

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ
по дисциплине «**Введение в искусственные нейронные сети**»
для студентов направления подготовки
02.03.01 «Математика и компьютерные науки»

ВВЕДЕНИЕ

Целью освоения дисциплины является формирование набора профессиональных компетенций будущего бакалавра по направлению подготовки 02.03.01 Математика и компьютерные науки, направленность (профиль) «Анализ данных и искусственный интеллект».

Задачи освоения дисциплины – изучение фундаментальных принципов построения и функционирования искусственных нейронных сетей, ознакомление студентов с основными архитектурами нейросетей, методами их обучения и оптимизации, а также их применением для решения практических задач в профессиональной сфере.

Дисциплина «Введение в искусственные нейронные сети» относится к дисциплинам части, формируемой участниками образовательных отношений. Ее освоение происходит в 7 семестре.

Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций, индикаторов
ПК-3. Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий, программирования и компьютерной техники	ИД-1_{ПК-3} разрабатывает новые математические модели в естественных науках, промышленности и бизнесе с использованием возможностей современных информационных и компьютерных технологий	Умеет применять при решении задач профессиональной деятельности специализированное программное обеспечение и методы искусственных нейронных сетей.
	ИД-2_{ПК-3} способен проводить исследования разработанных математических моделей с помощью современных информационных технологий и компьютерной техники	Способен разрабатывать и обучать основные модули нейросетевых систем, решать практические задачи в предметной области с использованием нейросетевых технологий
ПК-5. Способен принимать участие в	ИД-1_{ПК-5} оценивает свою роль и ответственность при	Готов разрабатывать и внедрять основные моду-

управлении проектами создания информационных систем и программных комплексов на стадиях их жизненного цикла	создании информационных систем	ли искусственных нейронных сетей
	ИД-2_{ПК-5} анализирует проблемы развития вычислительной техники и поддерживает информационные системы на всех стадиях их жизненного цикла	Владеет методами обучения искусственных нейронных сетей и применения их для решения практических задач в области информатики и математики.
ПК-6. Способен учитывать знания проблем и тенденций развития рынка ПО в профессиональной деятельности	ИД-1_{ПК-6} осуществляет выбор программного обеспечения для эффективного решения профессиональных задач	Способен учитывать знания о проблемах и тенденциях развития искусственных нейронных сетей и их применении в профессиональной деятельности, а также о новейших достижениях и разработках в области нейросетевых технологий.
	ИД-2_{ПК-6} применяет программное обеспечение в профессиональной деятельности с учетом современных тенденций его развития	Способен учитывать знания о проблемах и тенденциях развития искусственных нейронных сетей, их применении в профессиональной деятельности и выборе эффективных методов и технологий для решения задач в области нейросетевых систем.

Лабораторная работа № 1

ОДНОСЛОЙНЫЙ ПЕРСЕПТРОН

ЦЕЛЬ РАБОТЫ:

- провести обучение нейронной сети, построенной на принципе однослойного перцептрона, способной распознавать логические функции такие, как **OR** (логическое или), **AND** (логическое и), а также подбирать параметры парной регрессии.
- построить топологию сети;
- построить алгоритмический процесс для нахождения весов сети на основе любого доступного программного средства, такого как Excel, C++, Maple, VBA и др.;
- подобрать параметры процесса обучения.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Перцептрон, или **персептрон** (англ. *perceptron* от лат. *perceptio* – восприятие; нем. *perzeptron*) – математическая и компьютерная модель восприятия информации мозгом (кибернетическая *модель мозга*), предложенная Фрэнком Розенблаттом в 1957 г. и реализованная в виде электронной машины «Марк-1» в 1960 г. Перцептрон стал одной из первых моделей *нейросетей*, а «Марк-1» – первым в мире *нейрокомпьютером*. Несмотря на свою простоту, перцептрон способен обучаться и решать довольно сложные задачи. Основная математическая задача, с которой он справляется, – это линейное разделение любых нелинейных множеств, так называемое обеспечение линейной сепарабельности.

Перцептрон состоит из трёх типов элементов, а именно: поступающие от **сенсоров** сигналы передаются **ассоциативным** элементам, а затем **реагирующим** элементам. Таким образом, перцептроны позволяют создать набор «ассоциаций» между входными стимулами и необходимой реакцией на выходе. В биологическом плане это соответствует преобразованию, например, зрительной информации в физиологический ответ от двигательных нейронов. Согласно современной терминологии, перцептроны могут быть классифицированы как искусственные нейронные сети.

Логическая схема элементарного перцептрона (рис. 1.1). Веса S – A связей могут иметь значения -1 , $+1$ или 0 , т.е. отсутствие связи. Веса A – R связей W могут быть любыми.

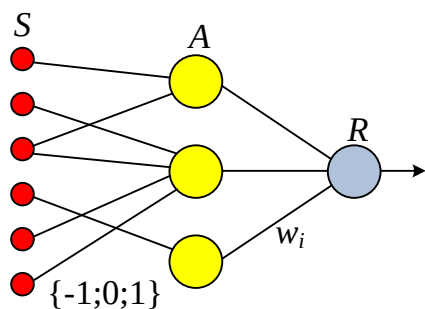


Рис. 1.1. Персептрон с одним выходом

Элементарный перцептрон состоит из элементов трех типов: *S*-элементов, *A*-элементов и *одного R*-элемента. *S*-элементы – это слой сенсоров, или рецепторов. Каждый рецептор может находиться в одном из двух состояний – *покоя* или *возбуждения*, и только в последнем случае он передаёт единичный сигнал в следующий слой, ассоциативным элементам.

A-элементы называются ассоциативными, потому что каждому такому элементу, как правило, соответствует целый набор (ассоциация) *S*-элементов. *A*-элемент активизируется, как только количество сигналов от *S*-элементов на его входе превысило некоторую величину θ . Таким образом, если набор соответствующих *S*-элементов располагается на сенсорном поле в форме буквы «Д», *A*-элемент активизируется, если достаточное количество рецепторов сообщило о появлении «белого пятна света» в их окрестности, то есть *A*-элемент будет как бы ассоциирован с наличием/отсутствием буквы «Д» в некоторой области.

Сигналы от возбуждившихся *A*-элементов, в свою очередь, передаются в сумматор *R*, причём сигнал от *i*-го ассоциативного элемента передаётся с коэффициентом w_i . Этот коэффициент называется *весом A–R* связи.

Так же как и *A*-элементы, *R*-элемент подсчитывает сумму значений входных сигналов, помноженных на веса (линейную форму):

$$net = \sum_{i=0}^n w_i x_i.$$

R-элемент, а вместе с ним и элементарный перцептрон, выдаёт «1», если линейная форма превышает порог θ , иначе на выходе будет «–1». Математически, функцию, реализуемую *R*-элементом, можно записать так:

$$y = \text{sign}(net - \theta) = \text{sign}(\sum_{i=0}^n w_i x_i - \theta). \quad (1.1)$$

Обучение элементарного перцептрона состоит в изменении весовых коэффициентов w_i связей *A–R*. Веса связей *S–A* (которые могут принимать значения $\{-1; 0; +1\}$) и значения порогов *A*-элементов выбираются случайным образом в самом начале и затем не изменяются. Помимо пороговой функции типа (1.1) часто используются их непрерывные аналоги. Например, логистическая функция:

$$f(net) = \frac{1}{1 + \exp(-net)}, \quad (1.2)$$

которая изменяется непрерывно в диапазоне значений $(0; +1)$ и показана на рис. 1.3 или функция тригонометрического тангенса:

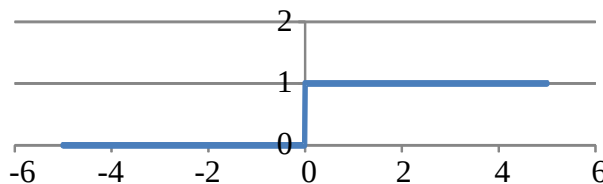


Рис. 1.2. Пороговая функция *sign*

$$f(net) = \tanh(net) = \frac{\exp(net) - \exp(-net)}{\exp(net) + \exp(-net)}, \quad (1.3)$$

изменяющаяся непрерывно в диапазоне значений $(-1; +1)$ и показана на рис. 1.4.

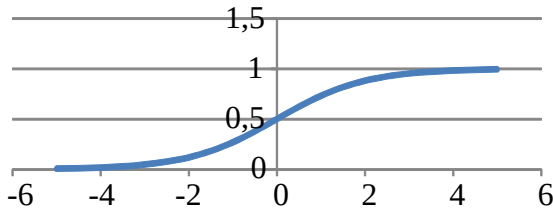


Рис. 1.3. Пороговая функция логистическая

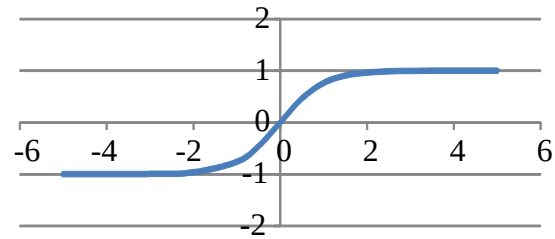


Рис. 1.4. Пороговая функция тригонометрического тангенса

Функция тригонометрического тангенса (1.3) предпочтительней по сравнению с логистической функцией (1.2) в вычислительном процессе, так как нижнее значение ее вместо значения «0» имеет «-1». Это ведет к лучшей работе метода градиентного спуска, поскольку при нулевых значениях переменных $f(x)$ значение градиента также равно нулю и это не приводит к движению алгоритма в сторону оптимума [3].

ОБУЧЕНИЕ НЕЙРОСЕТЕЙ

В основе обучения нейросетей лежит метод градиентной оптимизации – итерационное изменение синаптических весов, постепенно понижающее ошибку обработки нейросетью обучающих примеров. Эффективным методом нахождения этого градиента является так называемый алгоритм обратного распространения ошибки (error back-propagation). Отличия ответов нейросети от правильных, определяемые на выходном слое нейронов, распространяются по сети навстречу потоку сигналов. В итоге каждый нейрон способен определить вклад каждого своего веса в суммарную ошибку сети простым умножением невязки этого нейрона на значение соответствующего входа. Простейшее правило обучения соответствует методу наискорейшего спуска – изменения синаптических весов пропорциональны их (весов) вкладу в общую ошибку.

Можно считать, что при прохождении синапса сила импульса меняется в определенное число раз, которое мы будем называть весом синапса. Импульсы, поступившие к нейрону одновременно по нескольким дендритам, суммируются. Если суммарный импульс превышает некоторый порог, нейрон возбуждается, формирует собственный импульс и передает его далее по аксону. Важно отметить, что веса синапсов могут изменяться со временем, а значит, меняется и поведение соответствующего нейрона.

Нетрудно построить математическую модель описанного процесса. На рисунке изображена модель нейрона с тремя входами (дендритами), причем синапсы этих дендритов имеют веса w_1, w_2, w_3 . Пусть к синапсам поступают импульсы силы x_1, x_2, x_3 соответственно, тогда после прохождения синапсов и дендритов к нейрону поступают импульсы w_1x_1, w_2x_2, w_3x_3 . Нейрон преобразует полученный суммарный импульс $x = w_1x_1 + w_2x_2 + w_3x_3$ в соответствии с некоторой передаточной функцией $f(x)$. Сила выходного импульса равна $y = f(x) = f(w_1x_1 + w_2x_2 + w_3x_3)$.

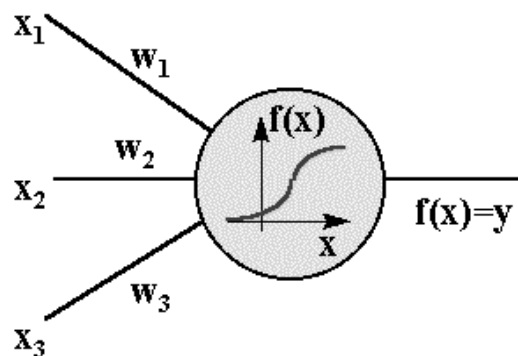


Рис. 1.5. Однослойный перцептрон с одним выходом

Таким образом, нейрон полностью описывается своими весами w_k и передаточной функцией $f(x)$. Получив набор чисел (вектор) x_k в качестве входов, нейрон выдает некоторое число y на выходе.

Классический метод обучения перцептрона – это *метод коррекции ошибки*. Он представляет собой такой вид обучения с учителем, при котором вес связи не изменяется до тех пор, пока текущая реакция перцептрона остается правильной. При появлении неправильной реакции вес изменяется, а знак (+/-) определяется противоположным от знака ошибки.

Допустим, мы хотим обучить перцептрон разделять два класса объектов так, чтобы при предъявлении объектов первого класса выход перцептрона был положителен (+1), а при предъявлении объектов второго класса – отрицательным (–1). Для этого выполним следующий алгоритм.

1. Случайным образом выбираем пороги для А-элементов и устанавливаем связи S–А (далее они изменяться не будут).

2. Начальные коэффициенты w_i полагаем равными случайным числам образом в диапазоне значений $[-0,3; 0,3]$.

3. Предъявляем **обучающую выборку**: объекты с указанием класса, к которым они принадлежат.

Показываем перцептрону объект первого класса. При этом некоторые А-элементы возбуждятся. Коэффициенты w_i , соответствующие этим возбужденным элементам, *увеличиваем* на Δw_i по правилу Видроу–Хоффа или дельта правилу [1, с. 26]:

$$\Delta w_i^{n+1} = \eta \delta x_i,$$

где параметр η называется нормой обучения и выбирается в диапазоне значений $[0, 1]$; а δ есть невязка между выходом сети и значением нашей функции на нашем объекте:

$$\delta = y^0 - f(x_1, x_2).$$

Предъявляем объект второго класса и коэффициенты w_{itex} А-элементов, которые возбуждятся при этом показе, *увеличиваем* на Δw_i .

4. Обе части шага 3 выполним для всей обучающей выборки. В результате обучения сформируются значения весов связей w_i .

Теорема сходимости перцептрона, описанная и доказанная Ф. Розенблаттом (с участием Блока, Джозефа, Кестена и других исследователей, работавших вместе с ним) [7], показывает, что элементарный перцептрон, обучаемый по такому алгоритму, независимо от начального состояния весовых коэффициентов и последовательности появления стимулов *всегда* приведет к достижению решения за конечный промежуток времени.

ПРАКТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Рассмотрим процесс обучения сети на примере логической функции AND(И).

Математический вид этой функции есть: $f(x_1, x_2) = x_1 \& x_2$.

Таблица 1.1

Классический вид функции AND(И)

x_1	x_2	$f(x_1, x_2)$
0	0	0
1	0	0
0	1	0
1	1	1

Таблица 1.2

Видоизмененный вид функции AND(И)

x_1	x_2	$f(x_1, x_2)$
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1

Табличный вид этой функции $f(x_1, x_2)$ представлен в табл. 1.1.

Это классический вид функции AND(И), однако в теории нейронных сетей используется видоизмененный вид этой функции. Вместо значения «0» используется «-1». Это обусловлено необходимостью лучшей работы метода градиентного спуска, поскольку при нулевых значениях переменных x значение градиента также равно нулю и это не приводит к движению алгоритма в сторону оптимума. Поэтому используется видоизмененный вид функции AND(И) (табл. 1.2).

В качестве топологии сети рассмотрим простейшую сеть как на рис. 1.5. Первые два входа сети совпадают с нашими переменными x_1 и x_2 . Переменная x_3 будет равна тождественной единице. В качестве выхода сети возьмем функцию сигнум:

$$y = \text{sign}(w_1 x_1 + w_2 x_2 + w_3 \cdot 1),$$

где функция сигнум есть: $\text{sign}(x) = \begin{cases} 1 & \text{если } x \geq 0 \\ -1 & \text{если } x < 0. \end{cases}$

Рассмотрим итерационный процесс следующего порядка. Пусть первоначальные значения весов w_1, w_2, w_3 будут выбраны случайным образом в диапазоне значений $[-1, 1]$ и равны w_1^0, w_2^0, w_3^0 . Значения переменными x_1 и x_2 выбраны из первой строки таблицы 2. x_3 будет равна тождественной единице. В качестве выхода сети будем использовать функцию сигнум:

$$y^0 = \text{sign}(w_1^0(-1) + w_2^0(-1) + w_3^0 \cdot 1).$$

Тогда невязка между выходом сети и значением нашей функции $f(x_1, x_2)$ есть: $\delta = y^0 - f(x_1, x_2)$.

Для корректировки весов воспользуемся правилом Видроу–Хоффа или дельта правилом [1, с. 26]: $\Delta w_i^{n+1} = \eta \delta x_i$, где параметр η называется нормой обучения и выбирается в диапазоне значений $[0, 1]$.

Повторим этот процесс несколько раз на основе первой строки табл. 1.2, пока значения невязки между выходом сети и значением нашей функции $f(x_1, x_2)$ не достигнет некоторого удовлетворительного уровня точности, например 10^{-5} : $|\delta| \leq 0,00001$.

Перейдем после этого к следующей строки табл. 1.2 и повторим этот процесс, пока значение невязки опять не достигнет некоторого удовлетворительного уровня точности.

Будем повторять этот процесс, до тех пор, пока суммарное значение модуля невязки на всех строках табл. 1.2. не достигнет некоторого удовлетворительного уровня точности.

В файле «Нейросеть конъюнкция и персептрон.xls» на листе «сигнум и» приведен пример построения итерационного процесса для функции AND(И) в программе Excel.

Отдельные фрагменты этой программы показаны в табл. 1.3 и 1.4.

Таблица 1.3

Организация процесса обучения нейронной сети в программе Excel

конъюнкция				норма обуч		nu	0.7			
x1	x2	f(x)	n	w0	w1	w2	сумма	y - сигн	делта	делта^2
-1	-1	-1	1	0.154	0.189	0.372	-0.4068	-1	0	0
-1	1	-1	2	0.154	0.189	0.37151	0.33623	1	-2.00000000	4.00000000
1	-1	-1	3	-1.246	1.589	-1.0285	1.37177	1	-2.00000000	4.00000000
1	1	1	4	-2.646	0.189	0.37151	-2.0852	-1	2.00000000	4.00000000
-1	-1	-1	5	-1.246	1.589	1.77151	-4.6068	-1	0.00000000	0.00000000
-1	1	-1	6	-1.246	1.589	1.77151	-1.0638	-1	0.00000000	0.00000000
1	-1	-1	7	-1.246	1.589	1.77151	-1.4282	-1	0.00000000	0.00000000
1	1	1	8	-1.246	1.589	1.77151	2.1148	1	0.00000000	0.00000000

Организация процесса обучения нейронной сети в программе Excel в формульном виде

	A	B	C	D	E	F	G	H	I	J	K
1	конъюн				норма обуч		nu	0.7			
2	x1	x2	f(x)	n	w0	w1	w2	сумма	y - сигн	делта	делта^2
3	-1	-1	-1	1	=СЛЧИС()*0.5	=СЛЧИС()*0.5	=СЛЧИС()*0.5	=E3+A3*F3+B3*G3	=ЕСЛИ(H3<0;-1;1)	=C3-I3	=J3*J3
4	-1	1	-1	2	=E3+\$J3*\$H\$1	=F3+\$J3*\$H\$1*A3	=G3+\$J3*\$H\$1*B3	=E4+A4*F4+B4*G4	=ЕСЛИ(H4<0;-1;1)	=C4-I4	=J4*J4
5	1	-1	-1	3	=E4+\$J4*\$H\$1	=F4+\$J4*\$H\$1*A4	=G4+\$J4*\$H\$1*B4	=E5+A5*F5+B5*G5	=ЕСЛИ(H5<0;-1;1)	=C5-I5	=J5*J5
6	1	1	1	4	=E5+\$J5*\$H\$1	=F5+\$J5*\$H\$1*A5	=G5+\$J5*\$H\$1*B5	=E6+A6*F6+B6*G6	=ЕСЛИ(H6<0;-1;1)	=C6-I6	=J6*J6
7	-1	-1	-1	5	=E6+\$J6*\$H\$1	=F6+\$J6*\$H\$1*A6	=G6+\$J6*\$H\$1*B6	=E7+A7*F7+B7*G7	=ЕСЛИ(H7<0;-1;1)	=C7-I7	=J7*J7
8	-1	1	-1	6	=E7+\$J7*\$H\$1	=F7+\$J7*\$H\$1*A7	=G7+\$J7*\$H\$1*B7	=E8+A8*F8+B8*G8	=ЕСЛИ(H8<0;-1;1)	=C8-I8	=J8*J8
9	1	-1	-1	7	=E8+\$J8*\$H\$1	=F8+\$J8*\$H\$1*A8	=G8+\$J8*\$H\$1*B8	=E9+A9*F9+B9*G9	=ЕСЛИ(H9<0;-1;1)	=C9-I9	=J9*J9
10	1	1	1	8	=E9+\$J9*\$H\$1	=F9+\$J9*\$H\$1*A9	=G9+\$J9*\$H\$1*B9	=E10+A10*F10+B10*G10	=ЕСЛИ(H10<0;-1;1)	=C10-I10	=J10*J10

На рис. 1.6 показан квадрат невязки δ между выходом сети и значением нашей функции $f(x_1, x_2)$

$$\delta^2 = (y - f(x_1, x_2))^2$$

в зависимости от номера итерации. Видно, что процесс достаточно быстро сходится к нулю. Уже в районе 7 итерации δ достигает малых значений, что подтверждает **теорему сходимости перцептрона**, описанную и доказанную Ф. Розенблаттом (для функции И).

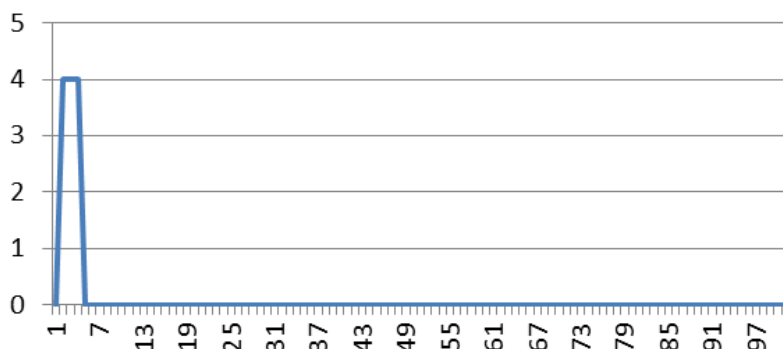


Рис. 1.6. График сходимости итерационного процесса

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Самостоятельно построить итерационный процесс для обучения нейронной сети с функцией AND(И). Повторить процесс обучения сети при различных значениях параметра η - нормы обучения. Выбрать оптимальное значение.
2. Построить графики значений невязки в зависимости от числа итераций процесса. Выбрать оптимальное число итераций.
3. Повторить процесс обучения сети при различных значениях начальных значениях весов.

4. Построить пример итерационного процесса для функции ИЛИ.
5. Построить пример построения итерационного процесса для функции ИЛИ на основе других передаточных функций, например логистической или тригонометрического тангенса.
6. Построить пример итерационного процесса для функции парной регрессии.

Выводы

Рассмотренная в лабораторной работе функция AND(И) является простейшим примером построения итерационного процесса для обучения нейронной сети. Вместе с тем на этом простейшем примере мы рассмотрели такие вопросы как построение и внутреннее функционирование нейронной сети. Рассмотрели построение процесса обучения нейронной сети с учителем. Рассмотрели различные типы передаточных функций. Эти все основные принципы сохраняются в более сложных типах нейронной сети и без их понимания сложно двигаться к более серьезным работам.

Лабораторная работа № 2 ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ. МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

ЦЕЛЬ РАБОТЫ:

- провести обучение нейронной сети, построенной на принципе многослойного персептрона, способной распознавать логические функции такие как **XOR**(исключающее или), на основе метода обратного распространения ошибки;
- построить топологию сети;
- построить алгоритмический процесс для нахождения весов сети на основе любого доступного программного средства, такого как Excel, C++, Maple, VBA и др.;
- подобрать параметры процесса обучения.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Многослойный персептрон (по Розенблатту) – это персептрон, в котором присутствуют дополнительные слои А-элементов. Его анализ провёл Розенблатт в третьей части своей книги.

Многослойный персептрон (по Румельхарту) – это персептрон, в котором присутствуют дополнительные слои А-элементов, причём, обучение такой сети проводится по методу обратного распространения ошибки, и обучаемыми являются все слои персептрона (в том числе S–А). Является частным случаем многослойного персептрона Розенблатта.

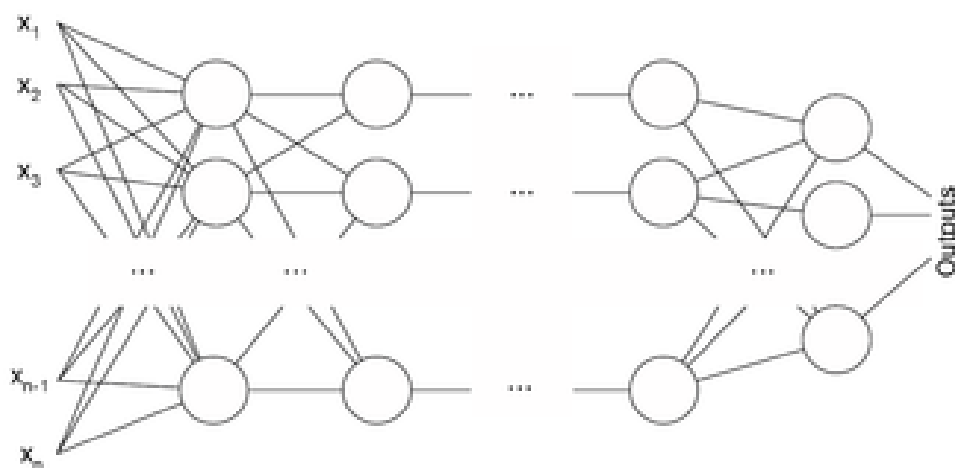


Рис. 2.1. Архитектура многослойного персептрона (обоих подтипов)

Метод обратного распространения ошибки – метод обучения многослойного персептрона. Впервые метод был описан в 1974 г. А.И. Галушкиным, а также независимо и одновременно Полем Дж. Вибросом. Далее существенно развит в 1986 г. Дэвидом И. Румельхартом, Дж. Е. Хинтоном и Рональдом Дж. Вильямсом и независимо и одновременно С.И. Барцевым и В.А. Охониным (Красноярская группа). Это итеративный градиентный алгоритм, который используется с целью минимизации ошибки работы многослойного персептрона и получения желаемого выхода.

Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы

Для возможности применения метода обратного распространения ошибки передаточная функция нейронов должна быть дифференцируема. Метод является модификацией классического метода градиентного спуска.

В тех случаях, когда удастся оценить работу сети, обучение нейронных сетей можно представить как задачу оптимизации. Оценить – означает указать количественно, хорошо или плохо сеть решает поставленные ей задачи. Для этого строится функция оценки. Она, как правило, явно зависит от выходных сигналов сети и неявно (через функционирование) – от всех её параметров. Простейший и самый распространенный пример оценки – сумма квадратов расстояний от выходных сигналов сети до их требуемых значений:

Алгоритм обратного распространения ошибки применяется для многослойного перцептрона. У сети есть множество входов $x_1, \dots, x_n$, множество выходов Outputs и множество внутренних узлов. Перенумеруем все узлы (включая входы и выходы) числами от 1 до N (сквозная нумерация, вне зависимости от топологии слоёв). Обозначим через w_j^i вес, стоящий на ребре, соединяющем i -й и j -й узлы, а через y^i – выход i -го узла. Если нам известен обучающий пример (правильные ответы сети t^k , то функция ошибки, полученная по методу наименьших квадратов, выглядит так:

$$E(\{w_j^i\}) = \frac{1}{2} \sum_{k \in \text{Output}} (y^k - t^k)^2, \quad (2.1)$$

где требуемое значение выходного сигнала на выходе k .

Метод наименьших квадратов далеко не всегда является лучшим выбором оценки. Тщательное конструирование функции оценки позволяет на порядок повысить эффективность обучения сети, а также получать дополнительную информацию – «уровень уверенности» сети в даваемом ответе [6].

Как модифицировать веса? Мы будем реализовывать стохастический градиентный спуск, то есть будем подправлять веса после каждого обучающего примера и, таким образом, «двигаться» в многомерном пространстве весов. Чтобы «добраться» до минимума ошибки, нам нужно «двигаться» в сторону, противоположную градиенту, то есть, на основании каждой группы правильных ответов, добавлять к каждому весу w_j^i

$$\Delta w_j^i = -\eta \frac{\partial E}{\partial w_j^i}, \quad (2.2)$$

где параметр η множитель, задающий скорость «движения» (называется нормой обучения) и выбирается в диапазоне значений $[0, 1]$.

Производная считается следующим образом. Пусть сначала выход $j \in \text{Outputs}$, то есть интересующий нас вес входит в нейрон последнего уровня. Сначала отметим, что w_j^i влияет на выход сети только как часть суммы $S_j = \sum_i w_{ij} x_i$, где сумма берется по входам j -го узла. Поэтому

$$\frac{\partial E}{\partial w_j^i} = \frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial w_j^i} = x_i \frac{\partial E}{\partial S_j}. \quad (2.3)$$

Аналогично, S_j влияет на общую ошибку только в рамках выхода j -го узла y_j (напоминаем, что это выход всей сети).

Поэтому

$$\frac{\partial E}{\partial S_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial S_j} = \frac{\partial (\frac{1}{2} \sum_{k \in \text{Output}} (y^k - t^k)^2)}{\partial y_j} \frac{\partial f}{\partial S_j} = (y^j - t^j) \frac{\partial f}{\partial S_j}. \quad (2.4)$$

Если функция активации нейрона j есть логистическая функция:

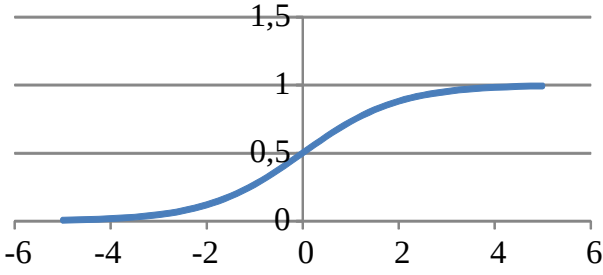


Рис. 2.2. Пороговая функция логистическая

$$f(S) = \frac{1}{1 + \exp(-\beta S)}, \quad (2.5)$$

изменяющаяся непрерывно в диапазоне значений $(0; +1)$ и показана на рис. 2.2, то ее производная есть

$$\frac{\partial f}{\partial S_j} = f(S)(1 - f(S))\beta.$$

И тогда формула (2.4) упрощается

$$\begin{aligned} \frac{\partial E}{\partial S_j} &= (y_j - t_j) \frac{\partial f}{\partial S_j} = (y_j - t_j) f(S)(1 - f(S))\beta = \\ &= (y_j - t_j) y_j(1 - y_j)\beta \end{aligned} \quad (2.6)$$

поскольку $y_j = f(S)$.

И окончательно для $\frac{\partial E}{\partial w_j^i}$ с учетом (2.3) и (2.6) имеем:

$$\frac{\partial E}{\partial w_j^i} = x_i \frac{\partial E}{\partial S_j} = x_i (y_j - t_j) y_j(1 - y_j) \beta. \quad (2.7)$$

Если же j -й узел – не на последнем уровне, то у него есть выходы; обозначим их через $\text{Children}(j)$. В этом случае

$$\begin{aligned} \frac{\partial E}{\partial S_j} &= \sum_{k \in \text{Children}(j)} \frac{\partial E}{\partial S_k} \frac{\partial S_k}{\partial S_j}; \\ \frac{\partial S_k}{\partial S_j} &= \frac{\partial S_k}{\partial y_j} \frac{\partial y_j}{\partial S_j} = W_{kj} \frac{\partial y_j}{\partial S_j} = W_{kj} y_j(1 - y_j) \beta. \end{aligned} \quad (2.8)$$

Ну а $\frac{\partial E}{\partial S_k}$ – это в точности аналогичная поправка, но вычисленная для узла следующего уровня. Будем обозначать её через δ_k – от Δ_k – она отличается отсутствием множителя $-\eta x_{ij}$.

Именно из-за этой особенности вычисления поправок алгоритм называется **алгоритмом обратного распространения ошибки** (backpropagation). Краткое резюме проделанной работы:

- для узла последнего уровня

$$\delta_j = (y_j - t_j) y_j(1 - y_j) \beta; \quad (2.9)$$

- для внутреннего узла сети

$$\delta_j = y_j(1 - y_j) \beta \sum_{k \in \text{Children}(j)} \delta_k w_{jk}; \quad (2.10)$$

- для всех узлов
$$\Delta w_j^i = -\eta \delta_j y_i . \quad (2.11)$$

Получающийся алгоритм представлен ниже. На вход алгоритму, кроме указанных параметров, нужно также подавать в каком-нибудь формате структуру сети. На практике очень хорошие результаты показывают сети достаточно простой структуры, состоящие из двух уровней нейронов – скрытого уровня (hidden units) и нейронов-выходов (output units); каждый вход сети соединен со всеми скрытыми нейронами, а результат работы каждого скрытого нейрона подается на вход каждому из нейронов-выходов. В таком случае достаточно подавать на вход количество нейронов скрытого уровня.

Алгоритм: BackPropagation

1. Инициализировать $\{w_j^i\}_{ij}$ маленькими случайными значениями,
2. Повторить N раз:
Для всех d от 1 до m :
 1. Подать $\{x_i^d\}_{id}$ на вход сети и подсчитать выходы y_j^d каждого узла.
 2. Для каждого узла $j \in \text{Outputs}$ последнего уровня

$$\delta_j = (y_j - t_j) y_j (1 - y_j) \beta . \quad (2.12)$$

3. Для каждого уровня l , начиная с предпоследнего.
Для каждого узла j уровня l вычислить:

$$\delta_j = y_j (1 - y_j) * \beta \sum_{k \in \text{Children}(j)} \delta_k w_{jk} . \quad (2.13)$$

4. Для каждого ребра сети $\{i, j\}$

$$\Delta w_j^i(n) = \alpha \Delta w_j^i(n - 1) + (1 - \alpha) \eta \delta_j y_i^d ; \quad (2.14)$$

$$w_j^i(n) = \Delta w_j^i(n) + w_j^i(n - 1) , \quad (2.15)$$

где α – коэффициент инерциальности для сглаживания резких скачков при перемещении по поверхности целевой функции

Если функция активации нейрона j есть функция тригонометрического тангенса:

$$\begin{aligned} f(S) &= \tanh(\beta S) = \\ &= \frac{\exp(\beta S) - \exp(-\beta S)}{\exp(\beta S) + \exp(-\beta S)}, \end{aligned} \quad (2.16)$$

изменяющаяся непрерывно в диапазоне значений $(-1; +1)$ и показана на рис. 2.3, то ее производная есть

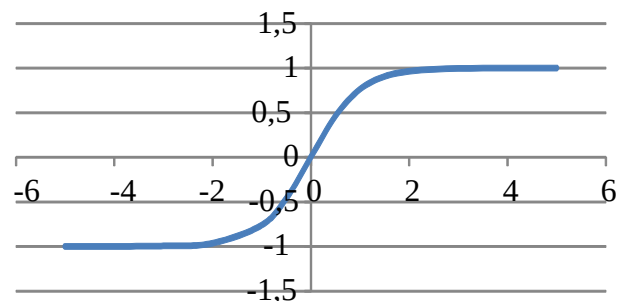


Рис. 2.3. Пороговая функция тригонометрического тангенса

$$\frac{\partial f}{\partial s_j} = (1 - f(S)^2) \beta.$$

И тогда в формулах (2.9) и (2.10), а также в (2.12) и (2.13), член $y_j(1 - y_j) \beta$ заменится на $(1 - y_j^2) \beta$.

Функция тригонометрического тангенса (2.16) предпочтительней по сравнению с логистической функцией (2.5) в вычислительном процессе, так как нижнее значение ее вместо значения «0» имеет «-1». Это ведет к лучшей работе метода градиентного спуска, поскольку при нулевых значениях $f(x)$ значение градиента также равно нулю и это не приводит к движению алгоритма в сторону оптимума [3].

ОБУЧЕНИЕ НЕЙРОСЕТЕЙ

В основе обучения нейросетей лежит метод градиентной оптимизации – итерационное изменение синаптических весов, постепенно понижающее ошибку обработки нейросетью обучающих примеров. Эффективным методом нахождения этого градиента является алгоритм обратного распространения ошибки (error back-propagation). Отличия ответов нейросети от правильных, определяемые на выходном слое нейронов, распространяются по сети навстречу потоку сигналов. В итоге каждый нейрон способен определить вклад каждого своего веса в суммарную ошибку сети простым умножением невязки этого нейрона на значение соответствующего входа. Простейшее правило обучения соответствует методу наискорейшего спуска – изменения синаптических весов пропорциональны их (весов) вкладу в общую ошибку

ПРАКТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Рассмотрим процесс обучения сети с помощью алгоритма обратного распространения ошибки на примере логической функции XOR (исключающее ИЛИ).

Таблица 2.1
**Классический вид
функции XOR**

x_1	x_2	$f(x_1, x_2)$
0	0	0
1	0	1
0	1	1
1	1	0

Математический вид этой функции есть:
 $f(x_1, x_2) = x_1 \text{XOR} x_2$.

Табличный вид этой функции $f(x_1, x_2)$ представлен в виде табл. 2.1.

Это классический вид функции XOR, однако в теории нейронных сетей используется видоизмененный вид этой функции. Вместо значения «0» используется «-1». Это обусловлено необходимостью лучшей

работы метода градиентного спуска, поскольку при нулевых значениях переменных x значение градиента также равно нулю и это не приводит к движению алгоритма в сторону оптимума. Поэтому используется видоизмененный вид функции XOR (табл. 2.2).

Таблица 2.2
**Видоизмененный вид
функции XOR**

x_1	x_2	$f(x_1, x_2)$
-1	-1	-1
1	-1	1
-1	1	1
1	1	-1

Несмотря на схожесть функции XOR с функцией AND по числу аргументов, области определения и размерности результата, простейшая сеть, состоящая из одного нейрона, трех входов и одного выхода (рассматривавшаяся в лабораторной № 1) не способна адекватно аппроксимировать эту функцию. Для правильной аппроксимации функции XOR необходима сеть уже с тремя нейронами и двумя слоями.

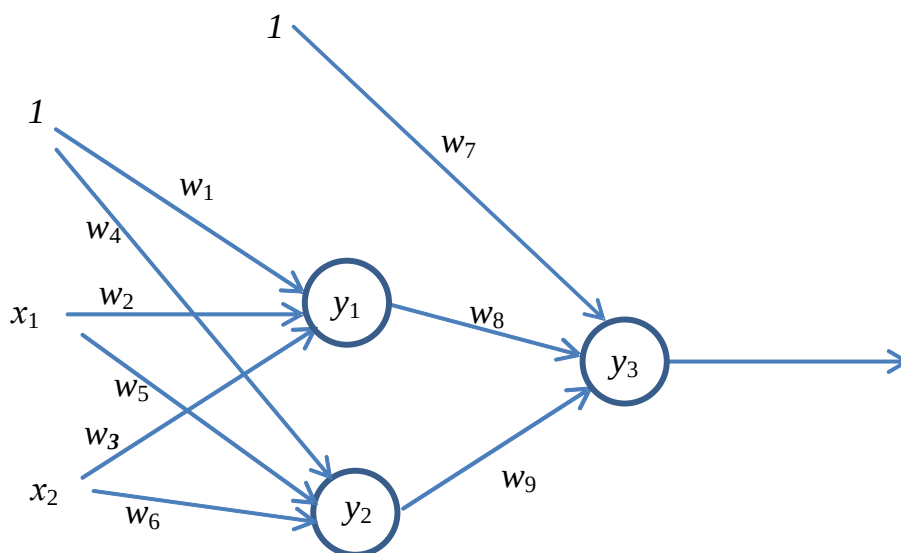


Рис. 2.4. Топология рассматриваемой сети

В качестве топологии сети рассмотрим сеть на рис. 2.4. Два входа сети совпадают с нашими переменными x_1 и x_2 . Переменная x_0 будет равна тождественной единице.

Выход сети является нейрон y_3 , который есть преобразование своих входов с весами w_7, w_8, w_9 :

$$y_3 = f(w_7 + w_8 y_1 + w_9 y_2).$$

В свою очередь нейроны y_1 и y_2 есть:

$$y_1 = f(w_1 + w_2 x_1 + w_3 x_2);$$

$$y_2 = f(w_4 + w_5 x_1 + w_6 x_2)$$

Задача обучения сети есть нахождение таких значений весов $w_1, w_2, \dots, w_9$, так чтобы наша сеть при поступлении на вход значений x_1 и x_2 выдавала на выходе значение y_3 , совпадающее со значением функции XOR из табл. 2.2.

В качестве функции преобразования сети возьмем функцию тригонометрического тангенса (2.16).

Рассмотрим итерационный процесс следующего порядка. Пусть первоначальные значения весов $w_1, w_2, \dots, w_3$ будут выбраны случайным образом в диапазоне значений $[-0,3; 0,3]$ и равны $w_1^0, w_2^0, \dots, w_9^0$. Значения переменными x_1 и x_2 выбраны из первой строки таблицы 2. x_0 будет равна тождественной единице. В качестве выхода сети получим функцию:

$$y_3^0 = \tanh(w_7^0 + w_8^0 * y_1 + w_3^0 y_2).$$

Тогда невязка между выходом сети и значением нашей функции $f(x_1, x_2)$ есть:

$$\delta = y_3^0 - f(x_1, x_2).$$

Для корректировки весов воспользуемся **алгоритмом обратного распространения ошибки** и рассчитаем по формулам (12) – (15) Δw_i^1 .

Повторим этот процесс несколько раз на основе первой строки табл. 2.2, пока значения невязки между выходом сети и значением нашей функции $f(x_1, x_2)$ не достигнет некоторого удовлетворительного уровня точности, например 10^{-5} :

$$|\delta| \leq 0,00001.$$

Перейдем после этого к следующей строки табл. 2.2 и повторим этот процесс, пока значение невязки опять не достигнет некоторого удовлетворительного уровня точности.

Будем повторять этот процесс, до тех пор, пока суммарное значение модуля невязки на всех строках табл. 2.2. не достигнет некоторого удовлетворительного уровня точности.

В файле «Нейросеть конъюнкция и персептрон.xls» на листе «персептрон XOR» приведен пример построения итерационного процесса для функции XOR в программе Excel.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Рассчитать по формулам (2.12)–(2.15) Δw_i^1 . Самостоятельно построить итерационный процесс для обучения нейронной сети с функцией XOR. Повторить процесс обучения сети при различных значениях параметра η -нормы обучения. Выбрать оптимальное значение.

2. Построить графики значений невязки в зависимости от числа итераций процесса. Выбрать оптимальное число итераций. Повторить процесс обучения сети при различных значениях начальных значений весов.

3. Построить пример построения итерационного процесса на основе других передаточных функций, например логистической.

Выводы

Рассмотренная в лабораторной работе функция XOR является дальнейшим усложнением построения итерационного процесса для обучения нейронной сети по сравнению с работой № 1. Она требует тщательного аналитического внимания для построения формул нахождения градиента весов при использовании **алгоритма обратного распространения ошибки**. Мы рассмотрели вопросы дальнейшего усложнения построения и внутреннее функционирование нейронной сети. Рассмотрели построение процесса обучения нейронной сети с учителем. Рассмотрели различные типы передаточных функций. Эти все основные принципы сохраняются в более сложных типах нейронной сети и без их понимания сложно двигаться к более серьезным работам.

Лабораторная работа № 3 СЕТЬ КОХОНЕНА

ЦЕЛЬ РАБОТЫ:

- изучить принципы работы сети Кохонена, алгоритм ее функционирования;
- провести обучение нейронной сети Кохонена, способной решать задачи кластеризации.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Сети (слои) Кохонена относятся к самоорганизующимся нейронным сетям. Самоорганизующаяся сеть позволяет выявлять кластеры входных векторов, обладающих некоторыми общими свойствами.

Кластеризация – это разделение исследуемого множества объектов на группы «похожих» объектов, называемых кластерами.

Кластеризация позволяет сгруппировать сходные данные, что облегчает решение ряда задач.

Изучение данных, облегчение анализа. Содержательный анализ полученных кластеров позволяет обнаружить закономерности. Например, можно выявить группы клиентов сети сотовой связи, для которых можно

предложить новый тарифный план. Анализ содержания кластера позволяет применить к объектам различных кластеров разные методы анализа.

Прогнозирование. Относя новый объект к одному из кластеров, можно прогнозировать поведение объекта, поскольку его поведение будет схожим с поведением объектов кластера.

Обнаружение аномалий. Содержательный анализ кластеров также помогает выявить аномалии. Обычно, это кластеры, в которые попадает мало объектов.

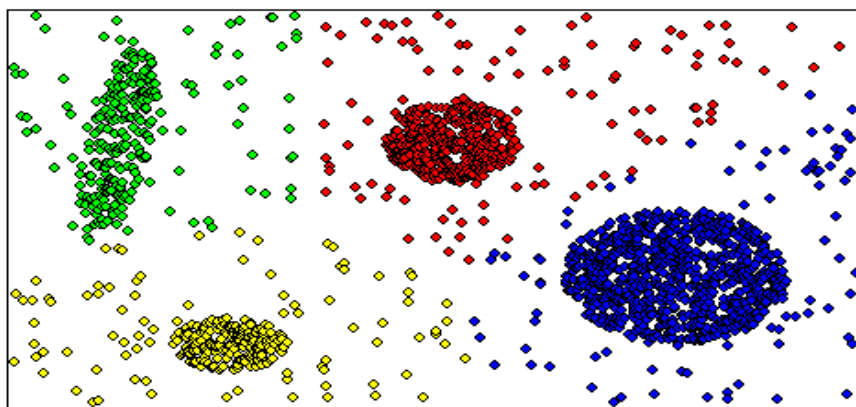


Рис. 3.1. Кластеризация исследуемого множества объектов

Сеть Кохонена – это однослойная сеть, построенная из нейронов типа WTA (Winner Takes All – победитель получает все).

Каждый нейрон сети соединен со всеми компонентами m -мерного входного вектора $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$. Входной вектор – это описание одного из объектов, подлежащих кластеризации. Количество нейронов совпадает с количеством кластеров, которое должна выделить сеть. В качестве нейронов сети Кохонена применяются линейные взвешенные сумматоры.

$$s_j = b_j + \sum_{i=1}^m w_{ij}x_i,$$

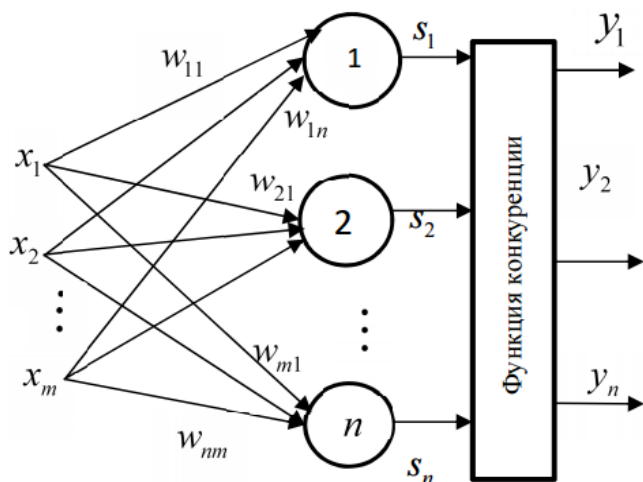


Рис. 3.2. Топология сети Кохонена

где j – номер нейрона; i – номер входа; s_j – выход j -го нейрона (адаптивного сумматора); w_{ij} – вес i -го входа j -го нейрона; b_j – порог.

Каждый j -й нейрон описывается вектором весов $w_j = (w_{1j}, w_{2j}, \dots, w_{mj})$, где m – число компонентов входных векторов. С выходов адаптивных сумматоров сигнал поступает на функцию конкуренции, работающую по правилу «победитель получает всё». Функция

конкуренции находит выход – адаптивный сумматор с минимальным значением выхода.

В сетях Кохонена используется обучение без учителя. Для обучения сети применяются механизмы конкуренции. При подаче на вход сети вектора x побеждает тот нейрон, вектор весов которого в наименьшей степени отличаются от входного вектора. Для нейрона-победителя выполняется соотношение:

$$d(x, w_j) = \min_{1 \leq i \leq n} d(x, w_i),$$

где n – количество нейронов; j – номер нейрона-победителя; $d(x, w_i)$ – расстояние (метрика) между векторами x и w . Чаще всего в качестве меры расстояния используется евклидова мера:

$$d(x, w_i) = \|x - w_i\| = \sqrt{\sum_{j=1}^m (x_j - w_{ji})^2}.$$

Конкурирующая функция активации анализирует значения сумматоров и формирует выходы нейронов, равные 0 для всех нейронов, кроме одного «нейрона-победителя», имеющего на выходе минимальное значение. Таким образом, вектор выхода имеет единственный элемент, равный 1, который соответствует нейрону-победителю, а остальные равны 0. Номер активного нейрона определяет ту группу (кластер), к которой наиболее близок входной вектор.

Процесс обучения сети Кохонена состоит из циклического повторения ряда шагов:

- 1) подача исходных данных на входы;
- 2) нахождение выхода каждого нейрона;
- 3) определение «выигравшего» нейрона или нейрона-победителя;
- 4) корректировка весов «выигравшего» нейрона по правилу Кохонена:

$$w_i^{(k+1)} = w_i^{(k)} + \eta_i^{(k)} [x - w_i^{(k)}],$$

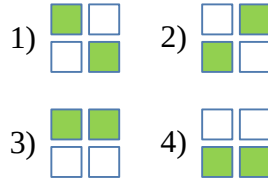
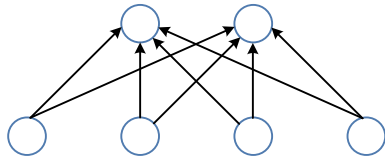
где x – входной вектор; k – номер цикла обучения; $\eta_i^{(k)}$ – коэффициент скорости обучения i -го нейрона в k -ом цикле обучения.

ЗАДАЧА

Для обучения сети SOFM, имеющей четыре входных элемента и два кластерных, используются четыре образца, закодированные 1 (закрашенный квадрат) и 0 (пустой квадрат) слева направо, сверху вниз. Даны начальные весовые значения W_{ij} и норма обучения η . Используя евклидову

метрику, определить принадлежность образцов кластерным элементам и вычислить весовые значения после первого цикла обработки данных.

$$\eta = 0.5$$



$$W = \begin{vmatrix} 0 & 0.3 & 0.1 & 0.9 \\ 0.1 & 0.5 & 0.9 & 0.1 \end{vmatrix}$$

Решение

Кодируем образцы в соответствии с заданным правилом: закрашенный квадрат как 1, пустой квадрат как 0. Получаем

$$X_1 = [1, 0, 0, 1], X_2 = [0, 1, 1, 0], X_3 = [1, 1, 0, 0], X_4 = [0, 0, 1, 1].$$

Определяем близость образца 1 со всеми кластерами $W_j, j = 1 \dots 2$. Близость кластера j и образца k определяется квадратом евклидова расстояния между ними, вычисляемый по формуле:

$$r_{kj} = \sum_i^n (X_{ki} - W_{ji})^2.$$

Таким образом, имеем:

$$r_{11} = (1 - 0)^2 + (0 - 0.3)^2 + (0 - 0.1)^2 + (1 - 0.9)^2 = 1.11.$$

Аналогично вычисляем, $r_{12} = 2.68$. Следовательно, кластер № 1 – победитель. Он ближе всех (в указанном смысле) оказался к первому образцу. Подвигаем кластер-победитель к образцу 1 с коэффициентом обучения $\eta = 0.5$: $W'_1 = W_1 + \eta \cdot (X_1 - W_1)$. В итоге

$$W'_1 = \begin{vmatrix} 0.5 & 0.15 & 0.05 & 0.95 \end{vmatrix}.$$

Повторим все для второго образца. Первый кластер берем новым. Вычисляем $r_{21} = 2.77, r_{22} = 0.28$. Теперь второй кластер имеет наибольшую близость с образцом. Подвигаем кластер-победитель к образцу 2 с коэффициентом обучения $\eta = 0.5$: $W'_2 = W_2 + \eta \cdot (X_2 - W_2)$. В итоге

$$W'_2 = \begin{vmatrix} 0.05 & 0.75 & 0.95 & 0.05 \end{vmatrix}.$$

Рассмотрим третий образец. Для него имеем $r_{31} = 1.8775, r_{32} = 1.87$. Побеждает второй кластер. Меняем его $\eta = 0.5$: $W''_2 = W'_2 + \eta \cdot (X_3 - W'_2)$:

$$W''_2 = \begin{vmatrix} 0.525 & 0.875 & 0.475 & 0.025 \end{vmatrix}.$$

Для четвертого образца $r_{41}=1.7775, r_{42}=2.2675$. Вновь побеждает первый кластер. Меняем его:

$$W''_1 = \begin{vmatrix} 0.25 & 0.075 & 0.525 & 0.975 \end{vmatrix}.$$

Таким образом, получаем последовательность кластеров-победителей после первого прохода: 1,2,2,1. Конечные значения координат кластеров:

$$W = \begin{vmatrix} 0.25 & 0.075 & 0.525 & 0.975 \\ 0.525 & 0.875 & 0.475 & 0.025 \end{vmatrix}.$$

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Для обучения сети SOFM, имеющей четыре входных элемента и два кластерных, используются четыре образца, закодированные 1 (закрашенный квадрат) и 0 (пустой квадрат). Даны начальные весовые значения W_{ij} и норма обучения $\eta = 0.5$. Используя евклидову метрику, определить принадлежность образцов кластерным элементам и вычислить весовые значения после первого цикла обработки данных.

Задача 3.1

$$X_1 = [1,0,0,1], X_2 = [0,1,1,0], X_3 = [1,1,0,0], X_4 = [0,0,1,1].$$

$$W = \begin{vmatrix} 0 & 0.3 & 0.1 & 0.9 \\ 0.1 & 0.5 & 0.9 & 0.1 \end{vmatrix}.$$

Задача 3.2

$$X_1 = [1,1,0,0], X_2 = [0,0,1,1], X_3 = [1,0,1,0], X_4 = [0,1,0,1].$$

$$W = \begin{vmatrix} 0.4 & 0.4 & 0.6 & 0.9 \\ 0.2 & 0.2 & 0.7 & 0.5 \end{vmatrix}.$$

Задача 3.3

$$X_1 = [1,1,0,0], X_2 = [0,0,1,1], X_3 = [1,0,1,0], X_4 = [0,1,0,1].$$

$$W = \begin{vmatrix} 0.3 & 0.1 & 0.3 & 0.7 \\ 0.4 & 0.4 & 0.6 & 0.4 \end{vmatrix}.$$

Задача 3.4

$$X_1 = [0,1,0,1], X_2 = [1,1,1,0], X_3 = [1,1,0,1], X_4 = [1,0,0,1].$$

$$W = \begin{vmatrix} 0.9 & 0.1 & 0.1 & 0.6 \\ 0.4 & 0.6 & 0.4 & 1 \end{vmatrix}.$$

Задача 3.5

$$X_1 = [0,1,0,1], X_2 = [1,1,0,0], X_3 = [1,0,1,0], X_4 = [0,0,1,1].$$

$$W = \begin{vmatrix} 0.1 & 0.7 & 0.2 & 0.6 \\ 0.3 & 0.2 & 0.7 & 0.2 \end{vmatrix}.$$

Задача 3.6

$$X_1 = [1,0,0,1], X_2 = [0,1,1,0], X_3 = [1,1,0,0], X_4 = [0,0,1,1].$$

$$W = \begin{vmatrix} 0 & 0.3 & 1 & 0.3 \\ 1 & 0.6 & 0.4 & 0.1 \end{vmatrix}.$$

Задача 3.7

$$X_1 = [1,1,1,0], X_2 = [0,0,1,1], X_3 = [1,0,1,0], X_4 = [0,1,0,1].$$

$$W = \begin{vmatrix} 0.5 & 0.8 & 0.8 & 0.9 \\ 0.9 & 0.5 & 0.9 & 0.6 \end{vmatrix}$$

Задача 3.8

$$X_1 = [0,1,0,1], X_2 = [1,1,0,0], X_3 = [0,0,1,1], X_4 = [1,0,1,0].$$

$$W = \begin{vmatrix} 0.2 & 0.1 & 0.7 & 0.5 \\ 0.3 & 0.6 & 0.6 & 0.3 \end{vmatrix}$$

Задача 3.9

$$X_1 = [1,1,0,0], X_2 = [0,0,1,1], X_3 = [1,0,1,0], X_4 = [0,1,0,1].$$

$$W = \begin{vmatrix} 0.4 & 0.7 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.9 & 0.5 \end{vmatrix}$$

$$W = \begin{vmatrix} 0.5 & 0.8 & 0.8 & 0.9 \\ 0.9 & 0.5 & 0.9 & 0.6 \end{vmatrix}$$

Задача 3.10

$$X_1 = [0,1,0,1], X_2 = [1,1,0,0], X_3 = [0,0,1,1], X_4 = [1,0,1,0].$$

$$W = \begin{vmatrix} 0.2 & 0.9 & 0.3 & 0.4 \\ 0.4 & 0.9 & 0.4 & 0.3 \end{vmatrix}$$

Лабораторная работа № 4 СЕТЬ ХОПФИЛДА

ЦЕЛЬ РАБОТЫ:

- изучить принципы работы сети Хопфилда, алгоритм ее функционирования;
- провести обучение нейронной сети Хопфилда, способной решать задачи фильтрации.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Сеть Хопфилда – полносвязная нейронная сеть с симметричной матрицей связей. В процессе работы динамика таких сетей сходится к одному из положений равновесия. Эти положения равновесия являются локальными минимумами функционала, называемого *энергией сети* (в простейшем случае – локальными минимумами отрицательно определённой квадра-

точной формы на n -мерном кубе). Такая сеть может быть использована как автоассоциативная память, как фильтр, а также для решения некоторых задач оптимизации.

Нейронная сеть Хопфилда состоит из искусственных нейронов. Каждый нейрон системы может принимать одно из двух состояний (что аналогично выходу нейрона с пороговой функцией активации):

$$y_i = \begin{cases} 1 \\ -1 \end{cases}.$$

Из-за их биполярной природы нейроны сети Хопфилда иногда называют спинами. Взаимодействие спинов сети описывается выражением:

$$E = \frac{1}{2} \sum_{i=0}^n w_{ij} x_i x_j,$$

где w_{ij} – элемент матрицы взаимодействий W , которая состоит из весовых коэффициентов связей между нейронами. В эту матрицу в процессе обучения записывается M «образов» – N -мерных бинарных векторов:

$$S_m = (s_{m1}, s_{m2}, \dots, s_{mN}).$$

Эти образы во время эксплуатации сети будут выражать отклик системы на входные сигналы, или иначе – окончательные значения выходов y_i после серии итераций.

В сети Хопфилда матрица связей является симметричной, а диагональные элементы матрицы полагаются равными нулю, что исключает эффект воздействия нейрона на самого себя.

ОБУЧЕНИЯ СЕТИ ХОПФИЛДА

Алгоритм обучения сети Хопфилда существенно отличается от таких классических алгоритмов обучения перцептронов, как метод коррекции ошибки или метод обратного распространения ошибки. Отличие заключается в том, что вместо последовательного приближения к нужному состоянию с вычислением ошибок, все коэффициенты матрицы рассчитываются по одной формуле, за один цикл, после чего сеть сразу готова к работе. Вычисление коэффициентов основано на следующем правиле: для всех запомненных образов матрица связи должна удовлетворять уравнению $X_i^T = W X_i$.

ЗАДАЧА

За один цикл определить образ $\bar{y}$ вектора y с помощью сети Хопфилда, обученной по образцам x_1, x_2, x_3 . Если вектор был распознан, указать номер соответствующего образца.

$$\begin{aligned}
x_1 &= [1, 1, -1, 1], \\
x_2 &= [-1, 1, -1, 1], \\
x_3 &= [1, -1, -1, 1], \\
y &= [-1, 1, 1, -1].
\end{aligned}$$

Решение

Составим матрицу взаимодействия W , которая состоит из весовых коэффициентов связей между нейронами. Для этого необходимо составить произведение, в котором первый множитель это вектор, составленный из значений входа i -го образца, а второй множитель транспонированный вектор i -го образца. Обозначим $\vec{x}_i = X_i$:

$$W_1 = X_1 \cdot X_1^T = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \cdot [1 \ 1 \ -1 \ 1] = \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \end{bmatrix}.$$

Аналогично вычисляем W_2 :

$$W_2 = X_2 \cdot X_2^T = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \cdot [-1 \ 1 \ -1 \ 1] = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}.$$

Найдем W_3 :

$$W_3 = X_3 \cdot X_3^T = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \cdot [1 \ -1 \ -1 \ 1] = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Найдем матрицу взаимодействия W :

$$W = W_1 + W_2 + W_3 = \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -1 & -1 & 1 \\ -1 & 3 & -1 & 1 \\ -1 & -1 & 3 & -3 \\ 1 & 1 & -3 & 3 \end{bmatrix}.$$

Для того чтобы исключить эффект воздействия нейрона на самого себя диагональные элементы полагаются равными нулю:

$$W = \begin{bmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & -1 & 1 \\ -1 & -1 & 0 & -3 \\ 1 & 1 & -3 & 0 \end{bmatrix}.$$

Вычислим новый вектор X_0 по формуле $X_0 = W \cdot y$. Тогда имеем:

$$X_0 = W \cdot y = \begin{bmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & -1 & 1 \\ -1 & -1 & 0 & -3 \\ 1 & 1 & -3 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -3 \\ -3 \\ 1 \\ -3 \end{bmatrix}.$$

К найденному вектору применим функцию $\text{sign}(x)$:

$$\bar{y} = \text{sign}(X_0) = \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}.$$

Таким образом, мы получили образ $\bar{y}$ вектора y . Сравним исходные образы с полученным вектором. Данный вектор не распознан, так как он не равен ни одному исходному образцу.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

За один цикл определить образ $\bar{y}$ вектора y с помощью сети Хопфилда, обученной по образцам x_1, x_2, x_3 . Если вектор был распознан, указать номер соответствующего образца.

Задача 4.1 $x_1 = [1, -1, 1, -1]$
 $x_2 = [-1, 1, 1, -1]$
 $x_3 = [-1, -1, -1, 1]$
 $y = [-1, -1, 1, 1]$

Задача 4.2 $x_1 = [1, -1, -1, 1]$
 $x_2 = [-1, -1, 1, 1]$
 $x_3 = [1, -1, -1, -1]$
 $y = [1, -1, 1, -1]$

Задача 4.3 $x_1 = [1, 1, -1, 1]$
 $x_2 = [-1, 1, 1, 1]$
 $x_3 = [1, 1, -1, -1]$
 $y = [1, -1, 1, -1]$

Задача 4.4 $x_1 = [-1, 1, -1, -1]$
 $x_2 = [-1, -1, 1, -1]$
 $x_3 = [1, 1, 1, -1]$
 $y = [1, -1, -1, 1]$

Задача 4.5 $x_1 = [-1, -1, 1, 1]$
 $x_2 = [1, -1, -1, -1]$
 $x_3 = [-1, 1, -1, -1]$
 $y = [1, -1, 1, -1]$

Задача 4.6 $x_1 = [-1, 1, -1, 1]$
 $x_2 = [1, 1, -1, 1]$
 $x_3 = [-1, 1, 1, 1]$
 $y = [1, -1, -1, -1]$

Задача 4.7 $x_1 = [-1, -1, 1, -1]$
 $x_2 = [-1, -1, -1, 1]$
 $x_3 = [1, -1, -1, 1]$
 $y = [1, 1, -1, 1]$

Задача 4.8 $x_1 = [1, 1, 1, -1]$
 $x_2 = [1, -1, -1, 1]$
 $x_3 = [1, 1, -1, 1]$
 $y = [-1, 1, 1, 1]$

Задача 4.9 $x_1 = [-1, -1, 1, 1]$
 $x_2 = [1, 1, 1, 1]$
 $x_3 = [-1, -1, 1, -1]$
 $y = [-1, 1, 1, -1]$

Задача 4.10 $x_1 = [1, -1, -1, -1]$
 $x_2 = [-1, -1, 1, -1]$
 $x_3 = [1, -1, 1, -1]$
 $y = [1, -1, -1, 1]$

Лабораторная работа № 5 СЕТЬ ХЕММИНГА

ЦЕЛЬ РАБОТЫ:

- изучить принципы работы сети Хопфилда, алгоритм ее функционирования;
- провести обучение нейронной сети Хопфилда, способной решать задачи классификации бинарных входных векторов.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Искусственная нейронная сеть Хэмминга используется для решения задач классификации бинарных входных векторов. В основе ее работы лежат процедуры, направленные на выбор в качестве решения задачи классификации одного из эталонных образов, наиболее близкого к поданному на вход сети зашумленному входному образу, и отнесение данного образа к соответствующему классу. Для оценки меры близости к каждому классу используется критерий, учитывающий расстояние Хэмминга – количество различающихся переменных у зашумленного и эталонного входных образов.

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m – число образцов. Нейроны первого слоя имеют по n синапсов, соединенных со входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$$w_{ik} = \frac{x_i^k}{2}; i = 0, \dots, n-1; k = 0, \dots, m-1; \quad (5.1)$$

$$T_k = n / 2; k = 0, \dots, m-1, \quad (5.2)$$

где x_i^k – i -й элемент k -го образца.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине $0 < \varepsilon < 1/m$. Синапс нейрона, связанный с его же аксоном имеет вес $+1$.

Алгоритм функционирования сети Хэмминга следующий.

1. На входы сети подается неизвестный вектор $X = \{x_i; i = 0, \dots, n-1\}$, исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j; j = 0, \dots, m-1. \quad (5.3)$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$$y_j^{(2)} = y_j^{(1)}; j = 0, \dots, m-1. \quad (5.4)$$

2. Вычислить новые состояния нейронов второго слоя:

$$s_j^{(2)}(p+1) = y_j^{(1)}(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), k \neq j, j = 0 \dots m-1 \quad (5.5)$$

и значения их аксонов:

$$y_j^{(2)}(p+1) = f[s_j^{(2)}(p+1)], j = 0 \dots m-1 \quad (5.6)$$

Активационная функция f имеет вид порога (рис. 5.1), причем величина F должна быть достаточно большой, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да – перейди к шагу 2. Иначе – конец.

Из оценки алгоритма видно, что роль первого слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

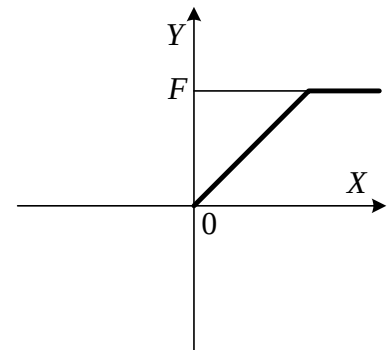
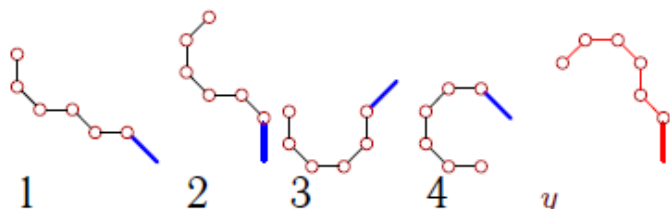


Рис. 5.1. Активационная функция

ЗАДАЧА

Даны четыре плоских образца $\bar{X}_k, k=1..4$ и искаженный образец y . Найти выход сети после первого цикла распознавания y сетью Хемминга. Образцы кодируются от основания (выделено утолщением). Левый поворот кодируется как -1, правый поворот +1. Во втором слое принять внедиагональные веса равными -1/4.



Решение

Кодируем образцы согласно заданным правилам: левый поворот кодируется как -1, правый поворот +1. Получаем:

$$X_1 = [-1, 1, -1, 1, 1], X_2 = [-1, -1, 1, 1, 1], X_3 = [-1, 1, 1, 1, 1],$$

$$X_4 = [-1, -1, -1, -1, -1], y = [-1, 1, -1, -1, -1].$$

Составим две вспомогательные матрицы.

W_1 – матрица образов:

$$W_1 = \begin{bmatrix} -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}.$$

W_2 – матрица второго слоя, где элементы на главной диагонали равны единице, а внедиагональные элементы согласно заданному весу:

$$W_2 = \begin{bmatrix} 1 & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & 1 & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & 1 & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 1 \end{bmatrix}.$$

Вычислим новый вектор по формуле $\vec{p} = \frac{\vec{b} - W_1 \cdot \vec{y}}{2}$:

$$\vec{p} = \frac{\vec{b} - W_1 \cdot \vec{y}}{2} = \begin{bmatrix} \frac{5}{2} \\ \frac{5}{2} \\ \frac{5}{2} \\ \frac{5}{2} \\ \frac{5}{2} \\ \frac{5}{2} \end{bmatrix} - \begin{bmatrix} -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 1 \end{bmatrix},$$

где вектор $\vec{p}$ вспомогательный вектор, все элементы которого составляют количество входов нейрона. А количество элементов равно количеству нейронов.

Элементы найденного вектора показывают, насколько заданный вектор отличается от исходных образцов. Элемент p_1 указывает на то, что 2 значения на входах образца X_1 и вектора y различны. Рассчитаем $\vec{p}$ по формуле:

$$\vec{p} = \vec{e} - \frac{\vec{p}}{N},$$

где $\vec{e}$ – единичный вектор; N – количество входов нейрона.

Получаем:

$$\vec{p} = \begin{bmatrix} \frac{3}{5} \\ \frac{1}{5} \\ \frac{2}{5} \\ \frac{4}{5} \\ \frac{4}{5} \end{bmatrix}.$$

Далее умножим найденный вектор на матрицу W_2 :

$$\hat{p} = W_2 \cdot \vec{p} = \begin{bmatrix} 1 & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & 1 & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & 1 & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{3}{5} \\ \frac{1}{5} \\ \frac{2}{5} \\ \frac{4}{5} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ -\frac{1}{4} \\ 0 \\ \frac{1}{2} \end{bmatrix}.$$

Обнулим отрицательные элементы в найденном векторе:

$$\hat{p} = \begin{bmatrix} 1 \\ \frac{1}{4} \\ 0 \\ 0 \\ \frac{1}{2} \end{bmatrix}.$$

Таким образом, мы получили выходной вектор, после одной итерации сети Хемминга.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Даны четыре плоских образца $\bar{X}_k, k=1..4$ и искаженный образец y . Найти выход сети после первого цикла распознавания y сетью Хемминга. Во втором слое принять внедиагональные веса равными $-1/4$.

Задача 5.1 $X_1 = [-1, 1, -1, 1, 1], X_2 = [-1, -1, 1, 1, 1],$
 $X_3 = [-1, 1, 1, 1, 1], X_4 = [-1, -1, -1, -1, 1], y = [-1, 1, -1, -1, -1].$

Задача 5.2 $X_1 = [1, -1, 1, -1, 1], X_2 = [1, 1, 1, -1, 1],$
 $X_3 = [1, -1, -1, 1, 1], X_4 = [1, 1, -1, 1, 1], y = [1, -1, 1, 1, 1].$

Задача 5.3 $X_1 = [-1, 1, -1, -1, 1], X_2 = [-1, -1, 1, 1, -1],$
 $X_3 = [-1, 1, 1, 1, 1], X_4 = [-1, -1, -1, 1, 1], y = [-1, -1, -1, -1, 1].$

Задача 5.4 $X_1 = [-1, 1, -1, -1, 1], X_2 = [-1, -1, -1, 1, -1],$
 $X_3 = [-1, 1, 1, -1, 1], X_4 = [-1, -1, -1, 1, 1], y = [-1, -1, 1, -1, 1].$

Задача 5.5 $X_1 = [1, -1, 1, 1, 1], X_2 = [1, 1, 1, -1, 1],$
 $X_3 = [1, -1, -1, 1, -1], X_4 = [-1, 1, -1, 1, 1], y = [1, -1, 1, -1, 1].$

Задача 5.6 $X_1 = [1, 1, 1, 1, 1], X_2 = [1, 1, 1, -1, 1],$
 $X_3 = [-1, -1, -1, 1, -1], X_4 = [-1, 1, -1, 1, 1], y = [1, -1, 1, -1, 1].$

Задача 5.7 $X_1 = [1, -1, 1, 1, 1], X_2 = [-1, 1, 1, -1, 1],$
 $X_3 = [-1, -1, -1, 1, -1], X_4 = [-1, 1, -1, 1, 1], y = [-1, -1, 1, -1, 1].$

Задача 5.8 $X_1 = [1, -1, 1, 1, 1], X_2 = [1, 1, -1, -1, 1],$
 $X_3 = [-1, -1, -1, 1, -1], X_4 = [-1, 1, -1, 1, 1], y = [1, 1, 1, 1, 1].$

Задача 5.9 $X_1 = [1, 1, 1, 1, -1], X_2 = [1, -1, -1, -1, 1],$
 $X_3 = [-1, -1, -1, 1, -1], X_4 = [-1, 1, -1, 1, 1], y = [-1, -1, -1, 1, 1].$

Задача 5.10 $X_1 = [1, 1, 1, 1, 1], X_2 = [1, 1, 1, -1, 1],$
 $X_3 = [-1, -1, -1, 1, -1], X_4 = [-1, 1, -1, 1, 1], y = [1, -1, 1, -1, 1].$

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень основной литературы:

1. Барский А. Б. Введение в нейронные сети: учебное пособие / А. Б. Барский. - 4-е изд. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 357 с. - ISBN 978-5-4497-2381-9. - Текст: электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - URL: <https://www.iprbookshop.ru/133929.html>
2. Цуканова Н. И. Применение нейронных сетей к решению практических задач обработки изображений и структурированных данных: учебное пособие / Н. И. Цуканова. - Москва : КУРС, 2024. - 256 с. - ISBN 978-5-907535-60-2. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/144809.html>
3. Барский А. Б. Логические нейронные сети: учебное пособие / А. Б. Барский. - 4-е изд. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 491 с. - ISBN 978-5-4497-3303-0. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/142272.html>

Перечень дополнительной литературы:

1. Горожанина Е. И. Нейронные сети: учебное пособие / Е. И. Горожанина. - Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. - 84 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/75391.html>
2. Вакуленко С. А. Нейронные сети: учебное пособие / С. А. Вакуленко, А. А. Жихарева. - Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2019. - 110 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/102447.html>
3. Лекун Я. Как учится машина: революция в области нейронных сетей и глубокого обучения / Я. Лекун; перевод Е. Арсенова; под редакцией В. Скворцова, М. Плещ. - Москва: Альпина ПРО, 2021. - 335 с. - ISBN 978-5-907394-92-6. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/131617.html>.
4. Павлова А. И. Искусственные нейронные сети: учебное пособие / А. И. Павлова. - Москва: Ай Пи Ар Медиа, 2021. - 190 с. - ISBN 978-5-4497-1165-6. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/108228.html>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- <http://www.machinelearning.ru/wiki/index.php?title> Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных.
- <https://archive.ics.uci.edu/ml/index.php> – UCI Machine learning repository. Center for Machine Learning and Intelligent Systems.
- <https://www.kaggle.com> –Your Machine Learning and Data Science Community.
- <https://Neurohive.io> сайт с новостями, туториалами, датасетами и другими материалами по теме нейронных сетей.
- <https://Ailibri.com> каталог нейросетей для различных задач, где представлены популярные модели, такие как ChatGPT, DALL-E, Midjourney и другие.
- <https://habr.com> ресурс с образовательными материалами в области нейронных сетей, а также проектами с открытым исходным кодом, связанными с нейросетями.
- <https://dtf.ru> на сайте есть подборка бесплатных курсов по нейросетям, как русских, так и зарубежных.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по организации и проведению самостоятельной работы
по дисциплине
«ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ»
для студентов направления подготовки
02.03.01 Математика и компьютерные науки

Общие положения

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);

- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);

- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине **«Введение в искусственные нейронные сети»** направлена на формирование следующих **компетенций**:

ПК-3. Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий, программирования и компьютерной техники

ПК-5. Способен принимать участие в управлении проектами создания информационных систем и программных комплексов на стадиях их жизненного цикла

ПК-6. Способен учитывать знания проблем и тенденций развития рынка ПО в профессиональной деятельности

1. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование соответствующего набора компетенций будущего бакалавра по направлению подготовки «Математика и компьютерные науки».

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

2. Технологическая карта самостоятельной работы студента

Коды реали- зуемых компе- тенций	Вид дея- тельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контакт- ная работа с препода- вателем	Все- го
7 семестр					
ПК-3 ПК-5 ПК-6	Само- стоятель- ное изу- чение ли- тературы	Собеседование	34	2	36
ПК-3 ПК-5 ПК-6	Подго- товка к лабора- торным работам	Отчет	34	2	36
Итого за 7 семестр			68	4	72
Итого			68	4	72

3. Порядок выполнения самостоятельной работы студентом

3.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста:**

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;

2. Выделите главное, составьте план;

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

3.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

3.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование

по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

3.4. Методические рекомендации по написанию научных текстов (докладов, рефератов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т. п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Реферат (доклад) - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Реферат не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в реферате должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки реферата студентом.

Выполнение реферата начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания реферата. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста реферата предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки реферат сдается на кафедру для его оценивания руководителем.

Требования к написанию реферата

Написание 1 реферата является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема реферата может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Реферат должен быть написан научным языком.

Объем реферата должен составлять 20-25 стр.

Структура реферата:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.
- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ог-

раничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

- Приложение (при необходимости).

Требования к оформлению:

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- реферат должен быть представлен в сброшюрованном виде.

Порядок защиты реферата:

Защита реферата проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту реферата отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите реферата приветствуется использование мультимедиа-презентации.

Оценка реферата

Реферат оценивается по следующим критериям:

- соблюдение требований к его оформлению;
- необходимость и достаточность для раскрытия темы приведенной в тексте реферата информации;
- умение студента свободно излагать основные идеи, отраженные в реферате;
- способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные мате-

риалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

Тематика рефератов по дисциплине «Введение в искусственные нейронные сети»

1. Основы и история нейронных сетей:

- История развития нейронных сетей: от персептрона до современных архитектур.
- Сравнительный анализ биологических и искусственных нейронных сетей.
- Основные парадигмы обучения нейронных сетей: обучение с учителем, без учителя, с подкреплением.
- Функции активации: свойства, применение, влияние на обучение сети.
- Потеря веса (Weight decay) и другие методы регуляризации в нейронных сетях.
- Проблема переобучения и способы ее решения.

2. Архитектуры нейронных сетей:

- Многослойный персептрон (MLP): архитектура, обучение, применение.
- Сверточные нейронные сети (CNN): архитектура, слои, особенности применения в обработке изображений.
- Рекуррентные нейронные сети (RNN): архитектура, типы (LSTM, GRU), применение в обработке последовательностей.
- Генеративно-состязательные сети (GAN): архитектура, обучение, применение для генерации данных.
- Трансформеры: архитектура, механизм внимания, применение в обработке текста и других задачах.
- Автокодировщики: архитектура, типы (Variational Autoencoder), применение для снижения размерности и генерации данных.
- Рекурсивные нейронные сети (Recursive Neural Networks).

3. Применение нейронных сетей в различных областях:

- Нейронные сети в компьютерном зрении: классификация изображений, обнаружение объектов, сегментация.
- Нейронные сети в обработке естественного языка: машинный перевод, анализ тональности, генерация текста.
- Нейронные сети в распознавании речи: распознавание речи, синтез речи.
- Нейронные сети в финансовом анализе: прогнозирование цен, обнаружение мошенничества.
- Нейронные сети в медицине: диагностика заболеваний, анализ медицинских изображений.
- Нейронные сети в робототехнике: управление роботами, планирование траекторий.
- Нейронные сети в играх: обучение игровых агентов, создание игрового контента.
- Нейронные сети в анализе социальных сетей: выявление трендов, анализ поведения пользователей.

4. Алгоритмы обучения и оптимизации:

- Градиентный спуск и его разновидности: стохастический градиентный спуск (SGD), Adam, RMSprop.
- Методы оптимизации второго порядка: L-BFGS, Hessian-free optimization.
- Обратное распространение ошибки (Backpropagation): принцип работы, проблемы и решения.
- Методы инициализации весов нейронных сетей.
- Обучение с подкреплением (Reinforcement Learning): алгоритмы Q-Learning, SARSA, Deep Q-Network (DQN).

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СРС

Перечень основной литературы:

1. Барский А. Б. Введение в нейронные сети: учебное пособие / А. Б. Барский. - 4-е изд. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 357 с. - ISBN 978-5-4497-2381-9. - Текст: электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - URL: <https://www.iprbookshop.ru/133929.html>
2. Цуканова Н. И. Применение нейронных сетей к решению практических задач обработки изображений и структурированных данных: учебное пособие / Н. И. Цуканова. - Москва : КУРС, 2024. - 256 с. - ISBN 978-5-907535-60-2. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/144809.html>
3. Барский А. Б. Логические нейронные сети: учебное пособие / А. Б. Барский. - 4-е изд. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 491 с. - ISBN 978-5-4497-3303-0. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/142272.html>

Перечень дополнительной литературы:

1. Горожанина Е. И. Нейронные сети: учебное пособие / Е. И. Горожанина. - Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. - 84 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/75391.html>
2. Вакуленко С. А. Нейронные сети: учебное пособие / С. А. Вакуленко, А. А. Жихарева. - Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2019. - 110 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/102447.html>
3. Лекун Я. Как учится машина: революция в области нейронных сетей и глубокого обучения / Я. Лекун; перевод Е. Арсенова; под редакцией В. Скворцова, М. Плещ. - Москва: Альпина ПРО, 2021. - 335 с. - ISBN 978-5-907394-92-6. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/131617.html>.
4. Павлова А. И. Искусственные нейронные сети: учебное пособие / А. И. Павлова. - Москва: Ай Пи Ар Медиа, 2021. - 190 с. - ISBN 978-5-4497-1165-6. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/108228.html>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- <http://www.machinelearning.ru/wiki/index.php?title> Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных.
- <https://archive.ics.uci.edu/ml/index.php>– UCI Machine learning repository. Center for Machine Learning and Intelligent Systems.
- <https://www.kaggle.com> –Your Machine Learning and Data Science Community.
- <https://Neurohive.io> сайт с новостями, туториалами, датасетами и другими материалами по теме нейронных сетей.
- <https://Ailibri.com> каталог нейросетей для различных задач, где представлены популярные модели, такие как ChatGPT, DALL-E, Midjourney и другие.
- <https://habr.com> ресурс с образовательными материалами в области нейронных сетей, а также проектами с открытым исходным кодом, связанными с нейросетями.
- <https://dtf.ru> на сайте есть подборка бесплатных курсов по нейросетям, как русских, так и зарубежных.