Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчёт**

**по лабораторной работе №15**

**Дисциплина: НЕЙРОСЕТЕВЫЕ И НЕЧЕТКИЕ МОДЕЛИ**

**Тема: «Настройка параметров персептронных нейронных сетей и реализация правил настройки в системе MATLAB»**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. А. Корнилов

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение

компьютерных технологий

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А. А. Крамаренко

**Тема:** Настройка параметров персептронных нейронных сетей и реализация правил настройки в системе Matlab.

**Цель:** Изучение процедуры настройки параметров персептронных нейронных сетей и реализация правил настройки в системе MATLAB.

**Задания:**

1. Для заданного преподавателем варианта задания выполнить ручной расчет настройки весов и смещений персептронной нейронной сети.



1. Разработать алгоритм создания и моделирования персептронной нейронной сети
2. Реализовать алгоритм в системе MATLAB
3. Определить параметры созданной нейронной сети и проверить правильность работы сети для последовательности входных векторов
4. Сравнить результаты ручных расчетов и расчетов, выполненных в системе MATLAB

**Ход работы:**

1. Был выполнен ручной расчет настройки весов и смещений персептронной нейронной сети с весами [1 -0.8] и смещением b=[1]

Эпоха 1:

1. x = [-2 1] target = 0

S = 1 \* -2 - 0.8\*1 + 1 = -1.8

Y = 0

e =target – Y = 0

w = w + 0.1 \* 0 \* [-2 1].T = w

b = b + 0.1 \* 0 = b

1. x = [0 1] target = 0

S = 1 \* 0 – 0.8 \* 1 + 1 = 0.2

Y = 1

e = target - Y = -1

w = w + 0.1 \* -1 \* [0, 1].T = [1 + 0 ,-0.8 -0.1 ] = [1,-0.9]

b = b + 0.1 \* -1 = 1 – 0.1 = 0.9

1. x = [2 -1] target = 1

S = 1 \* 2 – 0.9 \* - 1 + 0.9 = 3.8

Y = 1

e = target - Y = 0

w = w + 0.1 \* 0 \* X.T = w

b = b + 0.1 \* 0 = w

1. x = [-2 -1] target = 0

S = 1 \* -2 – 0.9 \* -1 + 0.9 = -2

Y = 0

e = target - Y = 0

w = w + 0.1 \* 0 \* X.T = w

b = b + 0.1 \* 0 = b

Эпоха 2:

1. x = [-2 1] target = 0

S = 1 \* -2 - 0.9\*1 + 0.9 = -2

Y = 0

e =target – Y = 0

w = w + 0.1 \* 0 \* [-2 1].T = w

b = b + 0.1 \* 0 = b

1. x = [0 1] target = 0

S = 1 \* 0 – 0.9\* 1 + 0.9 = 0

Y = 1

e = target - Y = -1

w = w + 0.1 \* -1 \* [0, 1].T = [1, -0.9-0.1] = [1, -1]

b = b + 0.1 \* -1 = 0.9 – 0.1 = 0.8

1. x = [2 -1] target = 1

S = 1 \* 2 – 1 \* - 1 + 0.8 = 3.8

Y = 1

e = target - Y = 0

w = w + 0.1 \* 0 \* X.T = w

b = b + 0.1 \* 0 = w

1. x = [-2 -1] target = 0

S = 1 \* -2 – 1 \* -1 + 0.8= -2.2

Y = 0

e = target - Y = 0

w = w + 0.1 \* 0 \* X.T = w

b = b + 0.1 \* 0 = b

Эпоха 3:

1. x = [-2 1] target = 0

S = 1 \* -2 - 1\*1 + 0.8 = -2.2

Y = 0

e =target – Y = 0

w = w + 0.1 \* 0 \* [-2 1].T = w

b = b + 0.1 \* 0 = b

1. x = [0 1] target = 0

S = 1 \* 0 – 1\* 1 + 0.8 = -0.2

Y = 0

e = target - Y = 0

w = w + 0.1 \* 0 \* [0, 1].T = w

b = b + 0.1 \* -0 = b

1. x = [2 -1] target = 1

S = 1 \* 2 – 1 \* - 1 + 0.8 = 3.8

Y = 1

e = target - Y = 0

w = w + 0.1 \* 0 \* X.T = w

b = b + 0.1 \* 0 = w

1. x = [-2 -1] target = 0

S = 1 \* -2 – 1 \* -1 + 0.8= -2.2

Y = 0

e = target - Y = 0

w = w + 0.1 \* 0 \* X.T = w

b = b + 0.1 \* 0 = b

Итоговые веса и смещение: [-1 1] [0.8]

1. Был реализован алгоритм обучения нейронной сети на базе MATLAB
2. X = [-2 1; 0 1; 2 -1; -2 -1;];
3. target = [ 0; 0; 1; 0;];
4. w = [1 -0.8];
5. b = [1];
6. max\_error = 0.01;
7. learning\_rate = 0.1;
8. num\_epochs = 10;
9. net = newp([-3 3; -3 3],1);
10. net.b{1} = b;
11. net.IW{1,1} = w;
12. for epoch = 1:num\_epochs
13. total\_error = 0;
14. for i = 1:size(X,1)
15. a = sim(net,X(i,:)')
16. e = target(i) - a
17. dw = learnp(w,X(i,:),[],[],[],[],e,[],[],[],[],[]);
18. w = w + learning\_rate \* dw';
19. net.b{1} = net.b{1} + learning\_rate \* e;
20. net.IW{1,1} = w;
21. total\_error = total\_error+abs(e);
22. end;
23. average\_error = total\_error/size(X,1);
24. disp(['Эпоха ' num2str(epoch) ', Средняя ошибка: ' num2str(average\_error)]);
25. if average\_error < max\_error
26. disp(['Обучение завершено на эпохе ' num2str(epoch)]);
27. break;
28. end;
29. end;
30. disp('Веса:');
31. disp(w);
32. disp('Смещение:');
33. disp(b);

Результат обучения с помощью алгоритма на matlab совпадает с ручным расчетом вплоть до количества эпох и значений весов и смещения.

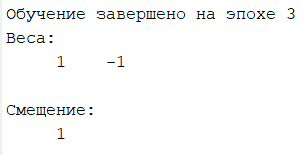


Рисунок 1 – Результат работы алгоритмы

3. Было проведено тестирование нейронной сети на входных векторах. Нейронная сеть полностю аппроксимировала функцию заданную входными обучающими значениями.

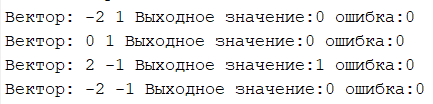


Рисунок 2 – Результаты тестирования.

**Вывод:** Были изучены процедуры настройки параметров персептронных нейронных сетей и реализация правил настройки в системе MATLAB.