МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

Кафедра систем автоматизированного проектирования
и информационных систем

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ»

для обучающихся по направлению 09.03.02 «Информационные системы и технологии», профиль «Технологии искусственного интеллекта» очной формы обучения

Воронеж 2024

УДК 681.3

Составитель: А.В. Питолин

Методические рекомендации по выполнению лабораторных работ по дисциплине «Нейросетевые технологии» для обучающихся по направлению 09.03.02 «Информационные системы и технологии», профиль «Технологии искусственного интеллекта» очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост.: А.В. Питолин. – Воронеж: Изд-во ВГТУ, 2024. – 22 с.

Приводится описание выполнения лабораторных работ по курсу «Нейросетевые технологии» для обучающихся по направлению 09.03.02. "Информационные системы и технологии", профиль "Технологии искусственного интеллекта" очной формы обучения

УДК 681.3

Рецензент - к.т.н., доцент Королев Е.Н.

Рекомендовано методическим семинаром кафедры САПРИС и методической комиссией ФИТКБ Воронежского государственного технического университета в качестве методических материалов

Лабораторная работа № 1

МОДЕЛИРОВАНИЕ РАБОТЫ ОДНОСЛОЙНОГО ПЕРСЕПТРОНА

1. ОБЩЕЕ ОПИСАНИЕ РАБОТЫ

1.1. Цель лабораторной работы - изучение основных принципов функционирования однослойного персептрона, получение практических навыков по его программной реализации.

1.2. Содержание работы

Лабораторная работа состоит из домашнего и лабораторного заданий. Домашнее задание заключается в изучении основных положений теории искусственных нейронных сетей, принципов функционирования и моделирования однослойного персептрона. Лабораторное задание включает разработку программных средств проектирования однослойного персептрона.

2. ДОМАШНЕЕ ЗАДАНИЕ

Основы теории искусственных нейронных сетей. Принципы функционирования и моделирования персептрона

Основу каждой искусственной нейронной сети (ИНС) составляют относительно простые, в большинстве случаев – однотипные, элементы (ячейки), имитирующие работу нейронов мозга. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рисунке 1.

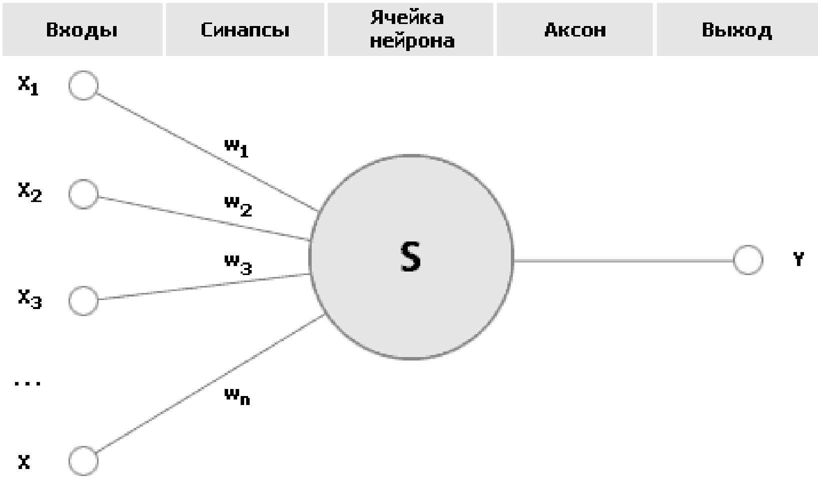

Рис. 1. Искусственный нейрон

Текущее состояние нейрона определяется, как взвешенная сумма его входов:

$$s = \sum_{i=1}^{n} x_i \cdot w_i \quad . \tag{1}$$

Выход нейрона есть функция его состояния:

$$y = f(s) \quad . \tag{2}$$

Нелинейная функция f называется активационной и может иметь различный вид, как показано на рисунке 2.

Одной из наиболее распространеных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т.е. функция S-образного вида):

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \, . \tag{3}$$

При уменьшении α сигмоид становится более пологим, в пределе при α=0 вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке x=0. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне [0,1].

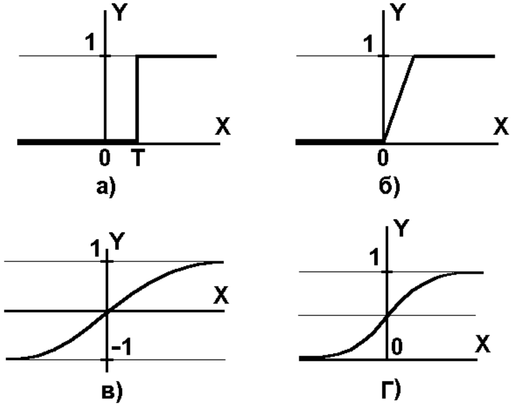

Рис. 2. а) функция единичного скачка; б) линейный порог (гистерезис); в) сигмоид – гиперболический тангенс; г) сигмоид – формула (3)

Одно из ценных свойств сигмоидной функции – простое выражение для ее производной:

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)). \quad (4)$$

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

В качестве примера простейшей ИНС рассмотрим трехнейронный перцептрон (рис.3), то есть такую сеть, нейроны которой имеют активационную функцию в виде единичного скачка. На n входов поступают некие сигналы, проходящие по синапсам на 3 нейрона, образующие единственный слой этой ИНС и выдающие три выходных сигнала:

$$y_j = f\left[\sum_{i=1}^{n} x_i \cdot w_{ij}\right], \quad j=1...3 \quad . \qquad (5)$$

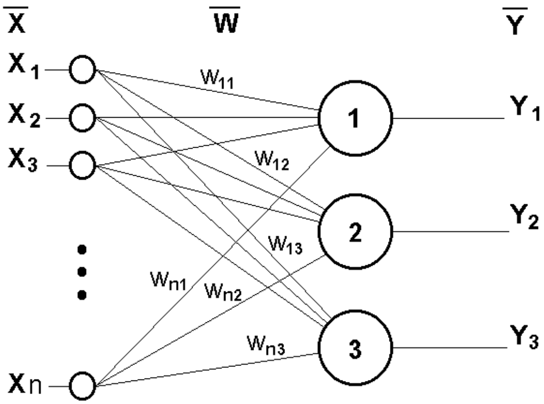

Рис.3. Однослойный перцептрон

Очевидно, что процесс функционирования ИНС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов Этот этап называется обучением ИНС. На этапе обучения кроме параметра качества подбора весов важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса.

Обучение ИНС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы ИНС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует великое множество различных алгоритмов обучения, которые, однако, делятся на два больших класса: детерминистские и стохастические. В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором – она производится на основе действий, подчиняющихся некоторому случайному процессу.

Из рисунка функции единичного скачка видно, что пороговое значение T, в общем случае, может принимать произвольное значение. Более того, оно должно принимать некое произвольное, неизвестное заранее значение, которое подбирается на стадии обучения вместе с весовыми коэффициентами. То же самое относится и к центральной точке сигмоидной зависимости, которая может сдвигаться вправо или влево по оси X, а также и ко всем другим активационным функциям. Это, однако, не отражено в формуле (1), которая должна была бы выглядеть так:

$$s = \sum_{i=1}^{n} x_i \cdot w_i - T \quad . \tag{6}$$

Такое смещение обычно вводится путем добавления к слою нейронов еще одного входа, возбуждающего дополнительный синапс каждого из нейронов, значение которого всегда равняется 1. Присвоим этому входу номер 0. Тогда

$$s = \sum_{i=0}^{n} x_i \cdot w_i \quad , \tag{7}$$

где $w_0 = -T$, $x_0 = 1$.

Очевидно, что различие формул (1) и (6) состоит лишь в способе нумерации входов.

Работа всех сетей сводится к классификации (обобщению) входных сигналов, принадлежащих n-мерному гиперпространству, по некоторому числу классов. С математической точки зрения это происходит путем разбиения гиперпространства гиперплоскостями (запись для случая однослойного перцептрона)

$$\sum_{i=1}^{n} x_i \cdot w_{ik} = T_k \ , \ \text{k=1...m} \quad . \tag{8}$$

Каждая полученная область является областью определения отдельного класса. Число таких классов для одной ИНС перцептронного типа не превышает 2^m, где m – число выходов сети. Однако не все из них могут быть разделимы данной ИНС.

Например, однослойный перцептрон, состоящий из одного нейрона с двумя входами, представленный на рисунке 4, не способен разделить плоскость (двумерное гиперпространоство) на две полуплоскости так, чтобы осуществить классификацию входных сигналов по классам A и B (таблица).

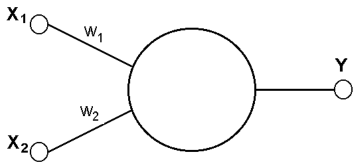

Рис. 4. Однонейронный перцептрон

x_1	x_2	
	0	1
0	A	B
1	B	A

Уравнение сети для этого случая

$$x_1 \cdot w_1 + x_2 \cdot w_2 = T \tag{9}$$

является уравнением прямой (одномерной гиперплоскости), которая ни при каких условиях не может разделить плоскость так, чтобы точки из множества входных сигналов, принадлежащие разным классам, оказались по разные стороны от прямой (рис. 5).

Рассмотрим вопрос обучения ИНС на примере однослойного перцептрона с рисунка 3.

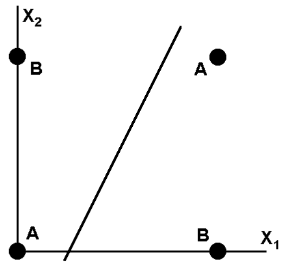

Рис.5. Визуальное представление работы ИНС

Рассмотрим алгоритм обучения с учителем.

1. Проинициализировать элементы весовой матрицы (обычно небольшими случайными значениями).

2. Подать на входы один из входных векторов, которые сеть должна научиться различать, и вычислить ее выход.

3. Если выход правильный, перейти на шаг 4.

Иначе вычислить разницу между идеальным и полученным значениями выхода:

$$\delta = Y_I - Y \quad . \tag{10}$$

Модифицировать веса в соответствии с формулой:

$$w_{ij}(t+1) = w_{ij}(t) + \nu \cdot \delta \cdot x_i \quad , \tag{11}$$

где t и t+1 – номера соответственно текущей и следующей итераций; ν – коэффициент скорости обучения, $0<\nu<1$; i – номер входа; j – номер нейрона в слое.

Очевидно, что если $Y_I > Y$ весовые коэффициенты будут увеличены и тем самым уменьшат ошибку. В противном случае они будут уменьшены, и Y тоже уменьшится, приближаясь к Y_I.

4. Цикл с шага 2, пока сеть не перестанет ошибаться.

3. ЛАБОРАТОРНОЕ ЗАДАНИЕ

Для выполнения лабораторной работы необходимо:

1. Изучить теоретическое введение.
2. Разработать программные процедуры моделирования

функционирования однослойного персептрона и его обучения на основе предложенного алгоритма
3. Протестировать программы на обучающей выборке данных.
4. Провести анализ влияния структуры и параметров сети на скорость обучения.

4. УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА И КОНТРОЛЬНЫЕ ВОПРОСЫ

4.1. Отчет по лабораторной работе должен содержать:
наименование и цель работы;
описание процесса проектирования однослойного персептрона;
листинг программы;
результаты моделирования.

4.2. Контрольные вопросы к лабораторной работе:
1. Объясните термины: нейрон, передаточная функция, входной нейрон, выходной нейрон, нейросеть.
2. Чем отличаются перцептроны с прямыми и обратными связями?
3. Чем отличается обучение с учителем от обучения без учителя?
4. Какие существуют методы обучения нейронных сетей?

Лабораторная работа № 2

ОБУЧЕНИЕ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА С ПОМОЩЬЮ АЛГОРИТМА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

1. ОБЩЕЕ ОПИСАНИЕ РАБОТЫ

1.1. Цель лабораторной работы - изучение моделей функционирования и обучения многослойного персептрона на основе алгоритма обратного распространения ошибки.

1.2. Содержание работы

Лабораторная работа состоит из домашнего и лабораторного заданий. Домашнее задание заключается в теоретическом изучении процесса функционирования многослойного персептрона и процедуры его обучения на основе алгоритма обратного распространения ошибки. Лабораторное задание заключается в разработке программных средств, реализующих алгоритм обратного распространения ошибки.

2. ДОМАШНЕЕ ЗАДАНИЕ

Многослойный персептрон. Алгоритм обратного распространения ошибки

Среди различных структур ИНС одной из наиболее известных является многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми аксонами нейронов предыдущего слоя или, в случае первого слоя, со всеми входами ИНС. Такие ИНС называются полносвязными. Когда в сети только один слой, алгоритм ее обучения с учителем довольно очевиден, так как правильные выходные состояния нейронов единственного слоя заведомо известны, и подстройка синаптических связей идет в направлении, минимизирующем ошибку на выходе сети. По этому принципу строится, например, алгоритм обучения однослойного перцептрона. В многослойных же сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, не известны, и двух или более слойный перцептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах ИНС. Один из вариантов решения этой проблемы – разработка наборов выходных сигналов, соответствующих входным, для каждого слоя ИНС, что, конечно, является очень трудоемкой операцией и не всегда осуществимо. Второй вариант – динамическая подстройка весовых коэффициентов синапсов, в ходе которой выбираются, как правило, наиболее слабые связи и изменяются на малую величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Очевидно, что данный метод "тыка", несмотря на свою кажущуюся простоту, требует громоздких рутинных вычислений. И, наконец, третий, более приемлемый вариант – распространение сигналов ошибки от выходов ИНС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения ИНС получил название процедуры обратного распространения. Именно он будет рассмотрен в дальнейшем.

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки ИНС является величина:

$$E(w) = \frac{1}{2}\sum_{j,p}(y_{j,p}^{(N)} - d_{j,p})^2 \ . \qquad (12)$$

где $y_{j,p}^{(N)}$ – реальное выходное состояние нейрона j выходного слоя N нейронной сети при подаче на ее входы p-го образа; d_{jp} – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad . \tag{13}$$

Здесь w_{ij} – весовой коэффициент синаптической связи, соединяющей i-ый нейрон слоя n-1 с j-ым нейроном слоя n, η – коэффициент скорости обучения, $0<\eta<1$.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{ij}} \quad . \tag{14}$$

Здесь под y_j, как и раньше, подразумевается выход нейрона j, а под s_j – взвешенная сумма его входных сигналов, то есть аргумент активационной функции. Так как множитель dy_j/ds_j является производной этой функции по ее аргументу, из этого следует, что производная активационной функция должна быть определена на всей оси абсцисс. В связи с этим функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых ИНС. В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид с экспонентой. В случае гиперболического тангенса

$$\frac{dy}{ds} = 1 - s^2 . \tag{15}$$

Третий множитель $\partial s_j/\partial w_{ij}$, очевидно, равен выходу нейрона предыдущего слоя $y_i^{(n-1)}$.

Что касается первого множителя в (14), он легко раскладывается следующим образом:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot w_{jk}^{(n+1)} . \tag{16}$$

Здесь суммирование по k выполняется среди нейронов слоя n+1.

Введя новую переменную

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \tag{17}$$

получается рекурсивная формулу для расчетов величин $\delta_j^{(n)}$ слоя n из величин $\delta_k^{(n+1)}$ более старшего слоя n+1.

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j} \quad . \tag{18}$$

Для выходного же слоя

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l} \tag{19}$$

Теперь мы можем записать (13) в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \quad . \tag{20}$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (20) дополняется значением изменения веса на предыдущей итерации

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) \ , \tag{21}$$

где μ – коэффициент инерционности, t – номер текущей итерации.

Таким образом, полный алгоритм обучения ИНС с помощью процедуры обратного распространения строится так:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования ИНС, когда сигналы распространяются от входов к выходам, рассчитать значения последних. Напомним, что

$$s_j^{(n)} = \sum_{i=0}^{M} y_i^{(n-1)} \cdot w_{ij}^{(n)} \ , \tag{22}$$

где M – число нейронов в слое n-1 с учетом нейрона с постоянным выходным состоянием +1, задающего смещение; $y_i^{(n-1)}=x_{ij}^{(n)}$ – i-ый вход нейрона j слоя n.

$$y_j^{(n)} = f(s_j^{(n)}), \text{ где } f() \text{ – сигмоид.} \tag{23}$$

$$y_q^{(0)}=I_q, , \tag{24}$$

где I_q – q-ая компонента вектора входного образа.

2. Рассчитать $\delta^{(N)}$ для выходного слоя по формуле (19).

Рассчитать по формуле (20) или (21) изменения весов $\Delta w^{(N)}$ слоя N.

3. Рассчитать по формулам (18) и (20) (или (18) и (21)) соответственно $\delta^{(n)}$ и $\Delta w^{(n)}$ для всех остальных слоев, n=N-1,...1.

4. Скорректировать все веса в ИНС:

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t)\,. \tag{25}$$

5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Сети на шаге 1 попеременно в случайном порядке предъявляются все тренировочные образы, чтобы сеть, образно говоря, не забывала одни по мере запоминания других.

3. ЛАБОРАТОРНОЕ ЗАДАНИЕ

Для выполнения лабораторной работы необходимо:

1. Изучить теоретические основы функционирования многослойного персептрона и алгоритм обучения с использованием процедуры обратного распространения ошибки.
2. Разработать программные средства моделирования работы многослойного персептрона и его обучения на основе обратного распространения ошибки.
3. Провести тестирование программы на обучающем множестве для решения задачи распознавания буквенных образов.
4. Проанализировать процесс обучения и сделать выводы.

4. УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА И КОНТРОЛЬНЫЕ ВОПРОСЫ

4.1. Отчет по лабораторной работе должен содержать:

наименование и цель работы;

описание алгоритма обучения многослойного персептрона на основе процедуры обратного распространения ошибки;

листинг программы;

результаты проектирования многослойного персептрона..

4.2. Контрольные вопросы к лабораторной работе:

1. В чем заключается обучение нейросети и зачем оно необходимо?
2. К какому классу обучения относится обучение методом обратного распространения ошибки – к обучению без учителя или с учителем?
3. В чем суть метода обратного распространения ошибки?
4. Почему порядок предъявления примеров в обучающей выборке может влиять на качество обучения?
5. В чем преимущества метода обратного распространения ошибки?
6. В чем его недостатки?
7. Как влияет уменьшение количества входных нейронов на функционирование сети?
8. Какими свойствами должна обладать функция активации при использовании алгоритма обратного распространения ошибки?

Лабораторная работа № 3

МОДЕЛИРОВАНИЯ АЛГОРИТМОВ ОБУЧЕНИЯ ИНС БЕЗ УЧИТЕЛЯ

1. ОБЩЕЕ ОПИСАНИЕ РАБОТЫ

1.1. Цель лабораторной работы – изучение и сравнительный анализ алгоритмов обучения ИНС без учителя

1.2. Содержание работы

Лабораторная работа состоит из домашнего и лабораторного заданий. Домашнее задание заключается в теоретическом изучении алгоритмов обучения ИНС без учителя и оценки параметров обучения ИНС. Лабораторное задание заключается в разработке программ моделирования алгоритмов обучения ИНС без учителя.

2. ДОМАШНЕЕ ЗАДАНИЕ

Алгоритмы обучения ИНС без учителя

Рассмотренный ранеее алгоритм обучения ИНС с помощью процедуры обратного распространения подразумевает наличие некоего внешнего звена, предоставляющего сети кроме входных 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. 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, построены алгоритмы обучения Хебба.

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

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot y_i^{(n-1)} \cdot y_j^{(n)} \quad , \tag{26}$$

где $y_i^{(n-1)}$ – выходное значение нейрона i слоя (n-1), $y_j^{(n)}$ – выходное значение нейрона j слоя n; $w_{ij}(t)$ 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 данному методу усиливаются связи между возбужденными нейронами.

Существует также и дифференциальный метод обучения Хебба:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot \left[y_i^{(n-1)}(t) - y_i^{(n-1)}(t-1) \right] \cdot \left[y_j^{(n)}(t) - y_j^{(n)}(t-1) \right]. \tag{27}$$

Здесь $y_i^{(n-1)}(t)$ и $y_i^{(n-1)}(t-1)$ – выходное значение нейрона i слоя n-1 соответственно на итерациях t и t-1; $y_j^{(n)}(t)$ и $y_j^{(n)}(t-1)$ – то же самое для нейрона j слоя n. Как видно из формулы (27), 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, 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 $y_i^{(n)}$, $i=0...M_i-1$, где M_i – число нейронов в слое i; $n=0...N-1$, а N – число слоев в сети.

3. На основании полученных выходных значений нейронов по формуле (26) или (27) производится изменение весовых коэффициентов.

4. Цикл с шага 2, пока выходные значения сети не застабилизируются с заданной точностью. Применение этого нового способа определения завершения обучения, отличного от использовавшегося для сети обратного распространения, 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 перцептрона необходимо заставить отображать выходные реакции сети в требуемые образы.

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

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot \left[y_i^{(n-1)} - w_{ij}(t-1) \right]. \qquad (28)$$

Из вышеприведенной формулы видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $y_i^{(n-1)}$, и весовыми коэффициентами его синапсов.

Полный 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 (28) 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 р-мерном пространстве, где р –

$$D_j = \sqrt{\sum_{i=0}^{p-1} (y_i^{(n-1)} - w_{ij})^2}\ , \qquad (29)$$

где j – индекс нейрона в слое n, i – индекс суммирования по нейронам слоя (n-1), w_{ij} – вес синапса, соединяющего нейроны; выходы нейронов слоя (n-1) являются входными значениями для слоя n. Корень в формуле (29) брать не обязательно, так как важна лишь относительная оценка различных D_j.

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

$$x_i = x_i \,/ \sqrt{\sum_{j=0}^{n-1} x_j^2}\ , \qquad (30)$$

где x_i – 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. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \alpha(t) \cdot x_i + (1 - \alpha(t)) \cdot \frac{1}{\sqrt{n}}, \tag{31}$$

где x_i – i-ая компонента входного образа, n – общее число его компонент, $\alpha(t)$ – 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

$$w_o = \frac{1}{\sqrt{n}}, \tag{32}$$

где n – размерность вектора весов для нейронов инициализируемого слоя.

3. ЛАБОРАТОРНОЕ ЗАДАНИЕ

Для выполнения лабораторной работы необходимо:

1. Изучить теоретические основы проектирования алгоритмов обучения ИНС без учителя.
2. Разработать программные процедуры, реализующие сигнальный и дифференциальный алгоритмы обучения Хебба, алгоритм Кохонена.
3. Провести тестирование разработанных программных средств на выборке данных.
4. Сравнить алгоритмы обучения без учителя по быстродействию и скорости сходимости к эталонам.

4. УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА И КОНТРОЛЬНЫЕ ВОПРОСЫ

4.1. Отчет по лабораторной работе должен содержать:
наименование и цель работы;
описание алгоритмов обучения ИНС без учителя;
листинг программы;
результаты работы программы проектирования алгоритмов обучения ИНС без учителя.

4.2. Контрольные вопросы к лабораторной работе:

1. Для каких целей проводится нормализация вектора входных параметров при обучении по алгоритму Кохонена?
2. Как меняются веса синапсов по правилу обучения Хебба?
3. Объясните в чем суть метода выпуклой комбинации при настройке весов в алгоритме Кохонена?
4. Каковы основные отличия метода интерполяции от метода аккредитации в алгоритме Кохонена?

СОДЕРЖАНИЕ