Лекция 10

Тема: Персептроны и зарождение искусственных нейронных сетей. Обучение персептрона

В качестве предмета исследования *искусственные нейронные сети* впервые заявили о себе в 1940-е годы. Стремясь воспроизвести функции человеческого мозга, исследователи создали простые аппаратные (а позже *программные) модели* биологического нейрона и системы его соединений. Когда нейрофизиологи достигли более глубокого понимания нервной системы человека, эти ранние попытки стали восприниматься как весьма грубые аппроксимации. Тем не менее, на этом пути были достигнуты впечатляющие результаты, стимулировавшие дальнейшие исследования, которые привели к созданию более изощренных сетей.

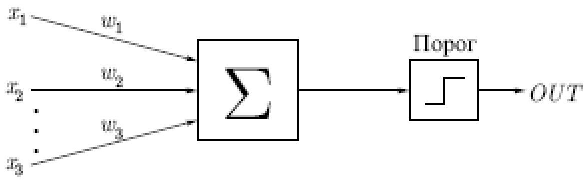

Рис. 1.

Первое систематическое изучение *искусственных нейронных сетей* было предпринято Маккалокком и Питтсом в 1943 г. Позднее они исследовали сетевые парадигмы для *распознавания изображений*, подвергаемых сдвигам и поворотам. Простая нейронная модель, показанная на рис. 1, использовалась в большей части их *работ*. Элемент $\sum$ умножает каждый вход x на *вес* w и суммирует взвешенные входы. Если полученная сумма больше заданного порогового значения, *выход* равен единице, в противном случае — нулю. Эти системы (и множество им подобных) получили название ***персептронов***. Они состоят из одного слоя *искусственных нейронов*, соединенных с помощью весовых коэффициентов с множеством входов (см. рис. 2), хотя, в принципе, описываются и более сложные системы. В 60-е годы *персептроны* вызвали большой интерес и оптимизм. Одной из первых искусственных сетей, способных к перцепции (восприятию) и формированию реакции на воспринятый раздражитель, явился *PERCEPTRON* Розенблатта (F.Rosenblatt, 1957). *Персептрон* рассматривался его автором не как конкретное техническое (вычислительное) устройство, а как модель работы мозга. Розенблатт называл такую нейронную *сеть* трехслойной, однако, *по* современной терминологии, представленная *сеть* обычно называется **однослойной**, так как имеет только один слой нейропроцессорных элементов.

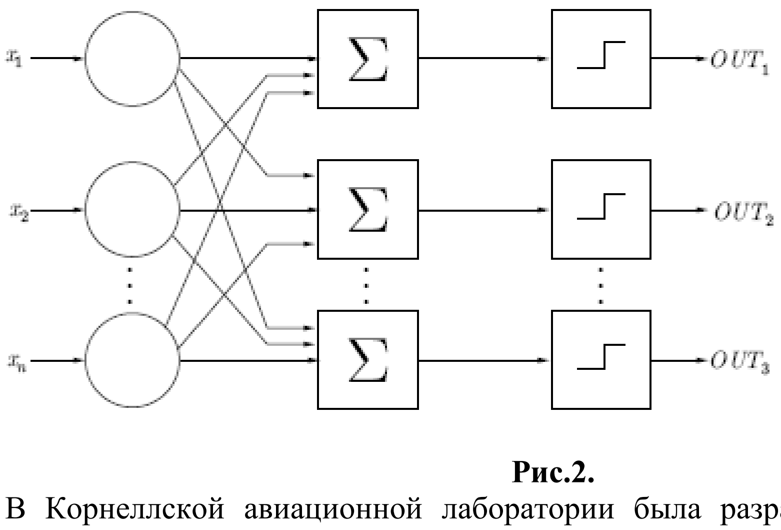

Рис.2.

В Корнеллской авиационной лаборатории была разработана электротехническая модель *персептрона* MARK-1, которая содержала 8 выходных элементов. На этом *персептроне* была проведена серия экспериментов *по* распознаванию букв алфавита и геометрических образов.

Персептронная представляемость

Доказательство теоремы обучения *персептрона* показало, что *персептрон* способен научиться всему, что он способен представлять. Важно при этом уметь различать представляемость и обучаемость. Понятие представляемости относится к способности *персептрона* (или другой сети) моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

Для иллюстрации проблемы представляемости допустим, что у нас есть множество карт, помеченных цифрами от 0 до 9. Допустим также, что мы обладаем гипотетической машиной, способной отличать карты с нечетным номером от карт с четным номером и зажигающей *индикатор* на своей панели при предъявлении карты с нечетным номером. Представима ли такая машина *персептроном*? То есть возможно ли сконструировать *персептрон* и настроить его веса (неважно, каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это достижимо, то говорят, что *персептрон* способен представлять желаемую машину. Мы увидим, что возможности представления однослойными *персептронами* весьма ограниченны. Имеется много простых машин, которые не могут быть представлены *персептроном*, независимо от того, как настраиваются его веса.

Эффективность запоминания

Серьезные вопросы существуют относительно эффективности запоминания информации в *персептроне* (или любых других нейронных сетях) *по* сравнению с обычной компьютерной памятью и методами поиска информации в ней. Например, в компьютерной памяти можно хранить все входные образы вместе с классифицирующими битами. *Компьютер* должен найти требуемый образ и дать его классификацию. Многочисленные и хорошо известные методы могли бы применяться для ускорения поиска. Если точное соответствие не найдено, то для ответа может быть использовано правило ближайшего соседа.

Число битов, необходимое для хранения этой же информации в весах *персептрона*, может быть значительно меньшим *по* сравнению с методом обычной компьютерной памяти, если образы допускают экономичную *запись*.

Обучение персептрона

Способность *искусственных нейронных сетей* к обучению является их наиболее интригующим свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами совершенствуют себя в результате попыток создать лучшую модель поведения.

Используя критерий *линейной разделимости*, можно решить, способна ли *однослойная нейронная сеть* реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесет мало пользы, если у нас нет способа найти нужные значения для весов и порогов. Чтобы *сеть* представляла практическую ценность, нужен *систематический* метод (*алгоритм*) для вычисления этих значений. Ф.Розенблатт создал такой метод в своем *алгоритме обучения персептрона* и доказал: *персептрон* может быть обучен всему, что он может реализовывать.

Обучение может быть с учителем или без него. Для обучения с учителем нужен "внешний" учитель, который оценивал бы поведение системы и управлял ее последующими модификациями. При обучении без учителя, которое будет рассмотрено на последующих лекциях, *сеть* путем *самоорганизации* делает требуемые изменения. Обучение *персептрона* является обучением с учителем.

Алгоритм обучения персептрона может быть реализован на цифровом компьютере или другом электронном устройстве, и *сеть* становится в определенном смысле самоподстраивающейся. *По* этой причине процедуру подстройки весов обычно называют "обучением" и говорят, что *сеть* "обучается". *Доказательство* Розенблатта стало основной вехой и дало мощный импульс исследованиям в этой области. Сегодня в той или иной форме элементы *алгоритма обучения персептрона* встречаются во многих сетевых парадигмах.

Алгоритм обучения однослойного персептрона

Персептрон должен решать задачу классификации *по* бинарным входным сигналам. Набор входных сигналов будем обозначать n -мерным вектором x. Все элементы вектора являются булевыми переменными (переменными, принимающими значения "*Истина*" или "*Ложь*"). Однако иногда полезно оперировать числовыми значениями. Будем считать, что значению "*ложь*" соответствует числовое *значение* 0, а значению "*Истина*" соответствует 1.

Персептроном будем называть устройство, вычисляющее следующую систему функций:

$$\psi = \left[\sum_{i=1}^{m} w_i x_i > \theta\right], \qquad (1)$$

где w_I — веса *персептрона*, θ — порог, x_I — значения входных сигналов, скобки $[\;]$ означают переход от булевых (логических) значений к числовым значениям *по* правилам, изложенным выше.

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

Предложенный Ф.Розенблаттом метод обучения 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 x^α, в результате формируется выходной образ $\tilde{y}^\alpha \neq y^\alpha$.
Шаг 2	Вычисляется вектор ошибки $\delta^\alpha = (y^\alpha - \tilde{y}^\alpha)$, 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: $W(t+\Delta T) = W(t) + \eta x^\alpha \cdot (\delta^\alpha)^T$. Здесь $0 < \eta < 1$ — 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, или б) когда полная, просуммированная по всем векторам абсолютная ошибка станет меньше некоторого малого значения.

Объясним данный *алгоритм* более подробно. Подаем на вход *персептрона* такой *вектор* x, для которого уже известен правильный ответ. Если выходной сигнал *персептрона* совпадает с правильным ответом, то никаких действий предпринимать не надо. В случае ошибки, необходимо обучить *персептрон* правильно решать данный пример. Ошибки могут быть двух типов. Рассмотрим каждый из них.

Первый тип ошибки: на выходе *персептрона* — 0, а правильный ответ — 1. Для того чтобы *персептрон* выдавал *правильный ответ*, необходимо, чтобы сумма в правой части (1) стала больше. Поскольку переменные принимают значения 0 или 1, увеличение суммы может быть достигнуто за счет увеличения весов w_i. Однако нет смысла увеличивать веса при переменных x_i , которые равны нулю. Таким образом, следует увеличить веса w_i при тех переменных x_i, которые равны 1.

Первое правило. Если на выходе *персептрона* получен 0, а *правильный ответ* равен 1, то необходимо увеличить веса связей между одновременно активными нейронами. При этом выходной *персептрон* считается активным. Второй тип ошибки: на выходе *персептрона* — 1, а правильный ответ равен нулю. Для обучения правильному решению данного примера следует уменьшить сумму в правой части (1). Следовательно, необходимо уменьшить веса связей w_i при тех переменных, которые равны 1 (поскольку нет смысла уменьшать веса связей при равных нулю переменных x_i). Необходимо также провести эту процедуру для всех активных нейронов предыдущих слоев. В результате получаем второе правило.

Второе правило. Если на выходе *персептрона* получена *единица*, а *правильный ответ* равен нулю, то необходимо уменьшить веса связей между одновременно активными нейронами.

Таким образом, процедура обучения сводится к последовательному перебору всех примеров обучающего *множества* с применением правил обучения для ошибочно решенных примеров. Если после очередного *цикла* предъявления всех примеров окажется, что все они решены правильно, то процедура обучения завершается.

Нерассмотренными остались два вопроса. Первый — о *сходимости* процедуры обучения. Второй — на сколько нужно увеличивать (уменьшать) веса связей при применении правил обучения.

Ответ на первый вопрос дают следующие теоремы.

Теорема о сходимости *персептрона*. Если существует *вектор* параметров w , при котором *персептрон* правильно решает все примеры обучающей выборки, то при обучении *персептрона по* вышеописанному алгоритму решение будет найдено за конечное число шагов.

Теорема о "зацикливании" *персептрона*. Если не существует вектора параметров w , при котором *персептрон* правильно решает все примеры обучающей выборки, то при обучении *персептрона по* данному правилу через конечное число шагов *вектор* весов начнет повторяться.

Таким образом, данные теоремы утверждают, что, запустив процедуру обучения *персептрона*, через конечное время мы либо получим обучившийся *персептрон*, либо ответ, что данный *персептрон* поставленной задаче обучиться не может.

Доказательства этих теорем в данное учебное пособие не включены.

Целочисленность весов персептронов

Для ответа на вопрос о количественных характеристиках вектора w рассмотрим следующую теорему.

Теорема. Любой *персептрон* можно заменить другим *персептроном* того же вида с целыми весами связей.

Доказательство.Обозначим множество примеров одного класса (*правильный ответ* равен 0) через X_0, а другого (*правильный ответ* равен 1) — через X_1. Вычислим максимальное и минимальное значения суммы в правой части (1):

$$s_0 = \max_{x \in X_0} \sum_i w_i x_i, \quad s_1 = \min_{x \in X_1} \sum_i w_i x_i.$$

Определим допуск s как *минимум* из s_0 и s_1. Положим $\delta = s/(m+1)$, где m — число слагаемых в (1). Поскольку *персептрон* (1) решает поставленную задачу классификации и множество примеров в *обучающей выборке* конечно, то $\delta > 0$. Из теории чисел известна теорема о том, что любое *действительное число* можно сколь угодно точно приблизить *рациональными числами*. Заменим веса w_i на *рациональные числа* так, чтобы выполнялись следующие неравенства: $|w_i - w_i'| < \delta$.

Из этих неравенств следует, что при использовании весов w_i' *персептрон* будет работать с теми же результатами, что и первоначальный *персептрон*. Действительно, если правильным ответом примера является 0, имеем $\sum_i w_i x_i \leqslant -s$.

Подставив новые веса, получим:

$$\sum_i w_i' x_i = \sum_i (w_i' - w_i) x_i + \sum_i w_i x_i \leqslant \sum_i |w_i' - w_i| x_i - $$
$$- s \leqslant$$
$$\leqslant \sum_i |w_i' - w_i| - s < (m+1)\delta - s = 0.$$

Откуда следует необходимое *неравенство*

$$\sum_i w_i' x_i < 0. \qquad (2)$$

Аналогично, в случае правильного ответа равного 1, имеем $\sum_i w_i x_i < s$, откуда, подставив новые веса и порог, получим:

$$\sum_i w_i' x_i = \sum_i (w_i' - w_i) x_i + \sum_i w_i x_i \geqslant s - \sum_i |w_i' - $$
$$- w_i| x_i \geqslant$$
$$\geqslant s - \sum_i |w_i' - w_i| > s - (m+1)\delta = 0.$$

Отсюда следует выполнение неравенства

$$\sum_i w_i' x_i > 0. \qquad (3)$$

Неравенства (2) и (3) доказывают возможность замены всех весов и порога любого *персептрона рациональными числами*. Очевидно также, что при умножении всех весов и порога на одно и то же ненулевое число *персептрон* не изменится. Поскольку любое *рациональное число* можно представить в виде отношения целого числа к натуральному числу, получим

$$\psi = \left[\sum_{i=1}^{m} w_i x_i > 0\right] = \left[\sum_{i=1}^{m} w_i' x_i > 0\right] = \left[\sum_{i=1}^{m} \frac{w_i''}{r_i} x_i > 0\right] \qquad (4)$$

где w_i'' — целые числа. Обозначим через r *произведение* всех знаменателей: $r = \prod_{i=0}^{m} r_i$. Умножим все веса и порог на r. Получим веса целочисленные $w''' = rw''$. Из (2), (3) и (4) получаем

$$\psi = \left[\sum_{i=1}^{m} w_i x_i > 0\right] = \left[\sum_{i=1}^{m} w_i' x_i > 0\right] = \left[\sum_{i=1}^{m} \frac{w_i''}{r_i} x_i > 0\right] = \left[\sum_{i=1}^{m} w_i''' x_i > 0\right],$$

что и завершает *доказательство* теоремы.

Поскольку из доказанной теоремы следует, что веса *персептрона* являются целыми числами, то вопрос о выборе шага при применении правил обучения решается просто: веса и порог следует увеличивать (уменьшать) на единицу.

Двуслойность персептрона

Как уже упоминалось в начале лекции, *алгоритм обучения персептрона* возможно использовать и для многослойных *персептронов*. Однако теоремы о *сходимости* и зацикливании *персептрона*, приведенные выше, верны только при обучении однослойного *персептрона* — или многослойного *персептрона* при условии, что обучаются только веса *персептрона*, стоящего в последнем слое сети. В случае произвольного многослойного *персептрона* они не работают. Следующий пример демонстрирует основную проблему, возникающую при обучении многослойных *персептронов*.

Пусть веса всех слоев *персептрона* в ходе обучения сформировались так, что все примеры обучающего *множества*, кроме первого, решаются правильно. При этом правильным ответом первого примера является 1. Все входные сигналы *персептрона* последнего слоя равны нулю. В этом случае первое правило не дает результата, поскольку все нейроны предпоследнего слоя не активны. Существует множество способов решать эту проблему. Однако все эти методы не являются регулярными и не гарантируют *сходимость* многослойного *персептрона* к решению, даже при условии, что такое решение существует.

Трудности с алгоритмом обучения персептрона

Иногда бывает сложно определить, выполнено ли условие разделимости для конкретного обучающего *множества*. Кроме того, во многих встречающихся на практике ситуациях входы часто меняются во времени и могут быть разделимы в один момент времени и неразделимы - в другой. В доказательстве *алгоритма обучения персептрона* ничего не говорится также о том, сколько шагов требуется для обучения сети. Мало утешительного знать, что обучение закончится за конечное число шагов, если необходимое для этого время сравнимо с геологической эпохой. Кроме того, не доказано, что *персептронный алгоритм обучения* более быстр *по* сравнению с простым перебором всех возможных значений весов, и в некоторых случаях этот примитивный подход может оказаться лучше.

На эти вопросы никогда не находилось удовлетворительного ответа, они относятся к природе обучающего материала. В различной форме они возникнут на последующих лекциях, где рассматриваются другие сетевые парадигмы. Ответы для современных сетей, как правило, не более удовлетворительны, чем для *персептрона*. Эти проблемы являются важной областью современных исследований.