## Методические указания к лабораторной работе № 3. Технология обучения нейронной сети.

## 1. Правило обучения Хебба

***Самоадаптация*** и ***самоорганизация*** нейронных сетей достигается в процессе их обучения, в ходе которого происходит определение синаптических связей между нейронными элементами. Обучающие правила определяют, как изменяются весовые коэффициенты в ответ на входное воздействие.

Правило обучения Хебба [12], как отмечалось во Введении, имеет биологические предпосылки. Оно является основой многих методов обучения нейронных сетей. Согласно правилу Хебба, обучение происходит в результате усиления силы связи (синаптического веса) между одновременно активными нейронами. Исходя из этого, часто используемые в сети связи усиливаются, что объясняет феномен обучения путем повторения и привыкания.

Пусть имеются два нейронных элемента *i* и *j*, между которыми существует сила связи, равная  (рис. 1.13):



Рис. 1.13. Взаимосвязь двух нейронных элементов.

Тогда правило обучения Хебба можно записать следующим образом:

 (1.6)

где *t*— время, и соответственно выходное значение *i*-го и *j*-го нейрона.

В начальный момент времени предполагается, что

.

Рассмотрим применение правила Хебба для простейшей нейронной сети, состоящей из двух входных и одного выходного нейрона (рис. 1.10). В такой сети порог выходного нейронного элемента является скрытым в этом элементе. При операциях с нейронными сетями порог нейронного элемента можно вынести за его пределы и изобразить, как синаптическую связь (рис. 1.14) с весовым коэффициентом, равным значению.

Так как входное значение, подаваемое на дополнительный нейрон, равняется -1, то взвешенная сумма определяется, как

 (1.7)

Данное выражение является эквивалентным выражению (1.3). Обучение нейронной сети происходит путем настройки весовых коэффициентов и порогов нейронных элементов. Поэтому рассмотренное выше преобразование позволяет настраивать весовые коэффициенты и пороги сети, как единое целое.

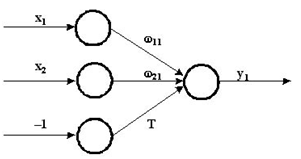


Рис. 1.14. Представление порогового значения в виде синаптической связи.

Правило Хебба для нейронной сети, изображенной на рис. 1.14, можно представить в виде следующих выражений:

 (1.8)

 (1.9)

 (1.10)

Аналогичным образом правило Хебба записывается для нейронной сети большей размерности.

Рассмотрим матричную формулировку правила Хебба. Пусть имеется  входных образов, подаваемых на нейронную сеть, и — размерность одного образа. Тогда совокупность входных образов можно представить в виде следующей матрицы:

,

где соответствует -му входному образу.

Весовые коэффициенты нейронной сети, как отмечалось в разделе 2, можно также представить в виде матрицы. Тогда матрица взвешенной суммы определяется, как

 (1.11)

а матрица выходных значений нейронной сети равняется:

 (1.12)

Правило Хебба в матричной форме можно представить в следующем виде:

 (1.13)

Правило Хебба может использоваться, как при обучении с учителем, так и без него. Если в качестве выходных значений нейронной сети используются эталонные значения, то правило Хебба будет соответствовать обучению с учителем. При использовании в качестве реальных значений, которые получаются при подаче на вход сети входных образов, правило Хебба соответствует обучению без учителя.

В последнем случае весовые коэффициенты нейронной сети в начальный момент времени инициализируются случайным образом. Обучение с использованием правила Хебба заканчивается после подачи всех имеющихся входных образов на нейронную сеть. Следует также отметить, что в общем случае правило Хебба не гарантирует сходимости процедуры обучения нейронной сети.

Рассмотрим решение задачи логическое "ИЛИ" при помощи процедуры обучения Хебба. Будем использовать биполярную кодировку двоичных сигналов. Пусть размерность входного сигнала *n*=2. Тогда общее количество входных образов, подаваемых на нейронную сеть, *L*=4. В таблице 1.1 изображена процедура обучения, соответствующая правилу Хебба.

Таблица 1.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *x1* | *x2* | *x3* | *y* | *w11* | *w21* | *T* |
| **-1** | **-1** | **-1** | **-1** | **1** | **1** | **1** |
| **1** | **-1** | **-1** | **1** | **2** | **0** | **0** |
| **-1** | **1** | **-1** | **1** | **1** | **1** | **-1** |
| **1** | **1** | **-1** | **1** | **2** | **2** | **-2** |

В данной таблице соответствует постоянному сигналу -1, подаваемому на третий вход нейронной сети (рис. 1.14). В данном случае происходит обучение с учителем. Поэтому в качестве используются эталонные значения. В начальный момент времени предполагается, что Обучение происходит согласно выражениям (1.8), (1.9), (1.10) и заканчивается после подачи всех входных образов на сеть. В результате обучения получается следующее уравнение разделяющей линии:



Графическая интерпретация решения задачи логическое "ИЛИ" приведена на рис. 1.15.

Как следует из рисунка, сеть обучилась поставленной задаче. Рассмотрим матричную формулировку процедуры обучения. Тогда

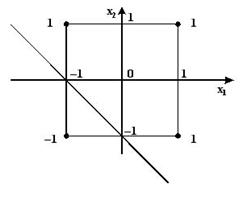


Рис. 1.15. Решение задачи логическое «ИЛИ».

,

что аналогично результату, приведенному в таблице 1.1.

Как уже отмечалось, правило Хебба не гарантирует сходимости процедуры обучения. Так, если взять, например, размерность входного сигнала *n*=5, то процедура обучения Хебба для задачи логического "ИЛИ" не приведет к желаемому результату. Данное правило, в основном, используется в различного рода нейросетевой памяти.

## 2. Процедура обучения Розенблатта

Данную процедуру предложил американский ученый Ф. Розенблатт в 1959 году для нейронной сети, которую он назвал персептроном [17].

***Персептрон*** — это сеть, состоящая из S, A и R нейронных элементов (рис. 1.16). Нейроны слоя S называются ***сенсорными*** и предназначены для формирования входных сигналов в результате внешних воздействий. Нейроны слоя A называются ***ассоциативными*** и предназначены для непосредственной обработки входной информации. Нейроны слоя R называются ***эффекторными*** . Они служат для передачи сигналов возбуждения к соответствующему объекту, например, к мышцам.

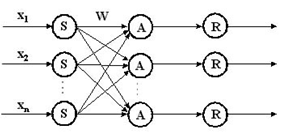


Рис. 1.16. Структура персептрона.

Таким образом, персептрон Розенблатта содержит один слой обрабатывающих нейронных элементов, в качестве которых используются нейронные элементы с пороговой функцией активации. Поэтому обучение персептрона происходит путем настройки весовых коэффициентов W между слоями S и A. В дальнейшем будем использовать стандартную интерпретацию нейронной сети Розенблатта, которая состоит из набора входных нейронов, выполняющих чисто распределительные функции, и набора выходных нейронов с пороговой функцией активации.

Математическую формулировку правила обучения Розенблатта можно   
 представить в следующем виде:

 (1.14)

где *tj* — эталонное значение *j*-го выхода нейронной сети; — характеризует скорость обучения сети.

Величина скорости обучения характеризуется следующими значениями:



Процедура обучения Розенблатта называется алгоритмом обучения с подкреплением. Она характеризуется тем, что весовые коэффициенты нейронной сети изменяются только в том случае, если выходная реакция сети не совпадает с эталонной. Алгоритм обучения Розенблатта состоит из следующих шагов:

1.Весовые коэффициенты нейронной сети инициализируются случайным образом или устанавливаются в нулевое состояние.

2. На входы сети поочередно подаются входные образы из обучающей выборки, которые трансформируются в выходные сигналы нейронных элементов .

3. Если реакция нейронной сети совпадает с эталонным значением , т. е. то весовой коэффициент не изменяется.

4. Если выходная реакция не совпадает с эталонной, т. е. , то производится модификация весовых коэффициентов по следующему правилу:

.

5. Алгоритм продолжается до тех пор, пока не станет для всех входных образов, или не перестанут изменяться весовые коэффициенты.

Согласно теореме сходимости персептрона [19], рассмотренный выше алгоритм сходится за конечное число шагов, если существует решение задачи.

Основные отличия процедуры обучения Розенблатта от правила Хебба заключаются в следующем:

1. В правиле обучения для персептрона присутствует скорость обучения .

2. Персептрон не изменяет весовые коэффициенты, если выходные сигналы совпадают с эталонными.

3. Входные образы из обучающей выборки в модели персептрона подаются до тех пор, пока не произойдет обучение сети.

4. Персептрон обучается за конечное число шагов, если существует решение задачи.

Линейная разделяющая поверхность, формируемая персептроном, ограничивает круг решаемых им задач. Это исследовали в 1969 г. Минский и Пайперт [19], которые показали, что персептрон не может решить задачу "ИСКЛЮЧАЮЩЕЕ ИЛИ". Выводы их по поводу перспектив персептронной модели были весьма пессимистичными. В связи с этим исследования в области нейронных сетей были почти полностью прекращены вплоть до конца 70-х годов.

## 3. Правило обучения Видроу-Хоффа

Используется для обучения нейронной сети, состоящей из распределительных нейронов и одного выходного нейрона, который имеет линейную функцию активации (рис.1.20):

Такая сеть называется адаптивным нейронным элементом или "ADALINE" (Adaptive Linear Element). Его предложили в 1960 г. Видроу (Widrow) и Хофф (Hoff) [18]. Выходное значение такой сети определяется, как

. (1.19)

Правило обучения Видроу Хоффа известно под названием ***дельта правила*** (delta rule). Оно предполагает минимизацию среднеквадратичной ошибки нейронной сети, которая для *L* входных образов определяется следующим образом:

 (1.20)

где — среднеквадратичная ошибка сети для *k*-го образа; и — соответственно выходное и эталонное значение нейронной сети для *k*-го образа.

Критерий (1.20) характеризуется тем, что при малых ошибках ущерб является также малой величиной, т. к. меньше чем величина отклонения . При больших ошибках ущерб возрастает, так как возрастает с ростом величины ошибки.

Среднеквадратичная ошибка нейронной сети для одного входного образа определяется, как

 (1.21)

Правило обучения Видроу-Хоффа базируется на методе градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Согласно этому правилу, весовые коэффициенты и пороги нейронной сети необходимо изменять с течением времени по следующим выражениям:

 (1.22)

 (1.23)

где ; — скорость или шаг обучения. Найдем производные   
среднеквадратичной ошибки по настраиваемым параметрам сети и . Тогда


где *xjk* — *j*-ая компонента *k*-го образа.

Отсюда получаем следующие выражения для обучения нейронной сети по дельта правилу:

 (1.24)

 (1.25)

где .

Выражение (1.24) эквивалентно правилу обучения Розенблатта (1.18), если на выходе линейного адаптивного элемента поставить пороговый элемент. Тогда в выражении (1.24) будет являться результатом нелинейного преобразования взвешенной суммы.

Видроу и Хофф доказали [18], что данный закон обучения всегда позволяет находить весовые коэффициенты нейронного элемента таким образом, чтобы минимизировать среднеквадратичную ошибку сети независимо от начальных значений весовых коэффициентов.

Алгоритм обучения, в основе которого лежит дельта правило состоит из следующих шагов:

1. Задается скорость обучения и минимальная среднеквадратичная ошибка сети , которой необходимо достичь в процессе обучения.

2. Случайным образом инициализируются весовые коэффициенты и порог нейронной сети.

3. Подаются входные образы на нейронную сеть и вычисляются векторы выходной активности сети.

4. Производится изменение весовых коэффициентов и порога нейронной сети согласно выражениям (1.24) и (1.25).

5. Алгоритм продолжается до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной, т. е. .

В алгоритме Видроу-Хоффа существует проблема выбора значения шага обучения . Если коэффициент слишком мал, то процесс обучения является очень длительным. В случае, когда шаг обучения большой, процесс обучения может оказаться расходящимся, то есть не привести к решению задачи. Таким образом сходимость алгоритма обучения не избавляет от разумного выбора значения шага обучения. В некоторых работах [26] предполагается выбирать значение , которое уменьшается в процессе обучения следующим образом:



где *k* — номер итерации в алгоритме обучения. Однако здесь важным фактором является скорость, с которой приближается к нулю. Если скорость слишком велика, то процесс обучения может закончиться до того, как будут получены оптимальные результаты.

Таким образом, временная сложность алгоритма обучения и достижение оптимального решения зависит от выбора величины шага обучения. В следующем разделе этот вопрос будет обсуждаться подробно.