

МЕТОДЫ ОБУЧЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

ВВЕДЕНИЕ

С давних пор люди пытались объяснить, как функционирует их собственное мышление. Нейробиологи и нейроанатомы достигли значительного прогресса в изучении работы мозга. Исследуя структуру и функции нервной системы человека, они многое поняли в «электропроводке» мозга, но мало узнали о его функционировании. В процессе исследований выяснилось, что мозг имеет ошеломляющую сложность: миллиарды нейронов, каждый из которых соединен с сотнями или тысячами других, образуют систему, далеко превосходящую наши самые смелые мечты о суперкомпьютерах.

Понятие «искусственная нейросеть» впервые было введено в 40-х годах прошлого века. Искусственная нейросеть моделирует на логическом уровне деятельность нервной системы человека и животных. В 43-м году была разработана формальная модель нейрона, однако вскоре оказалось, что такая модель способна решать лишь очень ограниченное число задач. Эти трудности можно преодолеть, объединяя формальные нейроны в сеть. Возможности таких систем намного шире: объединенные в сеть формальные нейроны могут решать задачи, которые традиционно относятся к области «человеческой деятельности» (например распознавание образов и даже принятие решений на основе неполной информации). Особенно интересна способность нейросетей обучаться и запоминать информацию, что напоминает мыслительные процессы человека. Именно поэтому в ранних работах по исследованию нейросетей часто упоминался термин «искусственный интеллект».

За последнее время интерес к искусственным нейронным сетям быстро возрос. Ими занялись специалисты из таких дале-

ких областей, как техническое конструирование, физиология, психология. Этот интерес понятен: так как искусственная нейронная сеть, по сути, представляет собой модель естественной нервной системы, поэтому создание и изучение таких сетей позволяет узнать многое о функционировании естественных систем.

Сама теория искусственных нейронных сетей появилась в 40-х годах благодаря последним на тот момент достижениям биологии, так как искусственные нейроны состоят из элементов, которые моделируют элементарные функции биологических нейронов. Эти элементы организуются по способу, который может соответствовать (или не соответствовать) анатомии мозга. Несмотря на такое поверхностное сходство, искусственные нейронные сети демонстрируют удивительные свойства, подобные свойствам естественного мозга. Например, искусственная нейронная сеть способна менять свое поведение в зависимости от внешней среды. Прочитав предъявленные ей входные сигналы (возможно, вместе с требуемыми выходами) она способна «обучиться» так, чтобы обеспечивать требуемую реакцию. После обучения сеть не реагирует на небольшие изменения входных сигналов. Эта способность видеть образ сквозь шум и искажения очень полезна, если нужно решать задачи распознавания образов. Стоит отметить, что нейронная сеть делает обобщения автоматически благодаря своей структуре, а не с помощью специально написанных программ.

Другим интересным свойством нейросетей является надежность: даже если несколько элементов будут работать неправильно или выйдут из строя, то сеть все равно будет способна выдавать правильные результаты, но с меньшей точностью.

Некоторые типы нейронных сетей обладают способностью генерировать абстрактный образ на основе нескольких входных сигналов. Например, можно обучить сеть, предъявляя ей последовательность искаженных изображений буквы «А». После обучения сеть сможет породить букву «А» без искажений, то есть сеть может сгенерировать то, что ей никогда не предъявлялось.

Тем не менее, стоит отметить, что искусственные нейронные сети не являются панацеей. Они очень плохо подходят для задач, где требуются точные и безошибочные математические расчеты.

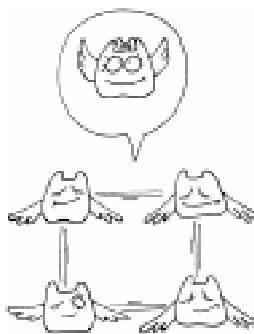
В данной статье рассматриваются формальные модели нейрона и перцептрона, основные отличия нейросетевой архитектуры от классической архитектуры фон Неймана, а также приведен обзор методов, применяемых для обучения нейросетей. В приложении можно найти примеры задач, решаемых нейронными сетями и детальное описание алгоритма обучения однослойного и многослойного перцептрона.

ИСТОРИЯ РАЗВИТИЯ

Первый шаг в исследовании нейросетей был сделан в 1943 г., когда вышла статья нейрофизиолога Уоррена Маккаллоха (Warren McCulloch) и математика Уолтера Питтса (Walter Pitts), посвященная искусственным нейронам, а также реализации модели нейронной сети с помощью электрических схем [17].

1949 г. Д. Хебб (D. Hebb) высказал идеи о характере соединений нейронов мозга и их взаимодействии, а также предложил правила обучения нейронной сети.

1950-е гг. Появились программные модели искусственных нейросетей. Первые работы были проведены Натаниелом Рочестером (Nathaniel Rochester) из исследовательской лаборатории ИВМ. И хотя дальнейшие реализации были успешными, эта модель потерпела неудачу, поскольку бурный рост разработок в области традиционных вычис-



лений оставил в тени нейронные исследования.

1957 г. Ф. Розенблатт (F. Rosenblatt) разработал принципы организации и функционирования перцептронов, а также предложил вариант технической реализации первого в мире нейромикрокомпьютера [18].

1958 г. Джон фон Нейман (John von Neumann) разработал систему на основе вакуумных трубок, имитирующую простые функции нейронов.

1959 г. Бернард Видроу (Bernard Widrow) и Марсиан Хофф (Marcelian Hoff) разработали модели ADALINE (Multiple ADaptive LINear Elements, Адаптивные Линейные Элементы) и MADALINE (Множественные Адаптивные Линейные Элементы). MADALINE использовалась в качестве адаптивного фильтра для устранения помех на телефонных линиях. Эта нейросеть используется и до сих пор [3; 20].

В том же году нейробиолог Фрэнк Розенблатт (Frank Rosenblatt) начал работу над моделью перцептрона. Однослойный перцептрон, построенный Розенблаттом, в настоящее время считается классической моделью нейросети. Розенблатт использовал свой перцептрон, чтобы разделять входные сигналы на два класса. К сожалению, однослойный перцептрон мог выполнять лишь ограниченный класс задач.

1969 г. Опубликована книга М. Минского (M. Minsky) и С. Пейперта (S. Papert) «Перцептроны», в которой доказывалась принципиальная ограниченность возможностей перцептронов [19].

Ранние успехи способствовали тому, что от нейронных сетей стали ожидать слишком много: больше, чем действительно можно было реализовать в рамках той модели. Чрезмерный оптимизм, процветающий в академическом и техническом сообществе, в конце концов привел множество специалистов к разочарованию, и все проекты исследования нейронных сетей были подвергнуты критике как бесперспективные. В результате, финансирование исследований было прекращено вплоть до 80-х годов.

1970–1976 гг. В СССР проводятся активные разработки в области перцептронов [22].

80-е годы. Благодаря работам Джона Хопфилда (John Hopfield), возродился интерес к нейроинформатике. Хопфилд показал, как преодолеть ограничения нейросетей первого поколения, а также разработал теорию нейронных сетей, моделирующих ассоциативную память.

1985 г. Появились первые коммерческие нейрокомпьютеры, например, Mark III фирмы TRW (США).

1986 г. Дэвид Румельхарт с соавторами предложил алгоритм обучения многослойного перцептрона. В то же время в Киото (Япония) состоялась объединенная американо-японская конференция по нейронным сетям. На этой конференции нейронные сети объявили пятым поколением ЭВМ.

С 1985 г. Американский Институт Физики начал ежегодные встречи – «Нейронные сети для вычислений».

1987 г. Начало широкомасштабного финансирования разработок в области искусственных нейронных сетей в США, Японии и Западной Европе (японская программа «Human Frontiers» и европейская программа «Basic Research in Adaptive Intelligence and Neurocomputing»).

1989 г. Разработки и исследования в области искусственных нейросетей ведутся практически всеми крупными электротехническими фирмами. Нейрокомпьютеры становятся одним из самых динамичных секторов рынка (за два года объем продаж вырос в пять раз). Агентством DARPA (Defence Advanced Research Projects Agency) министерства обороны США начато финансирование программы по созданию сверхбыстродействующих образцов нейрокомпьютеров для разнообразных применений.

1990 г. Активизация советских научных организаций в области исследования искусственных нейронных сетей и нейрокомпьютеров (Институт кибернетики им. Глушкова в Киеве, Институт многопроцессорных вычислительных систем в Таганроге, Институт нейрокибернетики в Ростове-на-Дону) [22].

1991 г. Годовой объем продаж на рынке нейрокомпьютеров приблизился к 140 млн. долларам. Были созданы центры исследования нейрокомпьютеров в Москве, Киеве, Минске, Новосибирске, Санкт-Петербурге [21].

1992 г. Работы в области нейросетей находятся в стадии интенсивного развития. Ежегодно проводится десятки международных конференций и форумов по нейронным сетям, число специализированных периодических научных изданий по указанной тематике достигает двух десятков наименований.

1997 г. Годовой объем продаж на рынке искусственных нейросетей и нейрокомпьютеров превысил 9 млрд. долларов, а ежегодный прирост составил 50%.

2000 г. Благодаря переходу на субмикронные и нанотехнологии, а также успехам молекулярной и биомолекулярной технологии, появляются принципиально новые архитектурные и технологические подходы к созданию нейрокомпьютеров.

ЧТО ТАКОЕ НЕЙРОСЕТЬ

Исследователи до сих пор не пришли к единому мнению относительно определения нейронной сети. В литературе можно встретить множество вариантов, например:

«Нейронная сеть – система, состоящая из множества простых вычислительных элементов, работающих параллельно. Результат работы сети определяется структурой сети, силой связей, а также видом вычислений, выполняемых каждым элементом» [15].

«Нейронная сеть – параллельный распределенный процессор, способный самостоятельно извлекать данные из поступающей информации. Работа такой сети напоминает работу мозга, так как знания получают с помощью процесса обучения, а полученные знания хранятся не в отдельном элементе, а распределены по всей сети» [12].

«Нейронная сеть – система, состоящая из большого числа простых вычислительных элементов. Результат работы каждого элемента зависит только от его внутреннего состояния. Все элементы работают незави-

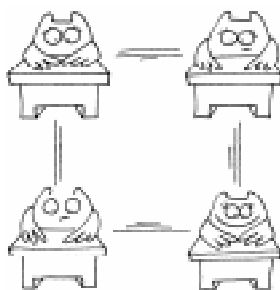
симо друг от друга, то есть без синхронизации с остальными элементами» [13].

«Искусственные нейронные сети – системы, способные получать, хранить и использовать знания» [14].

Тем не менее, большинство исследователей сходятся на том, что нейронная сеть – это система, состоящая из множества простых процессоров, каждый из которых обладает локальной памятью. Содержимое такой памяти принято называть состоянием процессора. Процессоры способны обмениваться между собой числовыми данными. Результат работы процессора зависит только от его состояния и данных, которые он получает на входе.¹

Перед тем как использовать нейросеть, необходимо провести так называемую процедуру обучения, во время которой на основании поступающих данных состояния каждого элемента корректируются таким образом, чтобы сеть вычисляла правильный ответ.

Другими словами, сеть «обучается» на примерах, подобно тому как ребенок учится по картинкам отличать кошку от собаки.



Отличия нейросетевой архитектуры от классической архитектуры фон Неймана

Можно провести следующую аналогию. Допустим, есть функция $y = (2 \cdot x + 1) / 2$. Как получить y при $x = 3$? Очень просто: два умножается на три, затем прибавляется единица и полученный результат делится на два. Получается 3,5. Эта последовательность действий и является простейшей программой.

Однако есть и другой способ решить ту же задачу. Можно построить график этой функции, а затем по графику найти решение. Хотя этот способ и кажется «не математическим», но его достоинства сразу становятся очевидными, когда есть только график функции, но не сама функция. Например, может быть известно изображение буквы. Очевидно, что поиск функции, описы-

вающей данное изображение, будет очень трудоемким.

Если продолжить эту аналогию, то процесс обучения нейросети является своего рода построением графика. То есть, мы сообщаем множество координат $\{(x_1, y_1), \dots, (x_n, y_n)\}$. По этим координатам строятся точки, после чего ближайшие точки соединяются прямыми линиями. Таким образом получается график, с помощью которого по любому заданному x можно выяснить значение y . При этом никаких вычислений не потребуется, результат будет найден на графике.

Правда, здесь возникает одна сложность.

Через заданные точки можно провести бесконечное число кривых. Следовательно, потом, пытаясь определить y по x , мы получим бесконечно много вариантов ответа. Но эту проблему можно решить: во-первых, значения y будут близки, во-вторых, существует способ минимизировать ошибку.

Это главное достоинство нейросетевой архитектуры. Для обработки в обычном компьютере всякую задачу нужно формализовать (изображение буквы превратить в функцию). При этом, если возникнет хотя бы небольшая ошибка в начальных данных или хотя бы одно из выражений будет повреждено, неверным окажется и конечный результат. В рассмотренном примере ($y = (2 \cdot x + 1) / 2$), если первая двойка из-за каких-то неполадок превратится в 4, в итоге мы получим при $x = 3$ $y = 6,5$. А для графика потеря части данных не играет решающей роли. Если нарисовано изображение буквы, то всегда можно восстановить утраченную часть графика, продолжив оставшиеся линии.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НЕЙРОНА

На сегодняшний день существует множество моделей нейрона, которые различаются вычислительной сложностью и степе-

¹ Следует отметить, что данное требование обычно нарушается на этапе обучения.

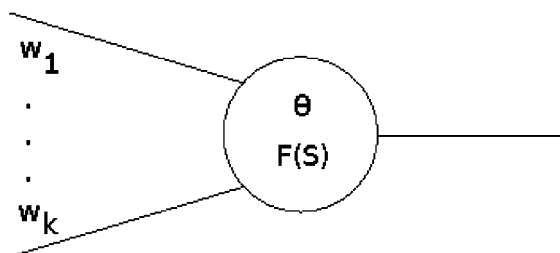


Рис. 1. Формальный нейрон

нию сходства с живым нейроном. Рассмотрим здесь классическую модель, называемую «формальный нейрон» (рис. 1).

У нейрона есть несколько входных каналов¹ и только один выходной канал. По входным каналам на нейрон поступают данные задачи, а на выходе формируется результат работы [2]

Нейрон вычисляет взвешенную сумму входных сигналов w_1, \dots, w_k , а затем преобразует полученную сумму с помощью заданной нелинейной функции $F(S)$.

Введем следующие обозначения:

x_i – значение входного сигнала,

θ – пороговый уровень нейрона,

w_i – весовой коэффициент нейрона (данное значение часто называют весом, силой связи или весом соединения),

F – функция активации, которая осуществляет преобразование,

y – выходное значение нейрона.²

Множество, состоящее из порогового уровня и всех весов, называют параметрами нейрона. Аналогично, параметрами сети называют множество параметров всех составляющих ее нейронов.

В этих обозначениях выход нейрона определяется следующей формулой:

$$y = F\left(\sum (x_i \cdot w_i) - \theta\right).$$

Существует модификация формального нейрона без порогового уровня. В этом случае к нейрону добавляется еще один входной канал (пусть его номер равен k), причем $x_k = 1$ для любого входного сигнала, а

$w_k = -\theta$. Очевидно, что эти модели эквивалентны:

$$\sum_{i \neq k} x_i w_i + x_k w_k = \sum x_i w_i - \theta.$$

Недостатки модели формального нейрона

Предполагается, что нейрон мгновенно вычисляет свой выход, поэтому с помощью таких нейронов нельзя моделировать непосредственно системы с внутренним состоянием.

- Формальные нейроны, в отличие от биологических, не могут обрабатывать информацию синхронно.

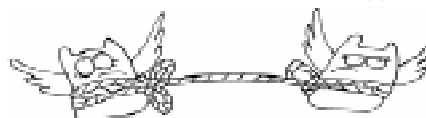
- Нет четких алгоритмов выбора функции активации.

- Невозможно регулировать работу всей сети.

- Излишняя формализация понятий «порог» и «весовые коэффициенты». У реальных нейронов порог меняется динамически, в зависимости от активности нейрона и общего состояния сети, а весовые коэффициенты изменяются в зависимости от проходящих сигналов.

Однослойный перцептрон

Один нейрон может выполнять простейшие вычисления, но основные функции нейросети обеспечиваются не отдельными



нейронами, а соединениями между ними. Однослойный перцептрон³ представляет собой простейшую сеть, которая состоит из группы нейронов, образующих слой, как показано на рис. 2. Входные данные кодируются вектором значений $X = (x_1, \dots, x_k)$, каждый элемент x_i подается на соответствующий вход каждого нейрона в слое. В свою

¹ Их часто называют синапсами по аналогии с биологическими нейронами.

² Или просто «выход нейрона».

³ От англ. «perceptron». Помимо термина «перцептрон» в русскоязычной литературе часто используется термин «персептрон».

очередь, нейроны вычисляют выход независимо друг от друга. Очевидно, что размерность выхода (то есть количество элементов) равна количеству нейронов, а количество синапсов у всех нейронов должно быть одинаково и совпадать с размерностью входного сигнала.

Несмотря на кажущуюся простоту, однослойный перцептрон может выполнять ряд полезных задач, например, классифицировать образы или вычислять значения логических функций [5].

МНОГОСЛОЙНЫЙ ПЕРЦЕПТРОН

Многослойный перцептрон способен рассчитать выходное значение Y для входного значения X . Другими словами, сеть вычисляет значение некоторой векторной функции: $Y = F(X)$. Таким образом, условие задачи, которая ставится перцептрон, должно быть сформулировано в виде множества векторов $\{x^1, \dots, x^s\}$. Решение задачи будет представлено в виде векторов $\{y^1, \dots, y^s\}$, причем для $\forall s \ y^s = F(x^s)$.

Все, что способен сделать перцептрон, – сформировать отображение $F: X \rightarrow Y$ для $\forall x \in X$. Мы не можем «извлечь» из перцептрона данное отображение полностью, а можем только посчитать образы произвольного числа точек.

Задача формализации, то есть выбора смысла, которым наделяются компоненты входного и выходного векторов, решается человеком на основе практического опыта. К сожалению, жестких рецептов формализации для нейронных сетей пока не создано.

Чтобы построить многослойный перцептрон, необходимо выбрать его параметры по следующему алгоритму [2]:

- Определить, какой смысл вкладывается в компоненты входного вектора X . Входной вектор должен содержать формализованное условие задачи, то есть всю информацию, необходимую для того, чтобы получить ответ.
- Выбрать выходной вектор Y таким образом, чтобы его компоненты содержали полный ответ для поставленной задачи.

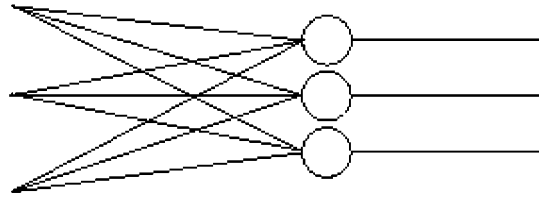


Рис. 2. Однослойный перцептрон

- Выбрать вид функции активации нейронов. При этом желательно учесть специфику задачи, так как удачный выбор увеличит скорость обучения.
- Выбрать количество слоев и нейронов в слое.
- Задать диапазон изменения входов, выходов, весов и пороговых уровней на основе выбранной функции активации.
- Присвоить начальные значения весам и пороговым уровням. Начальные значения не должны быть большими, чтобы нейроны не оказались в насыщении (на горизонтальном участке функции активации), иначе обучение будет очень медленным. Начальные значения не должны быть и слишком малыми, чтобы выходы большей части нейронов не были равны нулю, иначе обучение тоже замедлится.
- Провести обучение, то есть подобрать параметры сети так, чтобы задача решалась наилучшим образом. По окончании обучения сеть сможет решать задачи того типа, которым она обучена.
- Подать на вход сети условия задачи в виде вектора X . Рассчитать выходной вектор Y , который и даст формализованное решение задачи.

ОБЩИЙ ОБЗОР МЕТОДОВ ОБУЧЕНИЯ

Способность к обучению является фундаментальным свойством мозга. В контексте искусственных нейронных сетей процесс обучения может рассматриваться как настройка архитектуры сети, а также весов связей для эффективного выполнения поставленной задачи. Обычно нейронная сеть должна настроить веса по предоставленным обучающим примерам. Свойство сети обучаться на примерах делает их более привлекательными по сравнению с системами,

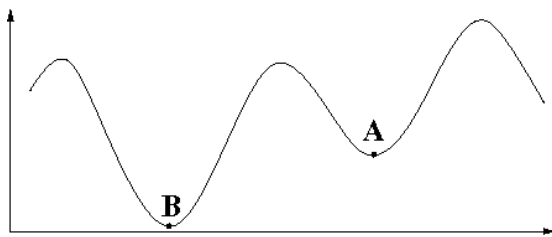


Рис. 3. Проблема локальных минимумов

которые работают по заранее заложенным правилам.

Среди всех существующих методов обучения можно выделить два класса: детерминированный и стохастический.

Детерминированный метод итеративно корректирует параметры сети, основываясь на ее текущих параметрах, величинах входов, фактических и желаемых выходов. Яркой иллюстрацией подобного метода является метод обратного распространения ошибки.

Стохастические методы обучения изменяют параметры сети случайным образом. При этом сохраняются только те изменения, которые привели к улучшениям. В качестве примера стохастического метода обучения можно привести следующий алгоритм:

1. Выбрать параметры сети случайным образом и подкорректировать их на небольшую случайную величину. Предъявить множество входов и вычислить получающиеся выходы.

2. Сравнить эти выходы с желаемыми и вычислить разницу между ними. Эта разница называется ошибкой. Цель обучения состоит в том, чтобы минимизировать ошибку.

3. Если ошибка уменьшилась, коррекция сохраняется, в противном случае коррекция отбрасывается и выбирается новая.

Шаги 2 и 3 повторяются до тех пор, пока сеть не обучится.

Необходимо отметить, что стохастический метод обучения может попасть в «ловушку» локального минимума (рис. 3).

Допустим, что первоначальное значение ошибки равно или очень близко к точке А.

Если случайные шаги корректировки малы, то любые отклонения от точки А увеличат ошибку и будут отвергнуты. Таким образом, наименьшее значение ошибки в точке В никогда не будет найдено. Когда случайные коррекции параметров сети очень велики, ошибка будет меняться так резко, что никогда не установится в одном из минимумов.

Чтобы избежать подобных проблем, можно постепенно уменьшать средний размер случайных шагов коррекции. Когда средний размер шагов велик, значение ошибки будет принимать все значения с равной вероятностью. Если плавно уменьшать размер шагов, то будет достигнуто условие, при котором значение ошибки будет на некоторое время «застревать» в точке В. Когда размер шага уменьшится еще больше, значение ошибки будет «останавливаться» на короткое время и в точке А, и в точке В. Если уменьшать размер шага непрерывно, то, в конце концов, будет достигнута величина шага, достаточная для того, чтобы преодолеть локальный минимум А, но не локальный минимум В.

Обучение с учителем

Алгоритм называется алгоритмом обучения с учителем, если во время обучения сеть располагает правильными ответами (выходами сети) на каждый входной пример, то есть заранее задается множество пар векторов $\{(x^s, d^s)\}^1$, где $x^s \in X$ – вектор, задающий условие задачи, а $d^s \in Y$ – известное решение задачи для вектора x^s . В процессе обучения сеть меняет свои параметры таким образом, чтобы давать нужное отображение $X \rightarrow Y$. Необходимо отметить, что размер множества $\{(x^s, d^s)\}$ должен быть достаточным для того, чтобы алгоритм обучения смог сформировать нужное отображение [16].

Обучение без учителя

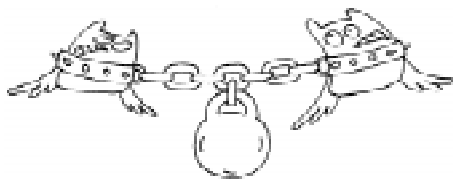
Хотя метод обучения с учителем успешно применяется для решения прикладных задач, многие исследователи критикуют его

¹ Множество, на котором обучается сеть, называется обучающим множеством или обучающей выборкой.

за биологическое неправдоподобие. Действительно, трудно вообразить, что в мозге есть некоторый механизм, который сравнивает действительные результаты с желаемыми. Алгоритм обучения без учителя может применяться тогда, когда известны только входные сигналы. На их основе сеть учится давать наилучшие значения выходов. Понятие «наилучшее значение» определяется алгоритмом обучения. Обычно алгоритм подстраивает параметры так, чтобы сеть выдавала одинаковые результаты для достаточно близких входных значений [2].

Метод Хэбба

Самым старым обучающим правилом (1949-й год) является постулат обучения Хэбба. На основании физиологических и психологических исследований Хэбб выдвинул гипотезу о том, как обучаются биологические нейроны. Он предположил, что вес соединения между двумя нейронами усиливается, если оба эти нейрона возбуждены.



Хэбб опирался на следующие нейрофизиологические наблюдения: если связанные между собой нейроны активизируются одновременно и регулярно, то сила связи возрастает. Важной особенностью этого правила является то, что изменение веса связи зависит только от активности нейронов, которые соединены данной связью.

Сам алгоритм выглядит следующим образом:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.
2. На вход сети подается входной сигнал и вычисляется выход.
3. На основании полученных выходных значений нейронов производится изменение весовых коэффициентов.
4. Повтор с шага 2 с новым образом из входного множества до тех пор, пока вы-

ходные значения сети не стабилизируются с заданной точностью.

Правило коррекции по ошибке

В 1957 г. Розенблатт разработал модель, которая вызвала большой интерес у исследователей. Модель использует алгоритм обучения с учителем, то есть обучающее множество состоит из множества входных векторов, для каждого из которых указан выходной вектор. Несмотря на некоторые ограничения, она стала основой для многих современных наиболее сложных алгоритмов обучения с учителем.

Суть алгоритма состоит в следующем: для каждого входного примера задается желаемый выход. Если реальный выход сети не совпадает с желаемым, то параметры сети будут скорректированы. Для вычисления величины коррекции используется разница между реальным и желаемым выходом сети, причем коррекция весов будет происходить только в случае ошибочного ответа [2].

Обучение методом соревнования

В отличие от обучения Хэбба, в котором множество выходных нейронов могут возбуждаться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. То



есть из всего множества выходных нейронов используется только один нейрон с самым большим выходом. Такой алгоритм напоминает процесс обучения биологических нейронных сетей. Обучение методом соревнования позволяет классифицировать входные данные: похожие примеры группируются сетью в один класс и представляются одним образцовым элементом. При этом каждый нейрон из множества выходных нейронов «отвечает» только за один класс. Очевидно, что общее число классов, с которыми способна работать сеть равно количеству выходных нейронов.

При обучении модифицируются только веса «победившего» нейрона. Это приводит к тому, что образцовый элемент становится чуть ближе к входному примеру.

Генетические алгоритмы

Генетические алгоритмы – группа алгоритмов, основанных на моделировании развития биологической популяции.

Назовем популяцией набор векторов $P = \{p^i\} = \{p^1, \dots, p^N\}$, где N – размер популяции. Элементы p^i – особи. Каждый вектор p^i содержит все параметры, с помощью которых можно описать каждую особь.

Пусть есть функция $E(p)$, зависящая от вектора p , с помощью которой вычисляется ошибка. Требуется найти минимум E .

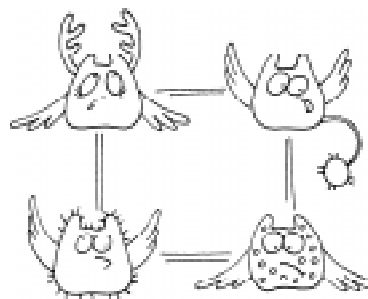
Элементы множества P способны эволюционировать по следующим правилам:

Если $E(p^0)$ – мало, то особь p^0 считается удачной и получает приоритет при размножении. Вероятность гибели этой особи уменьшается.

Если $E(p^0)$ – велико, то особь p^0 считается неудачной, вероятность размножения для этой особи снижается и повышается вероятность гибели.

Мутации: любая точка (особь) может мутировать, то есть ее значение может сместиться на небольшую величину $p^0: = p^0 + \Delta p$, где Δp – небольшой по модулю вектор, характеризующий величину мутации.

Размножение: в соответствии с вероятностью размножения (см. шаг 1), каждая точка делится. Законы размножения зависят от выбранной модели.



Гибель: в соответствии с вероятностью, определенной на шаге 2, точка может «погибнуть», то есть она может быть удалена из множества P .

Точная теория эволюции пока не построена, поэтому нет четких правил выбора численных значений вероятностей гибели, размножения. Оптимальность выбранных алгоритмов пока может быть оценена только экспериментально.

ЗАКЛЮЧЕНИЕ

Развитие теории искусственных нейронных сетей всегда вызывало немало энтузиазма и критики. Она развивалась быстрее, чем предсказывали пессимисты, но медленнее, чем надеялись оптимисты, – типичная ситуация. Тем не менее, нейронные сети уже сейчас с успехом используются в системах управления, распознавания образов, предсказания и диагностики, то есть там, где традиционные вычисления слишком трудны. К сожалению, для подобных задач пока не найдены оптимальные способы решения. Сравнительные исследования различных подходов (в том числе и без использования нейросетей) так и не привели к однозначным выводам. Очевидно, что в такой ситуации необходимо понять возможности, предпосылки и область применения всех существующих подходов и максимально использовать их преимущества для дальнейшего развития интеллектуальных систем. Подобные усилия могут привести к абсолютно новым алгоритмам, которые объединят искусственные нейросети с другими технологиями.

Литература

1. *Wasserman P.* Neural Computing – Theory and Practice. Van Nostrand Reinhold, 1989.
2. *Smith L.* An Introduction to Neural Networks. Unpublished draft, University of Stirling, 2001. <http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html>.
3. *Nilsson N.* Introduction to Machine Learning. Unpublished draft, Stanford University, 1996. P. 39–68. <http://robotics.stanford.edu/people/nilsson/mlbook.html>.
4. *Заенцев И.* Нейронные сети: основные модели. Учебное пособие. Воронежский государственный университет, 1999.

5. *Короткий С.* Нейронные сети: основные положения. <http://www.gotai.net/documents/doc-nn-002.aspx>.
6. *Короткий С.* Нейронные сети: алгоритм обратного распространения. <http://www.gotai.net/documents/doc-nn-003.aspx>.
7. *Короткий С.* Нейронные сети: обучение без учителя. <http://www.gotai.net/documents/doc-nn-004.aspx>.
8. *Короткий С.* Нейронные сети Хопфилда и Хэмминга. <http://www.gotai.net/documents/doc-nn-005.aspx>.
9. *Brown E.W.* Applying Neural Networks to Character Recognition. Northeastern University internal paper.
10. *Bishop C. M.* Neural Networks for Pattern Recognition. Oxford University Press Inc., 2003.
11. *Pinkus A.* Approximation theory of the MLP model in neural networks. Acta Numerica, 1999.
12. *Haykin S.* Neural Networks: A Comprehensive Foundation. NY: Macmillan, 1994.
13. *Nigrin A.* Neural Networks for Pattern Recognition. Cambridge, MA: The MIT Press, 1993.
14. *Zurada J.M.* Introduction To Artificial Neural Systems. Boston: PWS Publishing Company, 1992.
15. DARPA Neural Network Study, AFCEA International Press, 1988.
16. *Kröse B., Smagt P.* An introduction to Neural Networks. University of Amsterdam, 1996.
17. *McCulloch W., Pitts W.* A Logical Calculus of the Ideas Imminent in Nervous Activity. Bulletin of Mathematical Biophysics 5: 115–33, 1943.
18. *Rosenblatt F.* The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 1958.
19. *Minsky M., Papert S.* Perceptrons. Cambridge, VA: MIT Press, 1969.
20. *Widrow B., Hoff M.* Adaptive Switching Circuits. IRE Wescon Convention Record, W.Y., 1960.
21. *Галушкин А.И.* Нейрокомпьютеры. Кн. 3. М.: ИПРЖ, 2000.
22. *Логовский А.* Новейшая история нейрокомпьютинга в России. Открытые системы № 3, 2001. http://www.osp.ru/text/302/179978/_p1.html.

*Васенков Данила Валентинович,
аспирант математико-
механического факультета СПбГУ.*



Наши авторы, 2007
Our authors, 2007