

Персептроны и многослойная архитектура.

Персептрон - это один нейрон, который классифицирует набор входных значений в одну из двух категорий (обычно 1 или -1). Если входные значения представлены в виде сетки, персептрон может использоваться для распознавания визуальных образов фигур. Персептрон обычно использует ступенчатую функцию, которая возвращает 1, если взвешенная сумма входов превышает порог, а 0 - в противном случае.

Персептрон в общем виде.

В основе персептрона лежит математическая модель восприятия информации мозгом. Разные исследователи по-разному его определяют. В самом общем своем виде (как его описывал Розенблатт) он представляет систему из элементов трех разных типов: сенсоров, ассоциативных элементов и реагирующих элементов.

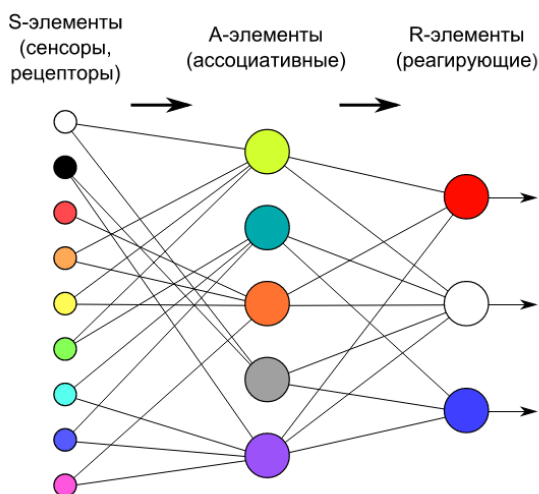


Рис.1 Персептрон в общем виде

Первыми в работу включаются S-элементы. Они могут находиться либо в состоянии покоя (сигнал равен 0), либо в состоянии возбуждения (сигнал равен 1).

Далее сигналы от S-элементов передаются А-элементам по так называемым S-A связям. Эти связи могут иметь веса, равные только -1, 0 или 1.

Затем сигналы от сенсорных элементов, прошедших по S-A связям попадают в А-элементы, которые еще называют ассоциативными элементами. Стоит заметить, что одному А-элементу может соответствовать несколько S-элементов. Если сигналы,

поступившие на А-элемент, в совокупности превышают некоторый его порог θ , то этот А-элемент возбуждается и выдает сигнал, равный 1. В противном случае (сигнал от S-элементов не превысил порога А-элемента), генерируется нулевой сигнал.

Почему А-элементы назвали ассоциативными? Дело в том, что А-элементы являются агрегаторами сигналов от сенсорных элементов. Например, у нас есть группа сенсоров, каждый из которых распознает кусок буквы «Д» на исследуемой картинке. Однако только их совокупность (то есть когда несколько сенсоров выдали сигнал, равный 1) может возбудить А-элемент целиком. На другие буквы А-элемент не реагирует, только на букву «Д». То есть он **ассоциируется** с буквой «Д». Отсюда и такое название.

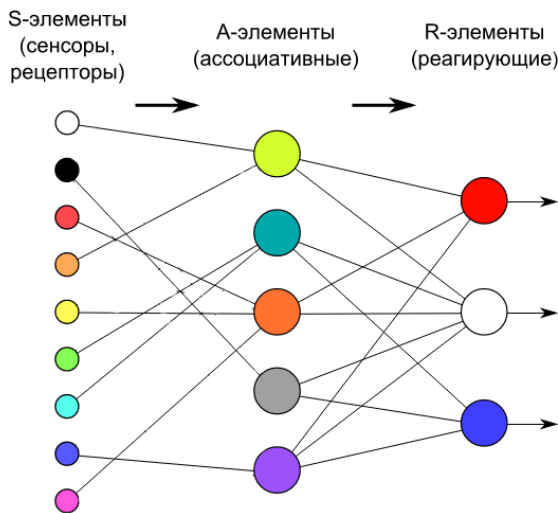
Далее сигналы, которые произвели возбужденные А-элементы, направляются к сумматору (R-элемент). Однако, чтобы добраться до R-элемента, они проходят по А-R связям, у которых тоже есть веса. Однако здесь они уже могут принимать любые значения (в отличие от S-A связей).

R-элемент складывает друг с другом взвешенные сигналы от A-элементов и, если превышен определенный порог, генерирует выходной сигнал, равный 1. Это означает, что в общем потоке информации мы распознали букву «Д».

Если порог не превышен, то выход персептрона равен -1. То есть это не буква «Д».

Так как R-элемент определяет выход персептрона в целом, его называли реагирующим.

Однослойный персептрон



Ключевая особенность состоит в том, что каждый S-элемент однозначно соответствует одному A-элементу, все S-A связи имеют вес, равный +1, а порог A элементов равен 1.

Рис.2 Однослойный персептрон

Многослойный персептрон

Когда слои персептрона объединены вместе, они образуют многослойную архитектуру, и это дает необходимую сложность обработки нейронной сети. Многослойные персептроны (MLP) являются наиболее широко используемой архитектурой для нейронных сетей.

Под многослойным персептроном понимают два разных вида: многослойный персептрон по Розенблатту и многослойный персептрон по Румельхарту.

Многослойный персептрон по Розенблатту — персептрон, у которого имеется более 1 слоя A-элементов.

Многослойный персептрон по Румельхарту — многослойный персептрон по Розенблатту, у которого обучению подлежат еще и S-A связи, а также само обучение производится по методу обратного распространения ошибки.

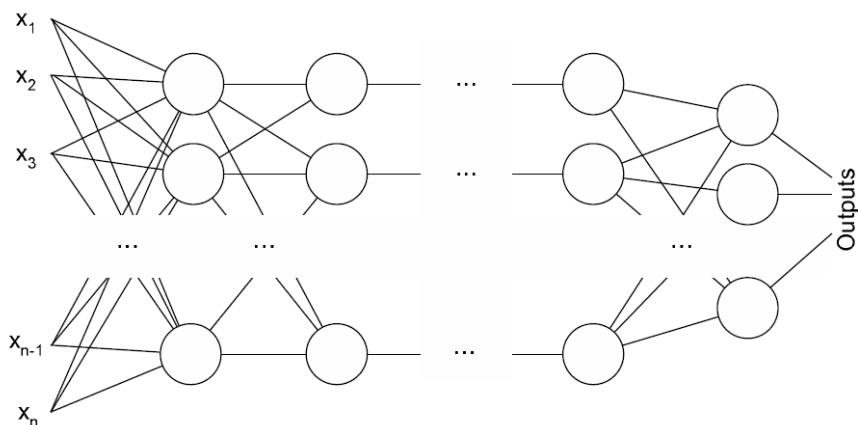


Рис.3 Многослойный персептрон

Распространение прямое и обратное.

Обработка от входного уровня до скрытого слоя (ев), а затем до выходного уровня называется прямым распространением. Сумма (вход * вес) + смещение применяется на каждом уровне, а затем значение функции активации распространяется на следующий уровень. Следующий слой может быть другим скрытым слоем или выходным слоем. В построении нейронных сетей используется большое количество скрытых слоев для создания Deep Neural Network (DNN).

Как только выход достигнут, на последнем слое (выходном слое) мы вычисляем ошибку (предсказанный выход за вычетом исходного вывода). Эта ошибка необходима для коррекции весов и смещений, используемых при прямом распространении. Здесь используется производная функция. Количество веса, которое необходимо изменить, определяется градиентным спуском.

В процессе обратного распространения используется частная производная каждой функции активации нейрона для определения наклона (или градиента) в направлении каждого из входящих весов. Градиент показывает, как круто ошибка будет уменьшена или увеличена для изменения веса. Обратное распространение продолжает изменять веса, пока не будет сильно уменьшено количество ошибок на, известное как скорость обучения.

Скорость обучения - это скалярный параметр, аналогичный размеру шага в численном интегрировании, который используется для установки скорости корректировок для более быстрого уменьшения ошибок. Скорость обучения используется при обратном распространении при корректировке весов и смещений.

Пошаговая иллюстрация нейронной сети и функции активации

Входной слой имеет один нейрон, и выход будет решать двоичную задачу классификации (предсказать 0 или 1).

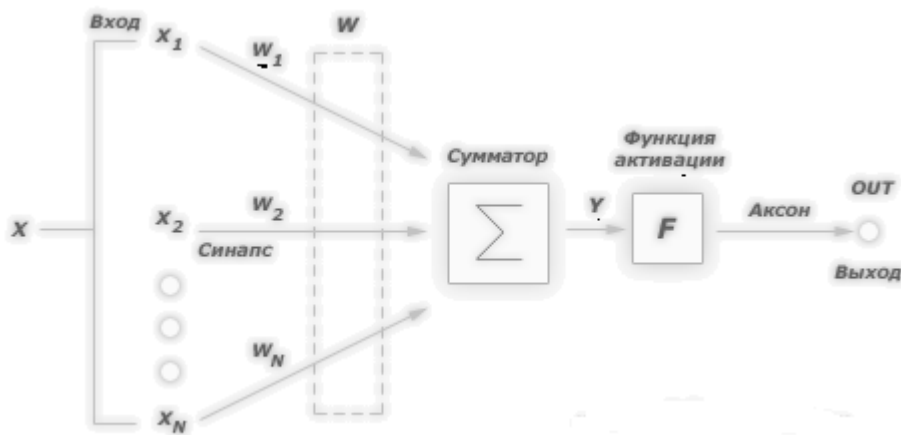
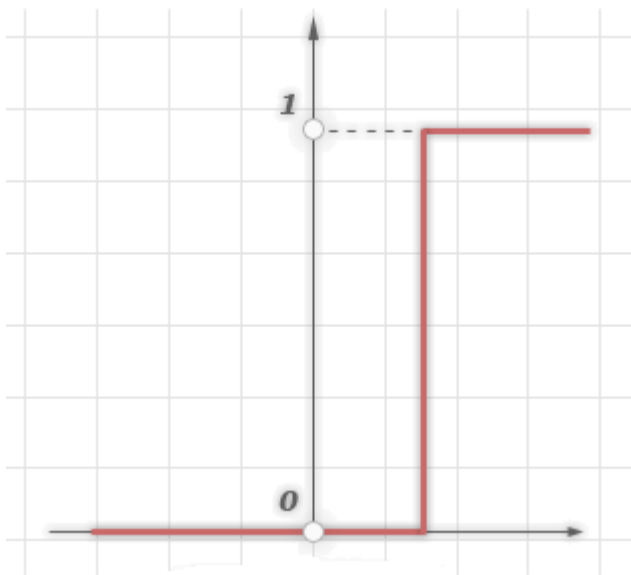


Рис.4 Модель нейрона

Функция активации (активационная функция, функция возбуждения) – функция, вычисляющая выходной сигнал искусственного нейрона. В качестве аргумента принимает сигнал Y , получаемый на выходе входного сумматора Σ . Наиболее часто используются следующие функции активации.

1. Единичный скачок или жесткая пороговая функция



Простая кусочно-линейная функция. Если входное значение меньше порогового, то значение функции активации равно минимальному допустимому, иначе – максимально допустимому.

Рис.5 Кусочно-линейная функция

2. Линейный порог или гистерезис

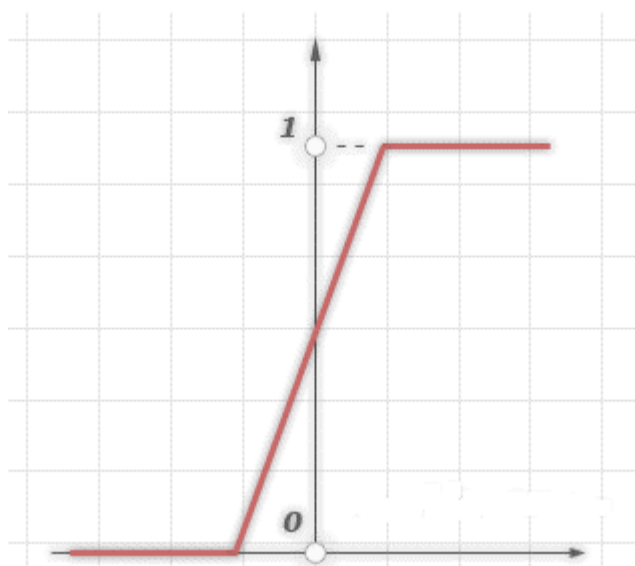


Рис.6 Несложно кусочно-линейная функция

3. Сигмоидальная функция или сигмоид

Моноotonно возрастающая всюду дифференцируемая S -образная нелинейная функция с насыщением. Сигмоид позволяет усиливать слабые сигналы и не насыщаться от сильных сигналов. Гроссберг (1973 год) обнаружил, что подобная нелинейная функция активации решает поставленную им дилемму шумового насыщения.

Слабые сигналы нуждаются в большом сетевом усилении, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления могут привести к насыщению выхода шумами усилителей, которые присутствуют в любой физически реализованной сети. Сильные входные сигналы в свою очередь также будут приводить к насыщению усилительных каскадов, исключая возможность полезного использования выхода.

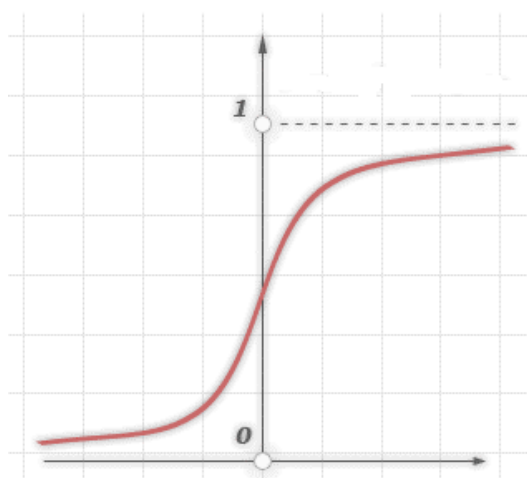


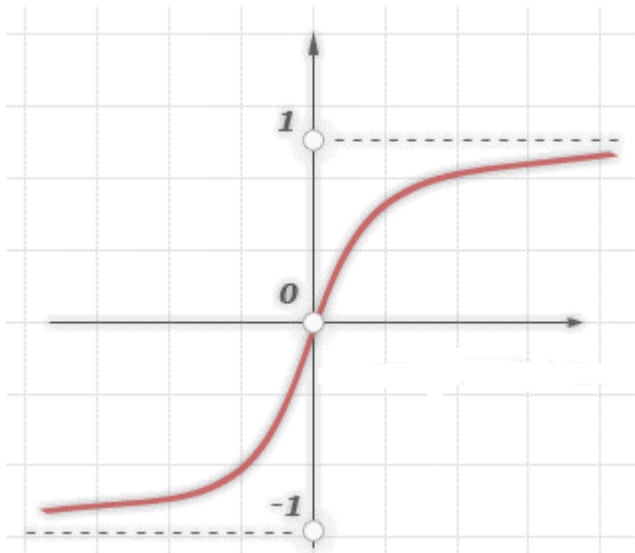
Рис.7 Несложно кусочно-линейная функция

Несложная кусочно-линейная функция. Имеет два линейных участка, где функция активации тождественно равна минимально допустимому и максимально допустимому значению и есть участок, на котором функция строго монотонно возрастает.

Примером сигмоидальной функции активации может служить **логистическая функция**, задаваемая следующим выражением:

$$OUT = \frac{1}{1 + \exp(-\alpha Y)}$$

где α – параметр наклона сигмоидальной функции активации. Изменяя этот параметр, можно построить функции с различной крутизной.



Еще одним примером сигмоидальной функции активации является **гиперболический тангенс**, задаваемая следующим выражением:

$$OUT = th\left(\frac{Y}{\alpha}\right)$$

где α – это также параметр, влияющий на наклон сигмоидальной функции.

Рис.8 Несложно кусочно-линейная функция

В заключение отметим, что функции активации типа единичного скачка и линейного порога встречаются очень редко и, как правило, используются на учебных примерах. В практических задач почти всегда применяется сигмоидальная функция активации.

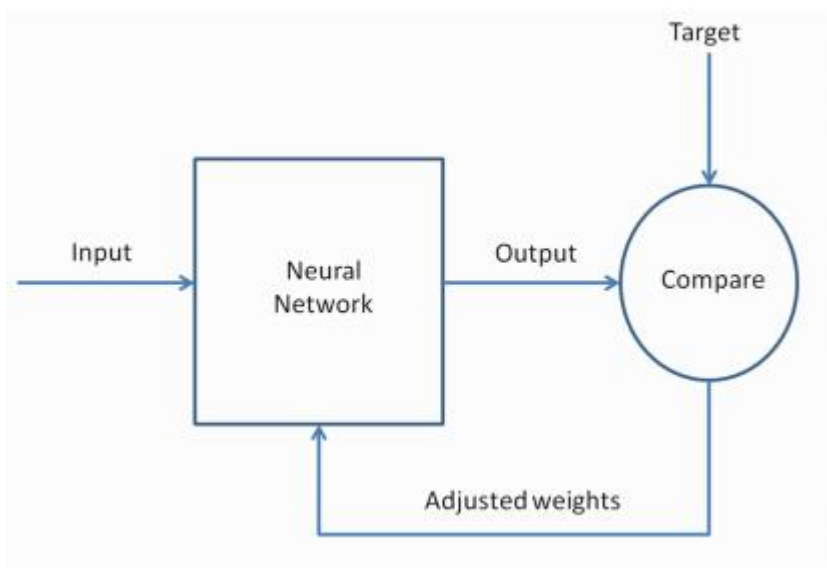


Рис.9 Прямой и обратный проход с одним скрытым слоем

Шаги, которые необходимо выполнить для сетевого обучения:

1. Представите входной сигнал, как матрицу.
2. Инициализировать веса и смещения со случайными значениями. Это один раз. Далее веса и смещения будут обновляться с помощью процесса распространения ошибок.
3. Повторить шаги с 4 по 9 для каждого шаблона обучения (представленного в случайном порядке), пока ошибка не будет сведена к минимуму.
4. Применить входы к сети.

5. Вычислить выход для каждого нейрона из входного слоя, через скрытые слой (и) к выходному слою.

6. Вычислить ошибку на выходах: фактический минус прогнозируемый.

7. Использовать ошибку вывода для вычисления сигналов ошибок для предыдущих уровней.

Производная от функции активации используется для вычисления сигналов ошибки.

8. Использовать сигналы ошибки, чтобы вычислить регулировки веса.

9. Применить регулировки веса.

Этапы 4 и 5 являются прямым распространением, а этапы с 6 по 9 являются обратным распространением.

Скорость обучения - это количество, которое обновляется весом, контролируется параметром конфигурации.

Полный проход назад и вперед называется циклом обучения или эпохой. Обновленные веса и предубеждения используются в следующем цикле. Нужно рекурсивно тренировать, пока ошибка не будет минимальной.

Нейронные сети прямого и обратного распространения

Поток сигналов в нейронных сетях может быть либо в одном направлении, либо в повторении. В первом случае мы называем динамическую структуру нейронной сети, поскольку входные сигналы подаются во входной уровень, а затем после обработки они перенаправляются на следующий уровень, как показано на следующем рисунке. MLP и базовые функции также являются хорошими примерами сетей передачи данных. На рисунке 10 показана архитектура MLP:

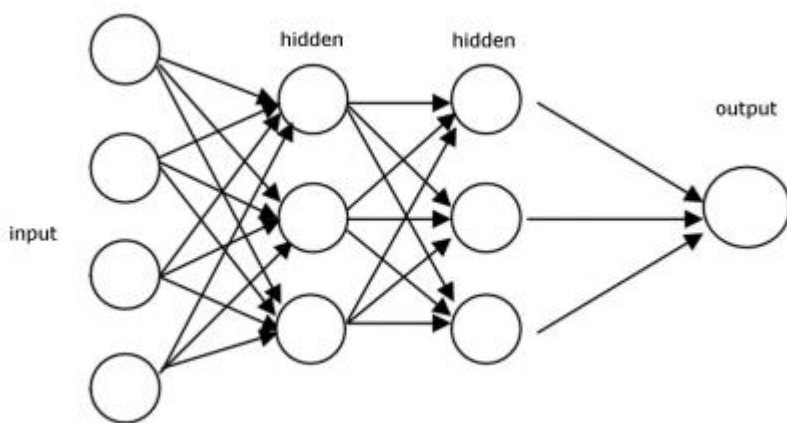
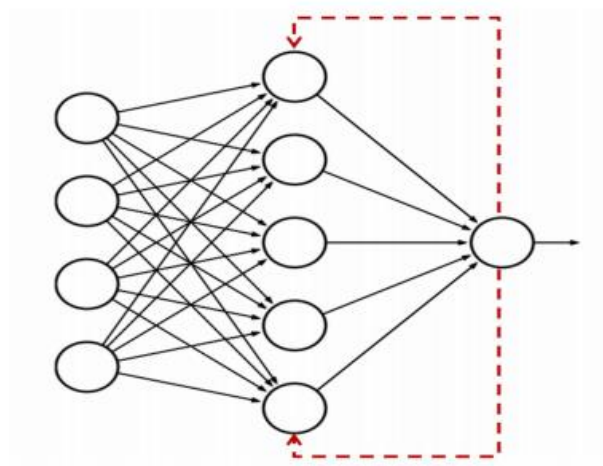


Рис.10 Нейронная сеть прямого распространения



Когда нейронная сеть имеет какой-то внутренний повтор, что означает, что сигналы возвращаются к нейрону или уровню, который уже получил и обработал этот сигнал, сеть имеет обратную связь типа, как показано на рисунке 11.

Особенной причиной добавления повторения в сети

Рис.10 Нейронная сеть обратного распространения

является создание динамического поведения, особенно когда сеть решает проблемы, связанные с временными рядами или распознаванием образов, для которых требуется внутренняя память, чтобы укрепить процесс обучения. Однако такие сети особенно трудны для обучения, в конечном счете неспособные учиться. Большинство сетей обратной связи являются одноуровневыми, такими как сети Эльмана и Хопфилда, но можно построить повторяющуюся многоуровневую сеть, такую как эхо-и повторяющиеся сети MLP.

Источники:

1. <https://edu.kpfu.ru/mod/resource/view.php?id=174779>
2. <http://www.aiportal.ru/articles/neural-networks/multi-perceptron.html>
3. <http://robocraft.ru/blog/algorithm/560.html>
4. <https://neuralnet.info/>