## Лабораторная работа №3. Нейронные сети: обучение без учителя

Рассмотренный в предыдущей работе алгоритм обучения нейронной сети подразумевает наличие некоего внешнего звена, предоставляющего сети кроме входных образов так же и целевые выходные образы. Алгоритмы, пользующиеся подобной концепцией, называются алгоритмами обучения с учителем. Для их успешного функционирования необходимо наличие экспертов, создающих на предварительном этапе для каждого входного образа эталонный выходной. Так как создание искусственного интеллекта движется по пути копирования природных прообразов, ученые не прекращают спор на тему, можно ли считать алгоритмы обучения с учителем натуральными или же они полностью искусственны. Например, обучение человеческого мозга, на первый взгляд, происходит без учителя: на зрительные, слуховые, тактильные и прочие рецепторы поступает информация извне, и внутри нервной системы происходит некая самоорганизация. Однако, нельзя отрицать и того, что в жизни человека не мало учителей – и в буквальном, и в переносном смысле, – которые координируют внешние воздействия. Вместе с тем, чем бы ни закончился спор приверженцев этих двух концепций обучения, они обе имеют право на существование.

**Обучение без учителя. Процедура Хэбба**

Главная черта, делающая обучение без учителя привлекательным, – это его "самостоятельность". Процесс обучения, как и в случае обучения с учителем, заключается в подстраивании весов синапсов. Некоторые алгоритмы, правда, изменяют и структуру сети, то есть количество нейронов и их взаимосвязи, но такие преобразования правильнее назвать более широким термином – самоорганизацией, и в рамках данной статьи они рассматриваться не будут. Очевидно, что подстройка синапсов может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весовых коэффициентов. Исходя из этого соображения и, что более важно, по аналогии с известными принципами самоорганизации нервных клеток [2], построены алгоритмы обучения Хэбба.

Сигнальный метод обучения Хебба заключается в изменении весов по следующему правилу:

 (1)

где *yi(n-1)* – выходное значение нейрона i слоя (n-1), *yj(n)* – выходное значение нейрона j слоя n; *wij(t)* и w*ij*(t-1) – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях t и t‑1 соответственно; α – коэффициент скорости обучения. Здесь и далее, для общности, под n подразумевается произвольный слой сети. При обучении по этому методу усиливаются связи между возбужденными нейронами.

Существует также и дифференциальный метод обучения Хебба.

 (2)

Здесь *yi(n-1)(t)* и *yi(n-1)(t-1)* – выходное значение нейрона i слоя n-1 соответственно на итерациях t и t-1; *yj(n)(t)* и *yj(n)(t-1)* – то же самое для нейрона j слоя n. Как видно из формулы (2), сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения с применением вышеприведенных формул будет выглядеть так:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.
2. На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямопоточных (feedforward) сетей[1], то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется активационная (передаточная) функция нейрона, в результате чего получается его выходное значение *yi(n)*, i=0...Mi-1, где Mi – число нейронов в слое i; n=0...N-1, а N – число слоев в сети.
3. На основании полученных выходных значений нейронов по формуле (1) или (2) производится изменение весовых коэффициентов.
4. Цикл с шага 2, пока выходные значения сети не станут стабильными с заданной точностью.

Применение этого нового способа определения завершения обучения, отличного от использовавшегося для сетей построенных в предыдущей работе, обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

На втором шаге цикла попеременно предъявляются все образы из входного набора.

Следует отметить, что вид откликов на каждый класс входных образов не известен заранее и будет представлять собой произвольное сочетание состояний нейронов выходного слоя, обусловленное случайным распределением весов на стадии инициализации. Вместе с тем, сеть способна обобщать схожие образы, относя их к одному классу. Тестирование обученной сети позволяет определить топологию классов в выходном слое. Для приведения откликов обученной сети к удобному представлению можно дополнить сеть одним слоем, который, например, по алгоритму обучения однослойного персептрона необходимо заставить отображать выходные реакции сети в требуемые образы.

**Алгоритм обучения Кохонена**

Другой алгоритм обучения без учителя – алгоритм Кохонена – предусматривает подстройку синапсов на основании их значений от предыдущей итерации.

 (3)

Из вышеприведенной формулы видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя *yi(n‑1)*, и весовыми коэффициентами его синапсов.

Полный алгоритм обучения с применением вышеприведенных формул будет выглядеть следующим образом:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.
2. На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямопоточных (feedforward) сетей [1], то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется активационная (передаточная) функция нейрона, в результате чего получается его выходное значение *yi(n)*, i=0...Mi-1, где Mi – число нейронов в слое i; n=0...N-1, а N – число слоев в сети.
3. На основании полученных выходных значений из всего слоя выбирается нейрон, значения синапсов которого максимально походят на входной образ (“нейрон-победитель”) и по формуле (3) производится изменение его весовых коэффициентов.
4. Цикл с шага 2, для “нейрона-победителя” пока выходные значения сети не станут стабильными с заданной точностью.

Приведенный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально походят на входной образ, и подстройка весов по формуле (3) проводится только для него. Эта, так называемая, аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение. Выбор такого нейрона может осуществляться, например, расчетом скалярного произведения вектора весовых коэффициентов с вектором входных значений. Максимальное произведение дает выигравший нейрон.

Другой вариант – расчет расстояния между этими векторами в p-мерном пространстве, где p – размер векторов.

, (4)

где j – индекс нейрона в слое n, i – индекс суммирования по нейронам слоя (n-1), *wij* – вес синапса, соединяющего нейроны; выходы нейронов слоя (n-1) являются входными значениями для слоя n. Корень в формуле (4) брать не обязательно, так как важна лишь относительная оценка различных *Dj*.

В данном случае, "побеждает" нейрон с наименьшим расстоянием. Иногда слишком часто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы "уравнять права" всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона.

При использовании алгоритма обучения Кохонена существует практика нормализации входных образов, а так же – на стадии инициализации – и нормализации начальных значений весовых коэффициентов.

, (5)

где *xi* – i-ая компонента вектора входного образа или вектора весовых коэффициентов, а n – его размерность. Это позволяет сократить длительность процесса обучения.

Инициализация весовых коэффициентов случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса. Для избежания такой ситуации используется метод выпуклой комбинации [3]. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

, (6)

где *xi* – i-ая компонента входного образа, n – общее число его компонент, *α(t)* – коэффициент, изменяющийся в процессе обучения от нуля до единицы, в результате чего вначале на входы сети подаются практически одинаковые образы, а с течением времени они все больше сходятся к исходным. Весовые коэффициенты устанавливаются на шаге инициализации равными величине

, (7)

где n – размерность вектора весов для нейронов инициализируемого слоя.

На основе рассмотренного выше метода строятся нейронные сети особого типа – так называемые самоорганизующиеся структуры – self-organizing feature maps (этот устоявшийся перевод с английского не очень удачен, так как, речь идет не об изменении структуры сети, а только о подстройке синапсов). Для них после выбора из слоя n нейрона j с минимальным расстоянием *Dj* (4) обучается по формуле (3) не только этот нейрон, но и его соседи, расположенные в окрестности R. Величина R на первых итерациях очень большая, так что обучаются все нейроны, но с течением времени она уменьшается до нуля. Таким образом, чем ближе конец обучения, тем точнее определяется группа нейронов, отвечающих каждому классу образов.

Необходимо отметить, что обучение без учителя гораздо более чувствительно к выбору оптимальных параметров, нежели обучение с учителем. Во-первых, его качество сильно зависит от начальных величин синапсов. Во-вторых, обучение критично к выбору радиуса обучения и скорости его изменения. И наконец, разумеется, очень важен характер изменения собственно коэффициента обучения. В связи с этим пользователю, скорее всего, потребуется провести предварительную работу по подбору оптимальных параметров обучения сети.

Несмотря на некоторые сложности реализации, алгоритмы обучения без учителя находят обширное и успешное применение. Например, в [4] описана многослойная нейронная сеть, которая по алгоритму самоорганизующейся структуры обучается распознавать рукописные символы. Возникающее после обучения разбиение на классы может в случае необходимости уточняться с помощью обучения с учителем. По сути дела, по алгоритму обучения без учителя функционируют и наиболее сложные из известных на сегодняшний день искусственных нейронных сетей – когнитрон и неокогнитрон, – максимально приблизившиеся в своем воплощении к структуре мозга. Однако они, конечно, существенно отличаются от рассмотренных ними сетей и намного более сложны. Тем не менее, на основе вышеизложенного материала можно создать реально действующие системы для распознавания образов, сжатия информации, автоматизированного управления, экспертных оценок и много другого.

#### Задания для работы в аудитории

* 1. Ознакомится с теоретическими сведениями, изложенными в методическом указании и ответить на контрольные вопросы.
  2. По изложенному в методическом указании алгоритму спроектировать алгоритм и создать программу, реализующую нейросетевой алгоритм обучения без учителя для распознавания четных и нечетных чисел
  3. на основе процедуры Хэбба;
  4. на основе процедуры Кохонена.
  5. Модифицировать алгоритм и программу метода Хэбба, применив дифференциальный алгоритм обучения.

#### Задания для самостоятельной работы

По заданию преподавателя на основе алгоритмов обучения без учителя реализовать НС для разделения пространств различных объектов. Результаты оформить в виде отчета и защитить на следующем занятии.

#### Контрольные вопросы

1. Охарактеризуйте алгоритмы обучения НС с учителем и без учителя.
2. Изложите сигнальную процедуру Хэбба для обучения НС.
3. Изложите дифференциальный метод обучения Хэбба.
4. Каковы недостатки алгоритмов Хэбба и как они преодолеваются?
5. Изложите алгоритм обучения Кохонена.
6. Каковы недостатки алгоритма обучения Кохонена?

#### Литература

1. С.Короткий, Нейронные сети: алгоритм обратного распространения.
2. Ф.Блум, А.Лейзерсон, Л.Хофстедтер, Мозг, разум и поведение, М., Мир, 1988.
3. Ф.Уоссермен, Нейрокомпьютерная техника, М., Мир, 1992.

Keun-Rong Hsieh and Wen-Tsuen Chen, A Neural Network Model which Combines Unsupervised and Supervised Learning, IEEE Trans. on Neural Networks, vol.4, No.2, march 1993.