

Применение нейроструктур в средствах измерений

Нейронная сеть (НС) - это набор нейронов, определенным образом связанных между собой. Как правило, передаточные функции всех нейронов в сети фиксированы, а веса являются параметрами сети и могут изменяться. Существуют разные типы нейронных сетей:

- **полносвязные сети** - в которых каждый нейрон связан со всеми остальными (на входы каждого нейрона подаются выходные сигналы остальных нейронов);
- **сети с обратными связями (рекуррентные)** - в них определенным образом выходы с последующих слоев нейронов подаются на вход предыдущим.

Эти сети имеют существенные различия, но отдельные типы НС обладают несколькими общими чертами.

Основу каждой НС составляют относительно простые, в большинстве случаев - однотипные, элементы (ячейки), имитирующие работу нейронов мозга. Далее под нейроном будет подразумеваться искусственный нейрон, то есть ячейка НС. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов.

Нейрон осуществляет взвешенное суммирование поступающих на его вход сигналов. Результат такого суммирования образует промежуточный выходной сигнал, который преобразуется активационной функцией в выходной сигнал нейрона. По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона, имеющей большой коэффициент усиления для слабых сигналов и с падающим усилением для больших возбуждений. Коэффициент усиления вычисляется как отношение выходного сигнала нейрона к вызвавшему его небольшому приращению взвешенной суммы входных сигналов. Кроме этого для обеспечения увеличения вычислительной мощности многослойными НС, по сравнению с однослойными, необходимо чтобы активационная функция между слоями была нелинейной, т.е. как показано в учитывая ассоциативность операции умножения матриц любую многослойную нейросеть без нелинейных активационных функций можно свести к эквивалентной однослойной нейросети, которые весьма ограничены по своим вычислительным возможностям.

Самыми простыми, базовыми элементами нейросетей являются:

Адаптивный сумматор. Элемент, вычисляющий скалярное произведение вектора входного сигнала x на вектор параметров w ;

Нелинейный преобразователь сигнала f , преобразующий скалярный сигнал x в $f(x)$;

Формальный нейрон.

Одним из основных достоинств нейровычислителя является то, что его основу составляют относительно простые, чаще всего - однотипные, элементы (ячейки), имитирующие работу нейронов мозга - "нейроны". Они состоят из элементов трех типов. Элементы нейрона — умножители (синапсы), сумматор и нелинейный преобразователь.

Общий вид нейрона приведен на рисунке 1. Каждый синапс характеризуется величиной синаптической связи или ее весом w_i , который по физическому смыслу эквивалентен электрической проводимости.

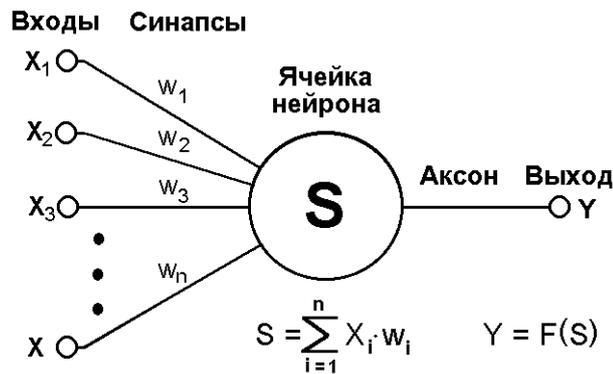


Рисунок 1. Искусственный нейрон.

Нейрон обладает группой синапсов - однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон - выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Каждый синапс характеризуется величиной синаптической связи или ее весом w_i , который по физическому смыслу эквивалентен электрической проводимости. Они осуществляют связь между нейронами, умножают входной сигнал на вес синапса.

Сумматор выполняет сложение сигналов, поступающих по синаптическим связям от других нейронов, и внешних входных сигналов.

Нелинейный преобразователь реализует нелинейную функцию одного аргумента - выхода сумматора. Эта функция называется "функция активации" или "передаточная функция" нейрона. Нейрон в целом реализует скалярную функцию векторного аргумента. В общем случае входной сигнал, весовые коэффициенты и значения смещения могут принимать действительные значения. Выход (y) определяется видом функции активации и может быть как действительным, так и целым. Во многих практических задачах входы, веса и смещения могут принимать лишь некоторые фиксированные значения.

Синаптические связи с положительными весами называют возбуждающими, с отрицательными весами - тормозящими.

Таким образом, нейрон полностью описывается своими весами w_i и передаточной функцией $f(x)$. Получив набор чисел (вектор) x_k в качестве входов, нейрон выдает некоторое число y на выходе.

Принцип параллельной обработки сигналов, достигается путем объединения большого числа нейронов в так называемые слои и соединения определенным образом нейронов различных слоев, а также, в некоторых конфигурациях, и нейронов одного слоя между собой, причем обработка взаимодействия всех нейронов ведется послойно.

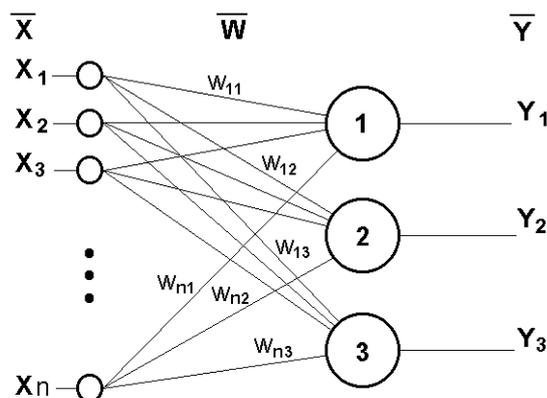


Рисунок 2. Трехнейронный перцептрон

В качестве примера простейшей НС рассмотрим трехнейронный перцептрон рисунок 2, то есть такую сеть, нейроны которой имеют активационную функцию в виде единичного скачка. На n входов поступают некие сигналы, проходящие по синапсам на 3 нейрона, образующие единственный слой этой НС и выдающие три выходных сигнала:

$$y_j = f \left[\sum_{i=1}^n x_i \cdot w_{ij} \right], \quad j=1...3$$

Очевидно, что все весовые коэффициенты синапсов одного слоя нейронов можно свести в матрицу \mathbf{W} , в которой каждый элемент w_{ij} задает величину i -ой синаптической связи j -ого нейрона.

$$u = \sum_{i=1}^n w_i x_i + w_0 x_0$$

Здесь x_i и w_i — соответственно сигналы на входах нейрона и веса входов, функция u называется индуцированным локальным полем, а $f(u)$ - передаточной функцией.

Также, процесс, происходящий в НС, может быть записан в матричной форме:

$$\mathbf{Y} = \mathbf{F}(\mathbf{XW})$$

где \mathbf{X} и \mathbf{Y} — соответственно входной и выходной сигнальные векторы, $\mathbf{F}(\mathbf{U})$ - активационная функция, применяемая поэлементно к компонентам вектора \mathbf{U} .

Возможные значения сигналов на входах нейрона считают заданными в интервале $[0,1]$. Они могут быть либо дискретными (0 или 1), либо аналоговыми. Дополнительный вход x_0 и соответствующий ему вес w_0 используется для инициализации нейрона. Под инициализацией подразумевается смещение активационной функции нейрона по горизонтальной оси, то есть формирование порога чувствительности нейрона. Кроме того, иногда к выходу нейрона специально добавляют некую случайную величину, называемую сдвигом. Сдвиг можно рассматривать как сигнал на дополнительном, всегда нагруженном, синапсе.

Передаточная функция $f(u)$ определяет зависимость сигнала на выходе нейрона от взвешенной суммы сигналов на его входах. В большинстве случаев она является монотонно возрастающей и имеет область значений $[-1,1]$ или $[0,1]$, однако существуют исключения. Также для некоторых алгоритмов обучения сети необходимо, чтобы она была непрерывно дифференцируемой на всей числовой оси. Искусственный нейрон полностью характеризуется своей передаточной функцией. Использование различных передаточных функций позволяет вносить нелинейность в работу нейрона и в целом нейронной сети.

В основном, нейроны классифицируют на основе их положения в топологии сети и разделяют на:

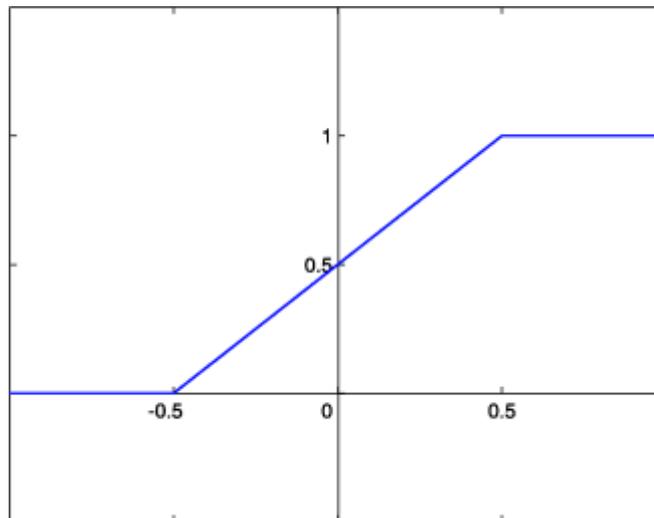
Входные нейроны - принимают исходный вектор, кодирующий входной сигнал. Как правило, эти нейроны не выполняют вычислительных операций, а просто передают полученный входной сигнал на выход, возможно, усилив или ослабив его;

Выходные нейроны - представляют из себя выходы сети. В выходных нейронах могут производиться какие-либо вычислительные операции;

Промежуточные нейроны - выполняют основные вычислительные операции.

Основные типы передаточных функций

Линейная передаточная функция

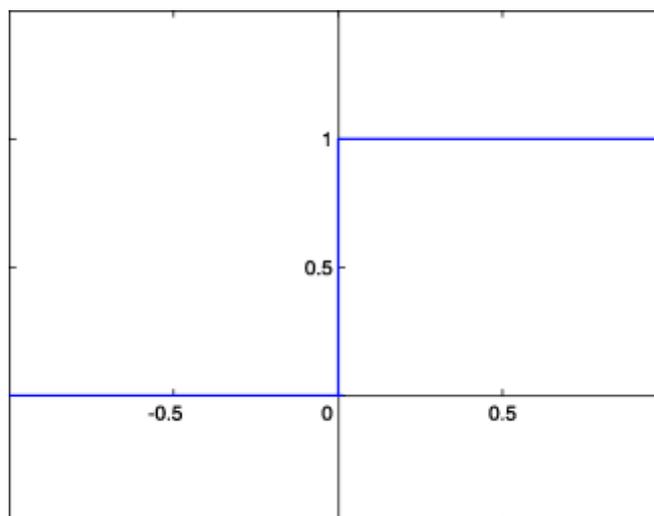


Сигнал на выходе нейрона линейно связан со взвешенной суммой сигналов на его входе. $f(\mathbf{x}) = \mathbf{Ax}$

В искусственных нейронных сетях со слоистой структурой нейроны с передаточными функциями такого типа, как правило, составляют входной слой. Кроме простой линейной функции могут быть использованы её модификации. Например полулинейная функция (если ее аргумент меньше нуля, то она равна нулю, а в остальных случаях, ведет себя как линейная) или шаговая (линейная функция с насыщением), которую можно выразить формулой:

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x \geq 1 \\ x & \text{else} \end{cases}$$

Пороговая передаточная функция



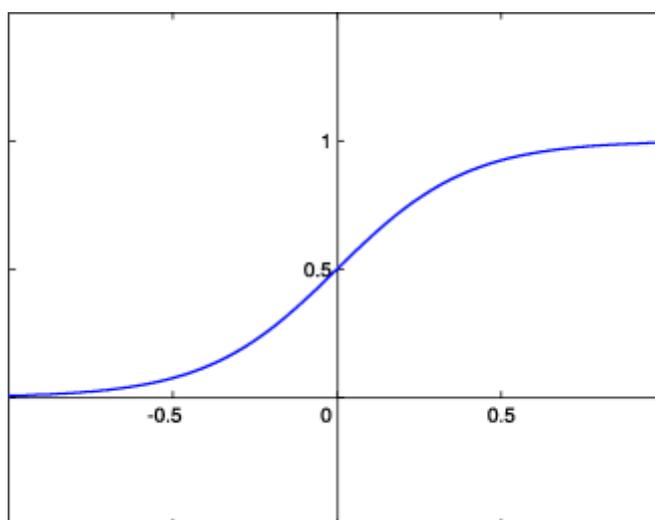
Другое название - Функция Хевисайда. Представляет собой перепад. До тех пор пока взвешенный сигнал на входе нейрона не достигает некоторого уровня T - сигнал на выходе равен нулю. Как только сигнал на входе нейрона превышает указанный уровень -

выходной сигнал скачкообразно изменяется на единицу. Самый первый представитель слоистых искусственных нейронных сетей - перцептрон состоял исключительно из нейронов такого типа. Математическая запись этой функции выглядит так:

$$f(x) = \begin{cases} 1 & \text{if } x \geq T \\ 0 & \text{else} \end{cases}$$

Здесь $T = -\mathbf{w}_0\mathbf{x}_0$ — сдвиг функции активации относительно горизонтальной оси, соответственно под x следует понимать взвешенную сумму сигналов на входах нейрона без учёта этого слагаемого. Ввиду того, что данная функция не является дифференцируемой на всей оси абсцисс, её нельзя использовать в сетях, обучающихся по алгоритму обратного распространения ошибки и другим алгоритмам, требующим дифференцируемости передаточной функции.

Сигмоидальная передаточная функция



Один из самых часто используемых, на данный момент, типов передаточных функций. Введение функций сигмоидального типа было обусловлено ограниченностью нейронных сетей с пороговой функцией активации нейронов — при такой функции активации любой из выходов сети равен либо нулю, либо единице, что ограничивает использование сетей не в задачах классификации. Использование сигмоидальных функций позволило перейти от бинарных выходов нейрона к аналоговым. Функции передачи такого типа, как правило, присущи нейронам, находящимся во внутренних слоях нейронной сети.

Другие функции передачи

Перечисленные выше функции составляют лишь часть от множества передаточных функций, используемых на данный момент. В число других передаточных функций входят такие как: логистическая функция, гиперболический тангенс, радиально-базисная функция передачи, экспонента $f(x) = \exp(-Ax)$, тригонометрический синус, модульная: $f(x) = |x|$, квадратичная.

Стохастический нейрон

Выше описана модель детерминистического искусственного нейрона, т.е. состояние на выходе нейрона однозначно определено результатом работы сумматора входных сигналов. Рассматривают также стохастические нейроны, где переключение нейрона происходит с вероятностью, зависящей от индуцированного локального поля, т.е. передаточная функция определена как

$$f(u) = \begin{cases} 1 & P(u) \\ 0 & 1 - P(u) \end{cases}$$

где распределение вероятности $P(u)$ обычно имеет вид сигмоида:

$$\sigma(u) = \frac{A(T)}{1 + \exp(-u/T)}$$

нормировочная константа $A(T)$ вводится для условия нормализации распределения вероятности:

$$\int_0^1 \sigma(u) du = 1$$

Таким образом, нейрон активируется с вероятностью $P(u)$. Параметр T - аналог температуры (но не температуры нейрона!) и определяет беспорядок в нейронной сети. Если T устремить к 0 , стохастический нейрон перейдет в обычный нейрон с передаточной функцией Хевисайда.

Моделирование формальных логических функций

Нейрон с пороговой передаточной функцией может моделировать различные логические функции. Изображения иллюстрируют, каким образом можно, задав веса входных сигналов и порог чувствительности, заставить нейрон выполнять конъюнкцию (логическое «И») и дизъюнкцию (логическое «ИЛИ») над входными сигналами, а также логическое отрицание входного сигнала. Этим трех операций достаточно, чтобы смоделировать абсолютно любую логическую функцию любого числа аргументов.

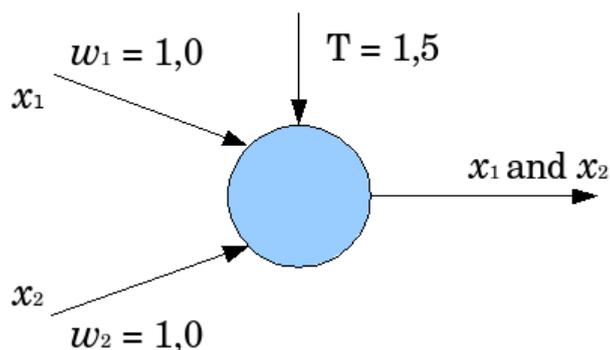


Схема нейрона, настроенного на моделирование логического «И».

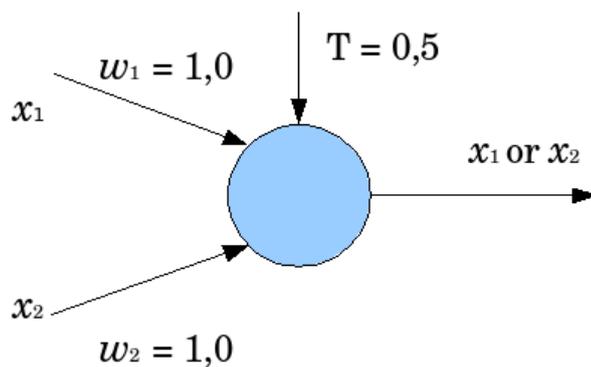


Схема нейрона, настроенного на моделирование логического «ИЛИ»

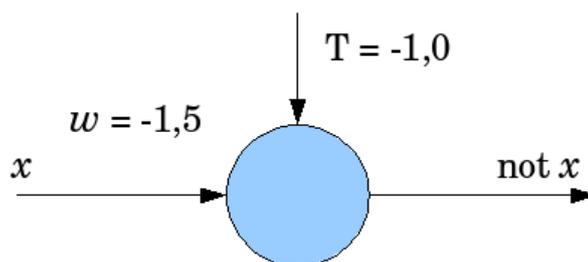
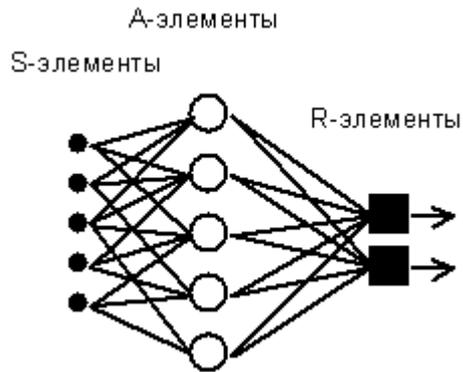


Схема нейрона, настроенного на моделирование логического «НЕ»

Перцептрон Розенблатта

Одной из первых искусственных сетей, способных к перцепции (восприятию) и формированию реакции на воспринятый стимул, явился PERCEPTRON Розенблатта (F.Rosenblatt, 1957). Перцептрон рассматривался его автором не как конкретное техническое вычислительное устройство, а как модель работы мозга. Нужно заметить, что после нескольких десятилетий исследований современные работы по искусственным нейронным сетям редко преследуют такую цель.



Простейший классический перцептрон содержит нейрободобные элементы трех типов, назначение которых в целом соответствует нейронам рефлекторной нейронной сети, рассмотренной в предыдущей лекции. S-элементы формируют сетчатку сенсорных клеток, принимающих двоичные сигналы от внешнего мира. Далее сигналы поступают в слой ассоциативных или A-элементов (для упрощения изображения часть связей от входных S-клеток к A-клеткам не показана). Только ассоциативные элементы, представляющие собой формальные нейроны, выполняют нелинейную обработку информации и имеют изменяемые веса связей. R-элементы с фиксированными весами формируют сигнал реакции перцептрона на входной стимул.

Теорема об обучении персептрона

Для того чтобы нейронная сеть работала - ее надо обучить. Обучение сети состоит в подстройке весовых коэффициентов каждого нейрона. Пусть имеется набор пар векторов $(\mathbf{x}_a, \mathbf{y}_a)$, $\mathbf{a}=\overline{1..p}$, называемый обучающей выборкой. Будем называть нейронную сеть обученной на данной обучающей выборке, если при подаче на входы сети каждого вектора \mathbf{x}_a на выходах всякий раз получается соответствующий вектор \mathbf{y}_a .

Один из методов обучения предложенный Ф.Розенблаттом состоит в итерационной подстройке матрицы весов, последовательно уменьшающей ошибку в выходных векторах. Алгоритм включает несколько шагов:

Шаг 0. Начальные значения весов всех нейронов $W(t=0)$ полагаются случайными.

Шаг 1. Сети предъявляется входной образ \mathbf{x}_a , в результате формируется выходной образ $\tilde{\mathbf{y}}_a \neq \mathbf{y}_a$

Шаг 2. Вычисляется вектор ошибки

$$\delta_a = \mathbf{y}_a - \tilde{\mathbf{y}}_a,$$

делаемой сетью на выходе. Дальнейшая идея состоит в том, что изменение вектора весовых коэффициентов в области малых ошибок должно быть пропорционально ошибке на выходе, и равно нулю, если ошибка равна нулю.

Шаг 3. Вектор весов модифицируется по следующей формуле:

$$W(t + \Delta t) = W(t) + \eta x_a \cdot \delta_a^T. \text{ Здесь } 0 < \eta < 1 - \text{ темп обучения.}$$

Шаг 4. Шаги 1-3 повторяются для всех обучающих векторов. Один цикл последовательного предъявления всей выборки называется эпохой. Обучение завершается по истечении нескольких эпох:

а) когда итерации сойдутся, т.е. вектор весов перестает изменяться

$$W(t + \Delta t) - W(t) \rightarrow 0$$

или

б) когда полная просуммированная по всем векторам абсолютная ошибка станет меньше некоторого малого значения.

Используемая на шаге 3 формула учитывает следующие обстоятельства:

а) модифицируются только компоненты матрицы весов, отвечающие ненулевым значениям входов;

б) знак приращения веса соответствует знаку ошибки, т.е. положительная ошибка ($\mathbf{d} > 0$), значение выхода меньше требуемого) приводит к усилению связи;

в) обучение каждого нейрона происходит независимо от обучения остальных нейронов, что соответствует важному с биологической точки зрения, принципу локальности обучения.

Данный метод обучения был назван Ф.Розенблаттом “методом коррекции с обратной передачей сигнала ошибки”. Позднее более широко стало известно название “**d - правило**”. Представленный алгоритм относится к широкому классу алгоритмов обучения с учителем, поскольку известны как входные вектора, так и требуемые значения выходных векторов (имеется учитель, способный оценить правильность ответа ученика).

Доказанная Розенблаттом теорема о сходимости обучения по **d - правилу** говорит о том, что персептрон способен обучиться любому обучающему набору, который он способен представить.

Теорема гарантирует успешное обучение только для персептронно представимых функций, однако ничего не говорит о том, как это свойство практически обнаружить до

обучения, сколько шагов потребуется при итерационном обучении? Другими словами, затянувшееся обучение может быть как следствием не представимости функции (и в этом случае оно никогда не закончится), так и просто особенностью алгоритма.

Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется НС. Чем сложнее НС, тем масштабнее задачи, подвластные ей. Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные, на сегодняшний день, конфигурации.

Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации. При этом он руководствуется несколькими основополагающими принципами: возможности сети возрастают с увеличением числа ячеек сети, плотности связей между ними и числом выделенных слоев.

Так как проблема синтеза НС сильно зависит от решаемой задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

Используя определенную структуру НС, отвечающую какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время эксплуатации.

Обучение НС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует великое множество различных алгоритмов обучения, которые однако делятся на два больших класса: детерминистские и стохастические. В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором — она производится на основе действий, подчиняющихся некоторому случайному процессу.

Итак, для того чтобы нейронная сеть работала - ее надо обучить. Обучить нейросеть — значит, сообщить ей, чего мы от нее добиваемся. От того, насколько качественно будет выполнено обучение сети зависит способность сети решать поставленные перед ней проблемы во время эксплуатации.

Этот процесс очень похож на обучение ребенка алфавиту. Показав ребенку изображение буквы "А", мы спрашиваем его: "Какая это буква?" Если ответ неверен, мы сообщаем ребенку тот ответ, который мы хотели бы от него получить: "Это буква А". Ребенок запоминает этот пример вместе с верным ответом, то есть в его памяти происходят некоторые изменения в нужном направлении. Мы будем повторять процесс предъявления букв снова и снова до тех пор, когда все 33 буквы будут твердо запомнены. Такой процесс называют "обучение с учителем".

При обучении сети мы действуем совершенно аналогично. Пусть у нас имеется некоторая база данных, содержащая примеры из разных классов, которые необходимо научиться распознавать (набор рукописных изображений букв). Предъявляя изображение буквы "А" на вход сети, мы получаем от нее некоторый ответ, не обязательно верный. Нам известен и верный (желаемый) ответ — в данном случае нам хотелось бы, чтобы на выходе с меткой "А" уровень сигнала был максимален. Обычно в качестве желаемого выхода в задаче классификации берут набор $(1,0,0,\dots)$, где 1 стоит на выходе с меткой "А", а 0 - на всех остальных выходах. Вычисляя разность между желаемым ответом и реальным ответом сети, мы получаем 33 числа — вектор ошибки. Далее применяя

различные алгоритмы по вектору ошибки вычисляем требуемые поправки для весов сети. Одну и ту же букву (а также различные изображения одной и той же буквы) мы можем предъявлять сети много раз. В этом смысле обучение скорее напоминает повторение упражнений в спорте - тренировку.

Оказывается, что после многократного предъявления примеров веса сети стабилизируются, причем сеть дает правильные ответы на все (или почти все) примеры из базы данных. В таком случае говорят, что "сеть выучила все примеры", "сеть обучена", или "сеть натренирована". В программных реализациях можно видеть, что в процессе обучения величина ошибки (сумма квадратов ошибок по всем выходам) постепенно уменьшается. Когда величина ошибки достигает нуля или приемлемого малого уровня, тренировку останавливают, а полученную сеть считают натренированной и готовой к применению на новых данных.

Важно отметить, что вся информация, которую сеть имеет о задаче, содержится в наборе примеров. Поэтому качество обучения сети напрямую зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают данную задачу. Так, например, бессмысленно использовать сеть для распознавания буквы "А", если в обучающей выборке она не была представлена. Считается, что для полноценной тренировки требуется хотя бы несколько десятков (а лучше сотен) примеров. Обучение сети — сложный и наукоемкий процесс. Алгоритмы обучения имеют различные параметры и настройки, для управления которыми требуется понимание их влияния.

Нейросетевые экспертные системы, создаваемые для конкретных областей применения, условно разделяются на два типа:

- системы первичного уровня
- системы вторичного уровня

Системы первичного уровня работают с прямыми (первичными) признаками исследуемого объекта. Например, при диагностике заболеваний организма, таковыми являются температура, давление, вес, параметры электрокардиограмм, количественные характеристики состояния крови и т.п. В инженерном мониторинге - сигналы, поступающие от датчиков и преобразователей. При этом, для принятия решения системой, эксперты не привлекаются.

Системы вторичного уровня (системы поддержки принятия решений СППР) работают не с непосредственными признаками, а с мнением группы экспертов. Каждый эксперт группы оценивает состояние объекта по первичным признакам в различных ситуациях и высказывает свое мнение. СППР на основании накопленного опыта эффективно объединяет решения экспертов и делает общий вывод. Системы вторичного уровня целесообразно применять в задачах с числом параметров, превышающим 500, когда нельзя охватить абсолютно все.

Применение нейросети

После того, как сеть обучена, ее можно применять для решения поставленной задачи. Важнейшая особенность человеческого мозга состоит в том, что, однажды обучившись определенному процессу, он может верно действовать и в тех ситуациях, в которых он не бывал в процессе обучения. Например, можно читать почти любой почерк, даже если видим его первый раз в жизни. Так же и нейросеть, грамотным образом обученная, может с большой вероятностью правильно реагировать на новые, не предъявленные ей ранее данные. Например, мы можем нарисовать букву "А" другим почерком, а затем предложить нашей сети классифицировать новое изображение. Веса обученной сети хранят достаточно много информации о сходстве и различиях букв, поэтому можно рассчитывать на правильный ответ и для нового варианта изображения

Применение нейронных технологий

Актуальность исследований в направлении нейронных технологий подтверждается массой различных применений НС, таких как:

- автоматизация процессов распознавания образов;
- адаптивное управление;
- аппроксимация функционалов;
- создание экспертных систем;
- организация ассоциативной памяти

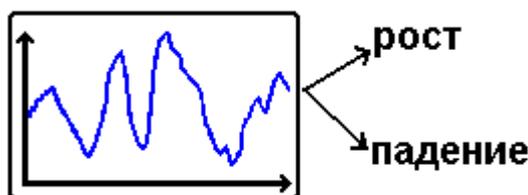
и многие другие приложения. С помощью НС можно, например:

- предсказывать показатели биржевого рынка;
- выполнять распознавание оптических или звуковых сигналов;
- создавать самообучающиеся системы, способные управлять автомашиной при парковке;
- синтезировать речь по тексту.

Примеры практического применения нейронных сетей

В качестве примеров рассмотрим наиболее известные классы задач, для решения которых в настоящее время широко применяются нейросетевые технологии.

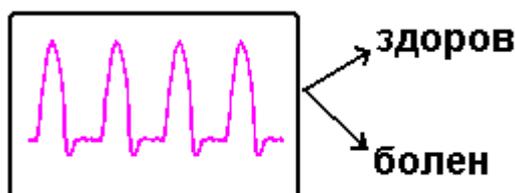
Прогнозирование



Прогноз будущих значений переменной, зависящей от времени, на основе предыдущих значений ее и/или других переменных. В финансовой области это, например, прогнозирование курса акций на 1 день вперед, или прогнозирование изменения курса валют на определенный период времени и т.д.

Распознавание или классификация

Определение, к какому из заранее известных классов принадлежит тестируемый объект. Следует отметить, что задачи классификации очень плохо алгоритмируются. Если в случае распознавания букв верный ответ очевиден для нас заранее, то в более сложных практических задачах обученная нейросеть выступает как эксперт, обладающий большим опытом и способный дать ответ на трудный вопрос.



Примером такой задачи служит медицинская диагностика, где сеть может учитывать большое количество числовых параметров (энцефалограмма, давление, вес и т.д.). Конечно, "мнение" сети в этом случае нельзя считать окончательным.



Классификация **предприятий по степени их перспективности** - это уже привычный способ использования нейросетей в практике крупных компаний. При этом сеть также использует множество экономических показателей, сложным образом связанных между собой.

Кластеризация и поиск закономерностей

Помимо задач классификации, нейросети широко используются для поиска зависимостей в данных и кластеризации.

Например, нейросеть на основе методики МГУА (метод группового учета аргументов) позволяет на основе обучающей выборки построить зависимость одного параметра от других в виде полинома. Такая сеть может не только мгновенно выучить таблицу умножения, но и найти сложные скрытые зависимости в данных (например, финансовых), которые не обнаруживаются стандартными статистическими методами.

Кластеризация - это разбиение набора примеров на несколько компактных областей (кластеров), причем число кластеров заранее неизвестно. Кластеризация позволяет представить неоднородные данные в более наглядном виде и использовать далее для исследования каждого кластера различные методы. Например, таким образом можно быстро выявить фальсифицированные страховые случаи или недобросовестные предприятия.

Несмотря на большие возможности, существует ряд недостатков, которые все же ограничивают применение нейросетевых технологий. Во-первых, нейронные сети позволяют найти только субоптимальное решение, и соответственно они неприменимы для задач, в которых требуется высокая точность. Функционируя по принципу черного ящика, они также неприменимы в случае, когда необходимо объяснить причину принятия решения. Обученная нейросеть выдает ответ за доли секунд, однако относительно высокая вычислительная стоимость процесса обучения как по времени, так и по объему занимаемой памяти также существенно ограничивает возможности их использования. И все же класс задач, для решения которых эти ограничения не критичны, достаточно широк.

Применение нейрокомпьютеров в средствах измерений

Длительное время считалось, что нейрокомпьютеры эффективны для решения так называемых неформализуемых и плохо формализуемых задач, требующих процесса обучения на реальном экспериментальном материале.

В первую очередь к таким задачам относилась задача аппроксимации частного вида функций, принимающих дискретное множество значений, т.е. задача распознавания образов.

В настоящее время к этому классу задач добавляется класс задач, иногда не требующий обучения на экспериментальном материале, но хорошо представимый в нейросетевом логическом базисе. К ним относятся задачи с ярко выраженным естественным параллелизмом обработки сигналов, обработка изображений и др. Подтверждением точки зрения, что в будущем нейрокомпьютеры будут более эффективными, чем прочие архитектуры, может, в частности, служить резкое расширение в последние годы класса общематематических задач, решаемых в нейросетевом логическом базисе. К ним, кроме перечисленных выше, можно отнести задачи решения:

- линейных и нелинейных алгебраических уравнений и неравенств большой размерности;
- систем нелинейных дифференциальных уравнений;
- уравнений в частных производных;
- задач оптимизации и других задач.

В целом формируется три раздела нейроматематики: общая, прикладная и специальная.

Общая нейроматематика

В последние годы резко расширился интерес к решению общих математических задач в нейросетевом логическом базисе. Даже такие, казалось бы, простые задачи, как сложение чисел, умножение, деление, извлечение корня, обращение чисел и другие, многие авторы пытаются решить с помощью нейрокомпьютеров, так как при ориентации на нейросетевую физическую реализацию алгоритмов эти операции можно выполнить значительно эффективнее, чем на известных булевских элементах. Во многих научных работах разрабатываются и исследуются нейросетевые алгоритмы решения следующих задач:

- системы линейных уравнений и неравенств;
- обращение матриц;
- задачи оптимизации (линейное и нелинейное программирование);
- сортировка;

Решение других задач общей нейроматематики, таких, например, как решение:

- обыкновенных дифференциальных уравнений с произвольной нелинейной правой частью;
- дифференциальных уравнений в частных производных;
- отдельным важным разделом нейроматематики так же является комплекс задач, связанных с формализацией с помощью графов.

Прикладная нейроматематика

Как правило, множество задач прикладной нейроматематики представляют задачи в принципе не решаемые известными типами вычислительных машин. Многие из них уже сейчас решаются с помощью нейрокомпьютеров.

Общие задачи:

- контроль кредитных карточек;
- системы скрытого обнаружения веществ с помощью устройств на базе тепловых нейронов и нейрокомпьютера на заказных цифровых нейрочипах;
- системы автоматизированного контроля безопасного хранения ядерных изделий и автоматизированного контроля установок и слежения за безопасностью АЭС.

Задачи обработки изображений:

- обработка аэрокосмических изображений, в частности: сжатие с восстановлением, сегментация, контрастирование, обработка текстур;
- формирование и обработка изображений, формируемых адаптивным составным телескопом (100-400 зеркал);
- выделение на изображении движущихся целей;
- поиск и распознавание на изображении объектов заданной формы;
- обработка информации в высокопроизводительных сканерах, ориентированных на применение в больших информационных системах.

Задачи обработки сигналов:

- прогнозирование финансовых показателей;
- упреждение мощности АЭС;
- обработка траекторных измерений;
- выделение речевого сигнала в негауссовском шуме значительного уровня;
- распознавание речи.

Задачи управления динамическими объектами:

- управление вертолетом, самолетом;
- управление зеркалами (100-400 зеркал) адаптивного составного телескопа;
- управление роботами.

Нейросетевые экспертные системы:

- система выбора воздушных маневров в процессе ведения воздушного боя;
- медицинская диагностическая экспертная система для оценки состояния летчика.

Системы виртуальной реальности.

Основное направление применения нейроподобных структур - исследования, где по каким-то причинам не может быть задействован человек. Это могут быть исследования вредные для человека, недоступные в данный момент или слишком дорогие.

Ярким примером является интеллектуальный мобильный робот (ИМР) - это техническая система, способная автономно двигаться к цели по заранее неизвестной среде без помощи человека. Принципиально новый подход к проблеме создания систем управления таких интеллектуальных мобильных роботов был предложен в НИИ многопроцессорных вычислительных систем Таганрогского государственного радиотехнического университета. Отличительной особенностью этого подхода является создание систем управления ИМР на базе однородных нейроподобных структур (ОНС), реализующих нецифровые методы обработки информации, присущие мозгу человека.

Использование таких нейроподобных структур позволяет добиться ряда важных преимуществ.

В результате проведенных исследований разработаны теоретические и практические основы построения систем управления ИМР на базе однородных нейроподобных структур. Для практической реализации развиваемого подхода была разработана и изготовлена специальная элементная база, включающая в себя СБИС фрагмента ОНС и многокристальный модуль ОНС. СБИС фрагмента ОНС содержит 128 нейропроцессоров на кристалле, объединенных в одно решающее поле, причем предусмотрена возможность стыковки данных СБИС друг с другом с целью наращивания размеров решающего поля. Многокристальный модуль содержит 8 СБИС фрагмента ОНС в одном корпусе, объединенных в одно решающее поле.

Прикладные результаты

Проведенные теоретические исследования послужили основой для создания целого ряда образцов нейропроцессорных систем управления ИМР различного назначения. В частности, в рамках российской космической программы, было создано несколько прототипов интеллектуальных мобильных роботов-планетоходов, предназначенных для исследования поверхности планет Солнечной системы, в частности, Марса.

Бортовая система управления данных ИМР была построена на базе нейроподобной структуры, содержащей 4096 элементарных нейропроцессоров. Ввод информации о среде движения в систему осуществлялся с помощью сканирующего лазерного дальномера и телекамеры. Данная система обеспечивала возможность автономного движения ИМР к последовательности из 16 целей по заранее неизвестной, пересеченной местности со скоростями до 10 км/час. Созданные прототипы ИМР- планетоходов прошли успешные испытания на естественных полигонах в условиях, приближенных к реальным, подтвердившие работоспособность и эффективность заложенных в них принципов.

Основные области применения

Робототехнические системы, предназначенные для функционирования в условиях опасных для жизнедеятельности человека, в частности в космосе или под водой, в зоне радиоактивного или химического загрязнения, при инспекции взрывоопасных зон и предметов, при проведении спасательных работ в зонах разрушения и т.д. Кроме того, полученные результаты могут также найти применение при создании интеллектуальных промышленных и домашних роботов.

Искусственный интеллект

Под интеллектом мы понимаем способность любого организма (или устройства) достигать некоторой измеримой степени успеха при поиске одной из многих возможных целей в обширном многообразии сред. Необходимо отличать знания от интеллекта, имея в виду, что знания - полезная информация, накопленная индивидуумом, а интеллект - это его способность предсказывать состояние внешней среды в сочетании с умением преобразовывать каждое предсказание в подходящую реакцию, ведущую к заданной цели. По-разному дается и определение искусственного интеллекта. Полагают, что о реализации искусственного интеллекта можно будет говорить лишь тогда, когда автомат начнет решать задачи, непосильные для человека, причем сделает это не в результате высокого быстродействия, а в результате применения нового найденного метода. Однако не все с этим согласны. В большинстве случаев исследований по искусственному интеллекту лишь соизмеримыми с результатами, полученными человеком, и не столь оригинальными.

Принято различать три основных пути моделирования интеллекта и мышления: классический, или (как его теперь называют) бионический; эвристического программирования и эволюционного моделирования. Рассмотрим их в этой последовательности.

Бионическое моделирование

Непосредственное моделирование человеческого мозга (т.е. моделирование каждой нервной клетки и связей между ними) с целью создания автоматов, обладающих интеллектом, чрезвычайно сложно. Мозг представляет собой самую сложную и лишь частично изученную структуру. Сложнейшее переплетение связей коры головного мозга практически не поддаются расшифровке. Известно лишь примерное расположение зон мозга, отвечающих за ту или иную функцию. В настоящее время не известен и принцип работы мозговых элементов нейронов, многочисленные связи которых имеют внешне хаотический характер. В головном мозге человека находится 70 миллиардов нейронов, а количество связей между ними достигает 60 триллионов. Однако за последние годы исследователи начали понимать, как "выглядят" многие процессы в мозге.

Попытки смоделировать работу головного мозга соединением между собой множества процессоров подобно нейронной сети, показали, что некоторое увеличение скорости и потока обрабатываемой информации идет лишь до уровня одного - двух десятков процессоров, а затем начинается резкий спад производительности. Процессоры как бы "теряются", перестают контролировать ситуацию или проводят большую часть времени в ожидании соседа. Некоторых успехов удалось добиться лишь в приборах, работающих в "двумерном варианте", т.е. обрабатывающих не последовательную, а параллельную информацию, например в системах распознавания образов. В них одна плоскость данных одновременно взаимодействует с другой, причем количество единиц информации может достигать нескольких миллионов. Таким образом, происходит одновременный охват изучаемого объекта, а не последовательное изучение его частей.

Эвристическое программирование

Второй подход к решению задачи искусственного интеллекта связан с эвристическим программированием и решает задачи, которые в общем можно назвать творческими.

Практичность этого метода заключается в радикальном уменьшении вариантов, необходимых при использовании метода проб и ошибок. Правда, всегда существует вероятность упустить наилучшее решение, так что говорят, что этот метод предлагает решения с некоторой вероятностью правильности.

Обычно используют два метода: метод анализа целей и средств и метод планирования. Первый заключается в выборе и осуществлении таких операций, которые последовательно уменьшают разницу между исходным и конечным состоянием задачи. Во втором методе вырабатывается упрощенная формулировка исходной задачи, которая также решается методом анализа целей и средств. Один из полученных вариантов дает решение исходной задачи.

Эволюционное моделирование

Третий подход является попыткой смоделировать не то, что есть, а то, что могло бы быть, если бы эволюционный процесс направлялся в нужном направлении и оценивался предложенными критериями.

Идея эволюционного моделирования сводится к экспериментальной попытке заменить процесс моделирования человеческого интеллекта моделированием процесса его эволюции. При моделировании эволюции предполагается, что разумное поведение предусматривает сочетание способности предсказывать состояние внешней среды с умением подобрать реакцию на каждое предсказание, которое наиболее эффективно ведет к цели.

Этот метод открывает путь к автоматизации интеллекта и освобождению от рутинной работы. Это высвобождает время для проблемы выбора целей и выявления параметров среды, которые заслуживают исследования. Такой принцип может быть применен для использования в диагностике, управлении неизвестными объектами, в игровых ситуациях.

Итак, существуют три пути моделирования интеллекта: бионический, эвристический и эволюционный. В зависимости от использованных средств можно выделить три фазы в исследованиях.

Первая фаза - создания устройств, выполняющих большое число логических операций с высоким быстродействием.

Вторая фаза включает разработку проблемно-ориентированных языков для использованного на оборудовании, созданном в первой фазе.

Третья фаза наиболее выражена в эволюционном моделировании. В ходе развития этой фазы отпадает необходимость в точной формулировке постановки задачи, т.е. задачу можно сформулировать в терминах цели и допустимых затрат, а метод решения будет найден самостоятельно по этим двум параметрам.

Работы по искусственному интеллекту во многом тесно связаны с философской проблемой кибернетического моделирования. Эти работы часто связывают с построением точной копии человеческого мозга. Однако такой подход можно назвать "некибернетическим". Каковы же черты кибернетического метода мышления, какие вопросы вносит кибернетика в человеческое познание?

Путь, который предлагает кибернетика, состоит в построении эскизных моделей, охватывающих все более и более широкий диапазон функций мышления. Задачи раскрыть "в лоб" "сущность мышления" не ставится, а ставится задача построения эскизных

моделей, позволяющих описать отдельные его стороны, воспроизведены отдельные его функции и, двигаясь в этом направлении, строить системы, все более приближающиеся к человеческому мозгу.

Уже сейчас существуют машины и программы, способные в процессе работы самообучаться, т.е. повышать эффективность приспособления к внешним факторам. В будущем, возможно, появятся машины, обладающие таким уровнем приспособляемости и надежности, что необходимость человеку вмешиваться в процесс отпадет. В этом случае возможна потеря самим человеком своих качеств, ответственных за поиск решений. Налицо возможная деградация способностей человека к реакции на изменение внешних условий и, возможно, неспособность принятия управления на себя в случае аварийной ситуации.

Встает вопрос о целесообразности введения некоторого предельного уровня в автоматизации процессов, связанных с тяжелыми аварийными ситуациями, т.к. нельзя доверять искусственному интеллекту полный контроль.

В этом случае у человека, "надзирающим" за управляющей машиной, всегда хватит умения и реакции таким образом воздействовать на ситуацию, чтобы погасить разгорающуюся аварийную ситуацию. Таковые ситуации возможны на транспорте, в ядерной энергетике. Особо стоит отметить такую опасность в ракетных войсках стратегического назначения, где последствия ошибки могут иметь фатальный характер. Несколько лет назад в США начали внедрять полностью компьютеризированную систему запуска ракет по командам суперкомпьютера, обрабатывающего огромные массивы данных, собранных со всего света. Однако оказалось, что даже при условии многократного дублирования и перепроверки, вероятность ошибки оказалась бы столь велика, что отсутствие контролирующего оператора привело бы к непоправимой ошибке. От системы отказались. Люди будут постоянно решать проблему искусственного интеллекта, постоянно сталкиваясь со все новыми проблемами. И, видимо, процесс этот бесконечен.