Дисциплина «Архитектура интеллектуальных систем»

ЛЕКЦИЯ №1.

Базовые понятия искусственного интеллекта. Философские аспекты проблемы систем искусственного интеллекта (СИИ) (возможность существования, безопасность, полезность). История развития систем ИИ.

Термин интеллект (intelligence)происходит от латинскогоintellectus — что означает ум, рассудок, разум; мыслительные способности человека. Соответственно искусственный интеллект (artificial intelligence) — ИИ (AI) обычно толкуется как свойство автоматических систем брать на себя отдельные функции интеллекта человека, например, выбирать и принимать оптимальные решения на основе ранее полученного опыта и рационального анализа внешних воздействий.

Мы, в нашем курсе, интеллектом будем называть способность мозга решать (интеллектуальные) задачи путем приобретения, запоминания и целенаправленного преобразования знаний в процессе обучения на опыте и адаптации к разнообразным обстоятельствам.

В этом определении под термином "знания" подразумевается не только ту информацию, которая поступает в мозг через органы чувств. Такого типа знания чрезвычайно важны, но недостаточны для интеллектуальной деятельности. Дело в том, что объекты окружающей нас среды обладают свойством не только воздействовать на органы чувств, но и находиться друг с другом в определенных отношениях. Ясно, что для того, чтобы осуществлять в окружающей среде интеллектуальную деятельность (или хотя бы просто существовать), необходимо иметь в системе знаний модель этого мира. В этой информационной модели окружающей среды реальные объекты, их свойства и отношения между ними не только отображаются и запоминаются, но и, как это отмечено в данном определении интеллекта, могут мысленно "целенаправленно преобразовываться". При этом существенно то, что формирование модели внешней среды происходит "в процессе обучения на опыте и адаптации к разнообразным обстоятельствам".

Мы употребили термин интеллектуальная задача. Для того, чтобы пояснить, чем отличается интеллектуальная задача от просто задачи, необходимо ввести термин "алгоритм" — один из краеугольных терминов кибернетики.

Под алгоритмом понимают точное предписание о выполнении в определенном порядке системы операций для решения любой задачи из некоторого данного класса (множества) задач. Термин "алгоритм" происходит от имени узбекского математика Аль-Хорезми, который еще в IXвеке предложил простейшие арифметические алгоритмы. В математике и кибернетике класс задач определенного типа считается решенным, когда для ее решения установлен алгоритм. Нахождение алгоритмов является естественной целью человека при решении им разнообразных классов задач. Отыскание алгоритма для задач некоторого данного типа связано с тонкими и сложными рассуждениями, требующими большой изобретательности и высокой квалификации. Принято считать, что подобного рода деятельность требует участия интеллекта человека. Задачи, связанные с отысканием алгоритма решения класса задач определенного типа, будем называть интеллектуальными.

Что же касается задач, алгоритмы решения которых уже установлены, то, как отмечает известный специалист в области ИИ М. Минский, "излишне приписывать им такое мистическое свойства, как "интеллектуальность". В самом деле, после того, как такой алгоритм уже найден, процесс решения соответствующих задач становится таким, что его могут в точности выполнить человек, вычислительная машина (должным образом запрограммированная) или робот, не имеющие ни малейшего представления о сущность самой задачи. Требуется только, чтобы лицо, решающее задачу, было способно выполнять те элементарные операции, их которых складывается процесс, и, кроме того, чтобы оно педантично и аккуратно руководствовалось предложенным алгоритмом. Такое лицо, действуя, как говорят в таких случаях, чисто машинально, может успешно решать любую задачу рассматриваемого типа.

Поэтому представляется совершенно естественным исключить их класса интеллектуальных такие задачи, для которых существуют стандартные методы решения. Примерами таких задач могут служить чисто вычислительные задачи: решение системы линейных алгебраических уравнений, численное интегрирование дифференциальных уравнений и т. д. Для решения подобного рода задач имеются стандартные алгоритмы, представляющие собой определенную последовательность элементарных операций, которая может быть легко реализована в виде программы для вычислительной машины. В противоположность этому для широкого класса интеллектуальных задач, таких, как распознавание образов, игра в шахматы, доказательство теорем и т. п., напротив это формальное разбиение процесса поиска решения на отдельные элементарные шаги часто оказывается весьма затруднительным, даже если само их решение несложно.

Таким образом, мы можем перефразировать определение интеллекта как универсальный сверхалгоритм, который способен создавать алгоритмы решения конкретных задач.

Еще интересным замечанием здесь является то, что профессия программиста, исходя из наших определений, является одной из самых интеллектуальных, поскольку продуктом деятельности программиста являются программы — алгоритмы в чистом виде. Именно поэтому, создание даже элементов ИИ должно очень сильно повысить производительность его труда.

Деятельность мозга (обладающего интеллектом), направленную на решение интеллектуальных задач, мы будем называть мышлением, или интеллектуальной деятельностью. Интеллект и мышление органически связаны с решением таких задач, как доказательство теорем, логический анализ, распознавание ситуаций, планирование поведения, игры и управление в условиях неопределенности. Характерными чертами интеллекта, проявляющимися в процессе решения задач, являются способность к обучению, обобщению, накоплению опыта (знаний и навыков) и адаптации к изменяющимся условиям в процессе решения задач. Благодаря этим качествам интеллекта мозг может решать разнообразные задачи, а также легко перестраиваться с решения одной задачи на другую. Таким образом, мозг, наделенный интеллектом, является универсальным средством решения широкого круга задач (в том числе неформализованных) для которых нет стандартных, заранее известных методов решения.

Следует иметь в виду, что существуют и другие, чисто поведенческие (функциональные) определения. Так, по А. Н. Колмогорову, любая материальная система, с которой можно достаточно долго обсуждать проблемы науки, литературы и искусства, обладает интеллектом. Другим примером поведенческой трактовки интеллекта может служить известное определение А. Тьюринга. Его смысл заключается в следующем. В разных комнатах находится люди и машина. Они не могут видеть друг друга, но имеют возможность обмениваться информацией (например, с помощью электронной почты). Если в процессе диалога между участниками игры людям не удается установить, что один из участников — машина, то такую машину можно считать обладающей интеллектом.

Кстати интересен план имитации мышления, предложенный А. Тьюрингом. "Пытаясь имитировать интеллект взрослого человека, — пишет Тьюринг, — мы вынуждены много размышлять о том процессе, в результате которого человеческий мозг достиг своего настоящего состояния… Почему бы нам вместо того, чтобы пытаться создать программу, имитирующую интеллект взрослого человека, не попытаться создать программу, которая имитировала бы интеллект ребенка? Ведь если интеллект ребенка получает соответствующее воспитание, он становится интеллектом взрослого человека… Наш расчет состоит в том, что устройство, ему подобное, может быть легко запрограммировано… Таким образом, мы расчленим нашу проблему на две части: на задачу построения "программы-ребенка" и задачу "воспитания" этой программы".

+Забегая вперед, можно сказать, что именно этот путь используют практически все системы ИИ. Ведь понятно, что практически невозможно заложить все знания в достаточно сложную систему. Кроме того, только на этом пути проявятся перечисленные выше признаки интеллектуальной деятельности (накопление опыта, адаптация и т. д.).

Лекция №2.

Знания и их классификация. Модели и формы знаний

1. Данные и знания

При изучении интеллектуальных систем традиционно возникает вопрос – что же такое знания и чем они отличаются от обычных данных, десятилетиями обрабатываемых компьютерами. Можно взять на вооружение следующие рабочие определения данных и знаний. Данные – это представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе. Знания – итог теоретической и практической деятельности человека, отражающий накопление предыдущего опыта и отличающийся высокой степенью структурированности. Правила использования этой информации для принятия решений. Главное отличие знаний от данных состоит в их активности, то есть появление новых фактов или установление новых связей может стать источником активности системы. Для хранения данных используются базы данных (БД), для которых характерны большой объем и относительно небольшая удельная стоимость информации. Для хранения знаний соответственно применяются базы знаний (БЗ), обладающие зачастую небольшим объемом, но являющиеся исключительно дорогими информационными массивами. База знаний – основа любой интеллектуальной системы. Раздел искусственного интеллекта, изучающий базы знаний и методы работы со знаниями, называется инженерией знаний.

2. Модели представления знаний

Существуют десятки моделей представления знаний для различных предметных областей. Большинство из них может быть сведено к следующим классам: • продукционные модели; • семантические сети; • фреймы; • формальные логические модели. Рассмотрим подробнее каждый класс представления знаний.

2.1. Продукционная модель

Продукционная модель (модель, основанная на правилах) позволяет представить знания в виде предложений, называемых продукциями, типа «Если (условие), то (действие)». Под условием (антецедентом) понимается некоторое предложение-образец, по которому осуществляется поиск в БЗ, а под «действием» (консеквентом) – операции, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее в качестве условий и терминальными или целевыми, завершающими работу системы). Чаще всего вывод на такой базе знаний бывает прямой (от данных к поиску цели) или обратный (от цели для ее подтверждения – к данным). Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода. Недостатком продукционной модели является то, что при накоплении достаточно большого количества (порядка нескольких сотен) продукций они начинают противоречить друг другу. Рост противоречивости продукционной модели может быть ограничен путем введения механизмов ограничений и возвратов. Механизм исключений означает, что вводятся специальные правила-исключения. Их отличает большая конкретность в сравнении с обобщенными правилами. При наличии исключения основное правило не применяется. Механизм возвратов означает, что логический вывод может продолжаться даже в том случае, если на каком-то этапе вывод привел к противоречию: просто необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию. Существует большое количество программных средств, реализующих продукционный подход: язык OPS 5, оболочки ЭС – EXSYS Professional, Kappa, ЭКСПЕРТ, инструментальные системы ПИЭС и СПЭИС и др.

2.2. Семантические сети Семантическая сеть – это ориентированный граф, вершины которого отображают некоторые понятия, а дуги – отношения между ними. Таким образом, семантическая сеть отражает семантику предметной области в виде понятий и отношений. Идея систематизации на основе каких-либо семантических отношений не раз возникала в ранние периоды развития науки. Примером этого может служить биологическая классификация Карла Линнея 1735 г. Прародителями современных семантических сетей можно считать экзистенциальные графы, предложенные Чарльзом Пирсом в 1909 г. Они использовались для представления логических высказываний в виде особых диаграмм. Пирс назвал этот способ «логикой будущего». Компьютерные семантические сети были детально разработаны Ричардом Риченсом в 1956 году в рамках проекта Кембриджского центра изучения языка по машинному переводу. Количество типов отношений в семантической сети определяется ее создателем исходя из конкретных целей. В реальном мире их число стремится к бесконечности. Наиболее часто возникает потребность в описании отношений между элементами, множествами и частями объектов. Отношение между объектом и множеством, обозначающее, что объект принадлежит этому множеству, называется отношением классификации (ISA). Связь ISA предполагает, что свойства объекта наследуются от множества. Обратное к ISA отношение используется для обозначения примером, поэтому так и называется – «Example». Отношение между надмножеством и подмножеством называется AKO (A Kind Of). Элемент подмножества называется гипонимом, а надмножества – гиперонимом, само же отношение называется отношением гипонимии. Это отношение определяет, что каждый элемент первого множества входит и во второе (выполняется ISA для каждого элемента), а также логическую связь между самими подмножествами: что первое не больше второго и свойства первого множества наследуются вторым. Объект, как правило, состоит из нескольких частей, или элементов. Важным отношением является HasPart, описывающее части/целые объекты (отношение меронимии). Мероним – это объект, являющийся частью для другого. Холоним – это объект, который включает в себя другое. Например, двигатель – это мероним для автомобиля, а дом – холоним для крыши. Часто в семантических сетях требуется определить отношения синонимии и антонимии. Используются также следующие отношения: • функциональные связи (определяемые обычно глаголами «производит», «влияет» и др.); • количественные (больше, меньше, равно); • пространственные (далеко от, близко к, за, под, над); • временные (раньше, позже, в течение • атрибутивные (иметь свойство, иметь значение); • логические (и, или, не); • лингвистические. Для всех семантических сетей справедливо разделение по арности и количеству типов отношений. По количеству типов, сети могут быть однородными и неоднородными. Однородные сети обладают только одним типом отношений (таковой является классификация биологических видов с единственным отношением АКО). В неоднородных сетях количество отношений больше двух. Классические иллюстрации данной модели представления знаний представляют именно такие сети. По арности, типичными являются сети с бинарными отношениями (связывающими ровно два понятия). Бинарные отношения просты и удобны в применении. На практике, однако, могут понадобиться отношения, связывающие более двух объектов – N-арные.

Рис. 1. Пример семантической сети

Недостатком данной модели представления знаний является сложность организации процедуры поиска вывода на семантической сети. Для реализации семантических сетей существуют специальные сетевые языки, например NET, язык реализации систем SIMER+MIR и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний – PROSPECTOR, CASNET, TORUS.

2.3. Фреймы Фрейм – это абстрактный образ для представления некоего стереотипа информации. В психологии известно понятие абстрактного образа. Например, произнесение вслух слова «комната» порождает у слушающих образ комнаты: «жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью, площадью 6-20 м2 ». Из этого описания ничего нельзя убрать (например, убрав окна, мы получим уже не комнату, а чулан), но в нем есть т.н. слоты – незаполненные значения некоторых атрибутов – например, количество окон, цвет стен, высота потолка, покрытие пола и др. В теории фреймов такой образ комнаты называется фреймом комнаты. Фреймом также называется и формализованная модель для отображения образа. Различают фреймы-образцы (прототипы), хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через: • фреймы-структуры, использующиеся для обозначения объектов и понятий (заем, залог, вексель); • фреймы-роли (менеджер, кассир, клиент); • фреймы-сценарии (банкротство, собрание акционеров, празднование именин); • фреймы-ситуации (тревога, авария, рабочий режим устройства) и др. Традиционно структура фрейма может быть представлена как список свойств: (ИМЯ ФРЕЙМА (имя 1-го слота: значение 1-го слота), (имя 2-го слота: значение 2-го слота), … (имя N-го слота: значение N-го слота)). Ту же запись можно представить в виде таблицы, дополнив ее двумя столбцами.

Таблица 1. Структура фрейма

Имя фрейма			
Имя слота	Значение слота	Способ получения значения	Присоединенная процедура

В табл. 1. дополнительные столбцы предназначены для описания способа получения слотом его значения и возможного присоединения к тому или иному слоту специальных процедур, что допускается в теории фреймов. В качестве значения слота может выступать имя другого фрейма: так образуются сети фреймов. Существует несколько способов получения слотом значений во фрейме-экземпляре:

• по умолчанию от фрейма-образца;

• через наследование свойств от фрейма, указанного в слоте АКО;

• по формуле, указанной в слоте;

• через присоединенную процедуру;

• явно из диалога с пользователем;

• из базы данных. Важнейшим свойством теории фреймов является заимствование из теории семантических сетей – так называемое наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО-связям. Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т.е. переносятся, значения аналогичных слотов.

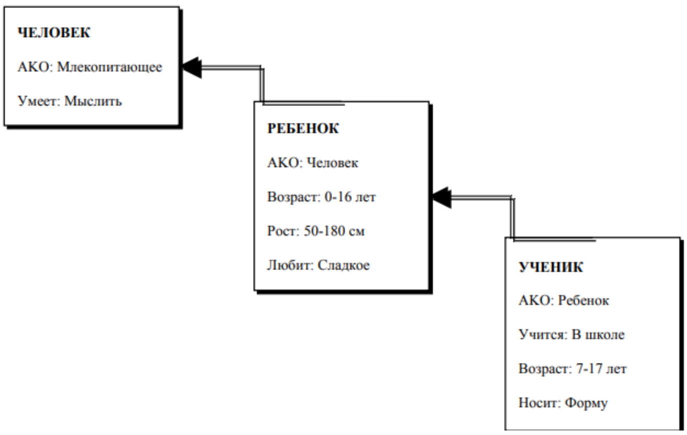

Рис. 2. Пример сети фреймов.

Основным преимуществом фреймов как модели представления знаний является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность. Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language), KRL (Knowledge Representation Language), фреймовая оболочка Kappa и другие программные средства позволяют эффективно строить промышленные ЭС. Широко известны такие фреймориентированные экспертные системы, как ANALYST, МОДИС, TRISTAN, ALTERID.

2.4. Формальные логические модели

Традиционно в представлении знаний выделяют формальные логические модели, основанные на классическом исчислении предикатов I-го порядка, когда предметная область или задача описываются в виде набора аксиом. Чаще всего эти логические модели строятся при помощи декларативных языков логического программирования, наиболее известным представителем которых является язык Пролог (Prolog). Начало истории языка относится к 70-м годам XX века. Интерес к Прологу поднимался и затихал несколько раз, энтузиазм сменялся жестким неприятием. Наибольшее внимание Пролог привлек к себе как к языку будущего во время разработок японской национальной программы «Компьютеры пятого поколения» в 1980-х годах, когда разработчики надеялись, что с помощью Пролога можно будет сформулировать новые принципы, которые приведут к созданию компьютеров более высокого уровня интеллекта. Неправильная оценка этой перспективы явилась одной из причин неудачи проекта. В настоящее время Пролог, несмотря на неоднократные пессимистические прогнозы, продолжает развиваться в разных странах и вбирает в себя новые технологии и концепции, а также парадигмы императивного программирования. Базовым принципом языка является равнозначность представления программы и данных (декларативность), отчего утверждения языка одновременно являются и записями, подобными записям в базах данных, и правилами, несущими в себе способы их обработки. Сочетание этих качеств приводит к тому, что по мере работы системы Пролога знания (и данные, и правила) накапливаются. Поэтому Пролог-системы считают естественной средой для накопления базы знаний.

2.5. Онтологии Онтология – это попытка всеобъемлющей и детальной формализации некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из иерархической структуры данных, содержащей все релевантные классы объектов, их связи и правила (теоремы, ограничения), принятые в этой области. Современные онтологии строятся по большей части одинаково, независимо от языка написания. Обычно они состоят из экземпляров, понятий, атрибутов и отношений. Экземпляры (или индивиды) – это основные, нижнеуровневые компоненты онтологии. Экземпляры могут представлять собой как физические объекты (люди, дома, планеты), так и абстрактные (числа, слова). Строго говоря, онтология может обойтись и без конкретных объектов. Однако одной из главных целей онтологии является классификация таких объектов, поэтому они тоже включаются. Понятия (или классы) – это абстрактные группы, коллекции или наборы объектов. Они могут включать в себя экземпляры, другие классы, либо же сочетания того и другого. Объекты в онтологии могут иметь атрибуты. Каждый атрибут имеет по крайней мере имя и значение, и используется для хранения информации, которая специфична для объекта и привязана к нему. Важная роль атрибутов заключается в том, чтобы определять зависимости (отношения) между объектами онтологии. Обычно отношением является атрибут, значением которого является другой объект.

Специализированные (предметно-ориентированные) онтологии – это представление какой-либо области знаний или части реального мира. В такой онтологии содержатся специальные для этой области значения терминов. Общие онтологии используются для представления понятий, общих для большого числа областей. Такие онтологии содержат базовый набор терминов, глоссарий или тезаурус, используемый для описания терминов предметных областей. Если использующая специализированные онтологии система развивается, то может потребоваться их объединение, и для инженера по онтологиям это серьезная задача. Подобные онтологии часто несовместимы друг с другом, хотя могут представлять близкие области. Разница может появляться из-за особенностей местной культуры, идеологии и т.п., или вследствие использования другого языка описания. Сегодня объединение онтологий приходится выполнять вручную: это трудоемкий, медленный и дорогостоящий процесс. Использование базисной онтологии – единого глоссария – несколько упрощает эту работу. Есть научные работы по технологиям объединения, но они по большей части носят чисто теоретический характер. Разработано несколько формальных языков для описания онтологий, в частности, следующие:

• OWL (Ontology Web Language), язык для поддержки семантической паутины (см. ниже);

• KIF (Knowledge Interchange Format) – основанный на т.н. S выражениях синтаксис для логики;

• CycL – онтологический язык, используемый в проекте Cyc, основан на исчислении предикатов с некоторыми расширениями более высокого порядка.

Для работы с языками онтологий существует несколько видов технологий: редакторы онтологий (для создания онтологий), хранилища онтологий (для работы с несколькими онтологиями) и др.

Лекция №3.

Современные архитектуры нейронных сетей. Научные и промышленные приложения

Архитектура нейронных сетей

Нейронные сети могут быть синхронные и асинхронные.

В синхронных *нейронных сетях* в каждый момент времени свое состояние меняет лишь один *нейрон*.

В асинхронных - состояние меняется сразу у целой группы нейронов, как правило, у всего *слоя*.

Можно выделить две базовые архитектуры - слоистые и полносвязные сети.

Ключевым в слоистых сетях является понятие *слоя*.

Слой - один или несколько нейронов, на входы которых подается один и тот же общий сигнал.

Слоистые нейронные сети - *нейронные сети*, в которых нейроны разбиты на отдельные группы (*слои*) так, что обработка информации осуществляется послойно.

В слоистых сетях нейроны i-го *слоя* получают входные сигналы, преобразуют их и через *точки ветвления* передают нейронам (i+1) *слоя*. И так до k-го *слоя*, который выдает выходные сигналы для интерпретатора и пользователя. Число нейронов в каждом *слое* не связано с количеством нейронов в других *слоях*, может быть произвольным.
В рамках одного *слоя* данные обрабатываются параллельно, а в масштабах всей сети обработка ведется последовательно - от *слоя* к *слою*. К слоистым *нейронным сетям* относятся, например, многослойные *персептроны*, сети радиальных *базисных функций*, *когнитрон*, *неокогнитрон*, сети ассоциативной памяти.
Однако сигнал не всегда подается на все нейроны *слоя*. В когнитроне, например, каждый *нейрон* текущего *слоя* получает сигналы только от близких ему нейронов предыдущего *слоя*.
Слоистые сети, в свою *очередь*, могут быть однослойными и многослойными [46].
Однослойная сеть - *сеть*, состоящая из одного *слоя*.
Многослойная сеть - *сеть*, имеющая несколько *слоев*.
В многослойной сети первый *слой* называется входным, последующие - внутренними или скрытыми, последний *слой* - выходным. Таким образом, промежуточные *слои* - это все *слои* в многослойной *нейронной сети*, кроме входного и выходного.
Входной слой сети реализует *связь* с входными данными, выходной - с выходными. Таким образом, нейроны могут быть *входными*, *выходными* и *скрытыми*.
Входной слой организован из ***входных нейронов*** (*input neuron*), которые получают данные и распространяют их на входы нейронов скрытого *слоя* сети.
Скрытый нейрон (*hidden neuron*) - это *нейрон*, находящийся в скрытом *слое нейронной сети*.
Выходные нейроны (*output neuron*), из которых организован выходной *слой* сети, выдает результаты работы нейронной сети.
В полносвязных сетях каждый *нейрон* передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам.
Перед использованием *нейронной сети* ее необходимо обучить.
Процесс обучения *нейронной сети* заключается в подстройке ее *внутренних параметров* под конкретную задачу.
Алгоритм работы *нейронной сети* является итеративным, его шаги называют ***эпохами*** или циклами.
Эпоха - одна *итерация* в процессе обучения, включающая предъявление всех примеров из обучающего *множества* и, возможно, проверку качества обучения на контрольном множестве.
Процесс обучения осуществляется на обучающей выборке.
Обучающая выборка включает входные значения и соответствующие им выходные значения набора данных. В ходе обучения *нейронная сеть* находит некие зависимости выходных полей от входных.

Таким образом, перед нами ставится вопрос - какие входные поля (признаки) нам необходимо использовать. Первоначально выбор осуществляется эвристически, далее количество входов может быть изменено.
Сложность может вызвать вопрос о количестве наблюдений в наборе данных. И хотя существуют некие правила, описывающие *связь* между необходимым количеством наблюдений и размером сети, их верность не доказана.
Количество необходимых наблюдений зависит от сложности решаемой задачи. При увеличении количества признаков количество наблюдений возрастает нелинейно, эта проблема носит название "проклятие размерности". При недостаточном количестве данных рекомендуется использовать *линейную модель*.
Аналитик должен определить количество *слоев* в сети и количество нейронов в каждом *слое*.
Далее необходимо назначить такие значения весов и смещений, которые смогут минимизировать ошибку решения. Веса и смещения автоматически настраиваются таким образом, чтобы минимизировать *разность* между желаемым и полученным на выходе сигналами, которая называется ***ошибка обучения***.
Ошибка обучения для построенной *нейронной сети* вычисляется путем сравнения выходных и целевых (желаемых) значений. Из полученных разностей формируется *функция ошибок*.
Функция ошибок - это *целевая функция*, требующая минимизации в процессе управляемого обучения *нейронной сети*.
С помощью *функции ошибок* можно оценить качество работы *нейронной сети* во время обучения. Например, часто используется сумма квадратов ошибок.
От качества обучения *нейронной сети* зависит ее способность решать поставленные перед ней задачи.
Переобучение нейронной сети
При обучении *нейронных сетей* часто возникает серьезная трудность, называемая проблемой *переобучения* (overfitting).
Переобучение, или чрезмерно близкая подгонка - излишне точное соответствие *нейронной сети* конкретному набору обучающих примеров, при котором *сеть* теряет способность к обобщению.
Переобучение возникает в случае слишком долгого обучения, недостаточного числа обучающих примеров или переусложненной структуры *нейронной сети*.
Переобучение связано с тем, что выбор обучающего (тренировочного) *множества* является случайным. С первых шагов обучения происходит уменьшение ошибки. На последующих шагах с целью уменьшения ошибки (целевой функции) параметры подстраиваются под особенности обучающего *множества*. Однако при этом происходит "подстройка" не под общие закономерности ряда, а под особенности его части - обучающего подмножества. При этом *точность прогноза* уменьшается.
Один из вариантов борьбы с *переобучением* сети - *деление* обучающей выборки на два *множества* (обучающее и тестовое).

На обучающем множестве происходит обучение *нейронной сети*. На тестовом множестве осуществляется проверка построенной модели. Эти *множества* не должны пересекаться.
С каждым шагом параметры модели изменяются, однако постоянное уменьшение значения целевой функции происходит именно на обучающем множестве. При разбиении *множества* на два мы можем наблюдать изменение ошибки прогноза на тестовом множестве параллельно с наблюдениями над обучающим множеством. Какое-то количество шагов ошибки прогноза уменьшается на обоих множествах. Однако на определенном шаге ошибка на тестовом множестве начинает возрастать, при этом ошибка на обучающем множестве продолжает уменьшаться. Этот момент считается концом реального или настоящего обучения, с него и начинается *переобучение*.

Стандартные архитектуры нейронных сетей

Сеть из одного нейрона

Рассмотрим возможности сети состоящей из одного нейрона. Пусть, для начала, этот формальный нейрон не имеет нелинейного преобразователя, т. е. функционирует по формуле $y = (\overline{w}, \overline{x}) + w_0$. Какие задачи можно решать с его помощью?

Линейная регрессия

Первая, и весьма важная задача, решаемая нейросетью из одного нейрона – это задача линейной регрессии, которая формулируется так: найти наилучшее линейное приближение функции *F*, заданной обучающей выборкой $(\overline{x}^{\alpha}, y^{\alpha})$, $F(\overline{x}^{\alpha}) = y^{\alpha}$, $\alpha = 1,...,m$. Для этого требуется найти линейную функцию $\varphi(\overline{x}) = (\overline{w}, \overline{x}) + w_0$, ближайшую к *F*.

Функцию ошибки можно определить как сумму квадратов разностей:

$$D(\varphi) = \sum_{i=1}^{m} (F(x^i) - \varphi(x^i))^2 = \sum_{i=1}^{m} (F(x^i) - (\overline{w}, \overline{x}^i) - w_0)^2 \equiv \sum_{i=1}^{m} \Delta_i^2 . \qquad (6.1)$$

Решение. Эту функцию необходимо минимизировать, для чего найдём производные по изменяемым параметрам:

$$\frac{\partial D}{\partial w_j} = -2\sum_{i=1}^{m} \Delta_i x_j^i, \; (j = 1,...,n); \;\; \frac{\partial H}{\partial w_0} = -2\sum_{i=1}^{m} \Delta_i . \qquad (6.2)$$

Приравнивая эти производные нулю и вводя в рассмотрение *(n+1)*-мерные вектора x^j, в которых все элементы те же, а $x_0^i \equiv 1$, мы можем записать

$$\begin{aligned} &\Delta_i = y^i - (\overline{w}, x^i), \\ &w_0 = \frac{1}{m}\left(\sum_{i=1}^{m} (y^i - (\overline{w}, x^i) \right). \end{aligned} \qquad (6.3)$$

Введем обозначения $\tilde{y}_m = \frac{1}{m}\sum_{i=1}^{m} y^i$; $\tilde{x}_m = \frac{1}{m}\sum_{i=1}^{m} \overline{x}^i$. В этих обозначениях (6.3) примет вид

$$w_0 = \tilde{y}^m - (\overline{w}, \tilde{x}^m). \qquad (6.4)$$

Подставив это значение в (6.2) и приравнивая к нулю, мы получим систему:

$$\sum_{k=1}^{n} w_k \cdot \left(\sum_{i=1}^{m} (x_j^i - \tilde{x}_j^m)(x_k^i - \tilde{x}_k^m) \right) = \sum_{i=1}^{m} (x_j^i - \tilde{x}_j^m)(y^i - \tilde{y}^m), \qquad (6.5)$$

или, в более короткой форме

$$\sum_{k=1}^{n} w_k r_{kj} = h_j. \qquad (6.6)$$

В векторно-матричной форме полученное равенство имеет вид

$$\overline{w}R = \overline{h}\ . \qquad (6.6')$$

Решение этой системы в случае невырожденности матрицы R может быть найдено как

$$\overline{w} = \overline{h} \cdot R^{-1}, \qquad (6.7)$$

в противном случае решения либо не существует, либо оно не единственно. При этом обычно ищется решение минимальной длины:

$$\overline{w} = \overline{h} \cdot R^{+}, \qquad (6.8)$$

где R^{+} - псевдообратная матрица. Способы решения систем линейных уравнений хорошо известны, и могут быть применены и в данном случае.

Пусть теперь нейрон кроме возможности скалярного умножения входного вектора на вектор весов (которое осуществляется адаптивным сумматором) имеет также возможность порогового преобразования, т. е. его функционирование описывается формулой

$$y = F((\overline{w},\overline{x}) + w_0) = \begin{cases} 1,\ (\overline{w},\overline{x}) + w_0 \ge 0, \\ 0,\ (\overline{w},\overline{x}) + w_0 < 0. \end{cases} \qquad (6.9)$$

Рассмотрим его возможности.

Задача линейного разделения двух классов

Эта задача ставится как задача построения решающего правила, в соответствии с которым, часть векторов из выборки $\{\overline{x}^{\alpha}\}_{\alpha=1,...,p}$ будет отнесена к одному классу, а остальные вектора выборки – к другому. В терминах нейросети из одного нейрона, это означает необходимость подобрать такие весовые коэффициенты w_j для нашего нейрона, что выход нейрона (6.9) равен единице, если входной вектор принадлежит первому классу и нулю, если входной вектор принадлежит другому классу.

Решение. Поиск такого решающего правила можно рассматривать как разделение точек посредством проекции их прямую. Вектор $\overline{w}$ задает прямую, на которую ортогонально проектируются все точки, а число w_0 - точку на этой прямой, отделяющую первый класс от второго.

Простейший и подчас очень удобный выбор состоит в проектировании на прямую, соединяющую центры масс выборок. Центр масс вычисляется в предположении, что массы всех точек одинаковы и равны 1. Это соответствует заданию $\overline{w}$ в виде

$$\overline{w} = \frac{1}{m}\sum_{i=1}^{m}\overline{x}^{\alpha_i} + \frac{1}{n}\sum_{i=m+1}^{p}\overline{x}^{\alpha_j}, \qquad (6.10)$$

где $\alpha_1, \alpha_2, ..., \alpha_m$ - номера векторов принадлежащих первому классу, $\alpha_{m+1}, \alpha_{m+2}, ..., \alpha_p$ - номера векторов принадлежащих второму классу. Простейший вариант выбора значения w_0 - посередине между центрами масс групп векторов:

$$w_0 = \frac{1}{2}\left(\left(\frac{1}{m}\sum_{i=1}^{m}\overline{x}^{\alpha_i}, \overline{w}\right) + \left(\frac{1}{n}\sum_{i=m+1}^{p}\overline{x}^{\alpha_j}, \overline{w}\right)\right). \qquad (6.11)$$

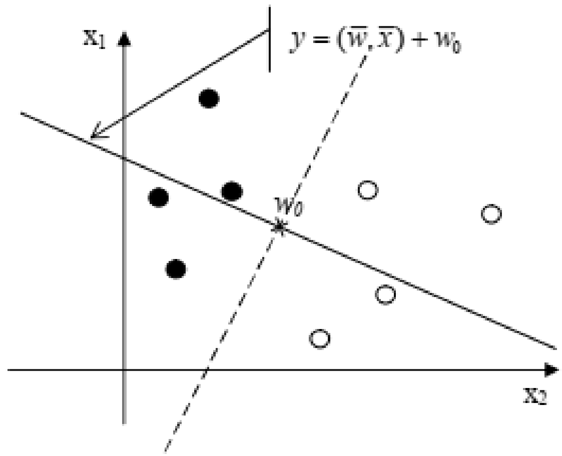

Рисунок 6.1. Разделение векторов на классы при помощи прямой

Более чувствительные методы выбора границы раздела классов w_0 учитывают различные вероятности появления объектов разных классов, и оценки плотности распределения точек классов на прямой.

Рассмотрим теперь более сложные структуры - сети, состоящие из нескольких нейронов.

Однослойный персептрон Розенблатта

Самым простым и исторически первым вариантом многослойного персептрона является персептрон Розенблатта – нейросеть с единственным слоем нейронов (F.Rosenblatt, 1957). Персептрон рассматривался его автором не как конкретное техническое вычислительное устройство, а как модель работы мозга.

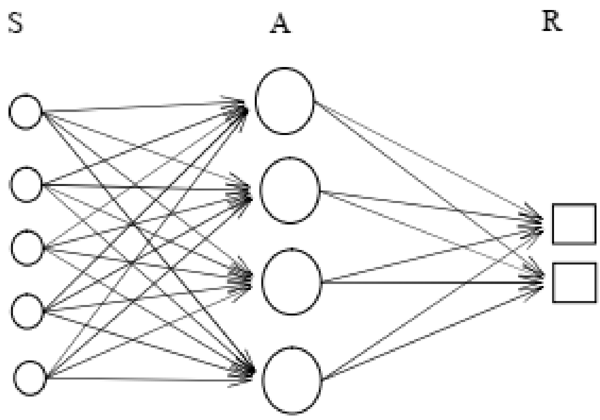

Рисунок 6.2.

Простейший классический персептрон содержит нейроподобные элементы трех типов. S-элементы формируют входной вектор двоичных сигналов, поступающих из внешнего мира. Далее сигналы поступают в слой ассоциативных или A-элементов. Только ассоциативные элементы, представляющие собой формальные нейроны, выполняют нелинейную обработку информации и имеют изменяемые веса связей. Нелинейная часть A-элемента реализуется посредством пороговой функции:

$$f(x)=\begin{cases}1, & \text{если } x \geq \theta,\\ 0, & \text{если } x < \theta.\end{cases} \qquad (6.12)$$

R-элементы с фиксированными весами формируют сигнал реакции персептрона на входной стимул.

Розенблатт называл такую нейронную сеть трехслойной, однако по современной терминологии, используемой в этой книге, представленная сеть обычно называется однослойной, так как имеет только один слой нейропроцессорных элементов. Однослойный персептрон характеризуется матрицей синаптических связей W от S- к A-элементам. Элемент матрицы W_{ij} отвечает связи, ведущей от i-го S-элемента к j-му A-элементу.

Алгоритм обучения персептрона

Обучение сети состоит в подстройке весовых коэффициентов каждого нейрона. Пусть имеется набор пар векторов $(\overline{x}^{\alpha},\overline{y}^{\alpha}), \alpha=1,...,p,\quad \overline{x}^{\alpha},\overline{y}^{\alpha}\in R^{N}$, называемый обучающей выборкой. Будем называть нейронную сеть обученной на данной обучающей выборке, если при подаче на входы сети каждого вектора $\overline{x}^{\alpha}$ на выходах всякий раз получается соответствующий вектор $\overline{y}^{\alpha}$.

1. Функционирование. Допустим, что вектор $\bar{x}^{\alpha}$ является образом распознаваемой демонстрационной карты, а значит, подаётся на каждый входной А-элемент. Например, для j-го элемента - компоненты $\bar{x}_i^{\alpha}$ – умножаются на соответствующие компоненты вектора весов w_{ij}. Эти произведения суммируются. Если сумма превышает порог Θ, то выход j-го нейрона равен единице, в противном случае он – ноль. Эта операция компактно записывается в векторной форме как

$$\bar{y}^{*} = F(\bar{x}^{\alpha} \cdot W), \qquad (6.13)$$

$$F(\bar{x}) = (f(x_1), f(x_2), ..., f(x_p)). \qquad (6.14)$$

2. Обучение. Для обучения сети образ $\bar{x}^{\alpha}$ подается на вход и вычисляется выход $\bar{y}^{*}$ (в соответствии с описанием в п. 1). Если $\bar{y}^{*}$ правильный, то ничего не меняется. Однако, если выход неправилен, то веса, присоединенные к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку: если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам; или если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса:

$$w_{ij}(t+1) = w_{ij}(t) + (y_j^{\alpha(t)} - y_j^{*}(t)) \cdot x_i^{\alpha(t)}. \qquad (6.15)$$

3. Если ошибка существенна, перейти на шаг 1.

В работах Розенблатта был сделано заключение о том, что нейронная сеть рассмотренной архитектуры будет способна к воспроизведению любой логической функции, однако, как было показано позднее М.Минским и С.Пейпертом (M.Minsky, S.Papert, 1969), этот вывод оказался неточным. Были выявлены принципиальные неустранимые ограничения однослойных персептронов, и в последствии стал в основном рассматриваться многослойный вариант персептрона, в котором имеются несколько слоев процессорных элементов.

Сегодня однослойный персептрон представляет только исторический интерес, однако на его примере могут быть изучены основные понятия и простые алгоритмы обучения нейронных сетей.

Слоистые архитектуры

Особый интерес в нейроинформатике представляют слоистые однородные нейросети, состоящие из однотипных формальных нейронов, каждый из которых может быть соединён с любыми нейронами следующего слоя не более чем одной связью. Нейросеть с такими свойствами называется *многослойным персептроном.*

Многослойная нейронная сеть и алгоритм обратного распространения ошибки

История многослойных нейронных сетей началась в 1960-х годах и связана с работами Розенблатта, Минского, Пейперта и др. Лишь в середине 1980-х несколькими исследователями независимо друг от друга был предложен эффективный алгоритм обучения многослойных персептронов, основанный на вычислении градиента функции ошибки. Алгоритм был назван "обратным распространением ошибки".

Алгоритм обратного распространения - это итеративный градиентный алгоритм, который используется с целью минимизации среднеквадратичного отклонения текущего выхода многослойного персептрона и желаемого выхода.

В сетях, обучаемых по методу обратного распространения ошибки могут быть целые или действительные входные сигналы. Выходные сигналы сети - это действительные числа из интервала, заданного передаточной функцией нейронов.

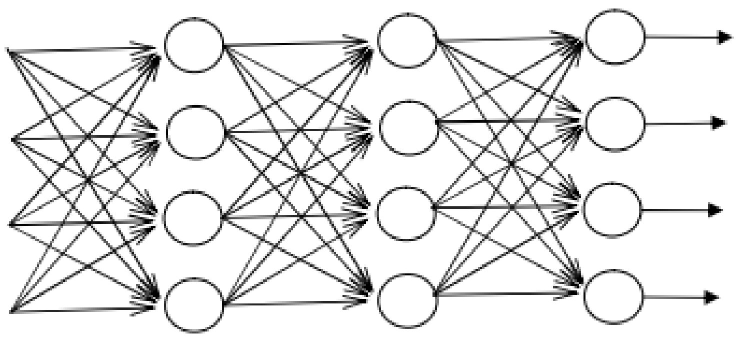

Рисунок 6.3. Многослойная сеть с последовательными связями

В нейронных сетях могут применяться несколько вариантов передаточных функций.

Дискретные:

$$f(s) = sign(s) = \begin{cases} 1, & s > 0, \\ -1, & s \le 0. \end{cases}, \text{ или } f(s) = \begin{cases} -1, & x < -1, \\ x, & -1 \le x \le 1, \\ 1, & x > 1. \end{cases}$$

Функция Ферми (экспоненциальная сигмоида):

$$f(s)=\frac{1}{1+e^{-\alpha s}}, \quad f'(s)=\alpha f(s)(1-f(s)),$$

где s - выход сумматора нейрона, α - некоторый параметр.

Рациональная сигмоида:

$$f(s)=\frac{s}{|s|+\alpha}, \quad f'(s)=(1-|f(s)|)^2.$$

Гиперболический тангенс:

$$f(s)=th\frac{s}{\alpha}, \quad f'(s)=\frac{1}{\alpha}(1-f^2(s)).$$

Гауссиан:

$$f(s)=e^{-\alpha s^2}, \quad f'(s)=-2\alpha s e^{-\alpha s^2}.$$

Сигмоидальные функции являются монотонно возрастающими и имеют отличные от нуля производные на всей области определения. Эти характеристики обеспечивают правильное функционирование и корректное обучение сети.

По скорости вычислений наиболее эффективной функцией активации является рациональная сигмоида. Для вычисления гиперболического тангенса требуется больше всего тактов работы процессора.

Задание. Нарисовать графики перечисленных выше функций.

Рассмотрим многослойный персептрон со стандартными элементами.

Функционирование многослойного персептрона

Функционирование в такой нейросети описывается следующими формулами:

$$y_i^0 = x_i^0,$$
$$s_i^k = \sum_{j=1}^{P_{k-1}} y_j^{k-1} w_{ij}^k, \qquad (6.16)$$
$$y_i^k = f(s_i^k),$$

где

x_i^0 - значение подаваемое в момент *t=0* на *i*-й синапс,

k - номер слоя ($k=1,\ldots,q$);

P_k - число нейронов в *k*-м слое;

y_i^k - выход *i*-го нейрона в *k*-м слое;

w_{ij}^k - вес синапса связи *i*-го нейрона в *(k*-1*)*-м слое с *j*-м нейроном в *k*-м слое;

s_i^k - выход сумматора *i*-го нейрона в *k*-м слое;

$f(s)$ - функция активации нейрона, одинаковая для всех нейронов сети.

Предполагается, что выход *i*-го нейрона в *(k*-1*)*-м подается на *i*-й вход *j*-го нейрона в *k*-м слое.

Очевидно, что формулы (6.16) можно записать и в матрично-векторном виде:

$$\begin{aligned} \bar{y}^0 &= \bar{x}^0, \\ \bar{s}^k &= W^k \bar{y}^{k-1}, \\ \bar{y}^k &= F(\bar{s}^k), \end{aligned} \qquad (6.16')$$

где

$\bar{x}^0$ - вектор, подаваемый на вход нейросети,

$W^k = (w_{ij}^k)$ - матрица размерности $(P_k \times P_{k-1})$, элементами которой являются веса связей нейронов $(k-1)$ -го слоя с нейронами k -го слоя,

$\bar{y}^k$ - вектор полученный на выходе нейронов k -го слоя,

$F(\bar{s}^k) := (f(s_1^k), f(s_2^k), \ldots, f(s_{P_k}^k))$ - функция нелинейного преобразования.

Функционирование сети происходит следующим образом:

1-й шаг: на вход сети подается вектор сигналов $X = (x_1, x_2, \ldots, x_{P_1})$. Происходит срабатывание нейронов первого слоя.

k-й шаг: выход каждого нейрона k-го слоя определяемый формулами (6.16) передается по синаптическим связям нейронам (k+1)-го слоя.

q-й шаг: на выходе нейронной сети снимается вектор сигналов $Y = (y_1, y_2, \ldots, y_{P_q})$, который и является результатом.

Алгоритм обучения многослойного персептрона

В многослойных сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, не известны, и многослойный перцептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах НС. Одним из вариантов решения этой проблемы является так называемое «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 далее.

Пусть имеется обучающая выборка $(\overline{x}(i),\overline{d}(i))$, $i=1,...,N$. Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки НС является величина:

$$D(\overline{W})=\frac{1}{2}\sum_{j,p}(y_j^q(p)-d_j(p))^2 \qquad (6.17)$$

где $y_j^q(p)$ – реальное выходное состояние нейрона *j* выходного слоя *q* нейронной сети при подаче на ее входы *p*-го образа; $d_j(p)$ – 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:

$$w_{ij}^k(t+1)=w_{ij}^k(t)+\Delta w_{ij}^k,$$

$$\Delta w_{ij}^k=-\eta\cdot\frac{\partial D}{\partial w_{ij}^k}. \qquad (6.18)$$

Здесь w_{ij}^k – весовой коэффициент синаптической связи, соединяющей *i*-ый нейрон слоя *k-1* с *j*-ым нейроном слоя *k*, η – коэффициент скорости обучения, $0<\eta<1$, *t* – номер итерации алгоритма

Из (6.17) и (6.18) следует равенство

$$\frac{\partial D}{\partial w_{ij}^k}=\frac{\partial D}{\partial y_j^k}\cdot\frac{dy_j^k}{ds_j^k}\cdot\frac{\partial s_j^k}{\partial w_{ij}^k}. \qquad (6.19)$$

Так как множитель $\frac{dy_j^k}{ds_j^k}$ является производной этой функции по ее аргументу, из этого следует, что производная активационной функция 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 $f(s)=th\frac{s}{\alpha}$,

$$\frac{dy}{ds}=1-s^2. \qquad (6.20)$$

В случае функции Ферми $f(s)=\frac{1}{1+e^{-\alpha s}}$, получим

$$\frac{dy}{ds} = \alpha \cdot f(s) \cdot (1 - f(s)) \tag{6.20'}$$

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

Что касается первого множителя в (6.19), он легко раскладывается следующим образом:

$$\frac{\partial D}{\partial y_j^k} = \sum_i \frac{\partial D}{\partial y_i^{k+1}} \cdot \frac{dy_i^{k+1}}{ds_i^{k+1}} \cdot \frac{\partial s_i^{k+1}}{\partial y_j^{k+1}} = \sum_i \frac{\partial D}{\partial y_i^{k+1}} \cdot \frac{dy_i^{k+1}}{ds_i^{k+1}} \cdot w_{ji}^{k+1}. \tag{6.21}$$

Здесь суммирование по i выполняется среди нейронов слоя k+1. Введя новую переменную

$$\delta_j^k = \frac{\partial D}{\partial y_j^k} \cdot \frac{dy_j^k}{ds_j^k} \tag{6.22}$$

мы получим рекурсивную формулу для расчетов величин δ_j^k слоя n из величин δ_i^{k+1} более старшего слоя k+1.

$$\delta_j^k = \left[\sum_i \delta_i^{k+1} \cdot w_{ji}^{k+1}\right] \cdot \frac{dy_j^k}{ds_j^k} \tag{6.23}$$

Для выходного же слоя

$$\delta_l^q = (y_l^q - d_l) \cdot \frac{dy_l^q}{ds_l^q}. \tag{6.24}$$

Теперь мы можем записать (6.18) в раскрытом виде:

$$\Delta w_{ij}^k = -\eta \cdot \delta_j^k \cdot y_i^{k-1}. \tag{6.25}$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (6.25) дополняется значением изменения веса на предыдущей итерации

$$\Delta w_{ij}^k(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^k(t-1) + (1-\mu) \cdot \delta_j^k \cdot y_i^{k-1}) \tag{6.26}$$

где μ – коэффициент инерционности, t – номер текущей итерации. Такой вариант алгоритма получил название «метод тяжелого шарика».

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

1. Подать на входы сети один из возможных образов $\overline{x}^0 = \overline{x}(p)$ и в режиме обычного функционирования НС, когда сигналы распространяются от входов к выходам, рассчитать значения последних по формулам (6.16).

2. Рассчитать δ^q для выходного слоя по формуле (6.24):

$$\delta_l^q = (y_l^q - d_l) \cdot \frac{dy_l^q}{ds_l^q}.$$

Рассчитать по формуле (6.25) или (6.26) изменения весов Δw^q слоя q:

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

3. Рассчитать по формулам (6.23) и (6.25) (или (6.23) и (6.26)) соответственно δ^k и Δw^k для всех остальных слоев, $k=q-1,...1$:

$$\delta_j^k = \left[\sum_i \delta_i^{k+1} \cdot w_{ji}^{k+1}\right] \cdot \frac{dy_j^k}{ds_j^k},$$

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

4. Скорректировать все веса в НС

$$w_{ij}^k(t) = w_{ij}^k(t-1) + \Delta w_{ij}^k(t). \qquad (6.27)$$

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 рисунком 1.

Из выражения (6.25) следует, что когда выходное значение y_i^{k-1} стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться, 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

Лекция №4.

Иерархическая организация нейросетевых архитектур. Многослойный перцептрон, сети обратного и встречного распространения ошибки, карта Кохоннена, модель Липмана Хемминга.

Элементом клеточной структуры мозга является нервная клетка - *нейрон*. Нейрон в своем строении имеет много общих черт с другими клетками биоткани: тело нейрона окружено плазматической мембраной, внутри которой находится цитоплазма, ядро и другие составляющие клетки. Однако нервная клетка существенно отличается от иных по своему *функциональному назначению*. Нейрон выполняет прием, элементарное преобразование и дальнейшую передачу информации другим нейронам. Информация переносится в виде импульсов нервной активности, имеющих электрохимическую природу.

Нейроны крайне разнообразны по форме, которая зависит от их местонахождения в нервной системе и особенностей фунционирования. На Рис. 6.1. приведена схема строения "типичного" нейрона. Тело клетки содержит множество ветвящихся отростков двух типов. Отростки первого типа, называемые *дендритами* за их сходство с кроной раскидистого дерева, служат в качестве входных каналов для нервных импульсов от других нейронов. Эти импульсы поступают в *сому* или *тело* клетки размером от 3 до 100 микрон, вызывая ее специфическое возбуждение, которое затем распространяется по выводному отростку второго типа - *аксону*. Длина аксонов обычно заметно превосходит размеры дендритов, в отдельных случаях достигая десятков сантиметров и даже метров. Гигантский аксон кальмара имеет толщину около миллиметра, и именно наблюдение за ним послужило выяснению механизма передачи нервных импульсов между нейронами.

Тело нейрона, заполненное проводящим ионным раствором, окружено мембраной толщиной около 75 ангстрем, обладающей низкой проводимостью. Между внутреннней поверхностью мембраны аксона и внешней средой поддерживается разность электрических потенциалов. Это осуществляется при помощи молекулярного механизма ионных насосов, создающих различную концентрацию положительных ионов K^+ и Na^+ внутри и вне клетки. Проницаемость мембраны нейрона селективна для этих ионов. Внутри аксона клетки, находящейся в состоянии покоя, активный транспорт ионов стремится поддерживать концентрацию ионов калия более высокой, чем ионов натрия, тогда как в жидкости, окружающей аксон, выше оказывается концентрация ионов Na^+. Пассивная диффузия более подвижных ионов калия приводит к их интенсивному выходу из клетки, что обуславливает ее общий отрицательный относительно внешней среды *потенциал покоя*, составляющий около -65 милливольт.

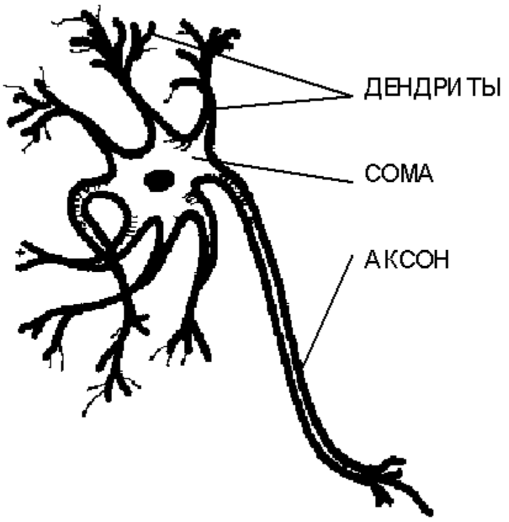

Рис. 1.Общая схема строения биологического нейрона.

Под воздействием стимулирующих сигналов от других нейронов мембрана аксона динамически изменяет свою проводимость. Это происходит, когда суммарный внутренний потенциал превышает пороговое значение масштаба -50 мв. Мембрана на короткое время, сотавляющее около 2 миллисекунд, изменяет свою полярность (деполяризуется) и достигает *потенциала действия* около +40 мв. На микроуровне это объясняется кратковременным повышением проницаемости мембраны для ионов Na^+ и активным поступлением их в аксон. В дальнейшем, по мере выхода ионов калия, положительный заряд с внутренней стороны мембраны меняется на отрицательный, и наступает так называемый *период рефрактерности*, длящийся около 200 мс. В течении этого времени нейрон является полностью пассивным, практически неизменно сохраняя потенциал внутри аксона на уровне около -70 мв.

Импульс деполяризации клеточной мембраны, называемый *спайком*, распространяется вдоль аксона практически без затухания, поддерживаясь локальными ионными градиентами. Скорость перемещения спайка является относительно невысокой и составляет от 100 до 1000 сантиметров в секунду.

Возбуждение нейрона в виде спайка передается другим нейронам, которые таким образом объеденены в проводящую нервные импульсы сеть. Участки мембраны на аксоне, где размещаются области контакта аксона данного нейрона с дендритами другими нейронов, называются *синапсами*. В области синапса, имеющего сложное строение, происходит обмен информацией о возбуждении между нейронами. Механизмы синаптической передачи достаточно сложны и разнообразны. Они могут иметь химическую и электрическую природу. В химическом синапсе в передаче импульсов участвуют специфические химические вещества - *нейромедиаторы*, вызывающие изменения проницаемости локального участка мембраны. В зависимости от типа вырабатываемого медиатора синапс может обладать возбуждающим (эффективно проводящим возбуждение) или тормозящим действием. Обычно на всех отростках одного нейрона вырабатывается один и тот же медиатор, и поэтому нейрон в целом функционально является тормозящим или возбуждающим. Это важное наблюдение о наличии нейронов различных типов в последующих главах будет существенно использоваться при проектировании искусственных систем.

Нейронные сети. Взаимодействующие между собой посредством передачи через отростки возбуждений нейроны формируют *нейронные сети*. Переход от рассмотрения отдельного нейрона к изучению нейронных сетей является естественным шагом в нейробиологической иерархии.

Общее число нейронов в центральной нервной системе человека достигает 10^{10} - 10^{11}, при этом каждая нервная клетка связана в среднем с 10^3 - 10^4 других нейронов. Установлено, что в головном мозге совокупность нейронов в объеме масштаба 1 мм^{3}формирует относительно независимую локальную сеть, несущую определенную функциональную нагрузку.

Выделяют несколько (обычно три) основных типов нейронных сетей, отличающихся структурой и назначением. Первый тип составляют *иерархические* сети, часто встречающиеся в сенсорных и двигательных путях. Информация в таких сетях передается в процессе последовательного перехода от одного уровня иерархии к другому.

Нейроны образуют два характерных типа соединений - *конвергентные*, когда большое число нейронов одного уровня контактирует с меньшим числом нейронов следующего уровня, и *дивергентные*, в которых контакты устанавливаются со все большим числом клеток последующих слоев иерархии. Сочетание конвергентных и дивергентных соединений обеспечивает многократное дублирование информационных путей, что является решающим фактором надежности нейронной сети. При гибели части клеток, сохранившиеся нейроны оказываются в состоянии поддерживать функционирование сети. Ко второму типу нейронных сетей относятся *локальные* сети, формируемые нейронами с ограниченными сферами влияния. Нейроны локальных сетей производят переработку информации в пределах одного уровня иерархии. При этом функционально локальная сеть представляет собой относительно изолированную тормозящую или возбуждающую структуру. Важную роль также играют так называемые дивергентные *сети с одним входом*. Командный нейрон, находящийся в основании такой сети может оказывать влияние сразу на множество нейронов, и поэтому сети с одним входом выступают согласующим элементом в сложном сочетании нейросетевых систем всех типов.

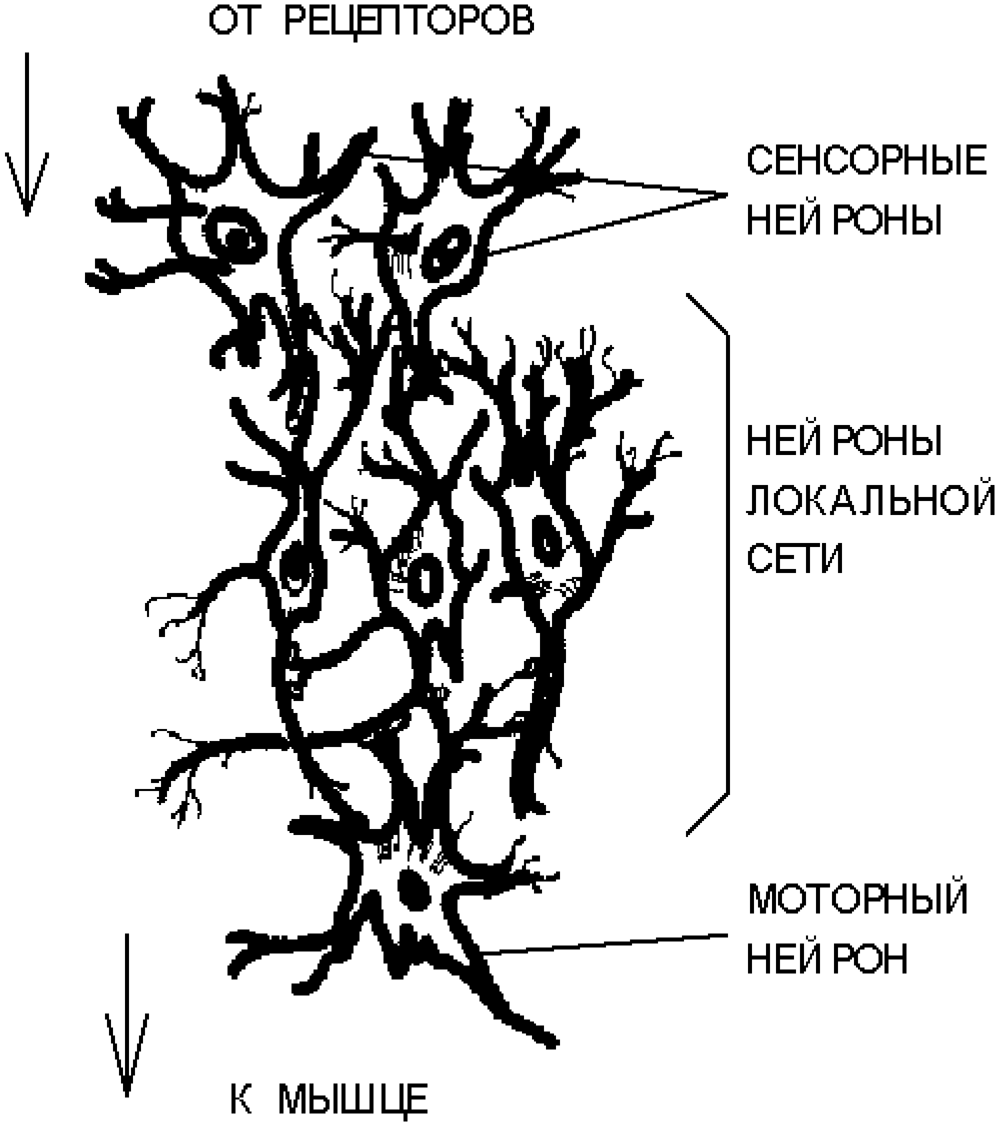

Рис. 2.Структура простой рефлекторной нейронной сети.

Рассмотрим схематически нейронную сеть, формирующую простой рефлекторную цепь с передачей возбуждения от раздражителя к двигательной мышце (рис. 2).

Сигнал внешнего раздражителя воспринимается *сенсорными* нейронами, связанными с чувствительными клетками-рецепторами. Сенсорные нейроны формируют первый (нижний) уровень иерархии. Выработанные ими сигналы передаются нейронам локальной сети, содержащим множество прямых и обратных связей с сочетанием дивергентных и конвергентных соединений. Характер преобразованного в локальных сетях сигнала определяет состояние возбуждения *моторных* нейронов. Эти нейроны, составляющие верхний в рассматриваемой сети уровень иерархии, образно говоря, «принимают решение», которое выражается в воздействии на клетки мышечной ткани посредством нервно-мышечных соединений.

Биологическая изменчивость и обучение нейронных сетей. Структура основных типов нейронных сетей генетически предопределена. При этом исследования в области сравнительной нейроанатомии говорят о том, что по фундаментальному плану строения мозг очень мало изменился в процессе эволюции. Однако детерминированные нейронные структуры демонстрируют свойства изменчивости, обуславливающие их адаптацию к конкретным условиям функционирования.

Генетическая предопределенность имеет место также и в отношении свойств отдельных нейронов, таких, например, как тип используемого нейромедиатора, форма и размер клетки. Изменчивость на клеточном уровне проявляется в *пластичности* синоптических контактов. Характер метаболической активности нейрона и свойства проницаемости синаптической мембраны могут меняться в ответ на длительную активизацию или торможение нейрона. Синаптический контакт «тренируется» в ответ на условия функционирования.

Изменчивость на уровне сети связана со спецификой нейронов. Нервная ткань практически лишена характерной для других типов тканей способности к регенерации путем деления клеток. Однако нейроны демонстрируют способность к формированию новых отростков и новых синаптических контактов. Ряд экспериментов с преднамеренным повреждением нервных путей указывает, что развитие нейронных ответвлений сопровождается конкуренцией за обладание синаптическими участками. Это свойство в целом обеспечивает устойчивость функционирования нейронный сетей при относительной ненадежности их отдельных компонент - нейронов.

Специфическая изменчивость нейронных сетей и свойств отдельных нейронов лежит в основе их способности к обучению - адаптации к условиям функционирования - при неизменности в целом их морфологической структуры. Следует заметить, однако, что рассмотрение изменчивости и обучаемости малых групп нейронов не позволяет в целом ответить на вопросы об обучаемости на уровне высших форм психической деятельности, связанных с интеллектом, абстрактным мышлением, речью.

Прежде чем перейти к рассмотрению моделей нейронов и искусственных нейронных сетей, сформулируем общие фактологические положения о биологических нейронных сетях.

Основными действующими элементами нервной системы являются отдельные клетки, называемые нейронами. Они имеют ряд общих с клетками других типов черт, при этом сильно отличаясь от них по своей конфигурации и функциональному назначению. Активность нейронов при передаче и обработке нервных импульсов регулируется свойствами мембраны, которые могут меняться под воздействием синаптических медиаторов. Биологические функции нейрона могут меняться и адаптироваться к условиям функционирования. Нейроны объединяются в нейронные сети, основные типы которых, а также схемы проводящих путей мозга являются генетически запрограммированными. В процессе развития возможно локальное видоизменение нейронных сетей с формированием новых соединений между нейронами. Отметим также, что нервная система содержит помимо нейронов клетки других типов.

Формальный нейрон

Исторически первой работой, заложившей теоретический фундамент для создания искусственных моделей нейронов и нейронных сетей, принято считать опубликованную в 1943 г. статью Уоррена С.Мак-каллока и Вальтера Питтса «Логическое исчисление идей, относящихся к нервной активности». Главный принцип теории Маккалока и Питтса заключается в том, что произвольные явления, относящиеся к высшей нервной деятельности, могут быть проанализированы и поняты, как некоторая активность в сети, состоящей из логических элементов, принимающих только два состояния («все или ничего»). При этом для всякого логического выражения, удовлетворяющего указанным авторами условиям, может быть найдена сеть логических элементов, имеющая описываемое этим выражением поведение. Дискуссионные вопросы, касающиеся возможности моделирования психики, сознания и т.п. находятся за рамками этой книги.

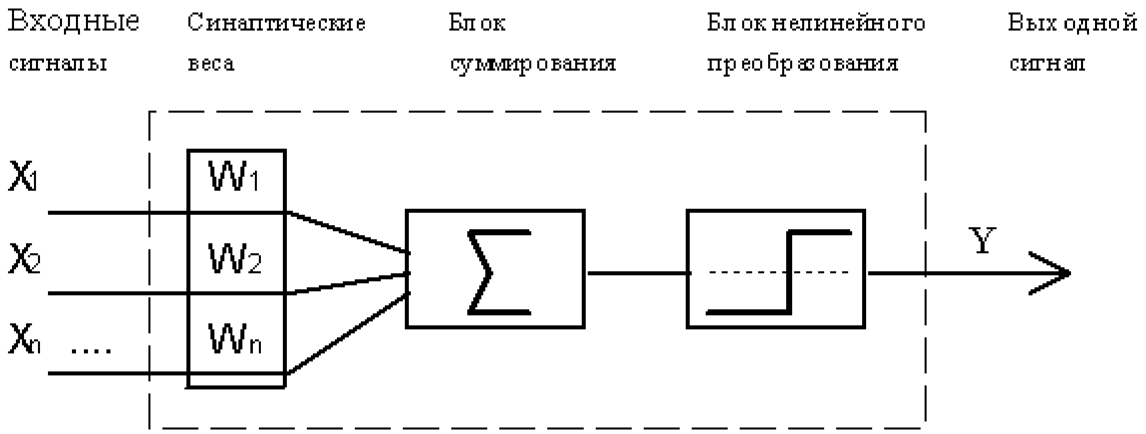

Рис. 3. Функциональная схема формального нейрона Маккалока и Пиитса.

В качестве модели такого логического элемента, получившего в дальнейшем название «формальный нейрон», была предложена схема, приведенная на рис. 6.3. С современной точки зрения, формальный нейрон представляет собой математическую модель простого процессора, имеющего несколько входов и один выход. Вектор входных сигналов (поступающих через «дендриты») преобразуется нейроном в выходной сигнал (распространяющийся по «аксону») с использованием трех функциональных блоков: локальной памяти, блока суммирования и блока нелинейного преобразования.

Вектор локальной памяти содержит информацию о весовых множителях, с которыми входные сигналы будут интерпретироваться нейроном. Эти переменные веса являются аналогом чувствительности пластических синаптических контактов. Выбором весов достигается та или иная интегральная функция нейрона.

В блоке суммирования происходит накопление общего входного сигнала (обычно обозначаемого символом ***net***), равного взвешенной сумме входов:

$$net = \sum_{i=1}^{n} W_i x_i$$

В модели Маккалока и Питтса отсутствуют временные задержки входных сигналов, поэтому значение ***net*** определяет полное внешненее возбуждение, воспринятое нейроном. Отклик нейрон далее описывается по принципу «все или ничего», т. е. переменная подвергается нелинейному пороговому преобразованию, при котором выход (состояние активации нейрона) Y устанавливается равным единице, если net > , и Y=0 в обратном случае. Значение порога (часто полагаемое равным нулю) также хранится в локальной памяти.

Формальные нейроны могут быть объединены в сети путем замыкания выходов одних нейронов на входы других, и по мысли авторов модели, такая кибернетическая система с надлежаще выбранными весами может представлять произвольную логическую функцию. Для теоретического описания получаемых нейронных сетей предлагался математический язык исчисления логических предикатов.

Нужно отметить, что сегодня исчерпывающей теории синтеза логических нейронных сетей с произвольной функцией, по-видимому, нет. Наиболее продвинутыми оказались исследования в области многослойных систем и сетей с симметричными связями. Большинство моделей опираются в своей основе на различных модификациях формального нейрона. Важным развитием теории формального нейрона является переход к аналоговым (непрерывным) сигналам, а также к различным типам нелинейных переходных функций. Опишем наиболее широко используемые типы переходных функций Y=f(net).

- Пороговая функция (рассмотренная Маккалоком и Питтсом):

$$Y = f(net) = \begin{cases} 1, & net > \Theta \\ 0, & net \le \Theta \end{cases}$$

- Линейная функция, а также ее вариант - линейная функция с погашением отрицательных сигналов:

$$Y = f(net) = \begin{cases} net, & net > \Theta \\ 0, & net \le \Theta \end{cases}$$

- Сигмоидальная функция:

$$Y = f(net) = \frac{1}{1 + \exp(-(net - \Theta))}$$

Сигмоидальная функция обладает избирательной чувствительностью к сигналам разной интенсивности, что соответствует биологическим данным. Наибольшая чувствительность наблюдается вблизи порога, где малые изменения сигнала ***net*** приводят к ощутимым изменениям выхода. Напротив, к вариациям сигнала в областях значительно выше или ниже порогового уровня сигмоидальная функция не чувствительна, так как ее производная при больших и малых аргументах стремится к нулю.

В последнее время также рассматриваются математические модели формальных нейронов, учитывающие нелинейные корреляции между входами. Для нейронов Маккалока и Питтса предложены электротехнические аналоги, позволяющие проводить прямое аппаратное моделирование.

Обучение нейрона детектированию границы «черное-белое». Способность формального нейрона к обучению проявляется в возможности изменения значений вектора весов W, соответствующей пластичности синапсов биологических нейронов. Рассмотрим обучение формального нейрона на примере простейшей задачи детектирования границы. Пусть имеется образ, составленный из одномерной цепочки черных и белых клеток. Зачерненные клетки соответствуют единичному сигналу, а белые клетки - нулевому. Сигнал на входах формального нейрона устанавливается равным значениям пар примыкающих клеток рассматриваемого образа. Нейрон обучается всякий раз возбуждаться и выдавать единичный выходной сигнал, если его первый вход (на рис. 4. - левый) соединен с белой клеткой, а второй (правый) - с черной. Таким образом, нейрон должет служить детектором границы перехода от светлого к темному тону образа.

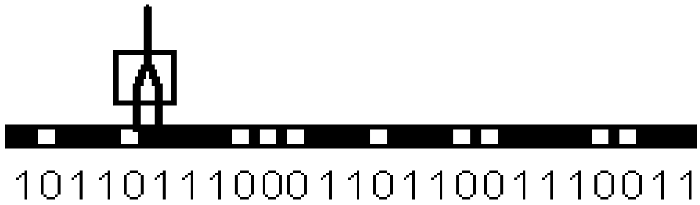

Рис. 4. Формальный нейрон с двумя входами, занятый обработкой образа в виде одномерной цепочки черных и белых клеток.

Функция, выполняемая нейроном, определяется следующей таблицей.

Вход 1	**Вход 2**	**Требуемый выход**
1	1	0
1	0	0
0	1	1
0	0	0

Для данной задачи значения весов и порога нейрона могут быть предъявлены и без специальной процедуры обучения. Легко убедиться, что нужным требованиям удовлетворяет набор □ = 0, W_1 = -1, W_1 = +1. В случае задачи детектирования границы перехода от темного к светлому веса нужно поменять местами.

В общем случае для подстройки весов при обучении нейрона разработаны различные алгоритмы, которые будут рассматриваться в применении к конкретным типам нейронных сетей, составленных из формальных нейронов.

Рассмотрим наиболее простые модели *нейронных сетей*: однослойный и многослойный *персептрон*.

Персептрон

Большое количество моделей *персептрона* рассмотрено в основополагающей работе Розенблатта [47]. Простейшая модель *нейронной сети* - однослойный *персептрон*.

Однослойный *персептрон* (персептрон Розенблатта) - однослойная *нейронная сеть*, все нейроны которой имеют жесткую пороговую функцию активации.

Однослойный *персептрон* имеет простой алгоритм обучения и способен решать лишь самые простые задачи. Эта модель вызвала к себе большой интерес в начале 1960-х годов и стала толчком к развитию искусственных *нейронных сетей*.

Классический пример такой *нейронной сети* - однослойный трехнейронный *персептрон* - представлен на рис.4.

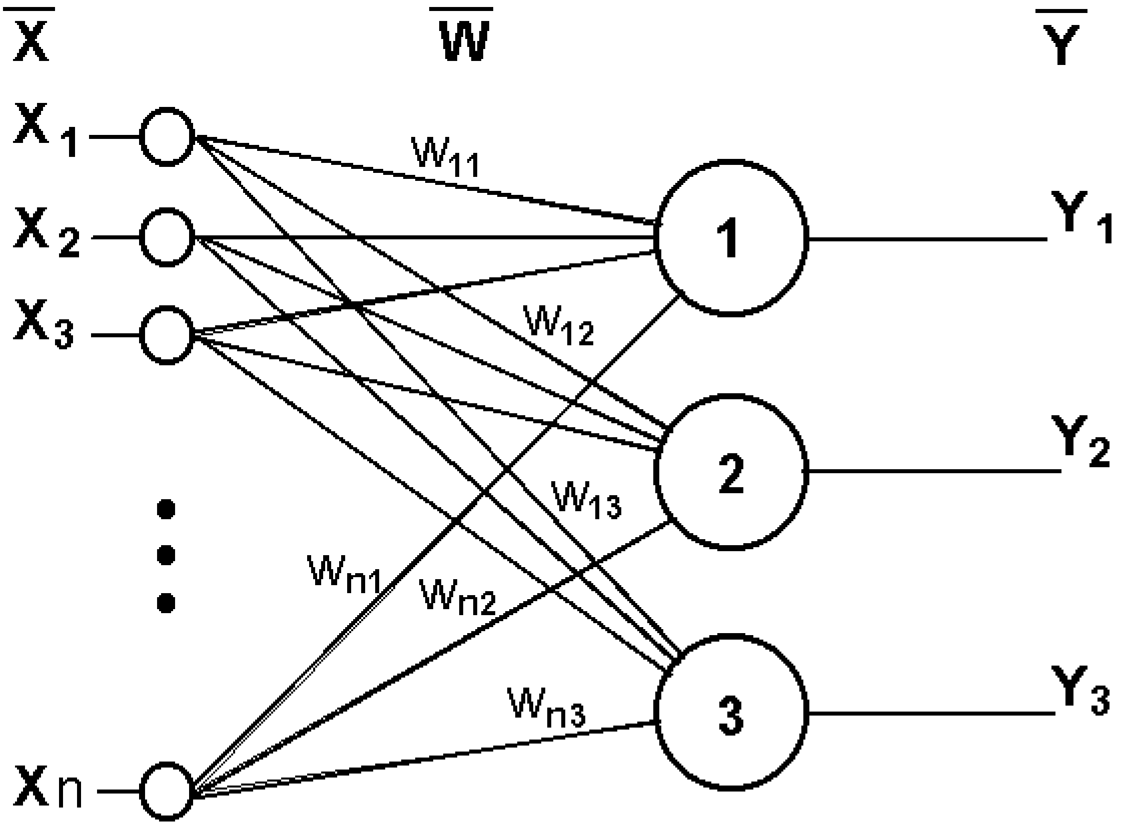

Рис. 4. Однослойный трехнейронный персептрон

Сеть, изображенная на рисунке, имеет n входов, на которые поступают сигналы, идущие по *синапсам* на 3 нейрона. Эти три нейрона образуют единственный *слой* данной сети и выдают три выходных сигнала.

Многослойный *персептрон* (*MLP*) - *нейронная сеть* прямого распространения сигнала (без обратных связей), в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько *слоев*.

Первый из таких *слоев* называют входным, последний - выходным.

Эти *слои* содержат так называемые вырожденные нейроны и иногда в количестве *слоев* не учитываются. Кроме входного и выходного *слоев*, в многослойном *персептроне* есть один или несколько промежуточных *слоев*, которые называют скрытыми.

В этой модели *персептрона* должен быть хотя бы один скрытый *слой*. Присутствие нескольких таких *слоев* оправдано лишь в случае использования нелинейных *функций активации*.

Пример двухслойного *персептрона* представлен на рис.4.

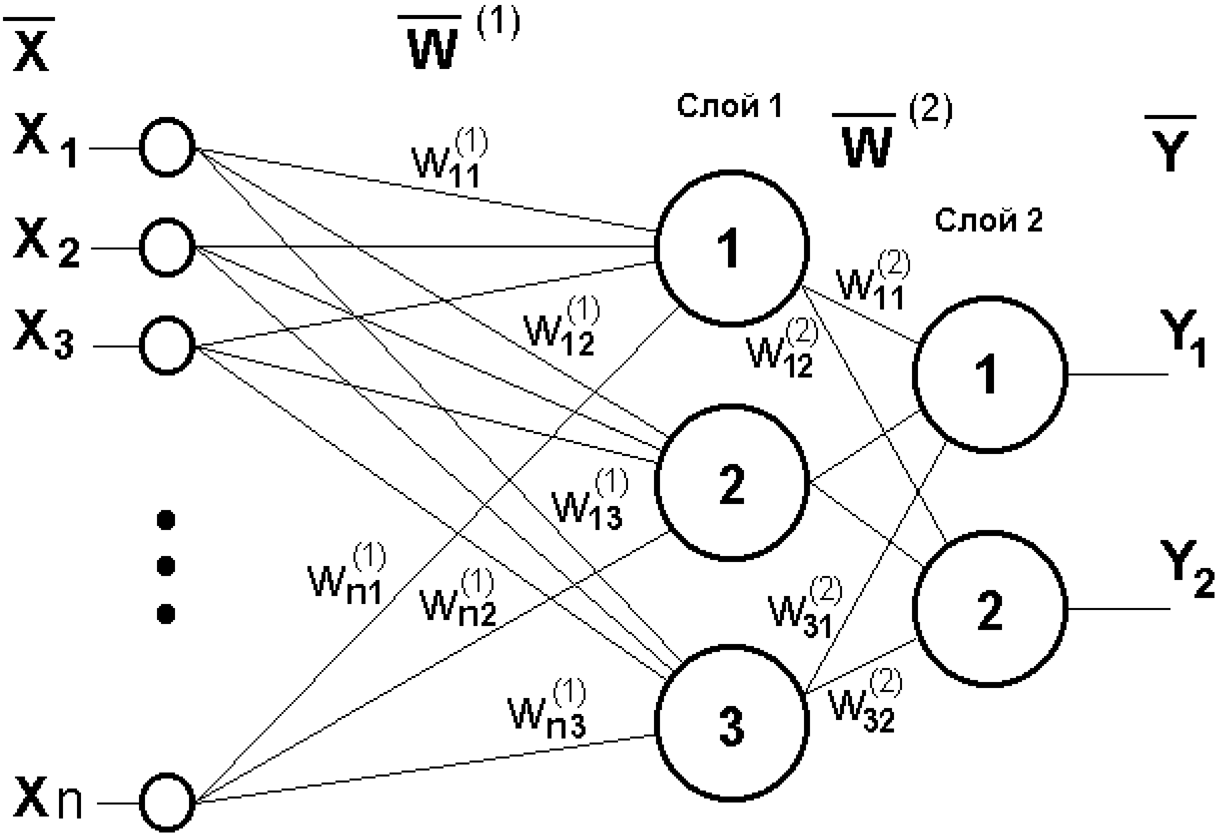

Рис. 4. Двухслойный перцептрон

Сеть, изображенная на рисунке, имеет n входов. На них поступают сигналы, идущие далее по *синапсам* на 3 нейрона, которые образуют первый *слой*. Выходные сигналы первого *слоя* передаются двум нейронам второго *слоя*. Последние, в свою очередь, выдают два выходных сигнала.

Метод обратного распространения ошибки (Back *propagation*, backprop) - алгоритм обучения многослойных *персептронов*, основанный на вычислении градиента *функции ошибок*. В процессе обучения веса нейронов каждого *слоя* нейросети корректируются с учетом сигналов, поступивших с предыдущего *слоя*, и *невязки* каждого *слоя*, которая вычисляется рекурсивно в обратном направлении от последнего *слоя* к первому.

Самоорганизу́ющаяся ка́рта Ко́хонена (англ. *Self-organizing map* — SOM) — нейронная сеть с обучением без учителя, выполняющая задачу визуализации и кластеризации. Идея сети предложена финским учёным Т. Кохоненом. Является методом проецирования многомерного пространства в пространство с более низкой размерностью (чаще всего, двумерное), применяется также для решения задач моделирования, прогнозирования, выявление наборов независимых признаков, поиска закономерностей в больших массивах данных, разработке компьютерных игр, квантизации цветов к их ограниченному числу индексов в цветовой палитре:

при печати на принтере и ранее на ПК или же на приставках с дисплеем с пониженным числом цветов, для архиваторов [общего назначения] или видео-кодеков, и прч. Является одной из версий нейронных сетей Кохонена.

Самоорганизующаяся карта состоит из компонентов, называемых узлами или нейронами. Их количество задаётся аналитиком. Каждый из узлов описывается двумя векторами. Первый — т. н. вектор веса m, имеющий такую же размерность, что и входные данные. Второй — вектор r, представляющий собой координаты узла на карте. Карта Кохонена визуально отображается с помощью ячеек прямоугольной или шестиугольной формы; последняя применяется чаще, поскольку в этом случае расстояния между центрами смежных ячеек одинаковы, что повышает корректность визуализации карты.

Изначально известна размерность входных данных, по ней некоторым образом строится первоначальный вариант карты. В процессе обучения векторы веса узлов приближаются к входным данным. Для каждого наблюдения (семпла) выбирается наиболее похожий по вектору веса узел, и значение его вектора веса приближается к наблюдению. Также к наблюдению приближаются векторы веса нескольких узлов, расположенных рядом, таким образом если в множестве входных данных два наблюдения были схожи, на карте им будут соответствовать близкие узлы. Циклический процесс обучения, перебирающий входные данные, заканчивается по достижении картой допустимой (заранее заданной аналитиком) погрешности, или по совершении заданного количества итераций. Таким образом, в результате обучения карта Кохонена классифицирует входные данные на кластеры и визуально отображает многомерные входные данные в двумерной плоскости, распределяя векторы близких признаков в соседние ячейки и раскрашивая их в зависимости от анализируемых параметров нейронов.

В результате работы алгоритма получаются следующие карты:

карта входов нейронов — визуализирует внутреннюю структуру входных данных путём подстройки весов нейронов карты. Обычно используется несколько карт входов, каждая из которых отображает один из них и раскрашивается в зависимости от веса нейрона. На одной из карт определенным цветом обозначают область, в которую включаются приблизительно одинаковые входы для анализируемых примеров.

карта выходов нейронов — визуализирует модель взаимного расположения входных примеров. Очерченные области на карте представляют собой кластеры, состоящие из нейронов со схожими значениями выходов.

специальные карты — это карта кластеров, полученных в результате применения алгоритма самоорганизующейся карты Кохонена, а также другие карты, которые их характеризуют.[1]

Работа сети

Инициализация карты, то есть первоначальное задание векторов веса для узлов.

Цикл:

Выбор следующего наблюдения (вектора из множества входных данных).

Нахождение для него лучшей единицы соответствия (best matching unit, BMU, или Winner) — узла на карте, вектор веса которого меньше всего отличается от наблюдения (в метрике, задаваемой аналитиком, чаще всего, евклидовой).

Определение количества соседей BMU и обучение — изменение векторов веса BMU и его соседей с целью их приближения к наблюдению.

Определение ошибки карты.

Алгоритм

Инициализация

Наиболее распространены три способа задания первоначальных весов узлов:

Задание всех координат случайными числами.

Присваивание вектору веса значение случайного наблюдения из входных данных.

Выбор векторов веса из линейного пространства, натянутого на главные компоненты набора входных данных.

Цикл

Пусть {\displaystyle t}t — номер итерации (инициализация соответствует номеру 0).

Выбрать произвольное наблюдение {\displaystyle x(t)}x(t) из множества входных данных.

Найти расстояния от него до векторов веса всех узлов карты и определить ближайший по весу узел {\displaystyle M_{c}(t)}M_c(t). Это — BMU или Winner. Условие на {\displaystyle M_{c}(t)}M_c(t):

{\displaystyle \|x(t)-m_{c}(t)\|\leq \|x(t)-m_{i}(t)\|} \| x(t)-m_c(t)\|\leq\| x(t)-m_i(t)\|,

для любого {\displaystyle m_{i}(t)} m_i(t), где {\displaystyle m_{i}(t)}m_i(t) — вектор веса узла {\displaystyle M_{i}(t)}M_i(t). Если находится несколько узлов, удовлетворяющих условию, BMU выбирается случайным образом среди них.

Определить с помощью функции {\displaystyle h}h (функции соседства) соседей {\displaystyle M_{c}}M_c и изменение их векторов веса.

Задание {\displaystyle h}h

Функция определяет «меру соседства» узлов {\displaystyle M_{i}}M_{i} и {\displaystyle M_{c}}M_c и изменение векторов веса. Она должна постепенно уточнять их значения, сначала у большего количества узлов и сильнее, потом у меньшего и слабее. Часто в качестве функции соседства используется гауссовская функция:

{\displaystyle h_{ci}(t)=\alpha (t)\cdot \exp(-{\frac {\|r_{c}-r_{i}\|^{2}}{2\sigma ^{2}(t)}})}h_{ci}(t)=\alpha(t)\cdot\exp(-\frac{\|r_c-r_i\|^2}{2\sigma^2(t)})

где {\displaystyle 0<\alpha (t)<1}0<\alpha(t)<1 — обучающий сомножитель, монотонно убывающий с каждой последующей итерацией (то есть определяющий приближение значения векторов веса BMU и его соседей к наблюдению; чем больше шаг, тем меньше уточнение);

{\displaystyle r_{i}}r_{i}, {\displaystyle r_{c}}r_{c} — координаты узлов {\displaystyle M_{i}(t)}M_i(t) и {\displaystyle M_{c}(t)}M_c(t) на карте;

{\displaystyle \sigma (t)}\sigma(t) — сомножитель, уменьшающий количество соседей с итерациями, монотонно убывает.

Параметры {\displaystyle \alpha }\alpha , {\displaystyle \sigma }\sigma и их характер убывания задаются аналитиком.

Более простой способ задания функции соседства:

{\displaystyle h_{ci}(t)=\alpha (t)}h_{ci}(t)=\alpha(t),

если {\displaystyle M_{i}(t)}M_i(t) находится в окрестности {\displaystyle M_{c}(t)}M_c(t) заранее заданного аналитиком радиуса, и 0 в противном случае.

Функция {\displaystyle h(t)}h(t) равна {\displaystyle \alpha (t)}\alpha(t) для BMU и уменьшается с удалением от BMU.

Изменение векторов веса

Изменить вектор веса по формуле:

{\displaystyle m_{i}(t)=m_{i}(t-1)+h_{ci}(t)\cdot (x(t)-m_{i}(t-1))}m_i(t)=m_i(t-1)+h_{ci}(t)\cdot(x(t)-m_i(t-1))

Т.о. вектора веса всех узлов, являющихся соседями BMU, приближаются к рассматриваемому наблюдению.

Вычисление ошибки карты

Например, как среднее арифметическое расстояний между наблюдениями и векторами веса соответствующих им BMU:

{\displaystyle {\frac {1}{N}}\sum _{i=1}^{N}\|x_{i}-m_{c}\|}\frac{1}{N}\sum_{i=1}^{N}\|x_{i}-m_{c}\|,

где N — количество элементов набора входных данных.

Особенности модели

Устойчивость к зашумленным данным, быстрое и неуправляемое обучение, возможность упрощения многомерных входных данных с помощью визуализации.[2]

Самоорганизующиеся карты Кохонена могут быть использованы для кластерного анализа только в том случае, если заранее известно число кластеров[2].

Важным недостатком является то, что окончательный результат работы нейронных сетей зависит от начальных установок сети. С другой стороны, нейронные сети теоретически могут аппроксимировать любую непрерывную функцию, что позволяет исследователю не принимать заранее какие-либо гипотезы относительно модели[2].

Лекция №5.

Методы и алгоритмы анализа структуры многомерных данных Эволюционные методы построения СИИ

Генетический алгоритм, составляющий основу эволюционных вычислений, – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путем последовательного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию. Сущность алгоритма состоит в следующем. Задача кодируется таким образом, чтобы ее решение могло быть представлено в виде вектора (такой вектор называется хромосомой). Случайным образом создается некоторое количество начальных векторов (начальная популяция). Они оцениваются с использованием т.н. функции приспособленности, в результате чего каждому вектору присваивается определенное значение (приспособленность), которое определяет вероятность выживания организма, представленного данным вектором. После этого с использованием полученных значений приспособленности выбираются вектора (селекция), допущенный к скрещиванию. К этим векторам применяются т.н. генетические операторы (в большинстве случаев – скрещивание и мутация), создавая таким образом следующее поколение. Особи следующего поколения также оцениваются, затем производится селекция, применяются генетические операторы и т.д. Так моделируется эволюционный процесс, продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий останова алгоритма. Генетические алгоримы служат, главным образом, для поиска решений в очень больших, сложных пространствах поиска, и применяются для решения следующих задач:

- Оптимизация функций
- Разнообразные задачи на графах (задача коммивояжера, раскраска, нахождение паросочетаний)
- Настройка и обучение нейронной сети
- Задачи компоновки
- Составление расписаний
- Игровые стратегии
- Аппроксимация функций
- Искусственная жизнь
- Биоинформатика

Многомерные методы анализа данных применяются с целью обнаружения скрытых закономерностей, выявления наиболее существенных связей между переменными путем анализа взаимосвязи между большим количеством этих переменных. К таким методам относят факторный анализ, кластерный анализ, дисперсионный анализ, регрессионный анализ, латентно-структурный анализ, многомерное шкалирование и др.

Факторный анализ заключается в выявлении и интерпретации факторов. Фактор – обобщенная переменная, которая позволяет свернуть часть информации, т.е. представить ее в удобообозримом виде. Например, факторная теория личности выделяет ряд обобщенных характеристик поведения, которые в данном случае называются чертами личности.

Кластерный анализ позволяет выделить ведущий признак и иерархию взаимосвязей признаков.
Дисперсионный анализ – статистический метод, используемый для изучения одной или нескольких одновременно действующих и независимых переменных на изменчивость наблюдаемого признака. Его особенность состоит в том, что наблюдаемый признак может быть только количественным, в то же время объясняющие признаки могут быть как количественными, так и качественными.

Регрессионный анализ позволяет выявить количественную (численную) зависимость среднего значения изменений результативного (объясняемого) признака (переменной) от изменений одного или нескольких объясняющих признаков (переменных). Как правило, данный вид анализа применяется в том случае, когда требуется выяснить насколько изменяется средняя величина одного признака при изменении на единицу другого признака.
Латентно-структурный анализ представляет собой совокупность аналитико-статистических процедур выявления скрытых переменных (признаков), а также внутренней структуры связей между ними. Он дает возможность исследовать проявления сложных взаимосвязей непосредственно наблюдаемых характеристик социально-психологических и педагогических феноменов. Латентный анализ может стать основой для моделирования указанных взаимосвязей.
Многомерное шкалирование обеспечивает наглядную оценку сходства или различия между некоторыми объектами, описываемыми большим количеством разнообразных переменных. Эти различия представляются в виде расстояния между оцениваемыми объектами в многомерном пространстве.
Математические и статистические методы исследования находят широкое применение при выборе количества подлежащих исследованию объектов, для получения нового фактического материала, для анализа и обобщения полученных эмпирическими методами данных.
Выбор оптимального количества подлежащих исследованию объектов.
Для достижения объективности, надежности и достоверности результатов (выводов) исследования важно обеспечить массовость и представительность (репрезентативность) объектов изучения. С этой целью психолого-педагогических исследования могут проводиться как сплошные или выборочные. В сплошных исследованиях изучаются все единицы (объекты) изучаемого явления. В выборочных исследованиях изучению подвергается только часть интересующих объектов (явлений, процессов), взятых по какому-либо признаку. В научной практике широко применяются именно выборочные исследования, т.к., во-первых, не всегда имеется изучить всю совокупность явлений (объектов, процессов), во-вторых, часто можно достичь достаточно точных выводов после изучения определенной части исследуемых

единиц. В основе выборочного способа исследования лежат положения теории вероятности.

Для соблюдения требований массовости необходимо определить достаточное количество фактов, наблюдений, объектов. С этой целью можно использовать методы вычисления требуемых выборочных совокупностей, которые разработаны математической статистикой. Наиболее простой способ из них: исследователь устанавливает величины вероятности и допускаемой ошибки, по значениям которых по таблице достаточно больших чисел, фрагмент которой приведен в таблице 1[3], находит минимально необходимое количество единиц исследования.

Таблица 1**Краткая таблица достаточно больших чисел**

Допустимая ошибка	Величина вероятности				
0,85	0,90	0,95	0,99	0,995	0,999
10,05					
10,04					
10,03					
10,02					
10,01					

Так, например, допускаемая ошибка в выводах по результатам наблюдений по сравнению с теоретическим предположениями принята равной 0,05 (т.е. мы можем ошибиться не более, чем в 5 случаях из 100). По таблице достаточно больших чисел (см. таблицу 1)можно определить, что правильное заключение может быть сделано в 9 случаях из 10 (с вероятностью 0,9), когда число единиц наблюдения будет не менее 270, или в 99 случаях из 100 (с вероятностью 0,99) – не менее 663 и т.д.

Как видно из таблицы величина требуемой выборки зависит от точности и вероятности, с которыми мы предполагаем делать выводы. В психолого-педагогических исследованиях она не должна быть чрезмерно большой. Для основательных выводов считается, как правило, достаточным 300-500 выбранных для наблюдения единиц.

Выбор объектов психолого-педагогического исследования, достаточно представительных для изучаемого класса явлений, может осуществляться с помощью различных методов отбора: простой случайный отбор, расслоенный случайный отбор, многоступенчатый отбор и др. Случайный выбор объектов

из общей совокупности позволяет получать результаты, которые не будут резко отличаться от результатов, получаемых в случае исследования всей совокупности объектов. Т.е. репрезентативность единиц наблюдения обеспечивается, прежде всего, случайным выбором с помощью таблиц случайных чисел.
Например, для проведения массового эксперимента требуется определить 20 учебных групп из имеющихся 200. Для этого составляется нумерованный список всех групп. Затем из таблицы случайных чисел выписываются 20 номеров, начиная с какого-либо числа, через определенный интервал. Эти 20 случайных чисел определяют те группы, которые выбираются для исследования.
Для получения нового фактического материала используются приемы шаблонирования и шкалирования. *Шаблонирование* повышает информативную емкость анкетного опроса. *Шкалирование* дает возможность: более точно оценивать действия исследователя и исследуемых; объективно и точно диагностировать и измерять интенсивность определенных психолого-педагогических явлений; упорядочить, количественно оценить, определить низшую и высшую ступени исследуемого процесса, явления. В психолого-педагогических исследованиях используются трехмерная, многомерная, двусторонняя, биполярная и другие оценочные шкалы. Они могут отражаться в виде перечня показателей с определенным числовым значением (таблица 2), в виде таблицы или графически.

Таблица 2 **Примеры оценочных шкал**

Трехмерная шкала	Многомерная шкала	Двусторонняя шкала	Биполярная шкала					
Очень активный		Очень активный		Очень интересуется		Дисциплинированность		Недисциплинированность
Активный		Среднеактивный		Достаточно интересуется				
Пассивный		Не слишком активный		Равнодушен				
		Пассивный		Не интересуется	-5			

		Полностью пассивный		Совершенно нет интереса	-10

Графическое изображение оценочных шкал выражает категории в наглядной форме, при этом каждая ступень шкалы характеризуется вербально.

Анализ и обобщение полученных эмпирическими методами данных. Математические и статистические методы позволяют установить различные соотношения, связи между фактами, проследить тенденции в развитии психолого-педагогических явлений. Так, определить сопоставимость и степень достоверности фактов, выбрать основания для их правильной группировки помогает теория группировок математической статистики. В зависимости от целей и задач исследования применяют типологические, вариационные или аналитические группировки.

Типологическая группировка применяется, когда необходимо разбить фактический материал на качественно однородные единицы. Например, распределение количества нарушений дисциплины между различными категориями студентов, разбивка показателей выполнения студентами физических упражнений по годам учебы.

Вариационная группировка применяется при необходимости сгруппировать материал по величине какого-либо изменяющегося (варьирующего) признака. Например, разбивка групп обучающихся по уровню успеваемости, по процентам выполнения заданий, однотипным нарушениям какого-либо порядка и т.п. Вариационная группировка дает возможность судить о структуре изучаемого явления.

При обобщении данных часто используется прием составления и изучения таблиц. Представление о статике явления дают вариационные ряды и таблицы. Сводка данных относительно одной статистической величины отображается как вариационный ряд распределения значения этой величины. По двум и более статистическим величинам составляется вариационная таблица распределения, раскрывающая распределение значений одной статической величины в соответствии со значениями, которые принимают другие величины. Динамику процесса или явления могут показать ряды развития, в которых первая строка содержит последовательные этапы или промежутки времени, а вторая – полученные на этих этапах значения изучаемой статистической величины. С помощью рядов развития выявляются возрастание, убывание или периодические изменения изучаемого явления, вскрываются его тенденции, закономерности. Таблицы, как правило, заполняются абсолютными или средними, относительными величинами. Результаты обобщения эмпирического материала, помимо и на основе таблиц, часто изображаются графически в виде диаграмм, графиков, фигур и т.п. способами точек, прямых и прямоугольников. Графическое изображение статистического материала таблиц позволяет глубже проникнуть в смысл цифровых величин, уловить их взаимозависимость и черты изучаемого явления. При этом графическое изображение рядов распределения значения одной статистической величины позволяет построить кривые распределения, двух и более статистических величин – поверхность распределения. Ряды развития при графическом исполнении образуют кривые развития. *Аналитическая группировка* помогает устанавливать взаимосвязь между изучаемыми явлениями, их взаимозависимость и взаимообусловленность в точном исчислении. Аналитический метод оперирует математическими формулами, с помощью которых выводятся так называемые обобщающие показатели, т.е. абсолютные величины, приведенные в сравнимый вид – относительные и средние величины, балансы и индексы. С помощью относительных величин (процентов) определяются качественные особенности анализируемых совокупностей, т.е. выявляются отношения части к целому (удельный вес), слагаемых к сумме (структура совокупности), одной части совокупности к другой ее части (характеристика динамики каких-либо изменений во времени) и др.

Литература

1. T. Kohonen, Self-Organizing Maps (Third Extended Edition), New York, 2001, 501 pages. ISBN 3-540-67921-9
2. Дебок Г., Кохонен Т. Анализ финансовых данных с помощью самоорганизующихся карт, Альпина Паблишер, 2001, 317 стр. ISBN 5-89684-013-6
3. Зиновьев А. Ю. Визуализация многомерных данных. — Красноярск: Изд. Красноярского государственного технического университета, 2000. — 180 с.
4. Чубукова И.А. Data Mining. — 2000. — 326 с.
5. Манжула В.Г., Федяшов Д.С. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных. — 2011.
6. Lakhmi C. Jain; N.M. Martin Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications. — CRC Press, CRC Press LLC, 1998.
7. Загвязинский В.И., Атаханов Р. Методология и методы психолого-педагогических исследований: Учебное пособие. – М.: Изд. центр «Академия», 2003.

8. Образцов П.И. Методы и методология психолого-педагогического исследования. – СПб.: Питер, 2004.
9. Давыдов В.П. Основы методологии, методики и технологии педагогического исследования: Научно-методическое пособие. – М.: Академия ФСБ, 1997.