**ЛАБОРАТОРНАЯ РАБОТА 2**

**Изучение основных функций пакета Neural Network Toolbox**

Цель: Ознакомиться с основными командами создания, обучения и применения многослойных нейронных сетей в Neural Network Toolbox для аппроксимации заданной функции.

**Задание**.   
  
1. Ознакомиться с демонстрационными программами персептрона.

Создать в рабочем пространстве MATLAB многослойную нейронную сеть с прямой передачей данных и исследовать ее структуру с оценкой качества обучения mse.   
2. Обучить сеть в заданной число циклов (до 100 циклов).

3. Построить графики зависимостей обучения нейросетей от количества циклов.

1. **Порядок выполнения лабораторной работы**

1.Для входа в программу Neural Network Toolbox в командной строке среды MatLab ввести:

1. >>nntool

>>help nndemos

Ознакомиться с примерами создания в среде MATLAB многослойной нейронной сети.

1. Создать скрипт-файл и выполнить моделирование многослойной нейронной сети для входных и выходных векторов, которые были сформированы в 7 семестре по дисциплине ТПР при разработке фрагмента нейросетевой экспертной системы в Excel.
2. Рекомендуемая оценка качества обучения нейронных сетей *mse* основана на функциях оценки качества, выбираемых из списка {mae| mse | sse}. Допускается для оценки качества применение функции RMSE.  
   mse - функция производительности сети. Это определяет эксплуатационные качества сети согласно среднему значению ошибок в квадрате.

mae - функция производительности сети. Это определяет эксплуатационные качества сети согласно среднему значению абсолютных ошибок.

sse - функция производительности сети. Это определяет эксплуатационные качества согласно сумме ошибок в квадрате.

5. Исследовать результаты многослойной нейронной сети для различных архитектур многослойной нейронной сети (трехслойная и четырех слойная нейронная сеть) с разными функциями активации и разными функциями обучения нейронной сети.

6. Подготовить отчёт по результатам выполнения лабораторной работы, включив в него созданные m-файлы, обучающую выборку и отразить в нем результаты исследования многослойной нейронной сети по п.5.

Пример отчета представлен в приложении А.

**2. Сведения о моделировании нейронных сетей и примеры моделирования**

### **2.1 Единичная функция активации с жестким ограничением *hardlim.***

Эта функция описывается соотношением

*a* = hardlim(*n*) = 1(*n*)

и равна 0, если n < 0,

и равна 1, если n ≥ 0.

Построим график этой функции в диапазоне значений входа от -5 до + 5:

n = -5:0.1:5;

plot(n,hardlim(n),'b+:');

### **Линейная функция активации *purelin*.**

Эта функция описывается соотношением

*a* = purelin(*n*) = *n*.

Построим график этой функции (рис.1) в диапазоне значений входа от -5 до + 5:

n=-5:0.1:5;

plot(n,purelin(n),'b+:');

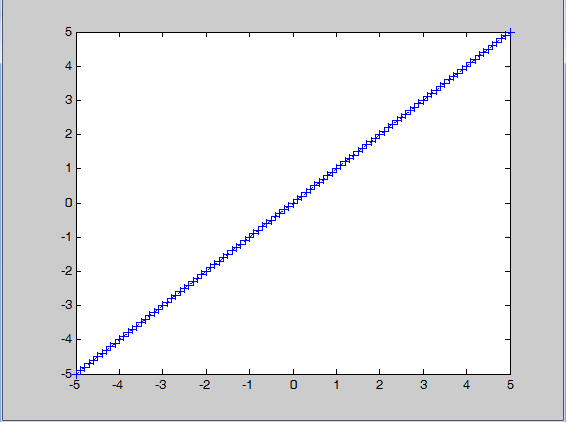


Рис.1

### C34. Логистическая функция активации *logsig*.

Эта функция описывается соотношением

*a* = logsig(*n*) = 1/(1 + exp(-*n*)).

Она принадлежит к классу сигмоидальных функций, и ее аргумент может принимать любое значение в диапазоне от -∞ до +∞, а выход изменяется в диапазоне от 0 до 1. Благодаря свойству дифференцируемости, эта функция часто используется в сетях с обучением на основе метода обратного распространения ошибки.

Построим график этой функции в диапазоне значений входа от -5 до + 5:

n=-5:0.1:5;

plot(n,logsig(n),'b+:');

### C42. Формирование архитектуры нейронной сети.

Следующий оператор создает сеть с прямой передачей сигнала (рис.2)

net = newff([-1 2; 0 5],[3,1],{'tansig','purelin'},'traingd');

gensim(net)

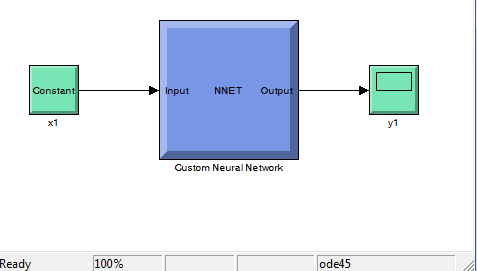


Рис.2

Эта сеть использует один вектор входа с двумя элементами, имеющими допустимые границы значений [-1 2] и [0 5];

- имеет 2 слоя с 3 нейронами в первом слое и 1 нейроном во втором слое;

- используемые функции активации: tansig - в первом слое, purelin – во втором слое;

* используемая функция обучения traingd.

### **2.3 Инициализация нейронной сети**

После того как сформирована архитектура сети, должны быть заданы начальные значения весов и смещений, или иными словами, сеть должна быть инициализирована. Такая процедура выполняется с помощью метода init для объектов класса network. Оператор вызова этого метода имеет вид

net = init(net);

Если мы хотим заново инициализировать веса и смещения в первом слое, используя функцию rands, то следует ввести следующую последовательность операторов:

net.layers{1}.initFcn = 'initwb';

net.inputWeights{1,1}.initFcn = 'rands';

net.biases{1,1}.initFcn = 'rands';

net.biases{2,1}.initFcn = 'rands';

net = init(net);

### **2.5** **Моделирование сети**

**Разработка структуры нейронной сети**

**Многослойная нейронная сеть**

В ППП NNT для создания многослойных сетей предусмотрена М-

функция newff, обращение к которой имеет следующий вид:

net = newff

net = newff(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF)

Описание:

net = newff создает новую сеть с использованием диалогового окна

NEWFF(PR, [S1 S2...SNl], {TF1 TF2...TFNl}, BTF, LF, PF) в качестве входных

параметров использует:

- PR - Rx2 матрица минимальных и максимальных значений строк

входной матрицы с размерностью RxQ. Для получения матрицы PR

можно использовать функцию minmax;

- S i – количество нейронов в i – ом слое, N1 – количество слоев;

- TF i - функция активации i - го слоя, по умолчанию = 'tansig';

- BTF – обучающая функция обратного распространения, по

умолчанию='trainlm';

- PF - функция оценки функционирования сети, по умолчанию = 'mse';

и возвращает однонаправленную сеть, состоящую из N слоев. По умолчанию

методом тренировки был взят метод градиентного спуска.

Создание многослойной нейронной сети для аппроксимации функции

cos(x):

clear

X=[-3.14:0.01:3.14]; // формирование входных массивов

Y=cos(X);

57

net=newff(minmax(X),[10,1,1],{'tansig','purelin','tansig'},'traingd'); // создание

сети с тремя слоями нейронов;

net.trainParam.epochs=3000; // максимальное количество циклов обучения

[net,tr]=train(net,X,Y); // обучение сети

test=sim(net,X); // обработка тестового массива

plot(X,Y,'+r',X,test,'-g'); hold on;

xx=[3.05 2.87 2.61 1.53 0.61]; // набор входных данных, взятых для

исследования

v=sim(net,xx); // обработка

plot(xx,v,'ob','MarkerSize',5,'LineWidth',2);

Приложение А

Пример отчета

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

ИЗУЧЕНИЕ ОСНОВНЫХ ФУНКЦИЙ ПАКЕТА NEURAL NETWORK TOOLBOX ДЛЯ ПОСТРОЕНИЯ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА

Отчёт

Лабораторная работа №2 по дисциплине

«Системы обработки знаний»

Вариант 7

Выполнил студент группы ВТб4301\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Степанов И.В./

Проверил доцент кафедры ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ростовцев В.С./

Киров 2021

1 Цель работы

Цель:Ознакомиться с основными командами создания, обучения и применения многослойных нейронных сетей с прямой передачей в Neural Network Toolbox для аппроксимации заданной функции.

2 Постановка задачи

Для выполнения лабораторной работы необходимо выполнить следующие задачи:

1. Выполнить моделирование многослойной нейронной сети для аппроксимации функции y=x17sin(x2+3), x1∈[-5 5], x2∈[-4 4]
2. Исследовать результаты аппроксимации для различных архитектур многослойной нейронной сети (двух и трехслойная нейронная сеть) с разными функциями активации и разными функциями обучения нейронной сети.
3. Вычислить среднеквадратичную ошибку аппроксимации (sse или mse) и сравнить по этому критерию разные типы нейронных сетей

3 Ход работы

Выполнить моделирование многослойной НС для аппроксимации функции y=x17sin(x2+3), x1∈[-5 5], x2∈[-4 4]. Вычислить показатель mse.

Для обучения НС использовалось три функции:

1. traingd – функция обучения НС с использованием алгоритма градиентного спуска GD.
2. trainlm – функция обучения НС с использованием алгоритма Левенберга-Марквардта LM.
3. trainbr – функция обучения НС с использованием алгоритма Левенберга-Марквардта, дополненная регуляцией по Байесу B

В соответствии со значениями средней квадратичной ошибки, полученными в результате предыдущей лабораторной работы, для нейронной сети были выбраны функции активации poslin и tansig.

Для создания нейронной сети была выбрана функция NEWFF(PR, [S1 S2...SNl], {TF1 TF2...TFNl}, BTF, LF, PF), которая в качестве входных параметров использует:

* PR - массив размера Rх2 минимальных и максимальных значений для R векторов входа.
* Si – количество нейронов в i – ом слое, N1 – количество слоев;
* TFi - функция активации i - го слоя, по умолчанию = 'tansig';
* BTF – обучающая функция обратного распространения, по умолчанию='trainlm';
* BLF – алгоритм подстройки весов и смещений (обучающий алгоритм), по умолчанию = 'learngdm';
* PF - функция оценки функционирования сети, по умолчанию = 'mse'.

Функция возвращает однонаправленную сеть, состоящую из N слоев.

Оценка качества обучения нейронных сетей основана на функциях оценки качества, выбираемых из списка {mae| mse | sse}.

mse - функция производительности сети. Это определяет эксплуатационные качества сети согласно среднему значению ошибок в квадрате.

mae - функция производительности сети. Это определяет эксплуатационные качества сети согласно среднему значению абсолютных ошибок.

sse - функция производительности сети. Это определяет эксплуатационные качества согласно сумме ошибок в квадрате.

%Формирование значений функций в этом интервале

x = 1;

for i = -5:.5:5;

for j = -4:.5:4;

P(1,x) = i;

P(2,x) = j;

T(x) = i^7\*sin(j+3);

x = x+1;

end;

end;

%Создание многослойных нейронных сетей для аппроксимации

net = newff([-5 5; -4 4],[13,1,1],{'tansig','tansig','tansig'},'traingd');

%Тренировка сети и получение результатов работы сети

net.trainParam.epochs = 1000;

[net,tr] = train(net, P, T);

Yn=sim(net,P);

plot(P,Yn,'ok','MarkerSize',7,'LineWidth',1);

%Построение графика функций

figure('NumberTitle','off','Name','Функция','ToolBar','none','MenuBar','none');

plot(P,T,'-k','LineWidth',1);hold on; plot(P,T,'--k','LineWidth',2);hold on;

A=net(P);

E=T-A;

SSE=sumsqr(E)

MSE=mse(E)

Результаты аппроксимации функции НС с различными архитектурами, функциями обучения и функциями активации приведены в таблице 1.

| Архитектура НС | | | Функция  активации | Обучающая  функция | SSE | MSE |
| --- | --- | --- | --- | --- | --- | --- |
| 1 слой | 2 слой | 3 слой |  |  |  |  |
| 15 | 1 | 1 | poslin | traingd | 5.0876e+12 | 43.1882 |
| poslin | trainlm | 5.0876e+12 | 43.1882 |
| ***poslin*** | ***trainbr*** | ***1.2035e+12*** | ***33.7131*** |
| tansig | traingd | 4.5149e+12 | 38.3265 |
| tansig | trainlm | 4.6442e+12 | 39.4244 |
| tansig | trainbr | 4.5156e+12 | 38,3325 |
| 15 | 5 | 1 | poslin | traingd | 5.0870e+12 | 43.1837 |
| poslin | trainlm | 5.0876e+12 | 43.1882 |
| poslin | trainbr | 5.0876e+12 | 43.1882 |
| **tansig** | **traingd** | ***1.2035e+12*** | ***33.7131*** |
| tansig | trainlm | 4.5301e+12 | 38.4554 |
| tansig | trainbr | 4.5118e+12 | 38.3008 |
| 5 | 1 | 1 | poslin | traingd | 5.0876e+12 | 43.1882 |
| ***poslin*** | ***trainlm*** | ***1.2035e+12*** | ***33.7131*** |
| poslin | trainbr | 5.0876e+12 | 43.1882 |
| tansig | traingd | 4.5387e+12 | 38.5285 |
| tansig | trainlm | 4.4907e+12 | 38.1212 |
| tansig | trainbr | 4.5161e+12 | 38.3372 |
| 21 | 1 | 1 | *poslin* | *traingd* | 5.0876e+12 | 43.1882 |
| ***poslin*** | ***trainlm*** | **1.2035e+12** | ***33.7131*** |
| poslin | trainbr | 5.0876e+12 | 43.1882 |
| tansig | traingd | 4.5073e+12 | 38.2627 |
| tansig | trainlm | 5.0215e+12 | 42.6277 |
| tansig | trainbr | 4.5156e+12 | 38.3324 |
| 32 | 1 | 1 | ***poslin*** | ***traingd*** | ***1.2035e+12*** | ***33.7131*** |
| poslin | trainlm | 4.3770e+12 | 37.1561 |
| poslin | trainbr | 5.0876e+12 | 43.1882 |
| tansig | traingd | 4.5022e+12 | 38.2190 |
| tansig | trainlm | 4.4905e+12 | 38.1200 |
| tansig | trainbr | 5.0180e+12 | 42.5977 |
| 32 | 2 | 1 | poslin | traingd | 5.0876e+12 | 43.1882 |
| poslin | trainlm | 5.0876e+12 | 43.1882 |
| poslin | trainbr | 5.0876e+12 | 43.1882 |
| **tansig** | **traingd** | ***1.2035e+12*** | ***33.7131*** |
| tansig | trainlm | 5.1697e+12 | 43.8854 |
| tansig | trainbr | 4.5011e+12 | 38.2096 |
| 32 | 7 | 1 | poslin | traingd | 5.0876e+12 | 43.1882 |
| ***poslin*** | ***trainlm*** | ***7.1841e+11*** | ***20.1248*** |
| poslin | trainbr | 5.0876e+12 | 43.1882 |
| tansig | traingd | 4.4974e+12 | 38.1785 |
| tansig | trainlm | 4.4905e+12 | 38.1196 |
| tansig | trainbr | 4.5011e+12 | 38.2097 |

Таблица 1 – Значения среднеквадратичной ошибки аппроксимации

Данные и результаты обучения для наиболее оптимальной сети представлены на рисунках 1-6.

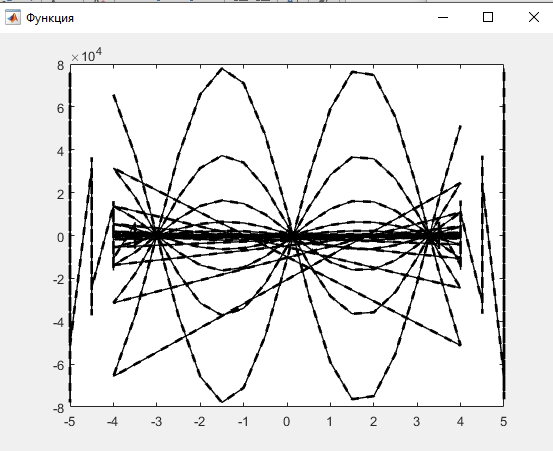


Рисунок 1 – График функции

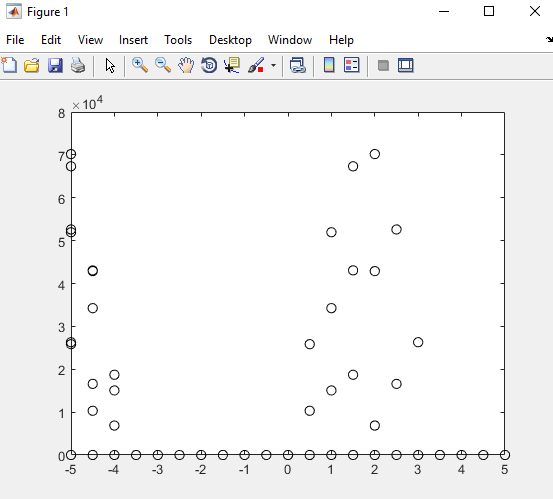


Рисунок 2 – Результат симуляции

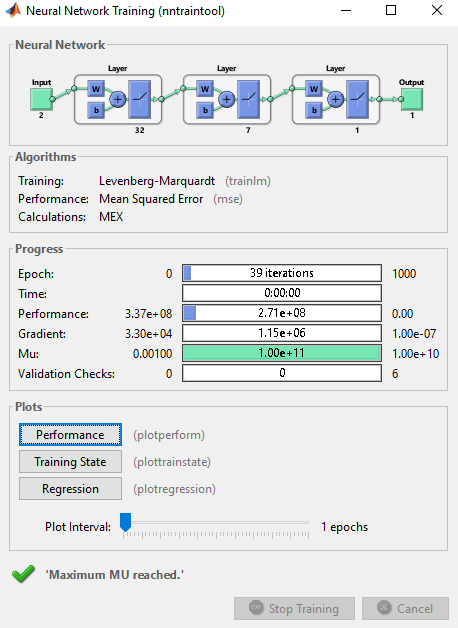


Рисунок 3 – Структура и параметры обучения наиболее производительной сети

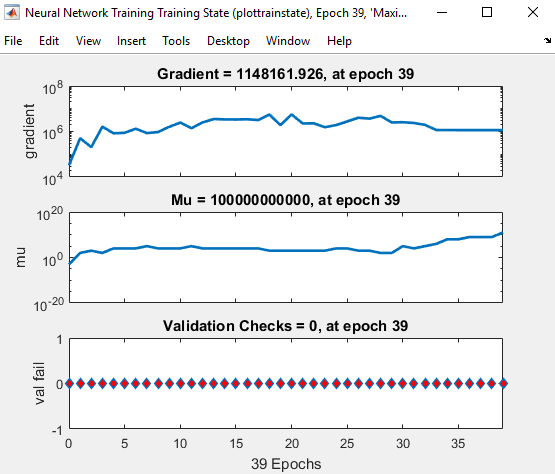


Рисунок 4 – Диаграммы тренировки сети

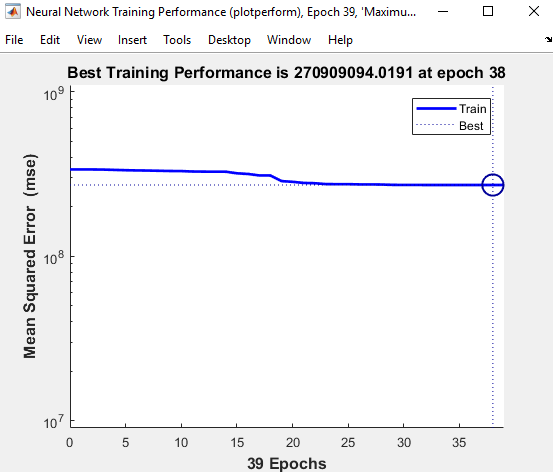


Рисунок 5 – Диаграмма производительности тренировки сети

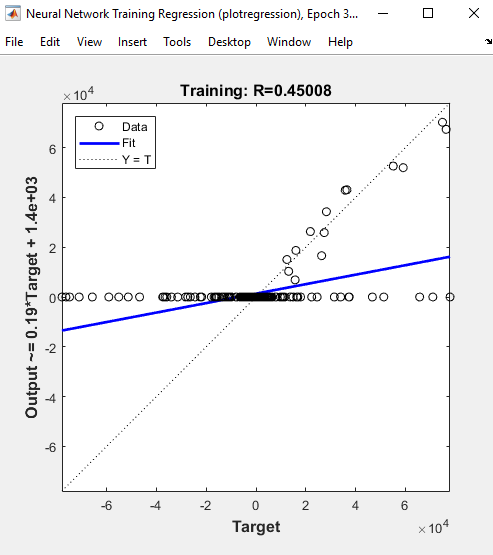


Рисунок 6 – Диаграммы регрессии созданной нейронной сети

Выводы

В ходе лабораторной работы были получены навыки построения многослойных нейронных сетей с помощью пакета Neural Toolbox для Matlab. В результате создана трехслойная нейронная сеть для задачи аппроксимации тригонометрической функции y=x17sin(x2+3) в области x1∈[-5; 5], x2∈[-4; 4]. Были использованы различные варианты архитектуры сети, функции обучения, функций активации и подсчитана средняя квадратичная ошибка.

При увеличении количества нейронов на втором слое или уменьшении на первом слое при использовании функции обучения trainbr качество аппроксимации ухудшается, а при использовании traingd улучшается. Для функций trainlm качество улучшается с увеличением количества нейронов на втором слое.

По результатам экспериментов минимальная среднеквадратичная ошибка у нейронной сети со архитектурой: 32 нейрона во входном слое, 7 нейронов в скрытом слое и 1 нейрон в выходном слое. В качестве функции активации для этой сети использовалась функция poslin, функции обучения - trainlm.