация объединяет такие состояния этого объекта, в которых следует применять одни и те же управляющие воздействия. Если объектом наблюдения является военная игра, то ситуация объединяет все состояния игры, которые требуют, например, мощного танкового удара при поддержке авиации.

Выбор исходного описания объектов является одной из центральных задач проблемы ОРО. При удачном выборе исходного описания (пространства признаков) задача распознавания может оказаться тривиальной и, наоборот, неудачно выбранное исходное описание может привести либо к очень сложной дальнейшей переработке информации, либо вообще к отсутствию решения. Например, если решается задача распознавания объектов, отличающихся по цвету, а в качестве исходного описания выбраны сигналы, получаемые от датчиков веса, то задача распознавания в принципе не может быть решена.

### ***Геометрический и структурный подходы.***

Каждый раз, когда сталкиваются с незнакомыми задачами, появляется естественное желание представить их в виде некоторой легко понимаемой модели, которая позволяла бы осмыслить задачу в таких терминах, которые легко воспроизводятся нашим воображением. А так как мы существуем в пространстве и во времени, наиболее понятной для нас является пространственно-временная интерпретация задач.

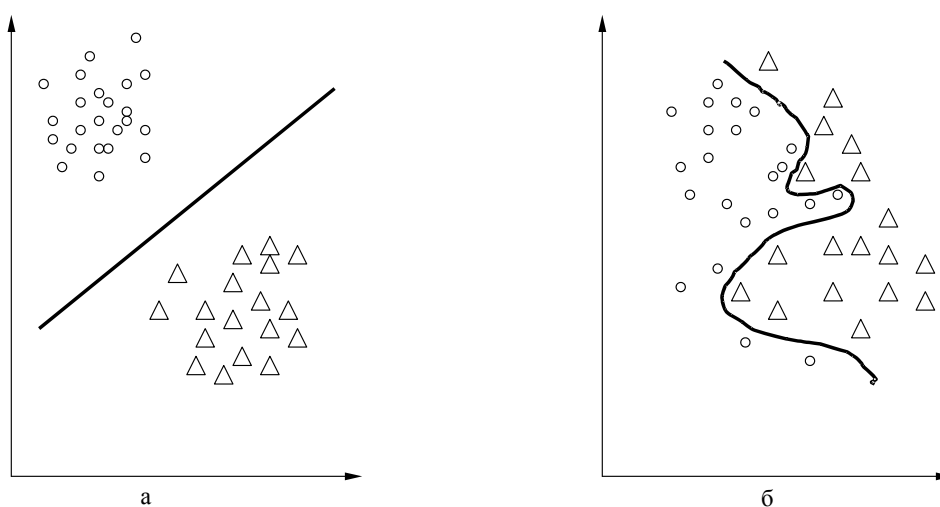
Любое изображение, которое возникает в результате наблюдения какого-либо объекта в процессе обучения или экзамена, можно представить в виде вектора, а значит и в виде точки некоторого пространства признаков. Если утверждается, что при показе изображений возможно однозначно отнести их к одному из двух (или нескольких) образов, то тем самым утверждается, что в некотором пространстве существует две (или несколько) области, не имеющие общих точек, и что изображения — точки из этих областей. Каждой такой области можно приписать наименование, т. е. дать название, соответствующее образу.

Проинтерпретируем теперь в терминах геометрической картины процесс обучения распознаванию образов, ограничившись пока случаем распознавания только двух образов. Заранее считается известным лишь только то, что требуется разделить две области в некотором пространстве и что показываються точки только из этих областей. Сами эти области заранее не определены, т. е. нет каких-либо сведений о расположении их границ или правил определения принадлежности точки к той или иной области.

В ходе обучения предъявляются точки, случайно выбранные из этих областей, и сообщается информация о том, к какой области принадлежат предъявляемые точки. Никакой дополнительной информации об этих областях, т. е. о расположении их границ, в ходе обучения не сообщается. Цель обучения состоит либо в построении поверхности, которая разделяла бы не только показанные в процессе обучения точки, но и все остальные точки, принадлежащие этим областям, либо в построении поверхностей, ограничивающих эти области так, чтобы в каждой из них находились только точки одного образа. Иначе говоря, цель обучения состоит в построении таких функций от векторов-изображений, которые была бы, например, положительны на всех точках одного и отрицательны на всех точках другого образа. В связи с тем, что области не имеют общих точек, всегда существует целое множество таких разделяющих функций, а в результате обучения должна быть построена одна из них.

Если предъявляемые изображения принадлежат не двум, а большему числу образов, то задача состоит в построении по показанным в ходе обучения точкам поверхности, разделяющей все области, соответствующие этим образам, друг от друга. Задача эта может быть решена, например, путем построения функции, принимающей над точками каждой из областей одинаковое значение, а над точками из разных областей значение этой функции должно быть различно.

**Рис. 2. Два образа.**



На первый взгляд кажется, что знание всего лишь некоторого количества точек из области недостаточно, чтобы отделить всю область. Действительно, можно указать бесчисленное количество различных областей, которые содержат эти точки, и как бы ни была построена по ним поверхность, выделяющая область, всегда можно указать другую область, которая пересекает поверхность и вместе с

тем содержит показанные точки. Однако известно, что задача о приближении функции по информации о ней в ограниченном множестве точек, существенно более узкой, чем все множество, на котором функция задана, является обычной математической задачей об аппроксимации функций. Разумеется, решение таких задач требует введения определенных ограничений на классе рассматриваемых функций, а выбор этих ограничений зависит от характера информации, которую может добавить учитель в процессе обучения. Одной из таких подсказок является гипотеза о компактности образов. Интуитивно ясно, что аппроксимация разделяющей функции будет задачей тем более легкой, чем более компактны и чем более разнесены в пространстве области, подлежащие разделению. Так, например, в случае, показанном на Рис. 2а, разделение заведомо более просто, чем в случае, показанном на Рис. 2б. Действительно, в случае, изображенном на Рис. 2а, области могут быть разделены плоскостью, и даже при больших погрешностях в определении разделяющей функции она все же будет продолжать разделять области. В случае же на Рис. 2б, разделение осуществляется замысловатой поверхностью и даже незначительные отклонения в ее форме приводят к ошибкам разделения. Именно это интуитивное представление о сравнительно легко делимых областях привело к гипотезе компактности.

Наряду с геометрической интерпретацией проблемы обучения распознаванию образов существует и иной подход, который назван структурным, или лингвистическим. Поясним лингвистический подход на примере распознавания зрительных изображений. Сначала выделяется набор исходных понятий — типичных фрагментов, встречающихся на изображениях, и характеристик взаимного расположения фрагментов — "слева", "снизу", "внутри" и т. д. Эти исходные понятия образуют словарь, позволяющий строить различные логические высказывания, иногда называемые предположениями. Задача состоит в том, чтобы из большого количества высказываний, которые могли бы быть построены с использованием этих понятий, отобрать наиболее существенные для данного конкретного случая.

Далее, просматривая конечное и по возможности небольшое число объектов из каждого образа, нужно построить описание этих образов. Построенные описания должны быть столь полными, чтобы решить вопрос о том, к какому образу принадлежит данный объект. При реализации лингвистического подхода возникают две задачи: задача построения исходного словаря, т. е. набор типичных фрагментов, и задача построения правил описания из элементов заданного словаря.

В рамках лингвистической интерпретации проводится аналогия между структурой изображений и синтаксисом языка. Стремление к этой аналогии было вызвано возможностью использовать аппарат математической лингвистики, т. е. методы по своей природе являются синтаксическими. Использование аппарата математической лингвистики для описания структуры изображений можно применять только после того, как произведена сегментация изображений на составные части, т. е. выработаны слова для описания типичных фрагментов и методы их поиска. После предварительной работы, обеспечивающей выделение слов, возникают собственно лингвистические задачи, состоящие из задач автоматического грамматического разбора описаний для распознавания

изображений. При этом проявляется самостоятельная область исследований, которая требует не только знания основ математической лингвистики, но и овладения приемами, которые разработаны специально для лингвистической обработки изображений.

### ***Гипотеза компактности***

Если предположить, что в процессе обучения пространство признаков формируется исходя из задуманной классификации, то тогда можно надеяться, что задание пространство признаков само по себе задает свойство, под действием которого образы в этом пространстве легко разделяются. Именно эти надежды по мере развития работ в области распознавания образов стимулировали появление гипотезы компактности, которая гласит: образам соответствуют компактные множества в пространстве признаков. Под компактным множеством пока будем понимать некие "сгустки" точек в пространстве изображений, предполагая, что между этими сгустками существуют разделяющие их разряжения.

Однако эту гипотезу не всегда удавалось подтвердить экспериментально, но, что самое главное, те задачи, в рамках которых гипотеза компактности хорошо выполнялась (Рис. 2а), все без исключения находили простое решение. И наоборот, те задачи, для которых гипотеза не подтверждалась (Рис. 2б), либо совсем не решались, либо решались с большим трудом с привлечением дополнительных ухищрений. Этот факт заставил по меньшей мере усомниться в справедливости гипотезы компактности, так как для опровержения любой гипотезы достаточно одного отрицающего ее примера. Вместе с этим, выполнение гипотезы всюду там, где удавалось хорошо решить задачу обучения распознаванию образов, сохраняло к этой гипотезе интерес. Сама гипотеза компактности превратилась в признак возможности удовлетворительного решения задач распознавания.

Формулировка гипотезы компактности подводит вплотную к понятию абстрактного образа. Если координаты пространства выбирать случайно, то и изображения в нем будут распределены случайно. Они будут в некоторых частях пространства располагаться более плотно, чем в других. Назовем некоторое случайно выбранное пространство абстрактным изображением. В этом абстрактном пространстве почти наверняка будут существовать компактные множества точек. Поэтому в соответствии с гипотезой компактности множества объектов, которым в абстрактном пространстве соответствуют компактные множества точек, разумно назвать абстрактными образами данного пространства.

### ***Обучение и самообучение. Адаптация и обучение***

Все картинки, представленные на Рис. 1, характеризуют задачу обучения. В каждой из этих задач задается несколько примеров (обучающая последовательность) правильно решенных задач. Если бы удалось подметить некое всеобщее свойство, не зависящее ни от природы образов, ни от их изображений, а определяющее лишь их способность к делимости, то наряду с обычной задачей обучения распознаванию, с использованием информации о принадлежности каждого объекта из обучающей последовательности тому или иному образу можно было бы поставить иную классификационную задачу — так называемую задачу обучения без учителя. Задачу такого рода на описательном уровне можно сформулировать следующим образом: системе одновременно или



последовательно предъявляются объекты без каких-либо указаний об их принадлежности к образам. Входное устройство системы отображает множество объектов на множество изображений и, используя некоторое заложенное в нее заранее свойство делимости образов, производит самостоятельную классификацию этих объектов. После такого процесса самообучения система должна приобрести способность к распознаванию не только уже знакомых объектов (объектов из обучающей последовательности), но и тех, которые ранее не предъявлялись. Процессом самообучения некоторой системы называется такой процесс, в результате которого эта система без подсказки учителя приобретает способность к выработке одинаковых реакций на изображения объектов одного и того же образа и различных реакций на изображения различных образов. Роль учителя при этом состоит лишь в подсказке системе некоторого объективного свойства, одинакового для всех образов и определяющего способность к разделению множества объектов на образы.

Оказывается, таким объективным свойством является свойство компактности образов. Взаимное расположение точек в выбранном пространстве уже содержит информацию о том, как следует разделить множество точек. Эта информация и определяет то свойство делимости образов, которое оказывается достаточным для самообучения системы распознаванию образов.

Большинство известных алгоритмов самообучения способны выделять только абстрактные образы, т. е. компактные множества в заданных пространствах. Различие между ними состоит, по-видимому, в формализации понятия компактности. Однако это не снижает, а иногда и повышает ценность алгоритмов самообучения, так как часто сами образы заранее никем не определены, а задача состоит в том, чтобы определить, какие подмножества изображений в заданном пространстве представляют собой образы. Хорошим примером такой постановки задачи являются социологические исследования, когда по набору вопросов выделяются группы людей. В таком понимании задачи алгоритмы самообучения генерируют заранее не известную информацию о существовании в заданном пространстве образов, о которых ранее никто не имел никакого представления.

Кроме того, результат самообучения характеризует пригодность выбранного пространства для конкретной задачи обучения распознаванию. Если абстрактные образы, выделяемые в процессе самообучения, совпадают с реальными, то пространство выбрано удачно. Чем сильнее абстрактные образы отличаются от реальных, тем "неудобнее" выбранное пространство для конкретной задачи.

Обучением обычно называют процесс выработки в некоторой системе той или иной реакции на группы внешних идентичных сигналов путем многократного воздействия на систему внешней корректировки. Такую внешнюю корректировку в обучении принято называть "поощрениями" и "наказаниями". Механизм генерации этой корректировки практически полностью определяет алгоритм обучения. Самообучение отличается от обучения тем, что здесь дополнительная информация о верности реакции системе не сообщается.

Адаптация — это процесс изменения параметров и структуры системы, а возможно, и управляющих воздействий на основе текущей информации с целью

достижения определенного состояния системы при начальной неопределенности и изменяющихся условиях работы.

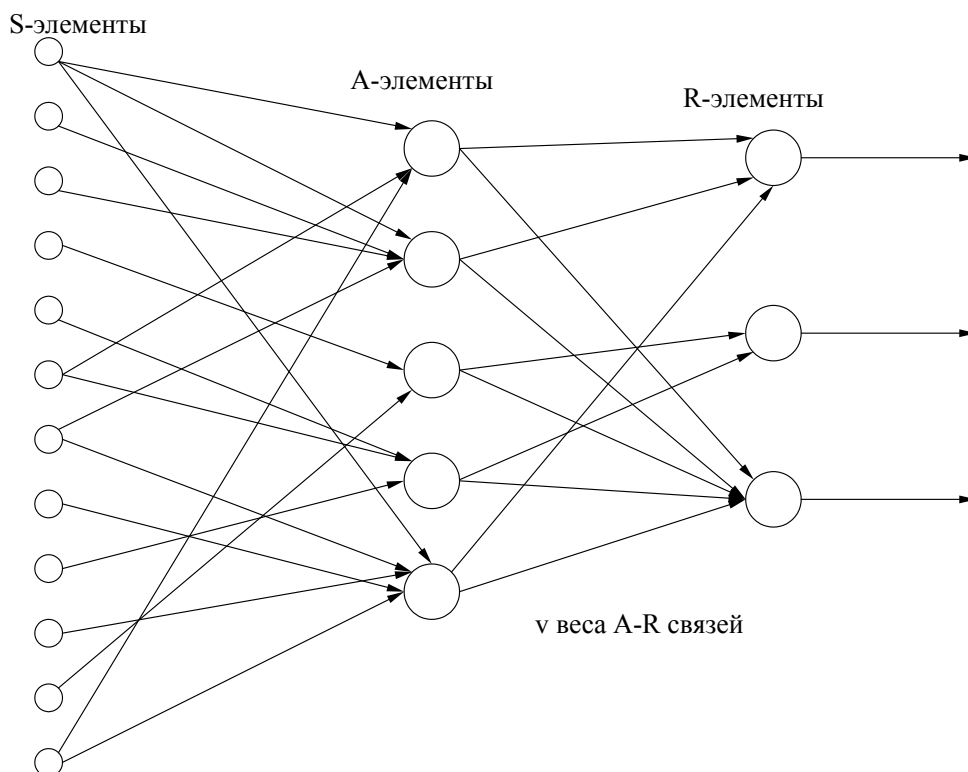
Обучение — это процесс, в результате которого система постепенно приобретает способность отвечать нужными реакциями на определенные совокупности внешних воздействий, а адаптация — это подстройка параметров и структуры системы с целью достижения требуемого качества управления в условиях непрерывных изменений внешних условий.

### ***Перцептроны***

Пока о проблеме обучения распознаванию образов удавалось говорить в общих чертах, не выделяя конкретные методы или алгоритмы, не возникало и трудностей, появляющихся всяких раз, когда приходится в огромном множестве конкретных примеров, характеризующиеся общими подходами к решению проблемы ОРО. Коварство самой проблемы состоит в том, что на первый взгляд все методы и алгоритмы кажутся совершенно различными и, что самое неприятное, часто никакой из них не годится для решения той задачи, которую крайне необходимо срочно решить. И тогда появляется желание выдумать новый алгоритм, который, может быть, достигнет цели. Очевидно, именно это привело к возникновению огромного множества алгоритмов, в котором не так-то легко разобраться.

Одним из методов решения задач обучения распознаванию образов основан на моделировании гипотетического механизма человеческого мозга. Структура модели заранее постулируется. При таком подходе уровень биологических знаний или гипотез о биологических механизмах является исходной предпосылкой, на которой базируются модели этих механизмов. Примером такого направления в теории и практике проблемы ОРО является класс устройств, называемых перцептронами. Нужно отметить, что перцептроны на заре своего возникновения рассматривались только как эвристические модели механизма мозга. Впоследствии они стали основополагающей схемой в построении кусочно-линейных моделей, обучающихся распознаванию образов.

**Рис. 3**



В наиболее простом виде перцептрон (Рис. 3) состоит из совокупности чувствительных (сенсорных) элементов (S-элементов), на которые поступают входные сигналы. S-элементы случайным образом связаны с совокупностью ассоциативных элементов (A-элементов), выход которых отличается от нуля только тогда, когда возбуждено достаточно большое число S-элементов, воздействующих на один A-элемент. A-элементы соединены с реагирующими элементами (R-элементами) связями, коэффициенты усиления ( $v$ ) которых переменны и изменяются в процессе обучения. Взвешенные комбинации выходов R-элементов составляют реакцию системы, которая указывает на принадлежность распознаваемого объекта определенному образу. Если распознаются только два образа, то в перцептроне устанавливается только один R-элемент, который обладает двумя реакциями — положительной и отрицательной. Если образов больше двух, то для каждого образа устанавливают свой R-элемент, а выход каждого такого элемента представляет линейную комбинацию выходов A-элементов:

$$R_j = \Theta_j + \sum_{i=1}^n v_{ij} x_i, \quad (\Phi. 1)$$

где  $R_j$  — реакция  $j$ -го R-элемента;  $x_i$  — реакция  $i$ -го A-элемента;  $v_{ij}$  — вес связи от  $i$ -го A-элемента к  $j$ -му R элементу;  $\Theta_j$  — порог  $j$ -го R-элемента.

Аналогично записывается уравнение  $i$ -го A-элемента:

$$x_i = \Theta_i + \sum_{k=1}^S y_k, \quad (\Phi. 2)$$

Здесь сигнал  $y_k$  может быть непрерывным, но чаще всего он принимает только два значения: 0 или 1. Сигналы от S-элементов подаются на входы A-элементов с постоянными весами равными единице, но каждый A-элемент связан только с группой случайно выбранных S-элементов. Предположим, что требуется обучить перцептрон различать два образа  $V_1$  и  $V_2$ . Будем считать, что в перцептроне существует два R-элемента, один из которых предназначен образу  $V_1$ , а другой — образу  $V_2$ . Перцептрон будет обучен правильно, если выход  $R_1$  превышает  $R_2$ , когда распознаваемый объект принадлежит образу  $V_1$ , и наоборот. Разделение объектов на два образа можно провести и с помощью только одного R-элемента. Тогда объекту образа  $V_1$  должна соответствовать положительная реакция R-элемента, а объектам образа  $V_2$  — отрицательная.

Перцептрон обучается путем предъявления обучающей последовательности изображений объектов, принадлежащих образам  $V_1$  и  $V_2$ . В процессе обучения изменяются веса  $v_i$  A-элементов. В частности, если применяется система подкрепления с коррекцией ошибок, прежде всего учитывается правильность решения, принимаемого перцептроном. Если решение правильно, то веса связей всех сработавших A-элементов, ведущих к R-элементу, выдавшему правильное решение, увеличиваются, а веса несработавших A-элементов остаются неизменными. Можно оставлять неизменными веса сработавших A-элементов, но уменьшать веса несработавших. В некоторых случаях веса сработавших связей увеличивают, а несработавших — уменьшают. После процесса обучения перцептрон сам, без учителя, начинает классифицировать новые объекты.

Если перцептрон действует по описанной схеме и в нем допускаются лишь связи, идущие от бинарных S-элементов к A-элементам и от A-элементов к единственному R-элементу, то такой перцептрон принято называть элементарным  $\alpha$ -перцептроном. Обычно классификация  $C(W)$  задается учителем. Перцептрон должен выработать в процессе обучения классификацию, задуманную учителем.

О перцептронах было сформулировано и доказано несколько основополагающих теорем, две из которых, определяющие основные свойства перцептрона, приведены ниже.

**Теорема 1.** Класс элементарных  $\alpha$ -перцептронов, для которых существует решение для любой задуманной классификации, не является пустым.

Эта теорема утверждает, что для любой классификации обучающей последовательности можно подобрать такой набор (из бесконечного набора) A-элементов, в котором будет осуществлено задуманное разделение обучающей последовательности при помощи линейного решающего правила).

**Теорема 2.** Если для некоторой классификации  $C(W)$  решение существует, то в процессе обучения  $\alpha$ -перцептрона с коррекцией ошибок, начинающегося с произвольного исходного состояния, это решение будет достигнуто в течение конечного промежутка времени.

Смысл этой теоремы состоит в том, что если относительно задуманной классификации можно найти набор A-элементов, в котором существует решение, то в рамках этого набора оно будет достигнуто в конечный промежуток времени.

Обычно обсуждают свойства бесконечного перцептрона, т. е. перцептрона с бесконечным числом A-элементов со всевозможными связями с S-элементами

(полный набор А-элементов). В таких перцептронах решение всегда существует, а раз оно существует, то оно и достижимо в  $\alpha$ -перцептронах с коррекцией ошибок.

Очень интересную область исследований представляют собой многослойные перцептроны и перцептроны с перекрестными связями, но теория этих систем практически еще не разработана.

### ***Нейронные сети***

#### **История исследований в области нейронных сетей**

Возвратимся немного назад, и рассмотрим историю исследований нейронных сетей.

В истории исследований в области нейронных сетей, как и в истории любой другой науки, были свои успехи и неудачи. Кроме того, здесь постоянно сказывается психологический фактор, проявляющийся в неспособности человека описать словами то, как он думает.

Способность нейронной сети к обучению впервые исследована Дж. Маккалоком и У. Питтом. В 1943 году вышла их работа “Логическое исчисление идей, относящихся к нервной деятельности”, в которой была построена модель нейрона, и сформулированы принципы построения искусственных нейронных сетей.

Крупный толчок развитию нейрокибернетики дал американский нейрофизиолог Френк Розенблатт, предложивший в 1962 году свою модель нейронной сети — персеptron. Воспринятый первоначально с большим энтузиазмом, он вскоре подвергся интенсивным нападкам со стороны крупных научных авторитетов. И хотя подробный анализ их аргументов показывает, что они оспаривали не совсем тот персеptron, который предлагал Розенблатт, крупные исследования по нейронным сетям были свернуты почти на 10 лет.

Несмотря на это в 70-е годы было предложено много интересных разработок, таких, например, как когнитрон, способный хорошо распознавать достаточно сложные образы независимо от поворота и изменения масштаба изображения.

В 1982 году американский биофизик Дж. Хопфилд предложил оригинальную модель нейронной сети, названную его именем. В последующие несколько лет было найдено множество эффективных алгоритмов: сеть встречного потока, двунаправленная ассоциативная память и др.

В киевском институте кибернетики с 70-х годов ведутся работы над стохастическими нейронными сетями.

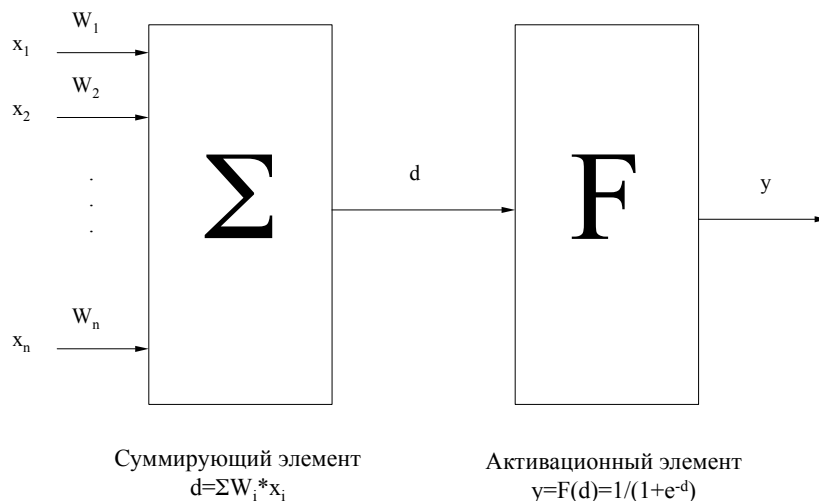
#### **Модель нейронной сети с обратным распространением ошибки (back propagation)**

В 1986 году Дж. Хинтон и его коллеги опубликовали статью с описанием модели нейронной сети и алгоритмом ее обучения, что дало новый толчок исследованиям в области искусственных нейронных сетей.

Нейронная сеть состоит из множества одинаковых элементов — нейронов, поэтому начнем с них рассмотрение работы искусственной нейронной сети.

Биологический нейрон моделируется как устройство, имеющее несколько входов (дендриты), и один выход (аксон). Каждому входу ставится в соответствие некоторый весовой коэффициент ( $w$ ), характеризующий пропускную способность канала и оценивающий степень влияния сигнала с этого входа на сигнал на

выходе. В зависимости от конкретной реализации, обрабатываемые нейроном сигналы могут быть аналоговыми или цифровыми (1 или 0). В теле нейрона происходит взвешенное суммирование входных возбуждений, и далее это значение является аргументом активационной функции нейрона, один из возможных вариантов которой представлен на Рис. 4.



**Рис. 4 Искусственный нейрон**

Будучи соединенными определенным образом, нейроны образуют нейронную сеть. Работа сети разделяется на обучение и адаптацию. Под обучением понимается процесс адаптации сети к предъявляемым эталонным образцам путем модификации (в соответствии с тем или иным алгоритмом) весовых коэффициентов связей между нейронами. Заметим, что этот процесс является результатом алгоритма функционирования сети, а не предварительно заложенных в нее знаний человека, как это часто бывает в системах искусственного интеллекта.

Среди различных структур нейронных сетей (НС) одной из наиболее известных является многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми аксонами нейронов предыдущего слоя или, в случае первого слоя, со всеми входами НС. Такие НС называются полносвязными. Когда в сети только один слой, алгоритм ее обучения с учителем довольно очевиден, так как правильные выходные состояния нейронов единственного слоя заведомо известны, и подстройка синаптических связей идет в направлении, минимизирующем ошибку на выходе сети. По этому принципу строится, например, алгоритм обучения однослойного перцептрона. В многослойных же сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, не известны, и двух или более слойный перцептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах НС. Один из вариантов решения этой проблемы – разработка наборов выходных сигналов, соответствующих входным, для каждого слоя НС, что, конечно, является очень трудоемкой операцией и не всегда осуществимо. Второй вариант – динамическая подстройка весовых коэффициентов синапсов, в ходе которой выбираются, как правило, наиболее слабые связи и изменяются на малую

величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Очевидно, что данный метод "тыка", несмотря на свою кажущуюся простоту, требует громоздких рутинных вычислений. И, наконец, третий, более приемлемый вариант – распространение сигналов ошибки от выходов НС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения НС получил название процедуры обратного распространения. Именно он будет рассмотрен в дальнейшем.

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки НС является величина:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2 \quad (1)$$

где  $y_{j,p}^{(N)}$  – реальное выходное состояние нейрона  $j$  выходного слоя  $N$  нейронной сети при подаче на ее входы  $p$ -го образа;  $d_{jp}$  – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad (2)$$

Здесь  $w_{ij}$  – весовой коэффициент синаптической связи, соединяющей  $i$ -ый нейрон слоя  $n-1$  с  $j$ -ым нейроном слоя  $n$ ,  $\eta$  – коэффициент скорости обучения,  $0 < \eta < 1$ .

Как показано в [2],

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{ij}} \quad (3)$$

Здесь под  $y_j$ , как и раньше, подразумевается выход нейрона  $j$ , а под  $s_j$  – взвешенная сумма его входных сигналов, то есть аргумент активационной функции. Так как множитель  $dy_j/ds_j$  является производной этой функции по ее аргументу, из этого следует, что производная активационной функции должна быть определена на всей оси абсцисс. В связи с этим функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых НС. В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид с экспонентой. В случае гиперболического тангенса

$$\frac{dy}{ds} = 1 - s^2 \quad (4)$$

Третий множитель  $\partial s_j / \partial w_{ij}$ , очевидно, равен выходу нейрона предыдущего слоя  $y_i^{(n-1)}$ .

Что касается первого множителя в (3), он легко раскладывается следующим образом [2]:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot w_{jk}^{(n+1)} \quad (5)$$

Здесь суммирование по  $k$  выполняется среди нейронов слоя  $n+1$ .

Введя новую переменную

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \quad (6)$$

мы получим рекурсивную формулу для расчетов величин  $\delta_j^{(n)}$  слоя n из величин  $\delta_k^{(n+1)}$  более старшего слоя n+1.

$$\delta_j^{(n)} = \left[ \sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j} \quad (7)$$

Для выходного же слоя

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l} \quad (8)$$

Теперь мы можем записать (2) в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \quad (9)$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (9) дополняется значением изменения веса на предыдущей итерации

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) \quad (10)$$

где  $\mu$  – коэффициент инерционности, t – номер текущей итерации.

Таким образом, полный алгоритм обучения НС с помощью процедуры обратного распространения строится так:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования НС, когда сигналы распространяются от входов к выходам, рассчитать значения последних. Напомним, что

$$s_j^{(n)} = \sum_{i=0}^M y_i^{(n-1)} \cdot w_{ij}^{(n)} \quad (11)$$

где M – число нейронов в слое n-1 с учетом нейрона с постоянным выходным состоянием +1, задающего смещение;  $y_i^{(n-1)} = x_{ij}^{(n)}$  – i-ый вход нейрона j слоя n.

$$y_j^{(n)} = f(s_j^{(n)}), \text{ где } f() \text{ – сигмоид} \quad (12)$$

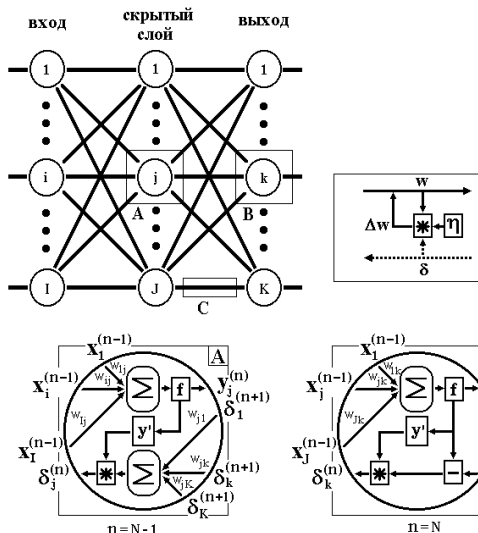
$$y_q^{(0)} = I_q, \quad (13)$$

где  $I_q$  – q-ая компонента вектора входного образа.

2. Рассчитать  $\delta^{(N)}$  для выходного слоя по формуле (8).

Рассчитать по формуле (9) или (10) изменения весов  $\Delta w^{(N)}$  слоя N.

**Рис. 5. Диаграмма сигналов в сети при обучении по алгоритму обратного распространения**



3. Рассчитать по формулам (7) и (9) (или (7) и (10)) соответственно  $\delta^{(n)}$  и  $\Delta w^{(n)}$  для всех остальных слоев, n=N-1,...1.

4. Скорректировать все веса в НС

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) \quad (14)$$

5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Сети на шаге 1 попеременно в случайном порядке предъявляются все тренировочные образы, чтобы сеть, образно говоря, не забывала одни по мере запоминания других. Алгоритм иллюстрируется Рис. 5.



Из выражения (9) следует, что когда выходное значение  $y_i^{(n-1)}$  стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться [3], поэтому область возможных значений выходов нейронов  $[0,1]$  желательно сдвинуть в пределы  $[-0.5,+0.5]$ , что достигается простыми модификациями логистических функций. Например, сигмоид с экспонентой преобразуется к виду

$$f(x) = -0.5 + \frac{1}{1 + e^{-\alpha x}} \quad (15)$$

Теперь коснемся вопроса емкости НС, то есть числа образов, предъявляемых на ее входы, которые она способна научиться распознавать. Для сетей с числом слоев больше двух, он остается открытым. Как показано в [4], для НС с двумя слоями, то есть выходным и одним скрытым слоем, детерминистская емкость сети  $C_d$  оценивается так:

$$N_w/N_y < C_d < N_w/N_y \cdot \log(N_w/N_y) \quad (16)$$

где  $N_w$  – число подстраиваемых весов,  $N_y$  – число нейронов в выходном слое.

Следует отметить, что данное выражение получено с учетом некоторых ограничений. Во-первых, число входов  $N_x$  и нейронов в скрытом слое  $N_h$  должно удовлетворять неравенству  $N_x + N_h > N_y$ . Во-вторых,  $N_w/N_y > 1000$ . Однако вышеприведенная оценка выполнялась для сетей с активационными функциями нейронов в виде порога, а емкость сетей с гладкими активационными функциями, например – (15), обычно больше. Кроме того, фигурирующее в названии емкости прилагательное "детерминистский" означает, что полученная оценка емкости подходит абсолютно для всех возможных входных образов, которые могут быть представлены  $N_x$  входами. В действительности распределение входных образов, как правило, обладает некоторой регулярностью, что позволяет НС проводить обобщение и, таким образом, увеличивать реальную емкость. Так как распределение образов, в общем случае, заранее не известно, мы можем говорить о такой емкости только предположительно, но обычно она раза в два превышает емкость детерминистскую.

В продолжение разговора о емкости НС логично затронуть вопрос о требуемой мощности выходного слоя сети, выполняющего окончательную классификацию образов. Дело в том, что для разделения множества входных образов, например, по двум классам достаточно всего одного выхода. При этом каждый логический уровень – "1" и "0" – будет обозначать отдельный класс. На двух выходах можно закодировать уже 4 класса и так далее. Однако результаты работы сети, организованной таким образом, можно сказать – "под завязку", – не очень надежны. Для повышения достоверности классификации желательно ввести избыточность путем выделения каждому классу одного нейрона в выходном слое или, что еще лучше, нескольких, каждый из которых обучается определять принадлежность образа к классу со своей степенью достоверности, например: высокой, средней и низкой. Такие НС позволяют проводить классификацию входных образов, объединенных в нечеткие (размытые или пересекающиеся) множества. Это свойство приближает подобные НС к условиям реальной жизни.

Рассматриваемая НС имеет несколько "узких мест". Во-первых, в процессе обучения может возникнуть ситуация, когда большие положительные или отрицательные значения весовых коэффициентов сместят рабочую точку на сигмоидах многих нейронов в область насыщения. Малые величины производной от логистической функции приведут в соответствие с (7) и (8) к остановке обучения, что парализует НС. Во-вторых, применение метода градиентного спуска не гарантирует, что будет найден глобальный, а не локальный минимум целевой функции. Эта проблема связана еще с одной, а именно – с выбором величины скорости обучения. Доказательство сходимости обучения в процессе обратного распространения основано на производных, то есть приращения весов и, следовательно, скорость обучения должны быть бесконечно малыми, однако в этом случае обучение будет происходить неприемлемо медленно. С другой стороны, слишком большие коррекции весов могут привести к постоянной неустойчивости процесса обучения. Поэтому в качестве  $\eta$  обычно выбирается число меньше 1, но не очень маленькое, например, 0.1, и оно, вообще говоря, может постепенно уменьшаться в процессе обучения. Кроме того, для исключения случайных попаданий в локальные минимумы иногда, после того как значения весовых коэффициентов стабилизируются,  $\eta$  кратковременно сильно увеличивают, чтобы начать градиентный спуск из новой точки. Если повторение этой процедуры несколько раз приведет алгоритм в одно и то же состояние НС, можно более или менее уверенно сказать, что найден глобальный максимум, а не какой-то другой.

Существует и иной метод исключения локальных минимумов, а заодно и парализа НС, заключающийся в применении стохастических НС, но о них лучше поговорить отдельно.

### **Нейронные сети: обучение без учителя**

Рассмотренный в предыдущей главе алгоритм обучения нейронной сети с помощью процедуры обратного распространения подразумевает наличие некоего внешнего звена, предоставляющего сети кроме входных так же и целевые выходные образы. Алгоритмы, пользующиеся подобной концепцией, называются алгоритмами обучения с учителем. Для их успешного функционирования необходимо наличие экспертов, создающих на предварительном этапе для каждого входного образа эталонный выходной. Так как создание искусственного интеллекта движется по пути копирования природных прообразов, ученые не прекращают спор на тему, можно ли считать алгоритмы обучения с учителем натуральными или же они полностью искусственны. Например, обучение человеческого мозга, на первый взгляд, происходит без учителя: на зрительные, слуховые, тактильные и прочие рецепторы поступает информация извне, и внутри нервной системы происходит некая самоорганизация. Однако, нельзя отрицать и того, что в жизни человека не мало учителей – и в буквальном, и в переносном смысле, – которые координируют внешние воздействия. Вместе в тем, чем бы ни закончился спор приверженцев этих двух концепций обучения, они обе имеют право на существование.

Главная черта, делающая обучение без учителя привлекательным, – это его "самостоятельность". Процесс обучения, как и в случае обучения с учителем, заключается в подстраивании весов синапсов. Некоторые алгоритмы, правда,

изменяют и структуру сети, то есть количество нейронов и их взаимосвязи, но такие преобразования правильнее назвать более широким термином – самоорганизацией, и в рамках данной главы они рассматриваться не будут. Очевидно, что подстройка синапсов может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весовых коэффициентов. Исходя из этого соображения и, что более важно, по аналогии с известными принципами самоорганизации нервных клеток, построены алгоритмы обучения Хебба.

Сигнальный метод обучения Хебба заключается в изменении весов по следующему правилу:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot y_i^{(n-1)} \cdot y_j^{(n)} \quad (1)$$

где  $y_i^{(n-1)}$  – выходное значение нейрона  $i$  слоя  $(n-1)$ ,  $y_j^{(n)}$  – выходное значение нейрона  $j$  слоя  $n$ ;  $w_{ij}(t)$  и  $w_{ij}(t-1)$  – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях  $t$  и  $t-1$  соответственно;  $\alpha$  – коэффициент скорости обучения. Здесь и далее, для общности, под  $n$  подразумевается произвольный слой сети. При обучении по данному методу усиливаются связи между возбужденными нейронами.

Существует также и дифференциальный метод обучения Хебба.

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot [y_i^{(n-1)}(t) - y_i^{(n-1)}(t-1)] \cdot [y_j^{(n)}(t) - y_j^{(n)}(t-1)] \quad (2)$$

Здесь  $y_i^{(n-1)}(t)$  и  $y_i^{(n-1)}(t-1)$  – выходное значение нейрона  $i$  слоя  $n-1$  соответственно на итерациях  $t$  и  $t-1$ ;  $y_j^{(n)}(t)$  и  $y_j^{(n)}(t-1)$  – то же самое для нейрона  $j$  слоя  $n$ . Как видно из формулы (2), сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения с применением вышеприведенных формул будет выглядеть так:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.

2. На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямопоточных (feedforward) сетей[1], то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется активационная (передаточная) функция нейрона, в результате чего получается его выходное значение  $y_i^{(n)}$ ,  $i=0...M_i-1$ , где  $M_i$  – число нейронов в слое  $i$ ;  $n=0...N-1$ , а  $N$  – число слоев в сети.

3. На основании полученных выходных значений нейронов по формуле (1) или (2) производится изменение весовых коэффициентов.

4. Цикл с шага 2, пока выходные значения сети не стабилизируются с заданной точностью. Применение этого нового способа определения завершения обучения, отличного от использовавшегося для сети обратного распространения, обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

На втором шаге цикла попеременно предъявляются все образы из входного набора.

Следует отметить, что вид откликов на каждый класс входных образов не известен заранее и будет представлять собой произвольное сочетание состояний нейронов выходного слоя, обусловленное случайным распределением весов на стадии инициализации. Вместе с тем, сеть способна обобщать схожие образы, относя их к одному классу. Тестирование обученной сети позволяет определить топологию классов в выходном слое. Для приведения откликов обученной сети к удобному представлению можно дополнить сеть одним слоем, который, например, по алгоритму обучения однослойного перцептрона необходимо заставить отображать выходные реакции сети в требуемые образы.

Другой алгоритм обучения без учителя – алгоритм Кохонена – предусматривает подстройку синапсов на основании их значений от предыдущей итерации.

$$w_{ij}(t) = w_{ij}(t - 1) + \alpha \cdot [y_i^{(n-1)} - w_{ij}(t - 1)] \quad (3)$$

Из вышеприведенной формулы видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя  $y_i^{(n-1)}$ , и весовыми коэффициентами его синапсов.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально походят на входной образ, и подстройка весов по формуле (3) проводится только для него. Эта, так называемая, аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение. Выбор такого нейрона может осуществляться, например, расчетом скалярного произведения вектора весовых коэффициентов с вектором входных значений. Максимальное произведение дает выигравший нейрон.

Другой вариант – расчет расстояния между этими векторами в  $p$ -мерном пространстве, где  $p$  – размер векторов.

$$D_j = \sqrt{\sum_{i=0}^{p-1} (y_i^{(n-1)} - w_{ij})^2}, \quad (4)$$

где  $j$  – индекс нейрона в слое  $n$ ,  $i$  – индекс суммирования по нейронам слоя  $(n-1)$ ,  $w_{ij}$  – вес синапса, соединяющего нейроны; выходы нейронов слоя  $(n-1)$  являются входными значениями для слоя  $n$ . Корень в формуле (4) брать не обязательно, так как важна лишь относительная оценка различных  $D_j$ .

В данном случае, "побеждает" нейрон с наименьшим расстоянием. Иногда слишком часто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы "уравнять права" всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона.

При использовании обучения по алгоритму Кохонена существует практика нормализации входных образов, а так же – на стадии инициализации – и нормализации начальных значений весовых коэффициентов.

$$x_i = x_i / \sqrt{\sum_{j=0}^{n-1} x_j^2}, \quad (5)$$

где  $x_i$  –  $i$ -ая компонента вектора входного образа или вектора весовых коэффициентов, а  $n$  – его размерность. Это позволяет сократить длительность процесса обучения.

Инициализация весовых коэффициентов случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса. Для избежания такой ситуации используется метод выпуклой комбинации[3]. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \alpha(t) \cdot x_i + (1 - \alpha(t)) \cdot \frac{1}{\sqrt{n}}, \quad (6)$$

где  $x_i$  –  $i$ -ая компонента входного образа,  $n$  – общее число его компонент,  $\alpha(t)$  – коэффициент, изменяющийся в процессе обучения от нуля до единицы, в результате чего вначале на входы сети подаются практически одинаковые образы, а с течением времени они все больше сходятся к исходным. Весовые коэффициенты устанавливаются на шаге инициализации равными величине

$$w_o = \frac{1}{\sqrt{n}}, \quad (7)$$

где  $n$  – размерность вектора весов для нейронов инициализируемого слоя.

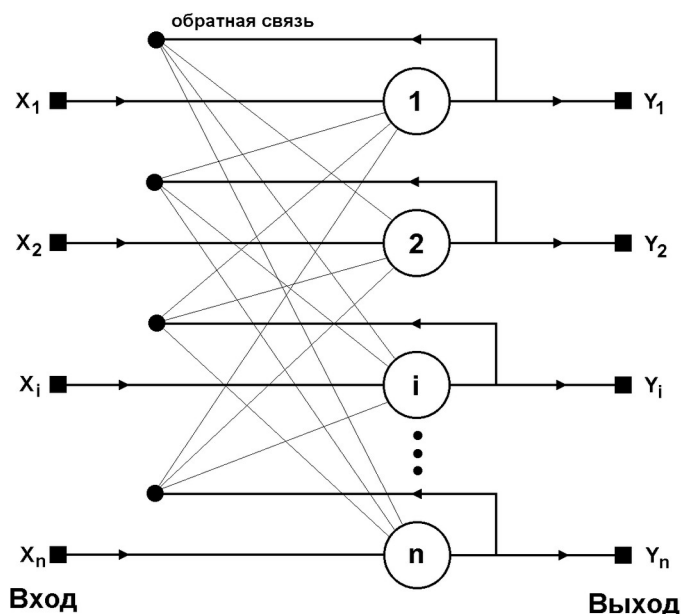
На основе рассмотренного выше метода строятся нейронные сети особого типа – так называемые самоорганизующиеся структуры – self-organizing feature maps (этот устоявшийся перевод с английского, на мой взгляд, не очень удачен, так как, речь идет не об изменении структуры сети, а только о подстройке синапсов). Для них после выбора из слоя  $n$  нейрона  $j$  с минимальным расстоянием  $D_j$  (4) обучается по формуле (3) не только этот нейрон, но и его соседи, расположенные в окрестности  $R$ . Величина  $R$  на первых итерациях очень большая, так что обучаются все нейроны, но с течением времени она уменьшается до нуля. Таким образом, чем ближе конец обучения, тем точнее определяется группа нейронов, отвечающих каждому классу образов.

### ***Нейронные сети Хопфилда и Хэмминга***

Среди различных конфигураций искусственных нейронных сетей (НС) встречаются такие, при классификации которых по принципу обучения, строго говоря, не подходят ни обучение с учителем, ни обучение без учителя. В таких сетях весовые коэффициенты синапсов рассчитываются только однажды перед началом функционирования сети на основе информации об обрабатываемых данных, и все обучение сети сводится именно к этому расчету. С одной стороны, предъявление априорной информации можно расценивать, как помощь учителя, но с другой – сеть фактически просто запоминает образцы до того, как на ее вход поступают реальные данные, и не может изменять свое поведение, поэтому говорить о звене обратной связи с "миром" (учителем) не приходится. Из сетей с подобной логикой работы наиболее известны сеть Хопфилда и сеть Хэмминга, которые обычно используются для организации ассоциативной памяти. Далее речь пойдет именно о них.

Структурная схема сети Хопфилда приведена на Рис. 6. Она состоит из единственного слоя нейронов, число которых является одновременно числом входов и выходов сети. Каждый нейрон связан синапсами со всеми остальными нейронами, а также имеет один входной синапс, через который осуществляется ввод сигнала. Выходные сигналы, как обычно, образуются на аксонах.

**Рис. 6. Структурная схема сети Хопфилда.**



Задача, решаемая данной сетью в качестве ассоциативной памяти, как правило, формулируется следующим образом. Известен некоторый набор двоичных сигналов (изображений, звуковых оцифровок, прочих данных, описывающих некие объекты или характеристики процессов), которые считаются образцовыми. Сеть должна уметь из произвольного неидеального сигнала, поданного на ее вход, выделить ("вспомнить" по частичной информации) соответствующий образец (если такой есть) или "дать заключение" о том, что входные данные не соответствуют ни одному из образцов. В общем случае, любой сигнал может быть описан вектором  $\mathbf{X} = \{x_i; i=0...n-1\}$ ,  $n$  – число нейронов в сети и размерность входных и выходных векторов. Каждый элемент  $x_i$  равен либо  $+1$ , либо  $-1$ . Обозначим вектор, описывающий  $k$ -ый образец, через  $\mathbf{X}^k$ , а его компоненты, соответственно, –  $x_i^k$ ,  $k=0...m-1$ ,  $m$  – число образцов. Когда сеть распознаёт (или "вспомнит") какой-либо образец на основе предъявленных ей данных, ее выходы будут содержать именно его, то есть  $\mathbf{Y} = \mathbf{X}^k$ , где  $\mathbf{Y}$  – вектор выходных значений сети:  $\mathbf{Y} = \{y_i; i=0,...n-1\}$ . В противном случае, выходной вектор не совпадет ни с одним образцовым.

Если, например, сигналы представляют собой некие изображения, то, отобразив в графическом виде данные с выхода сети, можно будет увидеть картинку, полностью совпадающую с одной из образцовых (в случае успеха) или же "вольную импровизацию" сети (в случае неудачи).

На стадии инициализации сети весовые коэффициенты синапсов устанавливаются следующим образом:

$$w_{ij} = \begin{cases} \sum_{k=0}^{m-1} x_i^k x_j^k, & i \neq j \\ 0, & i = j \end{cases} \quad (1)$$

Здесь  $i$  и  $j$  – индексы, соответственно, пресинаптического и постсинаптического нейронов;  $x_i^k, x_j^k$  –  $i$ -ый и  $j$ -ый элементы вектора  $k$ -ого образца.

Алгоритм функционирования сети следующий ( $p$  – номер итерации):

1. На входы сети подается неизвестный сигнал. Фактически его ввод осуществляется непосредственной установкой значений аксонов:

$$y_i(0) = x_i, \quad i = 0 \dots n-1, \quad (2)$$

поэтому обозначение на схеме сети входных синапсов в явном виде носит чисто условный характер. Ноль в скобке справа от  $y_i$  означает нулевую итерацию в цикле работы сети.

2. Рассчитывается новое состояние нейронов

$$s_j(p+1) = \sum_{i=0}^{n-1} w_{ij} y_i(p), \quad j=0 \dots n-1 \quad (3)$$

и новые значения аксонов

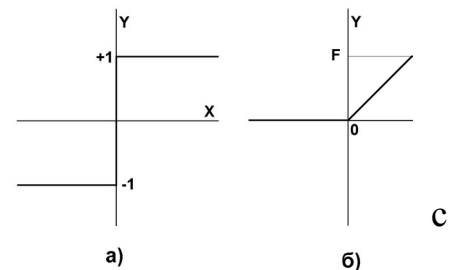
$$y_j(p+1) = f[s_j(p+1)] \quad (4)$$

где  $f$  – активационная функция в виде скачка, приведенная на Рис. 7а.

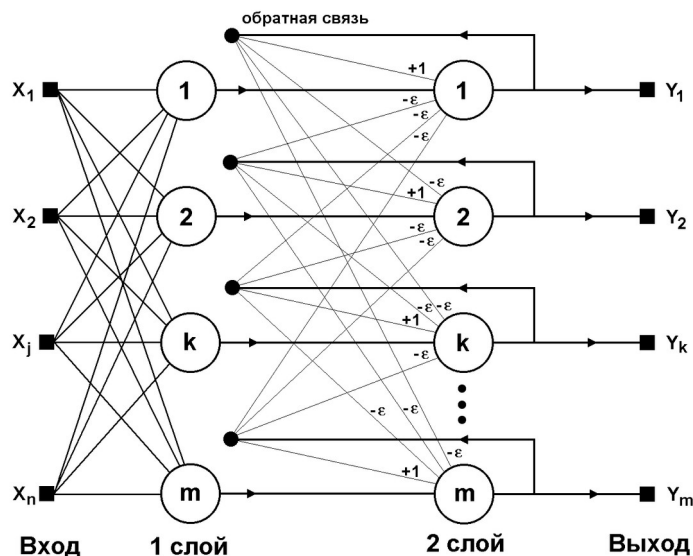
3. Проверка, изменились ли выходные значения аксонов за последнюю итерацию. Если да – переход к пункту 2, иначе (если выходы застabilizировались) – конец. При этом выходной вектор представляет собой образец, наилучшим образом сочетающийся входными данными.

Как говорилось выше, иногда сеть не может провести распознавание и выдает на выходе несуществующий образ. Это связано с проблемой ограниченности возможностей сети. Для сети Хопфилда число запоминаемых образов  $m$  не должно превышать величины, примерно равной  $0.15 \cdot n$ . Кроме того, если два образа А и Б сильно похожи, они, возможно, будут вызывать у сети перекрестные ассоциации, то есть предъявление на входы сети вектора А приведет к появлению на ее выходах вектора Б и наоборот.

**Рис. 7. Активационные функции.**



**Рис. 8. Структурная схема сети Хэмминга.**



Когда нет необходимости, чтобы сеть в явном виде выдавала образец, то есть достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, меньшими затратами на память и объемом вычислений, что становится очевидным из ее структуры (Рис. 8).

Сеть состоит из двух слоев. Первый и второй слои имеют по  $m$  нейронов, где  $m$  – число образцов. Нейроны первого слоя имеют по  $n$  синапсов, соединенных со входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$$w_{ik} = \frac{x_i^k}{2}, i=0...n-1, k=0...m-1 \tag{5}$$

$$T_k = n / 2, k = 0...m-1 \tag{6}$$

Здесь  $x_i^k$  –  $i$ -ый элемент  $k$ -ого образца.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине  $0 < \epsilon < 1/m$ . Синапс нейрона, связанный с его же аксоном имеет вес  $+1$ .

Алгоритм функционирования сети Хэмминга следующий:

1. На входы сети подается неизвестный вектор  $\mathbf{X} = \{x_i; i=0...n-1\}$ , исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):



$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, j=0...m-1 \quad (7)$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$$y_j^{(2)} = y_j^{(1)}, j = 0...m-1 \quad (8)$$

2. Вычислить новые состояния нейронов второго слоя:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), k \neq j, j = 0...m-1 \quad (9)$$

и значения их аксонов:

$$y_j^{(2)}(p+1) = f[s_j^{(2)}(p+1)], j = 0...m-1 \quad (10)$$

Активационная функция  $f$  имеет вид порога (рис. 2б), причем величина  $F$  должна быть достаточно большой, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да – перейди к шагу 2. Иначе – конец.

Из оценки алгоритма видно, что роль первого слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

### **Метод потенциальных функций**

Предположим, что требуется разделить два непересекающихся образа  $V_1$  и  $V_2$ . Это значит, что в пространстве изображений существует, по крайней мере, одна функция, которая полностью разделяет множества, соответствующие образам  $V_1$  и  $V_2$ . Эта функция должна принимать положительные значения в точках, соответствующих объектам, принадлежащим образу  $V_1$ , и отрицательные — в точках образа  $V_2$ . В общем случае таких разделяющих функций может быть много, тем больше, чем компактней разделяемые множества. В процессе обучения требуется построить одну из этих функций, иногда в некотором смысле наилучшую.

Метод потенциальных функций связан со следующей процедурой. В процессе обучения с каждой точкой пространства изображений, соответствующей единичному объекту из обучающей последовательности, связывается функция  $U(X, X_i)$ , заданная на всем пространстве и зависящая от  $X_i$  как от параметра. Такие функции называются потенциальными, так как они напоминают функции потенциала электрического поля вокруг точечного электрического заряда. Изменение потенциала электрического поля по мере удаления от заряда обратно пропорционально квадрату расстояния. Потенциал, таким образом, может служить мерой удаления точки от заряда. Когда поле образовано несколькими зарядами, потенциал в каждой точке этого поля равен сумме потенциалов, создаваемых в этой точке каждым из зарядов. Если заряды, образующие поле, расположены компактной группой, потенциал поля будет иметь наибольшее значение внутри группы зарядов и убывать по мере удаления от нее.

Обучающей последовательности объектов соответствует последовательность векторов  $X_1, X_2, \dots$ , в пространстве изображений с которыми связана последовательность  $U(X, X_1), U(X, X_2), \dots$  потенциальных функций,

используемых для построения функций  $f(X_1, X_2, \dots)$ . По мере увеличения числа объектов в процессе обучения функция  $f$  должна стремиться к одной из разделяющих функций. В результате обучения могут быть построены потенциальные функции для каждого образа:

$$U_1(X) = \sum_{X_i \in E_1} U(X, X_i), \quad U_2(X) = \sum_{X_i \in E_2} U(X, X_i), \quad (\Phi. 3)$$

В качестве разделяющей функции  $f(X)$  можно выбрать функцию вида:

$$f(X) = U_1(X) - U_2(X), \quad (\Phi. 4)$$

которая положительна для объектов одного образа и отрицательна для объектов другого.

В качестве потенциальной функции рассмотрим функцию вида

$$U(X, X_i) = \sum_{j=1}^{\infty} \lambda_j^2 \varphi_j(X) \varphi_j(X_i) = \sum_{j=1}^{\infty} \psi_j(X) \psi_j(X_i),$$

(Φ. 5)

где  $\varphi_j(X)$  — линейно независимая система функций;  $\lambda_j$  — действительные числа, отличные от нуля для всех  $j = 1, 2, \dots$ ;  $X_i$  — точка, соответствующая  $i$ -му объекту из обучающей последовательности. Предполагается, что  $\varphi_j(X)$  и  $U(X, X_i)$  ограничены при  $X \in V_1 \cup V_2$ ;  $\psi_j(X) = \lambda_j \varphi_j(X)$ .

В процессе обучения предъявляется обучающая последовательность и на каждом  $n$ -м такте обучения строится приближение  $f_n(X)$  характеризуется следующей основной рекуррентной процедурой:

$$f_{n+1}(X) = q_n f_n(X) + r_n U(X_{n+1}, X), \quad (\Phi. 6)$$

Разновидности алгоритмов потенциальных функций отличаются выбором значений  $q_n$  и  $r_n$ , которые являются фиксированными функциями номера  $n$ . Как правило,  $q_n \equiv 1$ , а  $r_n$  выбирается в виде:

$$r_n \equiv \gamma_n (S(f_n(X_{n+1}), f(X_{n+1}))), \quad (\Phi. 7)$$

где  $S(f_n, f)$  — невозрастающие функции, причем

$$S(f, f) \equiv 0 \quad S(f_n, f) \begin{cases} \leq 0, & f_n \geq f, \\ \geq 0, & f_n \leq f. \end{cases} \quad (\Phi. 8)$$

Коэффициенты  $\gamma_n$  представляют собой неотрицательную числовую последовательность, зависящую только от номера  $n$ . Кроме того,  $\sum_{n=1}^{\infty} \gamma_n = \infty$  и

$\sum_{n=1}^{\infty} \gamma_n^2 < \infty$  (например,  $\gamma_n = 1/n$ ) или  $\gamma_n = \text{const}$ .

Разработано несколько вариантов алгоритмов потенциальных функций, различие между которыми состоит в выборе законов коррекции разделяющей функции от шага к шагу, т. е. в выборе законов коррекции разделяющей функции от шага к шагу, т. е. в выборе коэффициентов  $r_n$ . Приведем два основных алгоритма потенциальных функций.

1. Будем считать, что  $f_0(X) \equiv 0$  (нулевое приближение). Пусть в результате применения алгоритма после  $n$ -го шага построена разделяющая функция  $f_n(X)$ , а на  $(n+1)$ -м шаге предъявлено изображение  $X_{n+1}$ , для которого известно действительное значение разделяющей функции  $f(X_{n+1})$ . Тогда функция  $f_{n+1}(X)$  строится по следующему правилу:

$$f_{n+1}(X) = f_n(X) + \gamma_{n+1} \text{sign}(f(X_{n+1}) - f_n(X_{n+1})) \cdot U(X, X_{n+1}). \quad (\Phi. 9)$$

2. Во втором алгоритме также принимается, что  $f_0(X) \equiv 0$ . Переход к следующему приближению, т. е. переход от функции  $f_n(X)$  к  $f_{n+1}(X)$ , осуществляется в результате следующей рекуррентной процедуры:

$$f_{n+1}(X) = f_n(X) + (f(X_{n+1}) - f_n(X_{n+1})) \cdot \frac{1}{\lambda} U(X, X_{n+1}). \quad (\text{ф. 10})$$

где  $\lambda$  — произвольная положительная константа, удовлетворяющая условию  $\lambda = (1/2) \cdot \max(X, X_i)$ .

Если в (ф. 5) принять

$$\psi_j(X) = \text{sign}\left(\sum_{v=1}^m \beta_{vj} x_v + \Theta_j\right),$$

и предположить, что  $x_v$  может иметь только два значения 0 и 1, то в этом случае алгоритм потенциальных функций будет совпадать со схемой перцептрона с индивидуальными порогами А-элементов и с коррекцией ошибок. Поэтому многие теоретические положения метода потенциальных функций могут быть успешно применены для анализа некоторых перцептронных схем.

### **Метод группового учета аргументов МГУА**

#### **Метод наименьших квадратов**

Перед тем, как начинать рассмотрение МГУА, было бы полезно вспомнить или узнать впервые метод наименьших квадратов — наиболее распространенный метод подстройки линейно зависимых параметров.

Рассмотрим для примера МНК для трех аргументов:

Пусть функция  $T = T(U, V, W)$  задана таблицей, то есть из опыта известны числа  $U_i, V_i, W_i, T_i$  ( $i = 1, \dots, n$ ). Будем искать зависимость между этими данными в виде:

$$T(U, V, W) = aU + bV + cW \quad (\text{ф. 11})$$

где  $a, b, c$  — неизвестные параметры.

Подберем значения этих параметров так, чтобы была наименьшей сумма квадратов отклонений опытных данных  $T_i$  и теоретических  $T_i = aU_i + bV_i + cW_i$ , то есть сумма:

$$\sigma = \sum_{i=1}^n (T_i - aU_i - bV_i - cW_i)^2 \rightarrow \min \quad (\text{ф. 12})$$

Величина  $\sigma$  является функцией трех переменных  $a, b, c$ . Необходимым и достаточным условием существования минимума этой функции является равенство нулю частных производных функции  $\sigma$  по всем переменным, то есть:

$$\frac{\partial \sigma}{\partial a} = 0, \quad \frac{\partial \sigma}{\partial b} = 0, \quad \frac{\partial \sigma}{\partial c} = 0 \quad (\text{ф. 13})$$

Так как:

$$\begin{aligned} \frac{\partial \sigma}{\partial a} &= -2 \sum_{i=1}^n (T_i - aU_i - bV_i - cW_i) U_i \\ \frac{\partial \sigma}{\partial b} &= -2 \sum_{i=1}^n (T_i - aU_i - bV_i - cW_i) V_i \\ \frac{\partial \sigma}{\partial c} &= -2 \sum_{i=1}^n (T_i - aU_i - bV_i - cW_i) W_i \end{aligned} \quad (\text{ф. 14})$$

то система для нахождения  $a, b, c$  будет иметь вид:

$$\begin{aligned}
a \sum_{i=1}^n U_i^2 + b \sum_{i=1}^n U_i V_i + c \sum_{i=1}^n U_i W_i &= \sum_{i=1}^n T_i U_i \\
a \sum_{i=1}^n U_i V_i + b \sum_{i=1}^n V_i^2 + c \sum_{i=1}^n V_i W_i &= \sum_{i=1}^n T_i V_i \\
a \sum_{i=1}^n U_i W_i + b \sum_{i=1}^n W_i V_i + c \sum_{i=1}^n W_i^2 &= \sum_{i=1}^n T_i W_i
\end{aligned}
\tag{ф. 15}$$

Данная система решается любым стандартным методом решения систем линейных уравнений (Гаусса, Жордана, Зейделя, Крамера).

Рассмотрим некоторые практические примеры нахождения приближающих функций:

$$1. \quad y = \alpha x^2 + \beta x + \gamma$$

Задача подбора коэффициентов  $\alpha$ ,  $\beta$ ,  $\gamma$  сводится к решению общей задачи при  $T=y$ ,  $U=x^2$ ,  $V=x$ ,  $W=1$ ,  $\alpha=a$ ,  $\beta=b$ ,  $\gamma=c$ .

$$2. \quad f(x, y) = \alpha \sin(x) + \beta \cos(y) + \gamma/x$$

Задача подбора коэффициентов  $\alpha$ ,  $\beta$ ,  $\gamma$  сводится к решению общей задачи при  $T=f$ ,  $U=\sin(x)$ ,  $V=\cos(y)$ ,  $W=1/x$ ,  $\alpha=a$ ,  $\beta=b$ ,  $\gamma=c$ .

Если мы распространим МНК на случай с  $m$  параметрами,

$$\sigma = \sum_{i=1}^n \left( T_i - \sum_{v=1}^m u_{iv} c_v \right)^2 \rightarrow \min \tag{ф. 16}$$

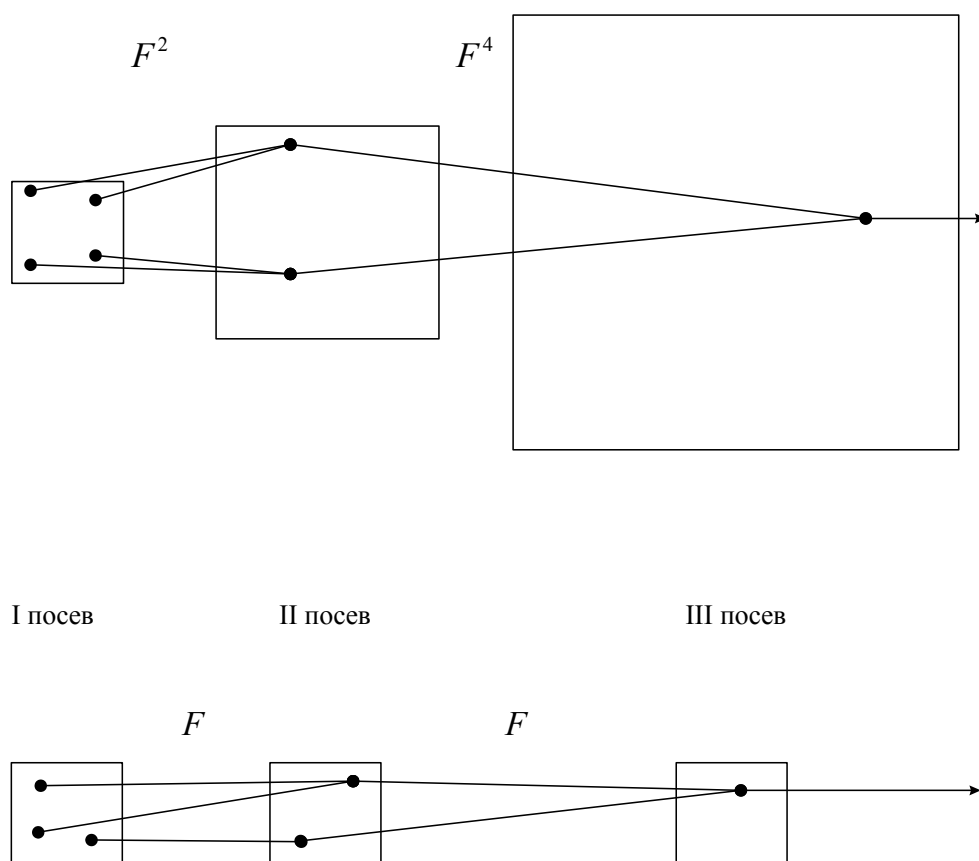
то путем рассуждений, аналогичных приведенным выше, получим следующую систему линейных уравнений:

$$\begin{cases}
c_1 \overline{u_1 u_1} + c_2 \overline{u_1 u_2} + \dots + c_m \overline{u_1 u_m} = \overline{T u_1} \\
c_1 \overline{u_2 u_1} + c_2 \overline{u_2 u_2} + \dots + c_m \overline{u_2 u_m} = \overline{T u_2} \\
\dots \\
c_1 \overline{u_m u_1} + c_2 \overline{u_m u_2} + \dots + c_m \overline{u_m u_m} = \overline{T u_m}
\end{cases}
\tag{ф. 17}$$

где  $\overline{T} = \{ T_i \}_{i=1}^n$ ,  $\overline{u}_v = \{ u_{iv} \}_{i=1}^n$

## Общая схема построения алгоритмов метода группового учета аргументов (МГУА).

Заимствование алгоритмов переработки информации у природы является одной из основных идей кибернетики. "Гипотеза селекции" утверждает, что алгоритм массовой селекции растений или животных является оптимальным алгоритмом переработки информации в сложных задачах. При массовой селекции высевается некоторое количество семян. В результате опыления образуются сложные наследственные комбинации. Селекционеры выбирают некоторую часть растений, у которых интересное их свойство выражено больше всего (эвристический критерий). Семена этих растений собирают и снова высевают для



**Рис. 9. Селекция самого черного тюльпана при расширяющемся опытном поле (эквивалент полного перебора), и при постоянном размере поля (эквивалент селекции при сохранении свободы выбора решений  $F = \text{const}$ ).**

образования новых, еще более сложных комбинаций. Через несколько поколений селекция останавливается и ее результат является оптимальным. Если чрезмерно продолжать селекцию, то наступит «инцухт» — вырождение растений. Существует оптимальное число поколений и оптимальное количество семян, отбираемых в каждом из них.

Алгоритмы МГУА воспроизводят схему массовой селекции [5], показанной на Рис. 9. В них есть генераторы усложняющихся из ряда в ряд комбинаций и

пороговые самоотборы лучших из них. Так называемое «полное» описание объекта

$$\varphi = f(x_1, x_2, x_3, \dots, x_m),$$

где  $f$  — некоторая элементарная функция, например степенной полином, заменяется несколькими рядами "частных" описаний:

$$1\text{-ряд селекции: } y_1 = f(x_1 x_2), y_2 = f(x_1 x_3), \dots, y_s = f(x_{m-1} x_m),$$

$$2\text{-ряд селекции: } z_1 = f(y_1 y_2), z_2 = f(y_1 y_3), \dots, z_p = f(y_{s-1} y_s), \text{ где } s=c^2, p=c_s^2 \text{ и т.д.}$$

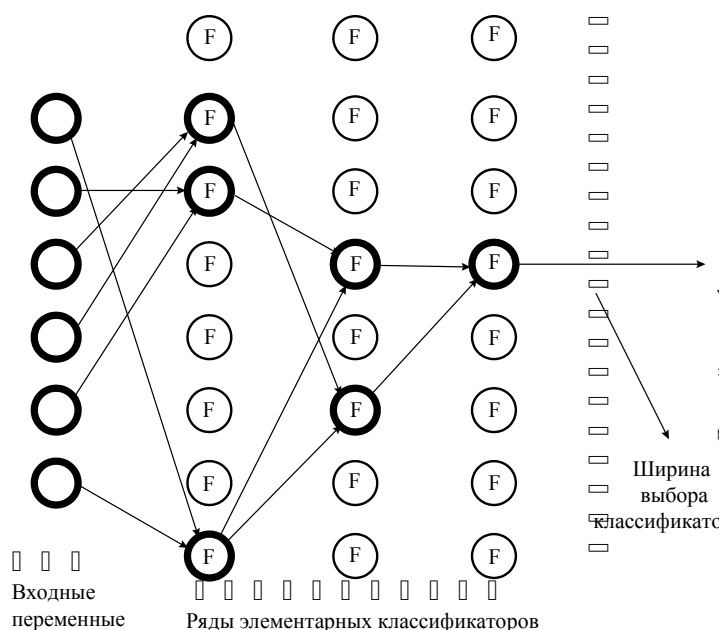
Входные аргументы и промежуточные переменные сопрягаются попарно, и сложность комбинаций на каждом ряду обработки информации возрастает (как при массовой селекции), пока не будет получена единственная модель оптимальной сложности.

Каждое частное описание является функцией только двух аргументов. Поэтому его коэффициенты легко определить по данным обучающей последовательности при малом числе узлов интерполяции [4]. Исключая промежуточные переменные (если это удастся), можно получить "аналог" полного описания. Математика не запрещает обе эти операции. Например, по десяти узлам интерполяции можно получить в результате оценки коэффициентов полинома сотой степени и т. д.

Из ряда в ряд селекции пропускается только некоторое количество самых регулярных переменных. Степень регулярности оценивается по величине среднеквадратичной ошибки (средней для всех выбираемых в каждом поколении переменных или для одной самой точной переменной) на отдельной проверочной последовательности данных. Иногда в качестве показателя регулярности используется коэффициент корреляции.

Ряды селекции наращиваются до тех пор, пока регулярность повышается. Как только достигнут минимум ошибки, селекцию, во избежание "инцухта", следует остановить. Практически рекомендуется остановить селекцию даже несколько раньше достижения полного минимума, как только ошибка начинает падать слишком медленно. Это приводит к более простым и более достоверным уравнениям.

## Алгоритм с ковариациями и с квадратичными описаниями.



**Рис. 10. МГУА как эквивалент массовой селекции.**

В этом алгоритме [5, 6] используются частные описания, представленные в следующих формулах:

$$y_i = a_0 + a_1x_i + a_2x_j + a_3x_ix_j;$$

$$y_k = a_0 + a_1x_i + a_2x_j + a_3x_ix_j + a_4x_i^2 + a_5x_j^2.$$

Сложность модели увеличивается от ряда к ряду селекции как по числу учитываемых аргументов, так и по степени. Степень полного описания быстро растет. На первом ряду — квадратичные описания, на втором — четвертой степени, на третьем — восьмой и т. д. В связи с этим минимум критерия селекции находится быстро, но не совсем точно. Кроме того, имеется опасность потери существенного аргумента, особенно на первых рядах селекции (в случае отсутствия протекции). Специальные теоремы теории МГУА определяют условия, при которых результат селекции не отличается от результата полного перебора моделей.

Для того чтобы степень полного уравнения повышалась с каждым рядом селекции на единицу, достаточно рассматривать все аргументы и их ковариации как обобщенные аргументы и пользоваться составленными для них линейными описаниями.

### **Метод предельных упрощений (МПУ)**

По тому, как организован процесс обучения распознающих систем, четко выделяются два подхода к проблеме ОРО. Первый основан на построении сложных разделяющих поверхностей в случайно выбранных пространствах, а во втором — центр тяжести проблемы переносится на достижение понимания принципов формирования такого описания объектов, в рамках которого сам процесс распознавания чрезвычайно прост. Обучение в этом случае

рассматривается как некий процесс конструирования пространств для решения конкретных задач.

В МПУ предполагается, что разделяющая функция задается заранее в виде линейного (самого простого) полинома, а процесс обучения состоит в конструировании такого пространства минимальной размерности, в котором заранее заданная наиболее простая разделяющая функция безошибочно разделяет обучающую последовательность. МПР назван так потому, что в нем строится самое простое решающее правило в пространстве небольшой размерности, т. е. в простом пространстве.

Пусть на некотором множестве объектов  $V$  заданы два подмножества  $V_1^*$  и  $V_2^*$ , определяющих собой образы на обучающей последовательности  $V$ . Рассмотрим  $i$ -е свойство объектов, такое, что некоторые объекты обучающей последовательности этим свойством обладают, а другие — нет. Пусть заданным свойством обладают объекты, образующие подмножество  $V_{1i}$ , а объекты подмножества  $V_{2i}$  этим свойством не обладают ( $V_{1i} \cup V_{2i} = V$ ). Тогда  $i$ -е свойство называют признаком первого типа относительно образа  $V_1^*$ , если выполняются соотношения

$$V_1^* \subseteq V_{1i} \text{ и } V_{1i} \cap V_{2i}^* \neq \emptyset \quad (\text{ф. 18})$$

и признаком второго типа, если выполняются

$$V_1^* \subseteq V_{1i} \text{ и } V_{1i} \cap V_{2i}^* = \emptyset \quad (\text{ф. 19})$$

Если же выполняются соотношения

$$V_2^* \subseteq V_{2i} \text{ и } V_{2i} \cap V_{1i}^* \neq \emptyset \quad (\text{ф. 20})$$

то  $i$ -е свойство считается признаком первого типа относительно образа  $V_2^*$ , а если выполняются

$$V_2^* \subseteq V_{2i} \text{ и } V_{2i} \cap V_{1i}^* = \emptyset \quad (\text{ф. 21})$$

то это же свойство объявляется признаком второго типа относительно образа  $V_2^*$ . Если свойство не обладает ни одной из приведенных особенностей, то оно вообще не относится к признакам и не участвует в формировании пространства.

Одинаковые признаки — это два признака  $x_i$  и  $x_j$ , порождающие подмножества  $V_{1j}$ ,  $V_{2j}$ ,  $V_{1i}$ ,  $V_{2i}$ , такие, что

$$V_{1j} = V_{1i} \text{ и } V_{2j} = V_{2i}. \quad (\text{ф. 22})$$

Доказано утверждение, смысл которого заключается в том, что если пространство конструировать из однотипных, но неодинаковых признаков, то в конце концов будет построено такое пространство, в котором обучающая последовательность будет безошибочно разделена на два образа линейным, т. е. самым простым, решающим правилом.

Метод предельных упрощений состоит в том, что в процессе обучения последовательно проверяются всевозможные свойства объектов и из них выбираются только такие, которые обладают хотя бы одной из особенностей, определяемых соотношениями (ф. 18), (ф. 21). Такой отбор однотипных, но неодинаковых признаков продолжается до тех пор, пока при некотором значении размерности пространства не наступит безошибочное линейное деление



образов на обучающей последовательности. В зависимости от того, из признаков какого типа строится пространство, в качестве разделяющей плоскости выбирается плоскость, описываемая уравнением

$$\sum_{i=1}^n x_i - (n - 0.5) = 0 \quad (\text{ф. 23})$$

либо уравнением

$$\sum_{i=1}^n x_i - 1 = 0 \quad (\text{ф. 24})$$

Каждый объект относится к одному из образов в зависимости от того, по какую сторону относительно плоскости находится соответствующий этому объекту вектор в пространстве признаков размерности  $n$ .

### ***Коллективы решающих правил***

Давно известны приемы повышения качества принимаемых решений, состоящие в объединении специалистов той или иной области знаний в коллектив, вырабатывающий совместное решение. Идею коллективного решения можно применить и к «коллективу» формальных алгоритмов, что позволит повысить эффективность решения многих задач.

Для рационального использования особенностей различных алгоритмов при решении задач распознавания возможно объединить различные по характеру алгоритмы распознавания в коллективы, формирующие классификационное решение на основе правил, принятых в теории коллективных решений. Пусть в некоторой ситуации  $X$  принимается решение  $S$ . Тогда  $S=R(X)$ , где  $R$ —алгоритм принятия решения в ситуации  $X$ . Предположим, что существует  $L$  различных алгоритмов решения задачи, т. е.  $S_l=R_l(X)$ ,  $l=1, 2, \dots, L$ , где  $S_l$ —решение, полученное алгоритмом  $R_l$ . Будем называть множество алгоритмов  $\{R\}=\{R_1, R_2, \dots, R_L\}$  коллективом алгоритмов решения задачи (коллективом решающих правил), если на множестве решений  $S_l$  в любой ситуации  $X$  определено решающее правило  $F$ , т. е.  $S=F(S_1, S_2, \dots, S_L, X)$ . Алгоритмы  $R_l$  принято называть членами коллектива,  $S_l$  — решением  $l$ -го члена коллектива, а  $S$  — коллективным решением. Функция  $F$  определяет способ обобщения индивидуальных решений в решения коллектива  $S$ . Поэтому синтез функции  $F$ , или способ обобщения, является центральным моментом в организации коллектива.

Принятие коллективного решения может быть использовано при решении различных задач. Так, в задаче управления под ситуацией понимается ситуация среды и целей управления, а под решением — самоуправление, приводящее объект в целевое состояние. В задачах прогноза  $X$  — исходное, а  $S$  — прогнозируемое состояние. В задачах распознавания ситуацией  $X$  является описание объекта  $X$ , т. е. его изображение, а решением  $S$  — номер образа, к которому принадлежит наблюдаемое изображение. Индивидуальное и коллективное решения в задаче распознавания состоят в отнесении некоторого изображения к одному из образов. Наиболее интересными коллективами распознающих алгоритмов являются такие, в которых существует зависимость веса каждого решающего правила  $R_l$  от распознаваемого изображения. Например, вес решающего правила  $R_l$  может определяться соотношением

$$\mu_i(X) = \begin{cases} 1, & \text{если } X \in B_i, \\ 0, & \text{если } X \notin B_i \end{cases}$$

(ф. 25)

где  $B_i$  — область компетентности решающего правила  $R_i$ . Веса решающих правил выбираются так, что

$$\sum_{i=1}^L \mu_i(X) = 1 \quad (\text{ф. 26})$$

для всех возможных значений  $X$ . Соотношение (ф. 25) означает, что решение коллектива определяется решением того решающего правила  $R_i$ , области компетентности которого принадлежит изображение объекта  $X$ . Такой подход представляет собой двухуровневую процедуру распознавания. На первом уровне определяется принадлежность изображения той или иной области компетентности, а уже на втором — вступает в силу решающее правило, компетентность которого максимальна в найденной области. Решение этого правила отождествляется с решением всего коллектива. Основным этапом в такой организации коллективного решения является обучение распознаванию областей компетентности. Практически постановкой этой задачи различаются правила организации решения коллектива. Области компетентности можно искать, используя вероятностные свойства правил коллектива, можно применить гипотезу компактности и считать, что одинаковым правилам должны соответствовать компактные области, которые можно выделить алгоритмами самообучения. В процессе обучения сначала выделяются компактные множества и соответствующие им области, а затем в каждой из этих областей восстанавливается свое решающее правило. Решение такого правила, действующего в определенной области, объявляется диктаторским, т. е. отождествляется с решением всего коллектива.

В перцептроне каждый  $A$ -элемент может интерпретироваться как член коллектива. В процессе обучения все  $A$ -элементы приобретают веса, в соответствии с которыми эти  $A$ -элементы участвуют в коллективном решении. Особенность каждого  $A$ -элемента состоит в том, что он действует в некотором подпространстве исходного пространства, характер которого определяется связями между  $S$ - и  $A$ -элементами. Решение, получаемое на выходе перцептрона, можно интерпретировать как средневзвешенное решение коллектива, состоящего из всех  $A$ -элементов.

### ***Методы и алгоритмы анализа структуры многомерных данных***

#### **Кластерный анализ**

Кластерный анализ предназначен для разбиения множества объектов на заданное или неизвестное число классов на основании некоторого математического критерия качества классификации (cluster (англ.) — гроздь, пучок, скопление, группа элементов, характеризующихся каким-либо общим свойством). Критерий качества кластеризации в той или иной мере отражает следующие неформальные требования:

- а) внутри групп объекты должны быть тесно связаны между собой;
- б) объекты разных групп должны быть далеки друг от друга;

в) при прочих равных условиях распределения объектов по группам должны быть равномерными.

Требования а) и б) выражают стандартную концепцию компактности классов разбиения; требование в) состоит в том, чтобы критерий не навязывал объединения отдельных групп объектов.

Узловым моментом в кластерном анализе считается выбор метрики (или меры близости объектов), от которого решающим образом зависит окончательный вариант разбиения объектов на группы при заданном алгоритме разбиения. В каждой конкретной задаче этот выбор производится по-своему, с учетом главных целей исследования, физической и статистической природы используемой информации и т. п. При применении экстенциональных методов распознавания, как было показано в предыдущих разделах, выбор метрики достигается с помощью специальных алгоритмов преобразования исходного пространства признаков.

Другой важной величиной в кластерном анализе является расстояние между целыми группами объектов. Приведем примеры наиболее распространенных расстояний и мер близости, характеризующих взаимное расположение отдельных групп объектов. Пусть  $w_i$  —  $i$ -я группа (класс, кластер) объектов,  $N_i$  — число объектов, образующих группу  $w_i$ , вектор  $\mu_i$  — среднее арифметическое объектов, входящих в  $w_i$  (другими словами  $\mu_i$  — «центр тяжести»  $i$ -й группы), а  $q(w_i, w_m)$  — расстояние между группами  $w_i$  и  $w_m$

**Рис. 11. Различные способы определения расстояния между кластерами  $w_1$  и  $w_m$ : 1 — по центрам тяжести, 2 — по ближайшим объектам, 3 — по самым далеким объектам**

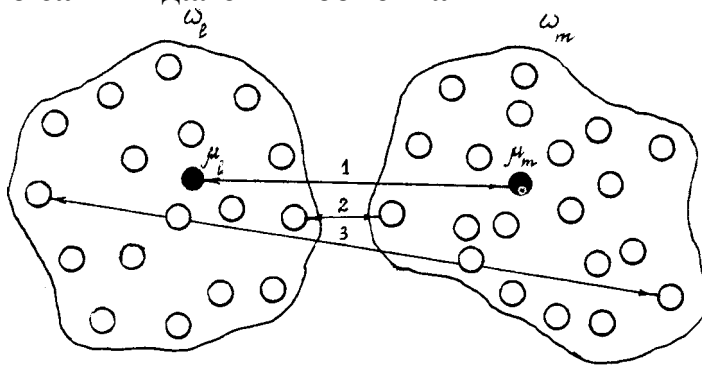


Рис. 3.11.

Расстояние ближайшего соседа есть расстояние между ближайшими объектами кластеров:

$$q_{\min}(w_1, w_m) = \min_{x_i \in w_1, x_j \in w_m} d(x_i, x_j)$$

Расстояние дальнего соседа — расстояние между самыми дальними объектами кластеров:

$$q_{\max}(w_1, w_m) = \max_{x_i \in w_1, x_j \in w_m} d(x_i, x_j)$$

Расстояние центров тяжести равно расстоянию между центральными точками кластеров:

$$q(w_1, w_m) = d(\mu_1, \mu_m)$$

Обобщенное (по Колмогорову) расстояние между классами, или обобщенное К-расстояние, вычисляется по формуле

$$q_{\tau}^{(K)}(w_1, w_m) = \left[ \frac{1}{N_1 N_m} \sum_{x_i * w_1 x_j * w_m} d^{\tau}(x_i, x_j) \right]^{\frac{1}{\tau}}$$

В частности, при  $\tau \rightarrow \infty$  и при  $\tau \rightarrow -\infty$  имеем

$$q_{\infty}^{(K)}(w_1, w_m) = q_{\max}(w_1, w_m)$$

$$q_{-\infty}^{(K)}(w_1, w_m) = q_{\min}(w_1, w_m)$$

Выбор той или иной меры расстояния между кластерами влияет, главным образом, на вид выделяемых алгоритмами кластерного анализа геометрических группировок объектов в пространстве признаков. Так, алгоритмы, основанные на расстоянии ближайшего соседа, хорошо работают в случае группировок, имеющих сложную, в частности, цепочечную структуру. Расстояние дальнего соседа применяется, когда искомые группировки образуют в пространстве признаков шаровидные облака. И промежуточное место занимают алгоритмы, использующие расстояния центров тяжести и средней связи, которые лучше всего работают в случае группировок эллипсоидной формы.

Нацеленность алгоритмов кластерного анализа на определенную структуру группировок объектов в пространстве признаков может приводить к неоптимальным или даже неправильным результатам, если гипотеза о типе группировок неверна. В случае отличия реальных распределений от гипотетических указанные алгоритмы часто «навязывают» данным не присущую им структуру и дезориентируют исследователя. Поэтому экспериментатор, учитывающий данный факт, в условиях априорной неопределенности прибегает к применению батареи алгоритмов кластерного анализа и отдает предпочтение какому-либо выводу на основании комплексной оценки совокупности результатов работы этих алгоритмов.

Алгоритмы кластерного анализа отличаются большим разнообразием. Это могут быть, например, алгоритмы, реализующие полный перебор сочетаний объектов или осуществляющие случайные разбиения множества объектов. В то же время большинство таких алгоритмов состоит из двух этапов. На первом этапе задается начальное (возможно, искусственное или даже произвольное) разбиение множества объектов на классы и определяется некоторый математический критерий качества автоматической классификации. Затем, на втором этапе, объекты переносятся из класса в класс до тех пор, пока значение критерия не перестанет улучшаться.

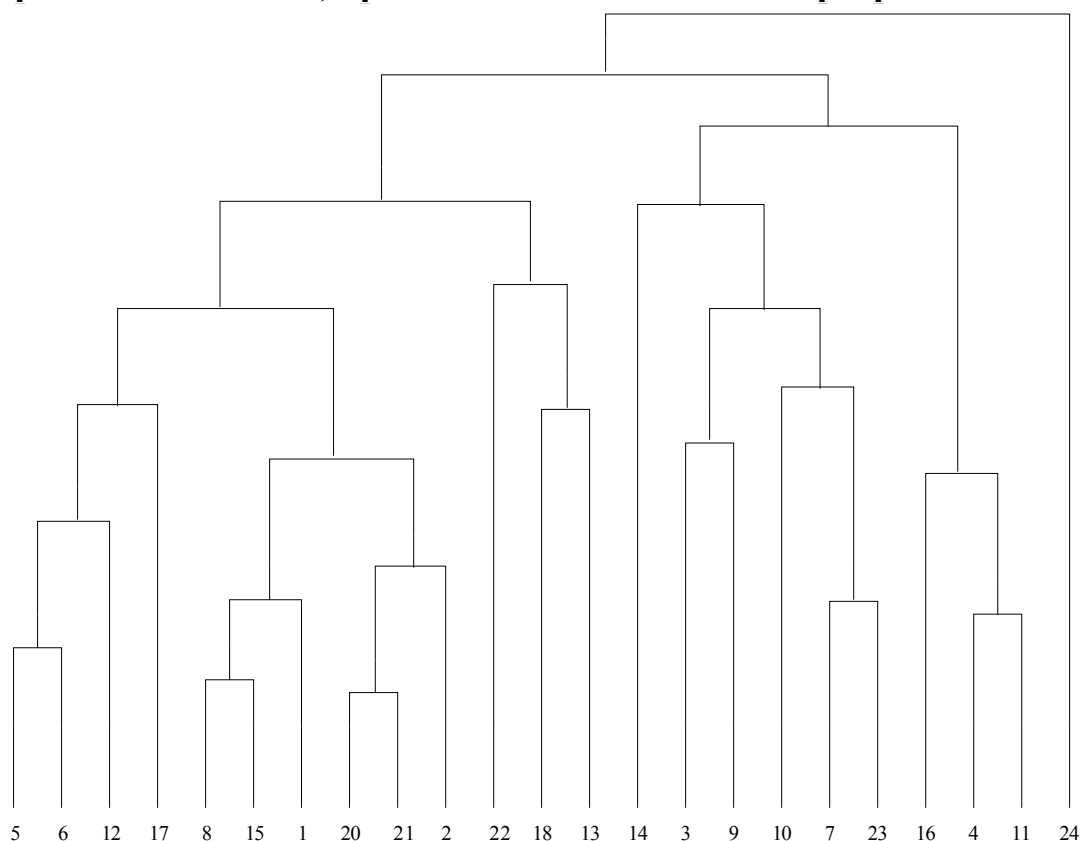
Многообразие алгоритмов кластерного анализа обусловлено также множеством различных критериев, выражающих те или иные аспекты качества автоматического группирования. Простейший критерий качества непосредственно базируется на величине расстояния между кластерами. Однако такой критерий не учитывает «населенность» кластеров — относительную плотность распределения объектов внутри выделяемых группировок. Поэтому другие критерии основываются на вычислении средних расстояний между объектами внутри кластеров. Но наиболее часто применяются критерии в виде отношений показателей «населенности» кластеров к расстоянию между ними. Это, например, может быть отношение суммы межклассовых расстояний к сумме внутриклассовых (между объектами) расстояний или отношение общей

дисперсии данных к сумме внутриклассовых дисперсий и дисперсии центров кластеров.

Функционалы качества и конкретные алгоритмы автоматической классификации достаточно полно и подробно рассмотрены в специальной литературе. Эти функционалы и алгоритмы характеризуются различной трудоемкостью и подчас требуют ресурсов высокопроизводительных компьютеров. Разнообразные процедуры кластерного анализа входят в состав практически всех современных пакетов прикладных программ для статистической обработки многомерных данных.

### Иерархическое группирование

**Рис. 12. Результаты работы иерархической агломеративной процедуры группирования объектов, представленные в виде дендрограммы.**



Классификационные процедуры иерархического типа предназначены для получения наглядного представления о стратификационной структуре всей исследуемой совокупности объектов. Эти процедуры основаны на последовательном объединении кластеров (агломеративные процедуры) и на последовательном разбиении (дивизимные процедуры). Наибольшее распространение получили агломеративные процедуры. Рассмотрим последовательность операций в таких процедурах.

На первом шаге все объекты считаются отдельными кластерами. Затем на каждом последующем шаге два ближайших кластера объединяются в один. Каждое объединение уменьшает число кластеров на один так, что в конце концов все объекты объединяются в один кластер. Наиболее подходящее разбиение выбирает чаще всего сам исследователь, которому предоставляется дендрограмма, отображающая результаты группирования объектов на всех шагах

алгоритма (Рис. 12). Могут одновременно также использоваться и математические критерии качества группирования.

Различные варианты определения расстояния между кластерами дают различные варианты иерархических агломеративных процедур. Учитывая специфику подобных процедур, для задания расстояния между классами оказывается достаточным указать порядок пересчета расстояний между классом  $w_1$  и классом  $w(m, n)$  являющимся объединением двух других классов  $w_m$  и  $w_n$  по расстояниям  $q_{mn} = q(w_m, w_n)$  и  $q_{ln} = q(w_1, w_n)$  между этими классами. В литературе предлагается следующая общая формула для вычисления расстояния между некоторым классом  $w_1$  и классом  $w(m, n)$ :

$$q_{l(m,n)} = q(w_1, w(m, n)) = \alpha q_{lm} + \beta q_{ln} + \gamma q_{mn} + \delta |q_{lm} - q_{ln}|$$

где  $\alpha$ ,  $\beta$ ,  $\gamma$  и  $\delta$  — числовые коэффициенты, определяющие нацеленность агломеративной процедуры на решение той или иной экстремальной задачи. В частности, полагая  $\alpha = \beta = -\delta = 1/2$  и  $\gamma = 0$ , приходим к расстоянию, измеряемому по принципу ближайшего соседа. Если положить  $\alpha = \beta = \delta = 1/2$  и  $\gamma = 0$ , то расстояние между двумя классами определится как расстояние между двумя самыми далекими объектами этих классов, то есть это будет расстояние дальнего соседа. И, наконец, выбор коэффициентов соотношения по формулам

$$\alpha = \frac{N_m}{N_m + N_n}, \quad \beta = \frac{N_n}{N_m + N_n}, \quad \gamma = \delta = 0$$

приводит к расстоянию  $q_{cp}$  между классами, вычисленному как среднее расстояние между всеми парами объектов, один из которых берется из одного класса, а другой из другого.

Использование следующей модификации формулы

$$q_{l(m,n)}^2 = \frac{N_1 + N_m}{N_1 + N_m + N_n} q_{lm}^2 + \frac{N_1 + N_n}{N_1 + N_m + N_n} q_{ln}^2 - \frac{N_1}{N_1 + N_m + N_n} q_{mn}^2$$

дает агломеративный алгоритм, приводящий к минимальному увеличению общей суммы квадратов расстояний между объектами внутри классов на каждом шаге объединения этих классов. В отличие от оптимизационных кластерных алгоритмов предоставляющих исследователю конечный результат группирования объектов, иерархические процедуры позволяют проследить процесс выделения группировок и иллюстрируют соподчиненность кластеров, образующихся на разных шагах какого-либо агломеративного или дивизимного алгоритма. Это стимулирует воображение исследователя и помогает ему привлекать для оценки структуры данных дополнительные формальные и неформальные представления.