

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Вятский государственный университет»**  
**(ФГБОУ ВО «ВятГУ»)**  
Факультет автоматизации и вычислительной техники  
Кафедра ЭВМ

Исследование многослойного персептрона с обучением по методу с  
обратным распространением ошибки

Отчёт  
Лабораторная работа № 2 по дисциплине  
«Системы обработки знаний»

Выполнила студентка группы ИВТб-42 \_\_\_\_\_/Рзаев А. Э./  
Проверил доцент кафедры ЭВМ \_\_\_\_\_/Ростовцев В. С./

Киров 2020

## Цель

Изучить алгоритм обратного распространения ошибки (ОРО) в процессе обучения нейронной сети при вариативных параметрах обучения. Работа выполняется с помощью программы BackPropagate 3.0.0.exe.

## 1 Задание

Обучающая выборка представлена в приложении А. Используется выходная функция  $y=x_1^7 \cdot \cos(x_2+2)$ . Переменная  $x_1$  изменяется в промежутке  $[-7;3]$  с шагом 0,05025, переменная  $x_2$  изменяется в промежутке  $[-7,4;4]$  с шагом 0,05729.

## 2 Протокол выполнения

Во всех таблицах ниже цветом выделен наилучший результат. Значение минимальной ошибки не учитывалось при выборе.

### 2.1 Влияние нормализации на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

В таблице 1 показаны результаты исследования.

Таблица 1 – Влияние нормализации на эффективность алгоритма ОРО

Тип нормализации	Результат
Без нормализации	Максимальная ошибка: 1476181.92000000 Минимальная ошибка: 10.00000000 Средняя ошибка: 740349.03363636 Среднеквадратичная ошибка: 52452033481360.53125
[0;1]	Максимальная ошибка: 700113.57821300 Минимальная ошибка: 10.00000000 Средняя ошибка: 57770.30542342 Среднеквадратичная ошибка: 1067691073785.042236
[-0.5;0.5]	Максимальная ошибка: 398234.04911600 Минимальная ошибка: 10.00000000 Средняя ошибка: 315016.03019154 Среднеквадратичная ошибка: 9580087043839.501953
[-1;1]	Максимальная ошибка: 398234.07575700 Минимальная ошибка: 10.00000000 Средняя ошибка: 315016.08984294 Среднеквадратичная ошибка: 9580091715941.306641

Применение нормализации значительно сокращает величину ошибок. Результат лучше примерно в 13 раз для средних и, приблизительно в 49 раз для среднеквадратичных ошибок при нормализации [0;1] (наиболее близкой к лучшей).

## 2.2 Влияние выбора примеров на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

Нормализация: [0;1].

В таблице 2 показаны результаты исследования.

Таблица 2 – Влияние выбора примеров на эффективность алгоритма ОРО

Выбор примеров	Результат
Последовательный	Максимальная ошибка: 702378.66533800 Минимальная ошибка: 10.00000000 Средняя ошибка: 56471.55202544 Среднеквадратичная ошибка: 1068233726262.251831
Случайный	Максимальная ошибка: 700115.89746300 Минимальная ошибка: 10.00000000 Средняя ошибка: 57769.00361650 Среднеквадратичная ошибка: 1067691350691.58691400

Случайный выбор примеров показывает слегка лучший результат по сравнению с последовательным выбором примеров обучения.

## 2.3 Влияние крутизны функции на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

Нормализация: [0;1].

Выбор примеров: случайный.

В таблице 3 показаны результаты исследования.

Таблица 3 – Влияние крутизны функции на эффективность алгоритма ОРО

Крутизна функции	Результат
0,1	Максимальная ошибка: 699749.13455200 Минимальная ошибка: 10.00000000 Средняя ошибка: 57982.79375657 Среднеквадратичная ошибка: 1067693548160.339844
0,5	Максимальная ошибка: 700109.42432900 Минимальная ошибка: 10.00000000 Средняя ошибка: 57772.70591974 Среднеквадратичная ошибка: 1067690775792.408691
1,0	Максимальная ошибка: 701558.62151300 Минимальная ошибка: 10.00000000 Средняя ошибка: 56940.79985290 Среднеквадратичная ошибка: 1067926311841.594971
3,0	Максимальная ошибка: 716843.05366900 Минимальная ошибка: 10.00000000 Средняя ошибка: 48638.85855966 Среднеквадратичная ошибка: 1094324154465.175293

Наилучшие результаты – при крутизне функции 0,5 и 0,1, при этом приоритет отдаем использованию крутизны функции 0,5, потому что у нее средняя и среднеквадратичная ошибки меньше, чем у 0,1.

## 2.4 Влияние смещения на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 0,5.

В таблице 4 показаны результаты исследования.

Таблица 4 – Влияние смещения на эффективность алгоритма ОРО

Смещение	Результат
0,1	Максимальная ошибка: 700003.87156200 Минимальная ошибка: 10.00000000 Средняя ошибка: 57834.25588502 Среднеквадратичная ошибка: 1067688835558.367432
0,5	Максимальная ошибка: 700034.52213600 Минимальная ошибка: 10.00000000 Средняя ошибка: 57816.50131751 Среднеквадратичная ошибка: 1067689925430.368286
1,0	Максимальная ошибка: 700112.48484900 Минимальная ошибка: 10.00000000 Средняя ошибка: 57771.02499060 Среднеквадратичная ошибка: 1067691484056.507446
2,0	Максимальная ошибка: 700466.54796100 Минимальная ошибка: 10.00000000 Средняя ошибка: 57565.73680297 Среднеквадратичная ошибка: 1067712819426.598511

Наилучшие результаты наблюдаются при смещениях 0,1 и 0,5, при этом приоритет отдаем использованию крутизны функции 0,1, потому что у нее максимальная и среднеквадратичная ошибки меньше, чем у 0,5.

## 2.5 Влияние скорости обучения на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 0,5.

Смещение: 0,1.

В таблице 5 показаны результаты исследования.

Таблица 5 – Влияние скорости обучения на эффективность алгоритма ОРО

Скорость обучения	Результат
0,01	Максимальная ошибка: 699942.97928700 Минимальная ошибка: 10.00000000 Средняя ошибка: 57869.69968871 Среднеквадратичная ошибка: 1067688658042.167725
0,1	Максимальная ошибка: 700003.82645600 Минимальная ошибка: 10.00000000 Средняя ошибка: 57834.26798417 Среднеквадратичная ошибка: 1067688715448.988647
0,5	Максимальная ошибка: 700446.17469900 Минимальная ошибка: 10.00000000 Средняя ошибка: 57577.27209238 Среднеквадратичная ошибка: 1067710467908.768921
1,0	Максимальная ошибка: 701229.61335800 Минимальная ошибка: 10.00000000 Средняя ошибка: 57129.06389978 Среднеквадратичная ошибка: 1067838787249.046265

Наилучшие результаты наблюдаются при скоростях 0,01 и 0,1, при этом приоритет отдаем использованию скорости 0,01, потому что у нее максимальная и среднеквадратичная ошибки меньше, чем у 0,1.

## 2.6 Влияние количества нейронов в скрытом слое на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 0,5.

Смещение: 0,1.

Скорость обучения: 0,01.

В таблице 6 показаны результаты исследования.

Таблица 6 – Влияние количества нейронов в скрытом слое на эффективность алгоритма ОРО

Количество нейронов в скрытом слое	Результат
1	Максимальная ошибка: 689051.41841200 Минимальная ошибка: 10.00000000 Средняя ошибка: 64355.31749626 Среднеквадратичная ошибка: 1078823983305.869
5	Максимальная ошибка: 699938.83880400 Минимальная ошибка: 10.00000000 Средняя ошибка: 57872.24419849 Среднеквадратичная ошибка: 1067689635727.307
10	Максимальная ошибка: 699980.91795900 Минимальная ошибка: 10.00000000 Средняя ошибка: 57847.71985383 Среднеквадратичная ошибка: 1067688885521.584

Приоритет был отдан количеству нейронов в скрытом слое, равному 10.

## 2.7 Влияние момента на эффективность алгоритма ОРО

Число входов: 2.

Число выходов: 1.

Циклов обучения: 1000

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 0,5.

Смещение: 0,1.

Скорость обучения: 0,01.

Количество нейронов в скрытом слое: 10.

В таблице 7 показаны результаты исследования.

Таблица 7 – Влияние момента на эффективность алгоритма ОРО

Момент	Результат
Без момента	Максимальная ошибка: 699980.86595100 Минимальная ошибка: 10.00000000 Средняя ошибка: 57847.64893262 Среднеквадратичная ошибка: 1067688862914.463135
0,1	Максимальная ошибка: 794034.88676800 Минимальная ошибка: 10.00000000 Средняя ошибка: 699930.66559208 Среднеквадратичная ошибка: 46873565128671.57031
0,5	Максимальная ошибка: 794063.21311300 Минимальная ошибка: 6.44208400 Средняя ошибка: 699958.68838775 Среднеквадратичная ошибка: 46877272672727.28906
1,0	Максимальная ошибка: 794066.52020900 Минимальная ошибка: 3.13502600 Средняя ошибка: 699961.96005125 Среднеквадратичная ошибка: 46877705538501.19531

Наилучшие результаты наблюдаются при обучении без момента.

## 3 Ручной расчет

Для более подробного изучения алгоритма ОРО в режиме трассировки был сделан один проход (включающий прямое и обратное распространение), а затем те же самые действия были произведены вручную.

На рисунке 1 показана трассировка первого прохода при обучении сети.

На рисунке 2 показана структура сети.

Условия останова обучения нейронной сети: Циклов обучения: 1000	Нейрон[1][9] Взвешенная сумма = 0,1272667 Аксон = 0,5159029717
Инициализация весов синапсов случайным образом...	
Нейрон[1][1] w[1, 1, 1] = -0,866 w[1, 1, 2] = 0,972 Вес смещения: w[1, 1, 3] = 1	Нейрон[1][10] Взвешенная сумма = 0,146030974 Аксон = 0,5182457664
Нейрон[1][2] w[1, 2, 1] = -0,276 w[1, 2, 2] = -0,968 Вес смещения: w[1, 2, 3] = 1	Нейрон[2][1] Взвешенная сумма = -0,7228196788 Аксон = 0,4106183265
Нейрон[1][3] w[1, 3, 1] = 0,896 w[1, 3, 2] = -0,53 Вес смещения: w[1, 3, 3] = 1	Обратная волна - подсчет локальной ошибки нейронов...
Нейрон[1][4] w[1, 4, 1] = 0,314 w[1, 4, 2] = 0,966 Вес смещения: w[1, 4, 3] = 1	Подсчет локальной ошибки нейронов на выходе нейронной сети... Желаемый сигнал на выходе: 0,073905 Прогнозируемый сигнал на выходе нейронной сети: 0,4106183265
Нейрон[1][5] w[1, 5, 1] = -0,94 w[1, 5, 2] = -0,736 Вес смещения: w[1, 5, 3] = 1	Нейрон[2][1] Локальная ошибка = 0,04074415036
Нейрон[1][6] w[1, 6, 1] = -0,132 w[1, 6, 2] = -0,954 Вес смещения: w[1, 6, 3] = 1	Подсчет локальной ошибки нейронов в скрытых слоях нейронной сети...
Нейрон[1][7] w[1, 7, 1] = 0,296 w[1, 7, 2] = 0,634 Вес смещения: w[1, 7, 3] = 1	Нейрон[1][1] Локальная ошибка = 0,001821977824
Нейрон[1][8] w[1, 8, 1] = 0,462 w[1, 8, 2] = -0,724 Вес смещения: w[1, 8, 3] = 1	Нейрон[1][2] Локальная ошибка = 0,003524351885
Нейрон[1][9] w[1, 9, 1] = 0,528 w[1, 9, 2] = -0,156 Вес смещения: w[1, 9, 3] = 1	Нейрон[1][3] Локальная ошибка = -0,00334783962
Нейрон[1][10] w[1, 10, 1] = 0,798 w[1, 10, 2] = -0,17 Вес смещения: w[1, 10, 3] = 1	Нейрон[1][4] Локальная ошибка = 0,0001219459046
Нейрон[2][1] w[2, 1, 1] = 0,358 w[2, 1, 2] = 0,692 w[2, 1, 3] = -0,658 w[2, 1, 4] = 0,024 w[2, 1, 5] = -0,978 w[2, 1, 6] = 0,196 w[2, 1, 7] = -0,728 w[2, 1, 8] = -0,27 w[2, 1, 9] = -0,646 w[2, 1, 10] = 0,408 Вес смещения: w[2, 1, 11] = 1	Нейрон[1][5] Локальная ошибка = -0,004980809868
Выбираем допустимый образ из обучающего множества... 0,073298 0,073299 0,073905 Подаем сигнал на вход нейронной сети...	Нейрон[1][6] Локальная ошибка = 0,0009982057268
Нейрон[0][1] Аксон = 0,073298	Нейрон[1][7] Локальная ошибка = -0,003701171913
Нейрон[0][2] Аксон = 0,073299	Нейрон[1][8] Локальная ошибка = -0,001374554192
Прямая волна...	Нейрон[1][9] Локальная ошибка = -0,003286761827
	Нейрон[1][10] Локальная ошибка = 0,002075184601
	Коррекция весов синапсов...
	w[1, 1, 1] = -0,8641780222 w[1, 1, 2] = 0,9738219778 Вес смещения: w[1, 1, 3] = 1,001821978
	w[1, 2, 1] = -0,2724756481 w[1, 2, 2] = -0,9644756481 Вес смещения: w[1, 2, 3] = 1,003524352
	w[1, 3, 1] = 0,8926521604 w[1, 3, 2] = -0,5333478396 Вес смещения: w[1, 3, 3] = 0,9966521604
	w[1, 4, 1] = 0,3141219459 w[1, 4, 2] = 0,9661219459 Вес смещения: w[1, 4, 3] = 1,000121946
	w[1, 5, 1] = -0,9449808099 w[1, 5, 2] = -0,7409808099 Вес смещения: w[1, 5, 3] = 0,9950191901
	w[1, 6, 1] = -0,1310017943 w[1, 6, 2] = -0,9530017943 Вес смещения: w[1, 6, 3] = 1,000998206
	w[1, 7, 1] = 0,2922988281 w[1, 7, 2] = 0,6302988281 Вес смещения:

Нейрон[1][1] Взвешенная сумма = 0,10777056 Аксон = 0,5134680613	$w[1, 7, 3] = 0,9962988281$ $w[1, 8, 1] = 0,4606254458$ $w[1, 8, 2] = -0,7253745542$ Вес смещения: $w[1, 8, 3] = 0,9986254458$
Нейрон[1][2] Взвешенная сумма = 0,00881632 Аксон = 0,5011020382	$w[1, 9, 1] = 0,5247132382$ $w[1, 9, 2] = -0,1592867618$ Вес смещения: $w[1, 9, 3] = 0,9967132382$
Нейрон[1][3] Взвешенная сумма = 0,126826538 Аксон = 0,5158480069	$w[1, 10, 1] = 0,8000751846$ $w[1, 10, 2] = -0,1679248154$ Вес смещения: $w[1, 10, 3] = 1,002075185$
Нейрон[1][4] Взвешенная сумма = 0,193822406 Аксон = 0,5242088567	$w[2, 1, 1] = 0,3987441504$ $w[2, 1, 2] = 0,7327441504$ $w[2, 1, 3] = -0,6172558496$ $w[2, 1, 4] = 0,06474415036$ $w[2, 1, 5] = -0,9372558496$ $w[2, 1, 6] = 0,2367441504$ $w[2, 1, 7] = -0,6872558496$ $w[2, 1, 8] = -0,2292558496$ $w[2, 1, 9] = -0,6052558496$ $w[2, 1, 10] = 0,4487441504$ Вес смещения: $w[2, 1, 11] = 1,04074415$
Нейрон[1][5] Взвешенная сумма = -0,022848184 Аксон = 0,4971440081	
Нейрон[1][6] Взвешенная сумма = 0,020397418 Аксон = 0,5025496552	
Нейрон[1][7] Взвешенная сумма = 0,168167774 Аксон = 0,5210085955	
Нейрон[1][8] Взвешенная сумма = 0,0807952 Аксон = 0,5100980267	

Рисунок 1 – Трассировка первого прохода

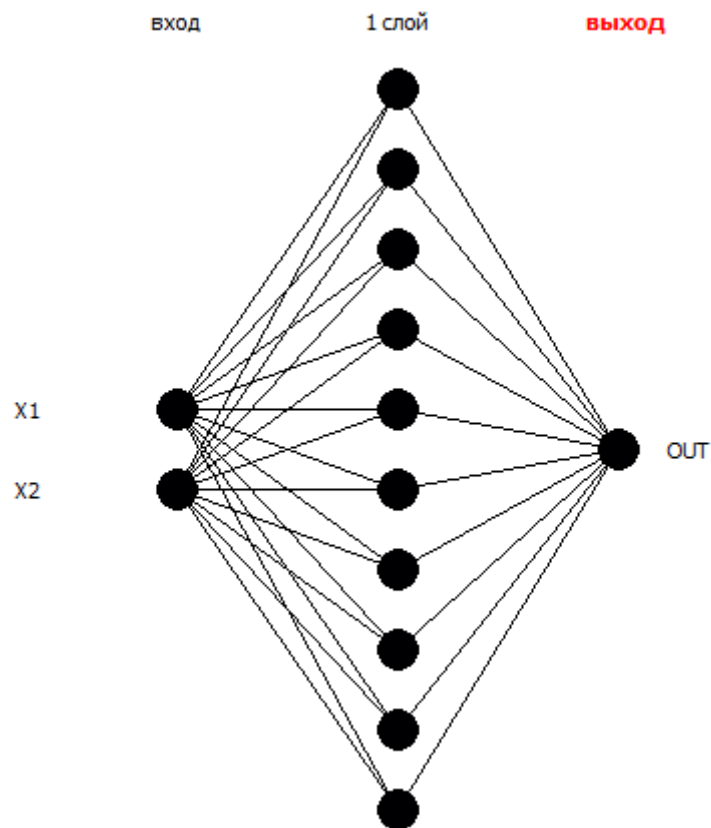


Рисунок 2 – Структура сети



В качестве активационной функции взята

$$F(S_i) = \frac{1}{1 + e^{-\alpha S_i}}, \quad (1)$$

где  $S_i$  – взвешенная сумма входов  $i$ -го нейрона (с учетом смещения);  
 $\alpha$  – коэффициент крутизны. Для всех слоев  $\alpha = 0,5$ .

В таблице 8 показан расчет прямой волны.

Таблица 8 – Расчет прямой волны

№ слоя	№ нейрона	№ выхода	Входной сигнал $x_j$	Весовой коэффициент $w_{ij}$	Смещение $w_{i0}$	Вес смещения	$w_{ij} \cdot x_j$	Взвешенная сумма $S_i$	Выход нейрона $y_i = F(S_i)$
Вход	1	1	0,177033	-	-	-	-	-	0,177033
	2	1	0,172249	-	-	-	-	-	0,172249
	3	1	0,129187	-	-	-	-	-	0,129187
1	1	1	0,073298	-0,866	0,1	1	-0,063476068	0,10777056	0,513468061
		2	0,073299	0,972			0,071246628		
	2	1	0,073298	-0,276	0,1	1	-0,020230248	0,00881632	0,501102038
		2	0,073299	-0,968			-0,070953432		
	3	1	0,073298	0,896	0,1	1	0,065675008	0,126826538	0,515848007
		2	0,073299	-0,53			-0,03884847		
	4	1	0,073298	0,314	0,1	1	0,023015572	0,193822406	0,524208857
		2	0,073299	0,966			0,070806834		
	5	1	0,073298	-0,94	0,1	1	-0,06890012	-0,022848184	0,497144008
		2	0,073299	-0,736			-0,053948064		
	6	1	0,073298	-0,132	0,1	1	-0,009675336	0,020397418	0,502549655
		2	0,073299	-0,954			-0,069927246		
	7	1	0,073298	0,296	0,1	1	0,021696208	0,168167774	0,521008595
		2	0,073299	0,634			0,046471566		
	8	1	0,073298	0,462	0,1	1	0,033863676	0,0807952	0,510098027
		2	0,073299	-0,724			-0,053068476		
	9	1	0,073298	0,528	0,1	1	0,038701344	0,1272667	0,515902972
		2	0,073299	-0,156			-0,011434644		
	10	1	0,073298	0,798	0,1	1	0,058491804	0,146030974	0,518245766
		2	0,073299	-0,17			-0,01246083		
Выход	1	1	0,513468061	0,358	0,1	1	0,183821566	-0,722819679	0,410618326
		2	0,501102038	0,692			0,34676261		
		3	0,515848007	-0,658			-0,339427989		
		4	0,524208857	0,024			0,012581013		
		5	0,497144008	-0,978			-0,48620684		
		6	0,502549655	0,196			0,098499732		
		7	0,521008595	-0,728			-0,379294258		
		8	0,510098027	-0,27			-0,137726467		
		9	0,515902972	-0,646			-0,33327332		
		10	0,518245766	0,408			0,211444273		

Для расчета ошибок необходимо найти производную функции (1) по  $S_i$ :

$$F'(S_i) = \frac{\alpha e^{-\alpha S_i}}{(e^{-\alpha S_i} + 1)^2}. \quad (2)$$

Тогда ошибка единственного нейрона выходного слоя будет найдена как

$$\gamma = (y - t) * F'(S),$$

где  $y$  – фактическое значение его выхода;  
 $t$  – желаемый сигнал на выходе.

Ошибка  $i$ -го нейрона скрытого слоя будет найдена как

$$\gamma_i = \gamma_j * F'(S_i) * \omega_{ij}.$$

где  $\gamma_j$  – ошибка выходного слоя;

$\omega_{ij}$  – синаптическая связь между  $i$ -м нейроном скрытого слоя и  $j$ -м нейроном выходного слоя.

В таблице 9 показан расчет ошибок.

Таблица 9 – Расчет ошибок

№ слоя	№ нейрона	$S_i$	$F'(S_i)$	Ошибка
Выход	1	-0,722819679	0,121005458	0,04074415
1	1	0,10777056	0,124909306	0,001821978
	2	0,00881632	0,124999393	0,003524352
	3	0,126826538	0,12487442	-0,00334784
	4	0,193822406	0,124706966	0,000121946
	5	-0,022848184	0,124995922	-0,00498081
	6	0,020397418	0,12499675	0,000998206
	7	0,168167774	0,124779319	-0,003701172
	8	0,0807952	0,124949015	-0,001374554
	9	0,1272667	0,124873548	-0,003286762
	10	0,146030974	0,124833546	0,002075185

Коррекция веса синапса производится по следующей формуле:

$$\omega_{ij}(t + 1) = \omega_{ij}(t) - \gamma_j.$$

Коррекция веса смещения производится по следующей формуле:

$$T_j(t + 1) = T_j(t) - \gamma_j.$$

В таблице 10 показан расчет новых весов.

Значения в таблицах 8, 9 полностью совпадают со значениями на рис. 1.

Скорректированные веса в таблице 10 совпадают с искомыми с точностью не менее трех знаков после запятой; погрешность можно объяснить ошибками округления и расчетов с плавающей запятой в разных средах.

Таким образом, ручной расчет выполнен верно.

Таблица 10 – Расчет новых весов

№ слоя	№ нейрона	№ выхода	Предыдущий весовой коэффициент $w_{ij}(t)$	Предыдущий вес смещения $T_j(t)$	Новый весовой коэффициент $w_{ij}(t+1)$	Новый вес смещения $T_j(t+1)$
1	1	1	-0,866	1	-0,86417802	1,001821978
		2	0,972		0,97382198	
	2	1	-0,276	1	-0,27247565	1,003524352
		2	-0,968		-0,96447565	
	3	1	0,896	1	0,89265216	0,99665216
		2	-0,53		-0,53334784	
	4	1	0,314	1	0,31412195	1,000121946
		2	0,966		0,96612195	
	5	1	-0,94	1	-0,94498081	0,99501919
		2	-0,736		-0,74098081	
	6	1	-0,132	1	-0,13100179	1,000998206
		2	-0,954		-0,95300179	
	7	1	0,296	1	-0,57000000	0,996298828
		2	0,634		0,63029883	
	8	1	0,462	1	0,46062545	0,998625446
		2	-0,724		-0,72537455	
	9	1	0,528	1	0,52471324	0,996713238
		2	-0,156		-0,15928676	
	10	1	0,798	1	0,80007518	1,002075185
		2	-0,17		-0,16792482	
Выход	1	1	0,358	1	0,39874415	1,040744150
		2	0,692		0,73274415	
		3	-0,658		-0,61725585	
		4	0,024		0,06474415	
		5	-0,978		-0,93725585	
		6	0,196		0,23674415	
		7	-0,728		-0,68725585	
		8	-0,27		-0,22925585	
		9	-0,646		-0,60525585	
		10	0,408		0,44874415	

#### 4 Выводы

В ходе выполнения лабораторной работы был исследован алгоритм ОРО при обучении многослойной НС, а также влияние различных параметров на качество обучения.

Наибольшее влияние на качество обучения показали нормализация, крутизна функции и величина момента. Влияние других параметров, таких как смещение, скорость обучения и количество нейронов в скрытом слое были значительно меньше. При этом случайный выбор примеров улучшил результаты работы, его использование предпочтительно для избегания привыкания сети.

Оптимальные параметры, выбранные для решения данной задачи:

Число входов: 2; Число выходов: 1; Циклов обучения: 1000; Нормализация: [0;1]; Выбор примеров: случайный; Крутизна функции: 0,5; Смещение: 0,1; Скорость обучения: 0,01; Количество нейронов в скрытом слое: 10.

Работе алгоритма ОРО была проверена ручным расчетом одного прохода. Вычисленные вручную значения совпали практически точно, разница с рассчитанными автоматически возникла из-за округлений и погрешностей в вычислениях.

Приложение А  
(обязательное)  
Обучающая выборка

X1	X2	OUT
-7	-7,4	-522696,8751
-6,94975	3,94271	-738090,9682
-6,8995	-7,28543	-403528,9204
-6,84925	3,82814	-635175,9701
-6,79899	-7,17085	-297229,1432
-6,74874	3,71357	-536941,4516
-6,69849	-7,05628	-204016,461
-6,64824	3,59899	-444844,3944
-6,59799	-6,94171	-123740,6191
-6,54774	3,48442	-359949,3605
-6,49749	-6,82714	-55978,64166
-6,44724	3,36985	-282966,8848
-6,39698	-6,71256	-74,96713
-6,34673	3,25528	-214296,9087
-6,29648	-6,59799	44787,75149
-6,24623	3,1407	-154072,2114
-6,19598	-6,48342	79569,25085
-6,14573	3,02613	-102196,7925
-6,09548	-6,36884	105309,4088
-6,04523	2,91156	-58376,86894
-5,99497	-6,25427	123080,3033
-5,94472	2,79698	-22167,97489
-5,89447	-6,1397	133976,1935
-5,84422	2,68241	6979,71511
-5,79397	-6,02513	139062,843
-5,74372	2,56784	29706,70016
-5,69347	-5,91055	139363,2579
-5,64322	2,45327	46700,03717
-5,59296	-5,79598	135829,992
-5,54271	2,33869	58670,69882
-5,49246	-5,68141	129346,793
-5,44221	2,22412	66327,02421
-5,39196	-5,56683	120702,6507
-5,34171	2,10955	70361,06727
-5,29146	-5,45226	110592,2695
-5,24121	1,99497	71429,63099
-5,19095	-5,33769	99614,74583
-5,1407	1,8804	70139,10354
-5,09045	-5,22312	88277,15253
-5,0402	1,76583	67046,21564
-4,98995	-5,10854	76990,31312
-4,9397	1,65126	62642,90954
-4,88945	-4,99397	66080,08719
-4,8392	1,53668	57358,69579
-4,78894	-4,8794	55792,2337
-4,73869	1,42211	51557,50109
-4,68844	-4,76482	46303,58843
-4,63819	1,30754	45544,10614
-4,58794	-4,65025	37726,37133
-4,53769	1,19296	39561,29899
-4,48744	-4,53568	30119,88063
-4,43719	1,07839	33797,78117
-4,38693	-4,42111	23498,85105
-4,33668	0,96382	28392,43606
-4,28643	-4,30653	17843,39968
-4,23618	0,84925	23441,85561

-4,18593	-4,19196	13105,48734
-4,13568	0,73467	19003,4949
-4,08543	-4,07739	9216,93683
-4,03518	0,6201	15104,20335
-3,98492	-3,96281	6096,17711
-3,93467	0,50553	11744,98162
-3,88442	-3,84824	3654,87135
-3,83417	0,39095	8907,70497
-3,78392	-3,73367	1801,03712
-3,73367	0,27638	6559,1404
-3,68342	-3,6191	444,1912
-3,63317	0,16181	4656,0058
-3,58291	-3,50452	-501,98615
-3,53266	0,04724	3148,96329
-3,48241	-3,38995	-1117,11818
-3,43216	-0,06734	1986,0849
-3,38191	-3,27538	-1473,10419
-3,33166	-0,18191	1115,32692
-3,28141	-3,1608	-1632,92887
-3,23116	-0,29648	486,6109
-3,1809	-3,04623	-1650,24003
-3,13065	-0,41106	53,47404
-3,0804	-2,93166	-1569,86407
-3,03015	-0,52563	-225,82462
-2,9799	-2,81709	-1427,86626
-2,92965	-0,6402	-387,934
-2,8794	-2,70251	-1252,46424
-2,82915	-0,75477	-464,01492
-2,77889	-2,58794	-1064,79792
-2,72864	-0,86935	-479,85114
-2,67839	-2,47337	-880,0853
-2,62814	-0,98392	-456,14676
-2,57789	-2,35879	-708,39597
-2,52764	-1,09849	-408,97507
-2,47739	-2,24422	-555,74867
-2,42714	-1,21307	-350,33418
-2,37688	-2,12965	-425,00151
-2,32663	-1,32764	-288,73175
-2,27638	-2,01508	-316,71158
-2,22613	-1,44221	-229,8625
-2,17588	-1,9005	-229,76888
-2,12563	-1,55678	-177,12586
-2,07538	-1,78593	-162,0524
-2,02513	-1,67136	-132,21563
-1,97487	-1,67136	-110,88735
-1,92462	-1,78593	-95,5844
-1,87437	-1,55678	-73,42696
-1,82412	-1,9005	-66,86833
-1,77387	-1,44221	-46,88857
-1,72362	-2,01508	-45,18989
-1,67337	-1,32764	-28,744
-1,62312	-2,12965	-29,43026
-1,57286	-1,21307	-16,81315
-1,52261	-2,24422	-18,40935
-1,47236	-1,09849	-9,30654
-1,42211	-2,35879	-11,01432
-1,37186	-0,98392	-4,81652
-1,32161	-2,47337	-6,26803
-1,27136	-0,86935	-2,28751
-1,22111	-2,58794	-3,36861
-1,17085	-0,75477	-0,96483
-1,1206	-2,70251	-1,69358
-1,07035	-0,6402	-0,33708

-1,0201	-2,81709	-0,78664
-0,96985	-0,52563	-0,07771
-0,9196	-2,93166	-0,33175
-0,86935	-0,41106	0,00681
-0,8191	-3,04623	-0,1239
-0,76884	-0,29648	0,02101
-0,71859	-3,1608	-0,03944
-0,66834	-0,18191	0,01458
-0,61809	-3,27538	-0,01003
-0,56784	-0,06734	0,00674
-0,51759	-3,38995	-0,00179
-0,46734	0,04724	0,00223
-0,41709	-3,50452	-0,00015
-0,36683	0,16181	0,0005
-0,31658	-3,6191	0,00002
-0,26633	0,27638	0,00006
-0,21608	-3,73367	0
-0,16583	0,39095	0
-0,11558	-3,84824	0
-0,06533	0,50553	0
-0,01508	-3,96281	0
0,03518	0,6201	0
0,08543	-4,07739	0
0,13568	0,73467	0
0,18593	-4,19196	0
0,23618	0,84925	-0,00004
0,28643	-4,30653	-0,00011
0,33668	0,96382	-0,00048
0,38693	-4,42111	-0,00098
0,43719	1,07839	-0,00305
0,48744	-4,53568	-0,00537
0,53769	1,19296	-0,01298
0,58794	-4,65025	-0,02141
0,63819	1,30754	-0,04252
0,68844	-4,76482	-0,06815
0,73869	1,42211	-0,11532
0,78894	-4,8794	-0,18374
0,8392	1,53668	-0,27055
0,88945	-4,99397	-0,43561
0,9397	1,65126	-0,5648
0,98995	-5,10854	-0,93123
1,0402	1,76583	-1,0692
1,09045	-5,22312	-1,82724
1,1407	1,8804	-1,85782
1,19095	-5,33769	-3,33312
1,24121	1,99497	-2,98383
1,29146	-5,45226	-5,70509
1,34171	2,10955	-4,43793
1,39196	-5,56683	-9,22311
1,44221	2,22412	-6,08784
1,49246	-5,68141	-14,14836
1,54271	2,33869	-7,592
1,59296	-5,79598	-20,65084
1,64322	2,45327	-8,28893
1,69347	-5,91055	-28,70461
1,74372	2,56784	-7,06055
1,79397	-6,02513	-37,9388
1,84422	2,68241	-2,17491
1,89447	-6,1397	-47,46008
1,94472	2,79698	8,88804
1,99497	-6,25427	-55,62039
2,04523	2,91156	29,61734
2,09548	-6,36884	-59,75805

2,14573	3,02613	64,63213
2,19598	-6,48342	-55,89512
2,24623	3,1407	119,83345
2,29648	-6,59799	-38,45149
2,34673	3,25528	202,49188
2,39698	-6,71256	0,07775
2,44724	3,36985	321,25855
2,49749	-6,82714	69,39518
2,54774	3,48442	486,05696
2,59799	-6,94171	181,59046
2,64824	3,59899	707,88496
2,69849	-7,05628	351,29544
2,74874	3,71357	998,39797
2,79899	-7,17085	595,64843
2,84925	3,82814	1369,32592
2,8995	-7,28543	934,14248
2,94975	3,94271	1831,56394
3	-7,4	1388,07332