

МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«Вятский государственный университет»
(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники
Кафедра ЭВМ

Исследование многослойного персептрона с обучением по методу с
обратным распространением ошибки

Отчёт
Лабораторная работа № 1-2 по дисциплине
«Системы обработки знаний»

Выполнил студент группы ИВТб-41 _____ /Крючков И.С./
Проверил доцент кафедры ЭВМ _____ /Ростовцев В.С./

Киров 2024

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 $Y = X_1 + X_2^2 + X_3$. Переменная x_1 изменяются в промежутке $[-5;4]$, переменная x_2 изменяется в промежутке $[-5;4]$, переменная x_3 изменяется в промежутке $[-6;4]$.

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: 3.

Число выходов: 1.

Циклов обучения: 10000

В таблице 1 показаны результаты исследования.

Таблица 1 – Влияние нормализации на эффективность алгоритма ОРО

Тип нормализации	Результат
Без нормализации	Максимальная ошибка: 62,00 Минимальная ошибка: 0,00 Средняя ошибка: 31,94 Среднеквадратичная ошибка: 629418
[0;1]	Максимальная ошибка: 2,28 Минимальная ошибка: 0,000305 Средняя ошибка: 0,28 Среднеквадратичная ошибка: 85,01
[-0.5;0.5]	Максимальная ошибка: 22,00 Минимальная ошибка: 9,30 Средняя ошибка: 8,31 Среднеквадратичная ошибка: 52270,43
[-1;1]	Максимальная ошибка: 44,00 Минимальная ошибка: 0,00 Средняя ошибка: 18,01 Среднеквадратичная ошибка: 221427,99

Применение нормализации [0;1] значительно сокращает величину ошибок. Результат лучше примерно в 8 раз для средних и, приблизительно в 6 раз для среднеквадратичных ошибок при нормализации [-0,5;0,5] (наиболее близкой к лучшей).

2.2 Влияние выбора примеров на эффективность алгоритма ОРО

Число входов: 3.

Число выходов: 1.

Циклов обучения: 10000.

Нормализация: [0;1].

В таблице 2 показаны результаты исследования.

Таблица 2 – Влияние выбора примеров на эффективность алгоритма ОРО

Выбор примеров	Результат
Последовательный	Максимальная ошибка: 2,41 Минимальная ошибка: 0,00051 Средняя ошибка: 0,32 Среднеквадратичная ошибка: 109,12
Случайный	Максимальная ошибка: 2,34 Минимальная ошибка: 0,0000037 Средняя ошибка: 0,25 Среднеквадратичная ошибка: 74,88

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

2.3 Влияние крутизны функции на эффективность алгоритма ОРО

Число входов: 3.

Число выходов: 1.

Циклов обучения: 10000.

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: 4,68 Минимальная ошибка: 0,00016 Средняя ошибка: 0,76 Среднеквадратичная ошибка: 603,19
0,5	Максимальная ошибка: 2,57 Минимальная ошибка: 0,00029 Средняя ошибка: 0,32 Среднеквадратичная ошибка: 104,45
1,0	Максимальная ошибка: 1,86 Минимальная ошибка: 0,00011 Средняя ошибка: 0,26 Среднеквадратичная ошибка: 68,89
3,0	Максимальная ошибка: 1,4 Минимальная ошибка: 0,0032 Средняя ошибка: 0,25 Среднеквадратичная ошибка: 41,13

Наилучшие результаты – при крутизне функции 3,0. Остальные параметры также лучше при [3;0].

2.4 Влияние смещения на эффективность алгоритма ОРО

Число входов: 3.

Число выходов: 1.

Циклов обучения: 10000.

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 1,0.

В таблице 4 показаны результаты исследования.

Таблица 4 – Влияние смещения на эффективность алгоритма ОРО

Смещение	Результат
0,1	Максимальная ошибка: 2,37 Минимальная ошибка: 0,00016 Средняя ошибка: 0,26 Среднеквадратичная ошибка: 79,99
0,5	Максимальная ошибка: 2,33 Минимальная ошибка: 0,00092 Средняя ошибка: 0,32 Среднеквадратичная ошибка: 102,02
1,0	Максимальная ошибка: 2,30 Минимальная ошибка: 0,00019 Средняя ошибка: 0,21 Среднеквадратичная ошибка: 62,01
2,0	Максимальная ошибка: 1,57 Минимальная ошибка: 0,00046 Средняя ошибка: 0,26 Среднеквадратичная ошибка: 61,41

Наилучшие результаты наблюдаются при смещениях 1,0 и 2,0. Наименьшие минимальная ошибка наблюдается при смещении 1,0, а наименьшие максимальная, среднеквадратическая и средняя ошибки – при смещении 2,0.

2.5 Влияние скорости обучения на эффективность алгоритма ОРО

Число входов: 3.

Число выходов: 1.

Циклов обучения: 10000.

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 1,0.

Смещение: 1,0.

В таблице 5 показаны результаты исследования.

Таблица 5 – Влияние скорости обучения на эффективность алгоритма ОРО

Скорость обучения	Результат
0,01	Максимальная ошибка: 4,27 Минимальная ошибка: 0,0000095 Средняя ошибка: 0,68 Среднеквадратичная ошибка: 480,39
0,1	Максимальная ошибка: 4,48 Минимальная ошибка: 0,00031 Средняя ошибка: 0,047 Среднеквадратичная ошибка: 283,80
0,5	Максимальная ошибка: 1,76 Минимальная ошибка: 1,09 Средняя ошибка: 0,13 Среднеквадратичная ошибка: 27,47
1,0	Максимальная ошибка: 1,60 Минимальная ошибка: 0,00057 Средняя ошибка: 0,19 Среднеквадратичная ошибка: 37,51

Увеличение скорости обучения с 0,01 до 0,5 ведет к снижению средней и среднеквадратичной ошибки. При увеличении с 0,5 до 1,0 – средняя и среднеквадратичная ошибка возрастает. Наилучшие результаты наблюдаются при скорости 0,5.

2.6 Влияние количества нейронов в скрытом слое на эффективность алгоритма ОРО

Число входов: 3.

Число выходов: 1.

Циклов обучения: 10000.

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 1,0.

Смещение: 1,0.

Скорость обучения: 0.5.

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: 19,08 Минимальная ошибка: 0,04 Средняя ошибка: 4,40 Среднеквадратичная ошибка: 18978,54
5	Максимальная ошибка: 1,74 Минимальная ошибка: 0,00021 Средняя ошибка: 0,15 Среднеквадратичная ошибка: 29,36
10	Максимальная ошибка: 1,47 Минимальная ошибка: 0,00028 Средняя ошибка: 0,085 Среднеквадратичная ошибка: 10,94

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

2.7 Влияние момента на эффективность алгоритма ОРО

Число входов: 3.

Число выходов: 1.

Циклов обучения: 10000.

Нормализация: [0;1].

Выбор примеров: случайный.

Крутизна функции: 1,0.

Смещение: 1,0.

Скорость обучения: 0,5.

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: 1,57 Минимальная ошибка: 0,00042 Средняя ошибка: 0,15 Среднеквадратичная ошибка: 25,63
0,1	Максимальная ошибка: 1,60 Минимальная ошибка: 0,00023 Средняя ошибка: 0,095 Среднеквадратичная ошибка: 14,82
0,5	Максимальная ошибка: 43,9 Минимальная ошибка: 0,00022 Средняя ошибка: 25,99 Среднеквадратичная ошибка: 414053,47
1,0	Максимальная ошибка: 43,99 Минимальная ошибка: 2,00 Средняя ошибка: 25,99 Среднеквадратичная ошибка: 414059,94

Наилучшие результаты наблюдаются при обучении без момента и обучении с моментом 0,1. Наименьшие максимальная, средняя и среднеквадратичная ошибки наблюдаются при обучении без момента, а наименьшая минимальная ошибка – при обучении с моментом 0,1.

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.

Инициализация весов синапсов случайным образом...	Обратная волна - подсчет локальной ошибки нейронов...
Нейрон[1][1] w[1, 1, 1] = -0,246 w[1, 1, 2] = 0,158 w[1, 1, 3] = -0,11 Вес смещения: w[1, 1, 4] = 1	Подсчет локальной ошибки нейронов на выходе нейронной сети... Желаемый сигнал на выходе: 0,340909 Прогнозируемый сигнал на выходе нейронной сети: 0,4747516288
Нейрон[1][2] w[1, 2, 1] = 0,942 w[1, 2, 2] = -0,676 w[1, 2, 3] = 0,246 Вес смещения: w[1, 2, 4] = 1	Нейрон[2][1] Локальная ошибка = 0,03337533518
Нейрон[1][3] w[1, 3, 1] = -0,226 w[1, 3, 2] = -0,194 w[1, 3, 3] = -0,41 Вес смещения: w[1, 3, 4] = 1	Подсчет локальной ошибки нейронов в скрытых слоях нейронной сети... Нейрон[1][1] Локальная ошибка = -0,0009672148606
Нейрон[1][4] w[1, 4, 1] = 0,814 w[1, 4, 2] = 0,968 w[1, 4, 3] = -0,202 Вес смещения: w[1, 4, 4] = 1	Нейрон[1][2] Локальная ошибка = -0,0006357329363
Нейрон[1][5] w[1, 5, 1] = 0,18 w[1, 5, 2] = 0,168 w[1, 5, 3] = -0,326 Вес смещения: w[1, 5, 4] = 1	Нейрон[1][3] Локальная ошибка = -0,004103828684
Нейрон[2][1] w[2, 1, 1] = -0,144 w[2, 1, 2] = -0,104 w[2, 1, 3] = -0,574 w[2, 1, 4] = -0,266 w[2, 1, 5] = -0,412 Вес смещения: w[2, 1, 6] = 1	Нейрон[1][4] Локальная ошибка = -0,001317145153
Выбираем допустимый образ из обучающего множества... 0,318182 0,295455 0,181818 0,340909 Подаем сигнал на вход нейронной сети... Нейрон[0][1] Аксон = 0,318182	Нейрон[1][5] Локальная ошибка = -0,00264350404
Нейрон[0][2] Аксон = 0,295455	Коррекция весов синапсов...
Нейрон[0][3] Аксон = 0,181818	w[1, 1, 1] = -0,2456922496 w[1, 1, 2] = 0,1582857685 w[1, 1, 3] = -0,1098241429 Вес смещения: w[1, 1, 4] = 1,000967215
Прямая волна...	w[1, 2, 1] = 0,9422022788 w[1, 2, 2] = -0,6758121695 w[1, 2, 3] = 0,2461155877 Вес смещения: w[1, 2, 4] = 1,000635733
Нейрон[1][1] Взвешенная сумма = 0,948409138 Аксон = 0,7207951303	w[1, 3, 1] = -0,2246942356 w[1, 3, 2] = -0,1927875033 w[1, 3, 3] = -0,4092538501 Вес смещения: w[1, 3, 4] = 1,004103829
Нейрон[1][2] Взвешенная сумма = 1,144727092 Аксон = 0,7585464813	w[1, 4, 1] = 0,8144190919 w[1, 4, 2] = 0,9683891571 w[1, 4, 3] = -0,2017605193 Вес смещения: w[1, 4, 4] = 1,001317145
Нейрон[1][3] Взвешенная сумма = 0,796227218 Аксон = 0,6891668686	w[1, 5, 1] = 0,1808411154 w[1, 5, 2] = 0,1687810365 w[1, 5, 3] = -0,3255193634 Вес смещения: w[1, 5, 4] = 1,002643504
Нейрон[1][4] Взвешенная сумма = 1,508273352 Аксон = 0,8188051767	w[2, 1, 1] = -0,1680567791 w[2, 1, 2] = -0,1293167431 w[2, 1, 3] = -0,5970011752 w[2, 1, 4] = -0,2933278972 w[2, 1, 5] = -0,4367084545 Вес смещения: w[2, 1, 6] = 0,9666246648
Нейрон[1][5] Взвешенная сумма = 1,047636532 Аксон = 0,7403207902	
Нейрон[2][1] Взвешенная сумма = -0,101079458 Аксон = 0,4747516288	

Рисунок 1 – Трассировка первого прохода

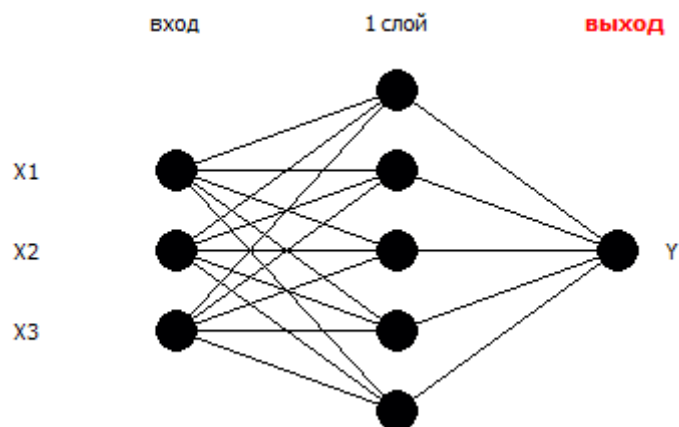


Рисунок 2 – Структура сети

В качестве активационной функции взята

$$F(S_i) = \frac{1}{1 + \exp[-s]} \quad (1)$$

где S_i – взвешенная сумма входов i -го нейрона (с учетом смещения);

В таблице показан 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	0,318182	-	-	-	-	-	0,318182
	2	1	0,295455	-	-	-	-	-	0,340909
	3	1	0,181818	-	-	-	-	-	0,340909
1	1	1	0,318182	-0,246	1	1	-0,078272772	0,948409138	0,72079513
		2	0,295455	0,158			0,04668189		
		3	0,181818	-0,11			-0,01999998		
	2	1	0,318182	0,942	1	1	0,299727444	1,144727092	0,758546481
		2	0,295455	-0,676			-0,19972758		
		3	0,181818	0,246			0,044727228		
	3	1	0,318182	-0,226	1	1	-0,071909132	0,796227218	0,689166869
		2	0,295455	-0,194			-0,05731827		
		3	0,181818	-0,41			-0,07454538		
	4	1	0,318182	0,814	1	1	0,259000148	1,508273352	0,818805177
		2	0,295455	0,968			0,28600044		
		3	0,181818	-0,202			-0,036727236		
	5	1	0,318182	0,18	1	1	0,05727276	1,047636532	0,74032079
		2	0,295455	0,168			0,04963644		
		3	0,181818	-0,326			-0,059272668		
Выход	1	1	0,72079513	-0,144	1	1	-0,103794499	-0,101079458	0,474751629
		2	0,75854648	-0,104			-0,078888834		
		3	0,68916687	-0,574			-0,395581783		
		4	0,81880518	-0,266			-0,217802177		
		5	0,74032079	-0,412			-0,305012166		

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

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

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 γ_j – ошибка выходного слоя;

ω_{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	-0,10107946	0,24936252	0,033375335
1	1	0,94840914	0,20124951	-0,000967215
	2	1,14472709	0,18315372	-0,000635733
	3	0,79622722	0,2142159	-0,004103829
	4	1,50827335	0,14836326	-0,001317145
	5	1,04763653	0,19224592	-0,002643504

Коррекция веса синапса производится по следующей формуле:

$$\omega_{ij}(t + 1) = \omega_{ij}(t) - \gamma_j F'(S_j) y_j.$$

Коррекция веса смещения производится по следующей формуле:

$$T_j(t + 1) = T_j(t) - \gamma_j F'(S_j).$$

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	-0,246	1	-0,245805348	1,000194652
		2	0,158		0,158194652	
		3	-0,11		-0,109805348	
	2	1	0,942	1	0,942116437	1,000116437
		2	-0,676		-0,675883563	
		3	0,246		0,246116437	
	3	1	-0,226	1	-0,225120895	1,000879105
		2	-0,194		-0,193120895	
		3	-0,41		-0,409120895	
	4	1	0,814	1	0,814195416	1,000195416
		2	0,968		0,968195416	
		3	-0,202		-0,201804584	
	5	1	0,18	1	0,180508203	1,000508203
		2	0,168		0,168508203	
		3	-0,326		-0,325491797	
Выход	1	1	-0,144	1	-0,152322558	0,991677442
		2	-0,104		-0,112322558	
		3	-0,574		-0,582322558	
		4	-0,266		-0,274322558	
		5	-0,412		-0,420322558	

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: 3; Число выходов: 1; Циклов обучения: 10000; Нормализация: [0;1]; Выбор примеров: случайный; Крутизна функции: 3,0; Смещение: 0,5; Скорость обучения: 1; Количество нейронов в скрытом слое: 5.

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	X3	OUT
2.00000	-1.00000	-6.00000	-3.00000
2.00000	4.00000	2.00000	20.00000
-4.00000	-1.00000	-6.00000	-9.00000
-4.00000	-4.00000	0.00000	12.00000
-2.00000	3.00000	4.00000	11.00000
4.00000	-1.00000	4.00000	9.00000
-1.00000	-3.00000	-2.00000	6.00000
-4.00000	-4.00000	-6.00000	6.00000
-3.00000	-4.00000	0.00000	13.00000
-4.00000	2.00000	0.00000	0.00000
3.00000	3.00000	-1.00000	11.00000
-3.00000	0.00000	1.00000	-2.00000
4.00000	-2.00000	0.00000	8.00000
2.00000	-1.00000	-3.00000	0.00000
3.00000	-5.00000	-5.00000	23.00000
-3.00000	-1.00000	2.00000	0.00000
-1.00000	-2.00000	2.00000	5.00000
2.00000	4.00000	1.00000	19.00000
1.00000	4.00000	-2.00000	15.00000
2.00000	-4.00000	-2.00000	16.00000
3.00000	-4.00000	0.00000	19.00000
-3.00000	1.00000	3.00000	1.00000
-1.00000	2.00000	-4.00000	-1.00000
-4.00000	0.00000	4.00000	0.00000
4.00000	3.00000	-4.00000	9.00000
2.00000	4.00000	3.00000	21.00000
2.00000	0.00000	1.00000	3.00000
3.00000	4.00000	4.00000	23.00000
2.00000	-2.00000	-3.00000	3.00000
0.00000	2.00000	4.00000	8.00000
4.00000	-1.00000	-1.00000	4.00000
-5.00000	-4.00000	4.00000	15.00000
-5.00000	-3.00000	-6.00000	-2.00000
2.00000	-1.00000	-6.00000	-3.00000
2.00000	4.00000	2.00000	20.00000
-4.00000	-1.00000	-6.00000	-9.00000
-4.00000	-4.00000	0.00000	12.00000
-2.00000	3.00000	4.00000	11.00000
4.00000	-1.00000	4.00000	9.00000
-1.00000	-3.00000	-2.00000	6.00000
-4.00000	-4.00000	-6.00000	6.00000
-3.00000	-4.00000	0.00000	13.00000
-4.00000	2.00000	0.00000	0.00000
3.00000	3.00000	-1.00000	11.00000
-3.00000	0.00000	1.00000	-2.00000
4.00000	-2.00000	0.00000	8.00000
2.00000	-1.00000	-3.00000	0.00000
3.00000	-5.00000	-5.00000	23.00000
-3.00000	-1.00000	2.00000	0.00000
-1.00000	-2.00000	2.00000	5.00000