

Лабораторная работа № 3

Многослойные искусственные нейронные сети.

Алгоритм обратного распространения ошибки

Вероятно, архитектура многослойных нейронных сетей используется сейчас наиболее часто. Она была предложена еще в работах Розенблатта и подробно обсуждается почти во всех учебниках по нейронным сетям. Обычно сеть состоит из множества сенсорных элементов (входных узлов), которые образуют входной слой; одного или нескольких скрытых слоев вычислительных нейронов и одного выходного слоя нейронов.

В литературе нет единообразия относительно того, как считать число слоев в многослойных нейронных сетях. Одни предлагают считать число слоев, включая несуммирующий входной слой, другие – считать, только слои, выполняющие суммирование. В данной работе предлагается использовать последнее определение. Согласно этому определению, многослойная нейронная сеть на рисунке 3.1 рассматривается как двухслойная. Вход распределительного слоя считается нулевым слоем.

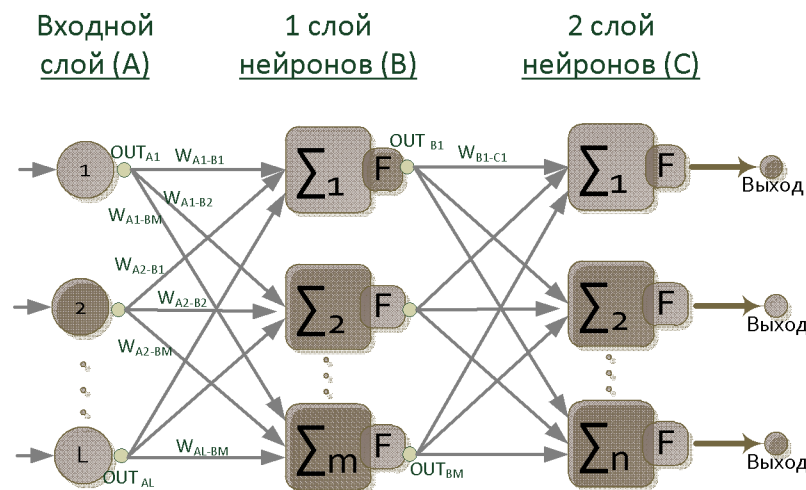


Рисунок 3.1 – Пример двухслойного персептрона

Многослойная нейронная сеть может моделировать функцию практически любой степени сложности, причем число слоев и число элементов в каждом слое определяют сложность функции. Определение числа промежуточных слоев и числа элементов в них является важным вопросом при конструировании.

Многослойными персептронами называют нейронные сети прямого распространения. Входной сигнал в таких сетях распространяется в прямом направлении, от слоя к слою. Многослойный персептрон в общем представлении состоит из следующих элементов:

- множества входных узлов, которые образуют входной слой;
- одного или нескольких скрытых слоев вычислительных нейронов;
- одного выходного слоя нейронов.

Многослойный персептрон представляет собой обобщение однослойного персептрона Розенблатта.

Количество входных и выходных элементов в многослойном персептроне определяется условиями задачи. Сомнения могут возникнуть в отношении того, какие входные значения использовать, а какие нет. Вопрос о том, сколько использовать промежуточных слоев и элементов в них, пока совершенно неясен. В качестве начального приближения можно взять один промежуточный слой, а число элементов в нем положить равным полусумме числа входных и выходных элементов.

Многослойные персептроны успешно применяются для решения разнообразных сложных задач и имеют следующих три отличительных признака.

Свойство 1. Каждый нейрон сети имеет нелинейную функцию активации

Важно подчеркнуть, что такая нелинейная функция должна быть гладкой (т.е. всюду дифференцируемой), в отличие от жесткой пороговой функции, используемой в персептроне Розенблатта. Самой популярной формой функции, удовлетворяющей этому требованию, является

сигмоидальная. Примером. Наличие нелинейности играет очень важную роль, так как в противном случае отображение «вход-выход» сети можно свести к обычному однослойному персептрону.

Свойство 2. Несколько скрытых слоев

Многослойный персептрон содержит один или несколько слоев скрытых нейронов, не являющихся частью входа или выхода сети. Эти нейроны позволяют сети обучаться решению сложных задач, последовательно извлекая наиболее важные признаки из входного образа.

Свойство 3. Высокая связность

Многослойный персептрон обладает высокой степенью связности, реализуемой посредством синаптических соединений. Изменение уровня связности сети требует изменения множества синаптических соединений или их весовых коэффициентов.

Комбинация всех этих свойств наряду со способностью к обучению на собственном опыте обеспечивает вычислительную мощность многослойного персептрона.

Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки является одним из методов обучения многослойных нейронных сетей прямого распространения, называемых также многослойными персептронами. Многослойные персептроны успешно применяются для решения многих сложных задач.

Обучение алгоритмом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе входной вектор подается на входной слой нейронной сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синаптические веса сети фиксированы. Во время обратного прохода все синаптические веса настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синаптических связей. Отсюда и название – алгоритм обратного распространения ошибки. Веса настраиваются с целью максимального приближения выходного сигнала сети к желаемому.

Рассмотрим работу алгоритма подробнее. Допустим необходимо обучить следующую нейронную сеть, применив алгоритм обратного распространения ошибки:

На рисунке 3.1 использованы следующие условные обозначения:

- каждому слою нейронной сети соответствует своя буква, например: входному слою соответствует буква A , а выходному – C ;
- все нейроны каждого слоя пронумерованы арабскими цифрами;
- w_{A1-B1} – синаптический вес между нейронами A_1 и B_1 ;
- OUT_{A1} – выход нейрона A_1 .

В качестве активационной функции в многослойных персептронах, как правило, используется сигмоидальная активационная функция, в частности логистическая:

$$y = F(net) = \frac{1}{1 + e^{-\alpha \cdot net}} \quad (3.1)$$

где α – параметр наклона сигмоидальной функции. Изменяя этот параметр, можно построить функции с различной крутизной. Оговоримся, что для всех последующих рассуждений будет использоваться именно логистическая функция активации, представленная формулой (3.1).

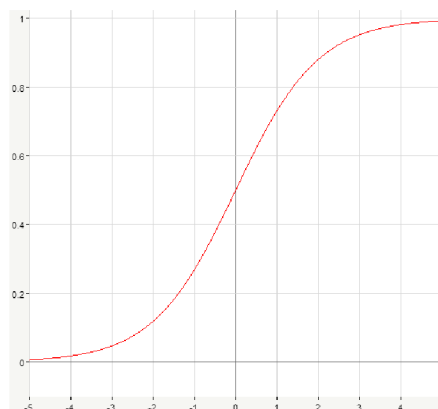


Рисунок 3.2 – Логистическая функция

Сигмоид сужает диапазон изменения так, что значение OUT лежит между нулем и единицей. Многослойные нейронные сети обладают большей представляющей мощностью, чем однослойные, только в случае присутствия нелинейности. Сжимающая функция обеспечивает требуемую нелинейность. В действительности имеется множество функций, которые могли бы быть использованы. Для алгоритма обратного распространения ошибки требуется лишь, чтобы функция была всюду дифференцируема. Сигмоид удовлетворяет этому требованию. Его дополнительное преимущество состоит в автоматическом контроле усиления. Для слабых сигналов (т.е. когда OUT близко к нулю) кривая вход-выход имеет сильный наклон, дающий большое усиление. Когда величина сигнала становится больше, усиление падает. Таким образом, большие сигналы воспринимаются сетью без насыщения, а слабые сигналы проходят по сети без чрезмерного ослабления.

Целью обучения сети алгоритмом обратного распространения ошибки является такая подстройка ее весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов. Для краткости эти множества входов и выходов будут называться векторами. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются обучающей парой. Сеть обучается на многих парах.

Алгоритм обратного распространения ошибки следующий:

1. Инициализировать синаптические веса маленькими случайными значениями.
2. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
3. Вычислить выход сети.
4. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
5. Подкорректировать веса сети для минимизации ошибки (как см. ниже).

Повторять шаги с 2 по 5 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 2 и 3, сходны с теми, которые выполняются при функционировании уже обученной сети, т.е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются послойно. В нейроне, изображенном на рисунке 3.1 сначала вычисляются выходы нейронов слоя B (слой A входной, а значит никаких вычислений в нем не происходит), затем они используются в качестве входов слоя C , вычисляются выходы OUT_{CN} нейронов слоя C , которые и образуют выходной вектор сети OUT . Шаги 2 и 3 образуют так называемый «проход вперед», так как сигнал распространяется по сети от входа к выходу.

Шаги 4 и 5 составляют «обратный проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов.

Рассмотрим подробнее 5 шаг – корректировка весов сети. Здесь следует выделить два нижеописанных случая.

Корректировка синаптических весов выходного слоя

Например, для модели нейронной сети на (рисунок 3.1), это будут веса имеющие следующие обозначения: w_{B1-C1} и w_{B2-C1} . Определимся, что индексом p будем обозначать нейрон, из которого выходит синаптический вес, а q – нейрон в который входит:

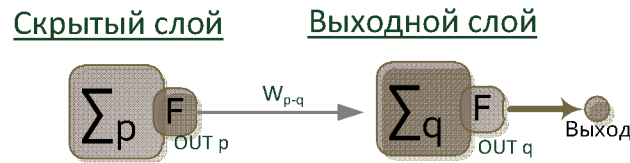


Рисунок 3.3 – Передача сигнала между нейронами многослойного персептрона

Введем величину δ , которая равна разности между требуемым T_q и реальным OUT_q выходами, умноженной на производную логистической функции активации:

$$\delta_q = OUT_q(1 - OUT_q)(T_q - OUT_q)$$

Тогда, веса выходного слоя после коррекции будут равны:

$$w_{p-q}(i+1) = w_{p-q}(i) + \eta \delta_q OUT_p$$

где:

i – номер текущей итерации обучения;

w_{p-q} – величина синаптического веса, соединяющего нейрон p с нейроном q ;

η (греческая буква «эта») – коэффициент «скорости обучения», позволяет управлять средней величиной изменения весов;

OUT_p – выход нейрона p .

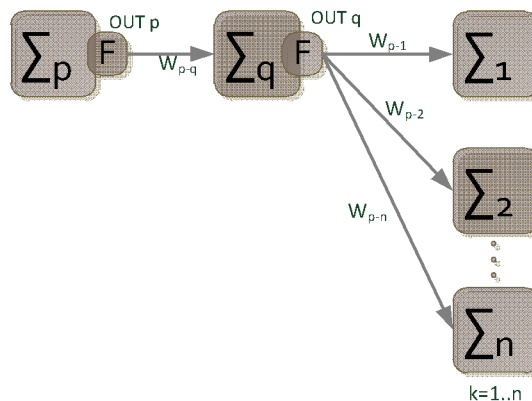
Пример вычислений для синаптического веса w_{B1-C1} :

$$\delta_{c1} = OUT_{c1}(1 - OUT_{c1})(T_1 - OUT_{c1})$$

$$w_{B1-C1}(i+1) = w_{B1-C1}(i) + \eta \delta_{c1} OUT_{B1}$$

Корректировка синаптических весов скрытого слоя

Для модели нейронной сети (рисунок 3.1), это будут веса соответствующие слоям A и B . Определимся, что индексом p будем обозначать нейрон из которого выходит синаптический вес, а q – нейрон в который входит (обратите внимание на появление новой переменной k):



Введем величину δ , которая равна:

$$\delta_q = OUT_q(1 - OUT_q) \sum_{k=1}^M \delta_k w_{q-k}$$

где $\sum_{k=1}^M \delta_k w_{q-k}$ – сумма от 1 по N.

Тогда, веса скрытых слоев после коррекции будут равны:

$$w_{p-q}(i+1) = w_{p-q}(i) + \eta \delta_q OUT_p$$

Пример вычислений для синаптического веса w_{A1-B1} :

$$\begin{aligned} \delta_{B1} &= OUT_{B1}(1 - OUT_{B1}) \\ w_{A1-B1}(i+1) &= w_{A1-B1}(i) + \eta \delta_{B1} OUT_{A1} \end{aligned}$$

Для каждого нейрона в скрытом слое должно быть вычислено δ и подстроены все веса, ассоциированные с этим слоем. Этот процесс повторяется слой за слоем по направлению к входу, пока все веса не будут подкорректированы.

Появление алгоритма обратного распространения ошибки стало знаковым событием в области развития нейронных сетей, так как он реализует вычислительно эффективный метод обучения многослойного персептрона. Будет неправильно утверждать, что алгоритм обратного распространения ошибки предлагает действительно оптимальное решение всех потенциально разрешимым проблем, однако он развеял пессимизм относительно обучения многослойных машин, воцарившийся в результате публикации Минского в 1969 году.

Задание

1. Постройте многослойный персептрон, позволяющий распознавать первые 5 букв вашей фамилии.
2. Предусмотрите возможность обучения нейронной сети еще 2-м буквам, не переучивая персептрон.
3. Приведите топологию созданной нейронной сети. Обоснуйте выбор количества слоев и количества нейронов в каждом слое.
4. Сделайте выводы по результатам работы.