

## Содержание:

image not found or type unknown



## Введение

Нейронные сети представляют собой новую и весьма перспективную вычислительную технологию, дающую новые подходы к исследованию динамических задач в финансовой области. Первоначально нейронные сети открыли новые возможности в области распознавания образов, затем к этому прибавились статистические и основанные на методах искусственного интеллекта средства поддержки принятия решений и решения задач в сфере финансов.

Способность к моделированию нелинейных процессов, работе с зашумленными данными и адаптивность дают возможности применять нейронные сети для решения широкого класса финансовых задач. В последние несколько лет на основе нейронных сетей было разработано много программных систем для применения в таких вопросах, как операции на товарном рынке, оценка вероятности банкротства банка, оценка кредитоспособности, контроль за инвестициями, размещение займов. Приложения нейронных сетей охватывают самые разнообразные области интересов: распознавание образов, обработка зашумленных данных, дополнение образов, ассоциативный поиск, классификация, оптимизация, прогноз, диагностика, обработка сигналов, абстрагирование, управление процессами, сегментация данных, сжатие информации, сложные отображения, моделирование сложных процессов, машинное зрение, распознавание речи.

## 1. История развития нейронных сетей

На заре развития электронно-вычислительной техники в середине XX века среди ученых и конструкторов еще не существовало единого мнения о том, как должна быть реализована и по какому принципу работать типовая электронно-вычислительная машина. Это сейчас мы с вами изучаем в курсах основ информатики архитектуру машины фон Неймана, по которой построены практически все существующие сегодня компьютеры. При этом в тех же учебниках

ни слова не говорится о том, что в те же годы были предложены принципиально иные архитектуры и принципы действия компьютеров. Одна из таких схем получила название нейросетевого компьютера, или просто нейросети. Рис. 1 Главные части нервной клетки - это ее тело, содержащее ядро и другие органеллы, единственный аксон, передающий импульсы от клетки, и дендриты, к которым приходят импульсы от других клеток. Первый интерес к нейросетям был обусловлен пионерской работой МакКаллока и Питса, изданной в 1943 году, где предлагалась схема компьютера, основанного на аналогии с работой человеческого мозга. Они создали упрощенную модель нервной клетки - нейрон. Мозг человека состоит из белого и серого веществ: белое - это тела нейронов, а серое - это соединительная ткань между нейронами, или аксоны и дендриты. Мозг состоит примерно из  $10^{11}$  нейронов,

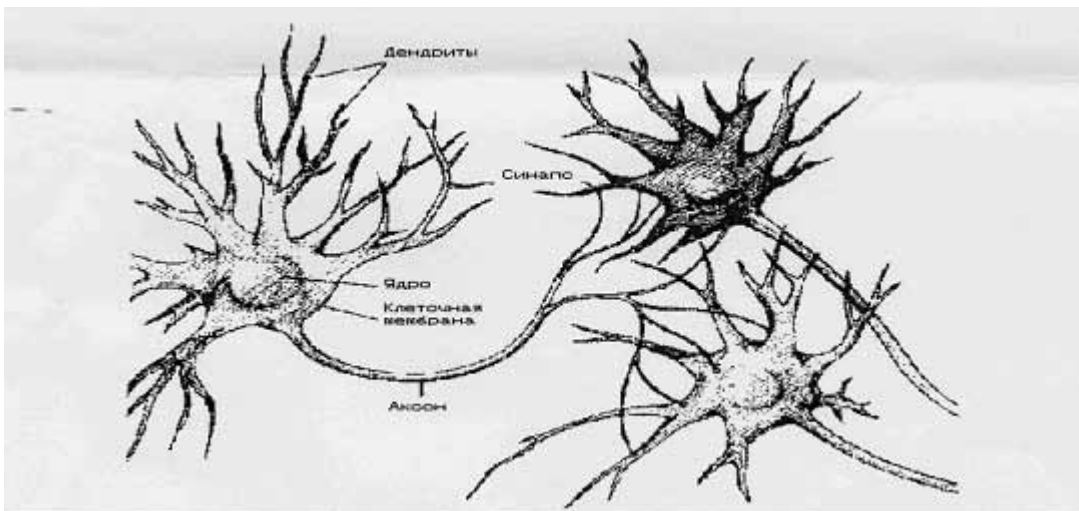


Рис. 1

Главными частями нервной клетки - это ее тело, содержащее ядро и другие органеллы, единственный аксон, передающий импульсы от клетки, и дендриты, к которым приходят импульсы от других клеток связанных между собой. Каждый нейрон получает информацию через свои дендриты, а передает ее дальше только через единственный аксон, разветвляющийся на конце на тысячи синапсов (см. рис. 1). Простейший нейрон может иметь до 10000 дендритов, принимающих сигналы от других клеток.

Таким образом, мозг содержит примерно  $10^{15}$  взаимосвязей. Если учесть, что любой нейрофизиологический процесс активизирует сразу множество нейронов, то можно представить себе то количество информации или сигналов, которое возникает в мозгу. Нейроны взаимодействуют посредством серий импульсов,

длящихся несколько миллисекунд, каждый импульс представляет собой частотный сигнал с частотой от нескольких единиц до сотен герц. Это невообразимо медленно по сравнению с современными компьютерами, но в тоже время человеческий мозг гораздо быстрее машины может обрабатывать аналоговую информацию, как-то: узнавать изображения, чувствовать вкус, узнавать звуки, читать чужой почерк, оперировать качественными параметрами. Все это реализуется посредством сети нейронов, соединенных между собой синапсами. Другими словами, мозг — это система из параллельных процессоров, работающая гораздо эффективнее, чем популярные сейчас последовательные вычисления.

Технология последовательных вычислений подошла к пределу своих технических возможностей, и в настоящее время остро стоит проблема развития методов параллельного программирования и создания параллельных компьютеров. Так что, может быть, нейросети являются только очередным шагом в этом направлении.

## 2. Устройство нейронных сетей

Искусственным нейроном называется простой элемент, сначала вычисляющий взвешенную сумму  $V$  входных величин  $x_i$ :

$$V = \sum_{i=1}^N W_i * x_i = \bar{W} * \bar{X}$$

Здесь  $N$  - размерность пространства входных сигналов. Затем полученная сумма сравнивается с пороговой величиной  $W_0$ , вслед за чем вступает в действие нелинейная функция активации  $f$ . Коэффициенты  $\{W_i\}$  во взвешенной сумме обычно называют синаптическими коэффициентами или весами. Саму же взвешенную сумму  $V$  мы будем называть потенциалом нейрона  $i$ . Выходной сигнал тогда имеет вид  $f(V)$ . Величину порогового барьера можно рассматривать как еще один весовой коэффициент при постоянном входном сигнале. В этом случае мы говорим о расширенном входном пространстве: нейрон с  $N$ -мерным входом имеет  $N+1$  весовой коэффициент. Если ввести в уравнение пороговую величину  $W_0$ , то оно переписется так:

$$V = \sum_{i=1}^N W * x + W_0$$

В зависимости от способа преобразования сигнала и характера активации возникают различные виды нейронных структур. Существуют детерминированные нейроны, когда активизирующая функция однозначно вычисляет выход по входу, и вероятностные нейроны, состояние которых в момент  $t$  есть случайная функция потенциала и состояния в момент  $t-1$ . Далее речь пойдет о детерминированных нейронах

### 3. Функции активации

В искусственных нейронах могут быть различные функции активации, но и в используемых программах, и в известной литературе указаны только следующие виды функций:

- \* Линейная: выходной сигнал нейрона равен его потенциалу,
- \* Пороговая: нейрон выбирает решение из двух вариантов: активен /неактивен,
- \* Многопороговая: выходной сигнал может принимать одно из  $q$  значений, определяемых  $(q-1)$  порогом внутри предельных значений.
- \* Сигмоидная: рассматриваются два вида сигмоидных функций:

$$s = f(V) = \frac{1}{1 + \exp(-bV)}$$

с выходными значениями в промежутке  $[0,1]$  и

$$s = f(V) = \frac{\exp(bV) - 1}{\exp(bV) + 1}$$

с выходными значениями в промежутке  $[-1,1]$ . Коэффициент  $b$  определяет крутизну сигмоида. Поскольку сигмоидная функция является гладким отображением  $(-\infty, \infty)$  на  $(-1,1)$ , то крутизну можно учесть через величины весов и порогов, и без ограничения общности можно полагать ее равной единице.

### 4. Типы архитектур нейросетей

Из точек на плоскости и соединений между ними можно построить множество графических фигур, называемых графами. Если каждую точку представить себе

как один нейрон, а соединения между точками - как дендриты и синапсы, то мы получим нейронную сеть. Но не всякое соединение нейронов будет работоспособно или вообще целесообразно. Поэтому на сегодняшний день существует только несколько работающих и реализованных программно архитектур нейросетей. Я только вкратце опишу их устройство и классы решаемых ими задач. Сеть прямого распространения По архитектуре связей нейросети могут быть сгруппированы в два класса: сети прямого распространения, в которых связи не имеют петель, и сети рекуррентного типа, в которых возможны обратные связи (см. рис. 2)

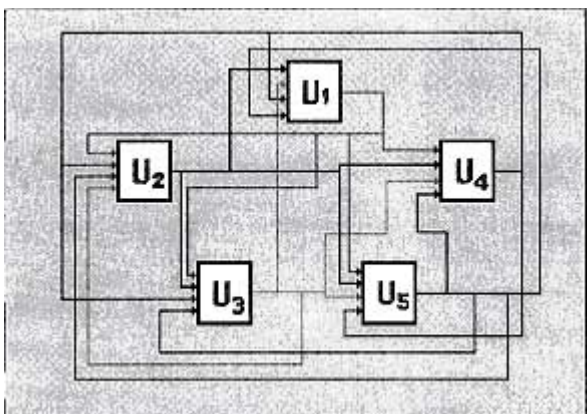
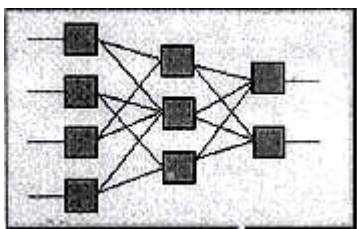


Рис. 2 Рекуррентная сеть

Сети прямого распространения подразделяются на однослойные перцептроны (сети) и многослойные перцептроны (сети). Название перцептрона для нейросетей придумал американский нейрофизиолог Ф. Розенблатт, придумавший в 1957 году первый нейропроцессорный элемент (НПЭ), то есть нейросеть. Он же доказал сходимости области решений для перцептрона при его обучении. Сразу после этого началось бурное исследование в этой области и был создан самый первый нейрокомпьютер Mark I. Многослойные сети отличаются тем, что между входными и выходными данными располагаются несколько так называемых скрытых слоев нейронов, добавляющих больше нелинейных связей в модель. Рассмотрим устройство простейшей многослойной нейросети.

Любая нейронная сеть состоит из входного слоя и выходного слоя. Соответственно подаются независимые и зависимые переменные. Входные данные преобразуются

нейронами сети и сравниваются с выходом. Если отклонение больше заданного, то специальным образом изменяются веса связей нейронов между собой и пороговые значения, нейронов. Снова происходит процесс вычисления выходного значения и его сравнение с эталоном. Если отклонения меньше заданной погрешности, то процесс обучения прекращается. Помимо входного и выходного слоев в многослойной сети существуют так называемые скрытые слои. Они представляют собой нейроны, которые не имеют непосредственных входов исходных данных, а связаны только с выходами входного слоя и с входом выходного слоя. Таким образом, скрытые слои дополнительно преобразуют информацию и добавляют нелинейности в модели.

Если однослойная нейросеть очень хорошо справляется с задачами классификации, так как выходной слой нейронов сравнивает полученные от предыдущего слоя значения с порогом и выдает значение либо ноль, то есть меньше порогового значения, либо единицу - больше порогового (для случая пороговой внутренней функции нейрона), и не способен решать большинство практических задач (что было доказано Минским и Пейпертом), то многослойный перцептрон с сигмоидными решающими функциями способен аппроксимировать любую функциональную зависимость (это было доказано в виде теоремы). Но при этом не известно ни нужное число слоев, ни нужное количество скрытых нейронов, ни необходимое для обучения сети время. Эти проблемы до сих пор стоят перед исследователями и разработчиками нейросетей. Лично мне кажется, что весь энтузиазм в применении нейросетей строится именно на доказательстве этой теоремы. Рассмотрим, как нейроны могут моделировать различные классы функций.

Класс рекуррентных нейросетей гораздо обширнее, да и сами сети сложнее по своему устройству. Поведение рекуррентных сетей описывается дифференциальными или разностными уравнениями, как правило, первого порядка. Это гораздо расширяет области применения нейросетей и способы их обучения. Сеть организована так, что каждый нейрон получает входную информацию от других нейронов, возможно, и от самого себя, и от окружающей среды. Этот тип сетей имеет важное значение, так как с их помощью можно моделировать нелинейные динамические системы. Среди рекуррентных сетей можно выделить сети Хопфилда и сети Кохонена. С помощью сетей Хопфилда можно обрабатывать неупорядоченные (рукописные буквы), упорядоченные во времени (временные ряды) или пространстве (графики) образцы. Рекуррентная нейросеть простейшего вида была введена Хопфилдом и построена она из  $N$

нейронов, связанных каждый с каждым кроме самого себя, причем все нейроны являются выходными. Нейросеть Хопфилда можно использовать в качестве ассоциативной памяти.

Сеть Кохонена еще называют "самоорганизующейся картой признаков". Сеть такого типа рассчитана на самостоятельное обучение. Во время обучения сообщать ей правильные ответы необязательно. В процессе обучения на вход сети подаются различные образцы. Сеть улавливает особенности их структуры и разделяет образцы на кластеры, а уже обученная сеть относит каждый вновь поступающий пример к одному из кластеров, руководствуясь некоторым критерием "близости". Сеть состоит из одного входного и одного выходного слоя. Количество элементов в выходном слое непосредственно определяет, сколько различных кластеров сеть сможет распознать. Каждый из выходных элементов получает на вход весь входной вектор. Как и во всякой нейронной сети, каждой связи приписан некоторый синоптический вес. В большинстве случаев каждый выходной элемент соединен также со своими соседями. Эти внутрисловные связи играют важную роль в процессе обучения, так как корректировка весов происходит только в окрестности того элемента, который наилучшим образом откликается на очередной вход. Выходные элементы соревнуются между собой за право вступить в действие и "получить урок". Выигрывает тот из них, чей вектор весов окажется ближе всех к входному вектору.

## **5. Обучение многослойной сети**

Главное отличие и преимущество нейросетей перед классическими средствами прогнозирования и классификации заключается в их способности к обучению. Так что же такое обучение нейросетей? На этапе обучения происходит вычисление синоптических коэффициентов в процессе решения нейронной сетью задач, в которых нужный ответ определяется не по правилам, а с помощью примеров, сгруппированных в обучающие множества. Так что нейросеть на этапе обучения сама играет роль эксперта в процессе подготовки данных для построения экспертной системы. Предполагается, что правила находятся в структуре обучающих данных. Для обучения нейронной сети требуются обучающие данные. Они должны отвечать свойствам представительности и случайности или последовательности. Все зависит от класса решаемой задачи. Такие данные представляют собой ряды примеров с указанием для каждого из них значением выходного параметра, которое было бы желательно получить. Действия, которые

при этом происходят, можно назвать контролируемым обучением: "учитель" подает на вход сети вектор исходных данных, а на выходной узел сообщает желаемое значение результата вычислений.

Контролируемое обучение нейросети можно рассматривать как решение оптимизационной задачи. Ее целью является минимизация функции ошибок  $E$  на данном множестве примеров путем выбора значений весов  $W$ . Достижение минимума называется сходимостью процесса обучения. Именно возможность этого и доказал Розенблатт. Поскольку ошибка зависит от весов нелинейно, получить решение в аналитической форме невозможно, и поиск глобального минимума осуществляется посредством итерационного процесса так называемого обучающего алгоритма. Разработано уже более сотни разных обучающих алгоритмов, отличающихся друг от друга стратегией оптимизации и критерием ошибок. Обычно в качестве меры погрешности берется средняя квадратичная ошибка (СКО):

$$E = \sqrt{\frac{\sum_{i=1}^M (d_i - y_i)^2}{M}}$$

где  $M$  - число примеров в обучающем множестве. Минимизация величины  $E$  осуществляется с помощью градиентных методов. Изменение весов происходит в направлении, обратном к направлению наибольшей крутизны для функции:

$$W(t+1) = W(t) + \varepsilon * \frac{\partial E}{\partial W}$$

Здесь  $\varepsilon$  - определяемый пользователем параметр, который называется коэффициентом обучения.

## 6. Обратное распространение ошибки

Одним из самых распространенных алгоритмов обучения нейросетей прямого распространения является алгоритм обратного распространения ошибки (Back Propagation, BP). Этот алгоритм был переоткрыт и популяризован в 1986 г. Румельхартом и МакКлелландом из группы по изучению параллельных распределенных процессов в Массачусетском технологическом институте. Здесь я хочу подробно изложить математическую суть алгоритма, так как очень часто в



литературе ссылаются на какой-то факт или теорему, но никто не приводит его доказательства или источника. Честно говоря, то же самое относится к теореме об отображении нейросетью любой функциональной зависимости, на которой основываются все попытки применить нейросети к моделированию реальных процессов. Приведём алгоритм работы нейросети. Итак, это алгоритм градиентного спуска, минимизирующий суммарную квадратичную ошибку:

$$E = \frac{1}{2} \sum_{k=1}^P \sum_i (d_k^i - y_k^i)^2$$

Здесь индекс  $i$  пробегает все выходы многослойной сети. Основная идея BP состоит в том, чтобы вычислять чувствительность ошибки сети к изменениям весов. Для этого нужно вычислить частные производные от ошибки по весам. Пусть обучающее множество состоит из  $P$  образцов, входы которого где  $\eta$  - длина шага в направлении, обратном к градиенту обозначены через  $\{x_{ik}\}$ . Вычисление частных производных осуществляется по правилу цепи: вес входа  $i$ -го нейрона, идущего от  $j$ -го нейрона, пересчитывается по формуле:

$$\Delta W_{ij} = -\eta * \sum_{k=1}^P \frac{\partial E_k}{\partial W_{ij}} = -\eta * \sum_{k=1}^P \frac{\partial E_k}{\partial V_k^i} * \frac{\partial V_k^i}{\partial W_{ij}} = -\eta * \sum_{k=1}^P \delta_k^i * x_{kj}^i$$

где  $\eta$  - длина шага в направлении, обратном к градиенту

Если рассмотреть отдельно  $k$ -тый образец, то соответствующее изменение весов равно:

$$\Delta W_{ij} = -\eta * \frac{\partial E_k}{\partial W_{ij}} = -\eta * \delta_k^i * x_{kj}^i$$

Множитель вычисляется через аналогичные множители из последующего слоя, и ошибка, таким образом, передается в обратном направлении. Для выходных элементов получим:

$$\delta_k^i = -\frac{\partial E_k}{\partial x_k^i} * \frac{\partial x_k^i}{\partial v_k^i} = -(d_k^i - x_k^i) f'(v_k^i)$$

Для скрытых элементов множитель определяется так:

$$\delta_k^i = -\frac{\partial E_k}{\partial v_k^i} = -\sum_h \frac{\partial E_k}{\partial v_k^h} \frac{\partial v_k^h}{\partial v_k^i}$$

где индекс  $h$  пробегает номера всех нейронов, на которые воздействует  $i$ -ый нейрон.

## 7. Организация процесса обучения

Из теоремы об отображении практически любой функции с помощью многослойной нейросети следует, что обучаемая нами нейронная сеть в принципе способна сама подстроиться под любые данные с целью минимизации суммарной квадратичной ошибки. Чтобы этого не происходило, при обучении нейросетей используют следующий способ проверки сети. Для этого обучающую выборку еще перед началом обучения разбивают случайным образом на две подвыборки: обучающую и тестовую. Обучающую выборку используют собственно для процесса обучения, при этом изменяются веса нейронов. А тестовую используют в процессе обучения для проверки на ней суммарной квадратичной ошибки, но при этом не происходит изменение весов. Если нейросеть показывает улучшение аппроксимации и на обучающей, и на тестовой выборках, то обучение сети происходит в правильном направлении. Иначе может снижаться ошибка на обучающей выборке, но происходит ее увеличение на тестовой. Последнее означает, что сеть "переобучилась" и уже не может быть использована для прогнозирования или классификации. В этом случае немного изменяются веса нейронов, чтобы вывести сеть из окрестности локального минимума ошибки.