Обучение однослойного перцептрона

Обучение однослойного перцептрона

Обучение перцептрона

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 перцептрона и доказал: перцептрон 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 перцептрона является обучением с учителем.

Алгоритм обучения перцептрона 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 перцептрона встречаются во многих сетевых парадигмах.

Алгоритм обучения однослойного перцептрона

Перцептрон 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.

Перцептроном будем называть устройство, вычисляющее следующую систему функций:

|  |  |
| --- | --- |
| \psi=\Bigl[\sum_{i=1}^m w_i x_i>\theta\Bigr],\quad | ( 1) |

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

Обучение перцептрона состоит в подстройке весовых коэффициентов. Пусть имеется набор пар векторов (x^\alpha,
y^\alpha), \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^\alpha на выходах всякий раз получается соответствующий вектор y^\alpha.

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^\alpha, в результате формируется выходной образ \widetilde y^\alpha\ne y^\alpha. |
| Шаг 2 | Вычисляется вектор ошибки \delta ^\alpha=(y^\alpha-\widetilde 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 перцептрона такой вектор x, для которого уже известен правильный ответ. Если выходной сигнал перцептрона совпадает с правильным ответом, то никаких действий предпринимать не надо. В случае ошибки, необходимо обучить перцептрон правильно решать данный пример. Ошибки могут быть двух типов. Рассмотрим каждый из них.

Первый тип ошибки: на выходе перцептрона — 0, а правильный ответ — 1. Для того чтобы перцептрон 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 xi, которые равны 1.

Первое правило. Если на выходе перцептрона получен 0, а правильный ответ равен 1, то необходимо увеличить веса связей между одновременно активными нейронами. При этом выходной перцептрон считается активным. Второй тип ошибки: на выходе перцептрона — 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 перцептрона 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.

Не рассмотренными 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 перцептрона. Если существует вектор параметров w, при котором перцептрон правильно решает все примеры обучающей выборки, то при обучении перцептрона по вышеописанному алгоритму решение будет найдено за конечное число шагов.

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

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

Доказательства этих теорем в данное учебное пособие не включены.

Целочисленность весов перцептронов

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

Теорема. Любой перцептрон можно заменить другим перцептроном 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_ix_i, \quad
s_1=\min_{x\in X_1}\sum_i w_ix_i.

Определим допуск s как минимум из s_0 и s_1. Положим \delta=s/(m+1), где m — число слагаемых в (1). Поскольку перцептрон (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' перцептрон будет работать с теми же результатами, что и первоначальный перцептрон. Действительно, если правильным ответом примера является 0, имеем \sum_i w_ix_i\le -s.

Подставив новые веса, получим:

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

Откуда следует необходимое неравенство

|  |  |
| --- | --- |
| \sum_i w_i'x_i<0. | ( 2) |

Аналогично, в случае правильного ответа равного 1, имеем \sum_i
w_ix_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\ge s-\sum_i|w_i'-
w_i|x_i\ge\\
\ge s-\sum_i|w_i'-w_i|>s-(m+1)\delta=0.

Отсюда следует выполнение неравенства

|  |  |
| --- | --- |
| \sum_i w_i' x_i>0. | ( 3) |

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

|  |  |
| --- | --- |
| \psi=\Bigl[\sum_{i=1}^m w_i x_i>0\Bigr]= \Bigl[\sum_{i=1}^m w_i' x_i>0\Bigr]= \Bigl[\sum_{i=1}^m \frac{w_i''}{r_i} x_i>0\Bigr], | ( 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=\Bigl[\sum_{i=1}^m w_i x_i>0\Bigr]=
\Bigl[\sum_{i=1}^m w_i' x_i>0\Bigr]=
\Bigl[\sum_{i=1}^m \frac{w_i''}{r_i} x_i>0\Bigr]=
\Bigl[\sum_{i=1}^m w_i''' x_i>0\Bigr],

что и завершает доказательство теоремы.

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

Двуслойность перцептрона

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

Пусть веса всех слоев перцептрона 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 перцептрона 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 перцептрона к решению, даже при условии, что такое решение существует.

В действительности, проблема настройки (обучения) многослойного перцептрона решается следующей теоремой.

Теорема о двуслойности перцептрона. Любой многослойный перцептрон может быть представлен в виде двуслойного персептрона с необучаемыми весами первого слоя.

Для доказательства этой теоремы потребуется одна теорема из математической логики.

Теорема о дизъюнктивной нормальной форме. Любая булева функция булевых аргументов может быть представлена в виде дизъюнкции конъюнкций элементарных высказываний и отрицаний элементарных высказываний:

f=\vee (\&\; x_i \;\& \;\neg x_j).

Напомним некоторые свойства дизъюнктивной нормальной формы.

Свойство 1. В каждый конъюнктивный член (слагаемое) входят все элементарные высказывания либо в виде самого высказывания, либо в виде его отрицания.

Свойство 2. При любых значениях элементарных высказываний в дизъюнктивной нормальной форме может быть истинным не более одного конъюнктивного члена (слагаемого).

Доказательство теоремы о двуслойности перцептрона. Из теоремы о дизъюнктивной нормальной форме следует, что любой многослойный перцептрон может быть представлен в следующем виде:

|  |  |
| --- | --- |
| \psi=|\vee (\&\; x_i\;\& \neg  x_j)|. | ( 5) |

В силу второго свойства дизъюнктивной нормальной формы, равенство (5) можно переписать в виде

|  |  |
| --- | --- |
| \psi=[\vee(\&\;x_i\;\&\;\neg x_j)]=\left[\sum[(\&\;x_i\;\&\;\neg x_j)]>0\right]. | ( 6) |

Переведем в арифметическую форму все слагаемые в выражении (6). Конъюнкцию заменяем на умножение, а отрицание на разность: \neg x_j=1-
x_j. Произведя эту замену и приведя подобные члены, получим:

|  |  |
| --- | --- |
| \psi=\left[\sum_l \alpha_l\prod_{i\in I_l}x_i>0\right], | ( 7) |

где I_l — множество индексов сомножителей в l -м слагаемом, \alpha_l — число, указывающее, сколько раз такое слагаемое встретилось в выражении (6) после замены и раскрытия скобок (число подобных слагаемых).

Заменим i -е слагаемое в формуле (7) перцептроном следующего вида:

|  |  |
| --- | --- |
| \varphi_i=\prod_{l\in I_l} x_l=\left[\sum_{l\in I_l}x_l>|I_l|-1\right]. | ( 8) |

Подставив выражение (8) в формулу (7), получим равенство (1), то есть произвольный многослойный перцептрон представлен в виде (1) с целочисленными коэффициентами. В качестве перцептронов первого слоя используются перцептроны вида (8) с необучаемыми весами. Теорема доказана.

Подводя итоги данной лекции, следует отметить следующие основные свойства перцептронов:

1. Любой перцептрон может содержать один или два слоя. В случае двухслойного перцептрона веса первого слоя не обучаются.
2. Веса любого перцептрона можно заменить на целочисленные.
3. При обучении после конечного числа итераций возможны два исхода: перцептрон обучится или вектор весов перцептрона будет повторяться ( перцептрон зациклится).

Знание этих свойств позволяет избежать "усовершенствований" типа модификации скорости обучения и других, столь же "эффективных" модернизаций.

Трудности с алгоритмом обучения перцептрона

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 перцептрона 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 перцептронный 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 перцептрона. Эти проблемы являются важной областью современных исследований.

* Просмотр
* [Редактировать](https://moodle.bgpu.ru/mod/lesson/edit.php?id=71563)
* [Отчеты](https://moodle.bgpu.ru/mod/lesson/report.php?id=71563)
* [Оценить эссе](https://moodle.bgpu.ru/mod/lesson/essay.php?id=71563)

Обучение перцептрона

Обучение перцептрона

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 перцептрона и доказал: перцептрон 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 перцептрона является обучением с учителем.

Алгоритм обучения перцептрона 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 перцептрона встречаются во многих сетевых парадигмах.

Алгоритм обучения однослойного перцептрона

Перцептрон 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.

Перцептроном будем называть устройство, вычисляющее следующую систему функций:

|  |  |
| --- | --- |
| \psi=\Bigl[\sum_{i=1}^m w_i x_i>\theta\Bigr],\quad | ( 1) |

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

Обучение перцептрона состоит в подстройке весовых коэффициентов. Пусть имеется набор пар векторов (x^\alpha,
y^\alpha), \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^\alpha на выходах всякий раз получается соответствующий вектор y^\alpha.

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^\alpha, в результате формируется выходной образ \widetilde y^\alpha\ne y^\alpha. |
| Шаг 2 | Вычисляется вектор ошибки \delta ^\alpha=(y^\alpha-\widetilde 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 перцептрона такой вектор x, для которого уже известен правильный ответ. Если выходной сигнал перцептрона совпадает с правильным ответом, то никаких действий предпринимать не надо. В случае ошибки, необходимо обучить перцептрон правильно решать данный пример. Ошибки могут быть двух типов. Рассмотрим каждый из них.

Первый тип ошибки: на выходе перцептрона — 0, а правильный ответ — 1. Для того чтобы перцептрон 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 xi, которые равны 1.

Первое правило. Если на выходе перцептрона получен 0, а правильный ответ равен 1, то необходимо увеличить веса связей между одновременно активными нейронами. При этом выходной перцептрон считается активным. Второй тип ошибки: на выходе перцептрона — 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 перцептрона 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.

Не рассмотренными 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 перцептрона. Если существует вектор параметров w, при котором перцептрон правильно решает все примеры обучающей выборки, то при обучении перцептрона по вышеописанному алгоритму решение будет найдено за конечное число шагов.

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

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

Доказательства этих теорем в данное учебное пособие не включены.

Целочисленность весов перцептронов

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

Теорема. Любой перцептрон можно заменить другим перцептроном 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_ix_i, \quad
s_1=\min_{x\in X_1}\sum_i w_ix_i.

Определим допуск s как минимум из s_0 и s_1. Положим \delta=s/(m+1), где m — число слагаемых в (1). Поскольку перцептрон (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' перцептрон будет работать с теми же результатами, что и первоначальный перцептрон. Действительно, если правильным ответом примера является 0, имеем \sum_i w_ix_i\le -s.

Подставив новые веса, получим:

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

Откуда следует необходимое неравенство

|  |  |
| --- | --- |
| \sum_i w_i'x_i<0. | ( 2) |

Аналогично, в случае правильного ответа равного 1, имеем \sum_i
w_ix_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\ge s-\sum_i|w_i'-
w_i|x_i\ge\\
\ge s-\sum_i|w_i'-w_i|>s-(m+1)\delta=0.

Отсюда следует выполнение неравенства

|  |  |
| --- | --- |
| \sum_i w_i' x_i>0. | ( 3) |

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

|  |  |
| --- | --- |
| \psi=\Bigl[\sum_{i=1}^m w_i x_i>0\Bigr]= \Bigl[\sum_{i=1}^m w_i' x_i>0\Bigr]= \Bigl[\sum_{i=1}^m \frac{w_i''}{r_i} x_i>0\Bigr], | ( 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=\Bigl[\sum_{i=1}^m w_i x_i>0\Bigr]=
\Bigl[\sum_{i=1}^m w_i' x_i>0\Bigr]=
\Bigl[\sum_{i=1}^m \frac{w_i''}{r_i} x_i>0\Bigr]=
\Bigl[\sum_{i=1}^m w_i''' x_i>0\Bigr],

что и завершает доказательство теоремы.

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

Двуслойность перцептрона

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

Пусть веса всех слоев перцептрона 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 перцептрона 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 перцептрона к решению, даже при условии, что такое решение существует.

В действительности, проблема настройки (обучения) многослойного перцептрона решается следующей теоремой.

Теорема о двуслойности перцептрона. Любой многослойный перцептрон может быть представлен в виде двуслойного персептрона с необучаемыми весами первого слоя.

Для доказательства этой теоремы потребуется одна теорема из математической логики.

Теорема о дизъюнктивной нормальной форме. Любая булева функция булевых аргументов может быть представлена в виде дизъюнкции конъюнкций элементарных высказываний и отрицаний элементарных высказываний:

f=\vee (\&\; x_i \;\& \;\neg x_j).

Напомним некоторые свойства дизъюнктивной нормальной формы.

Свойство 1. В каждый конъюнктивный член (слагаемое) входят все элементарные высказывания либо в виде самого высказывания, либо в виде его отрицания.

Свойство 2. При любых значениях элементарных высказываний в дизъюнктивной нормальной форме может быть истинным не более одного конъюнктивного члена (слагаемого).

Доказательство теоремы о двуслойности перцептрона. Из теоремы о дизъюнктивной нормальной форме следует, что любой многослойный перцептрон может быть представлен в следующем виде:

|  |  |
| --- | --- |
| \psi=|\vee (\&\; x_i\;\& \neg  x_j)|. | ( 5) |

В силу второго свойства дизъюнктивной нормальной формы, равенство (5) можно переписать в виде

|  |  |
| --- | --- |
| \psi=[\vee(\&\;x_i\;\&\;\neg x_j)]=\left[\sum[(\&\;x_i\;\&\;\neg x_j)]>0\right]. | ( 6) |

Переведем в арифметическую форму все слагаемые в выражении (6). Конъюнкцию заменяем на умножение, а отрицание на разность: \neg x_j=1-
x_j. Произведя эту замену и приведя подобные члены, получим:

|  |  |
| --- | --- |
| \psi=\left[\sum_l \alpha_l\prod_{i\in I_l}x_i>0\right], | ( 7) |

где I_l — множество индексов сомножителей в l -м слагаемом, \alpha_l — число, указывающее, сколько раз такое слагаемое встретилось в выражении (6) после замены и раскрытия скобок (число подобных слагаемых).

Заменим i -е слагаемое в формуле (7) перцептроном следующего вида:

|  |  |
| --- | --- |
| \varphi_i=\prod_{l\in I_l} x_l=\left[\sum_{l\in I_l}x_l>|I_l|-1\right]. | ( 8) |

Подставив выражение (8) в формулу (7), получим равенство (1), то есть произвольный многослойный перцептрон представлен в виде (1) с целочисленными коэффициентами. В качестве перцептронов первого слоя используются перцептроны вида (8) с необучаемыми весами. Теорема доказана.

Подводя итоги данной лекции, следует отметить следующие основные свойства перцептронов:

1. Любой перцептрон может содержать один или два слоя. В случае двухслойного перцептрона веса первого слоя не обучаются.
2. Веса любого перцептрона можно заменить на целочисленные.
3. При обучении после конечного числа итераций возможны два исхода: перцептрон обучится или вектор весов перцептрона будет повторяться ( перцептрон зациклится).

Знание этих свойств позволяет избежать "усовершенствований" типа модификации скорости обучения и других, столь же "эффективных" модернизаций.

Трудности с алгоритмом обучения перцептрона

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 перцептрона 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 перцептронный 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 перцептрона. Эти проблемы являются важной областью современных исследований