Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования

информационных систем

Направление подготовки математическое обеспечение и администрирование

информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе № 9**

«Многослойные нейронные сети. Система МАТЛАБ.»

дисциплина «Основы теории нейронных сетей»

Выполнила:

студентка группы 413 Мусонда Салиму

Проверил:

Профессор кафедры ПОиАИС Добрица В. П.

Курск, 2020

Вариант 3

Задача № 1. Выбрать параметры и провести обучение нейронной сети для прогнозирования временного ряда задачи 1 из лабораторной работы №8 в системе MATLAB. Функция .

Функция активации нейронов скрытого слоя:

Функция активации нейрона выходного слоя:

Формулы:

Среднеквадратичная ошибка, где L – количество входных образов, – эталонное значение, а - вычисленное:

Выход нейрона скрытого слоя, где n – количество нейронов входного слоя, в данном случае n = 6:

Выход сети, где h – количество нейронов скрытого слоя, в данном случае h = 10:

Изменение весовых коэффициентов и смещения выходного слоя:

Где под понимается значения весовых коэффициентов и смещения на предыдущем шаге.

Изменение весовых коэффициентов и смещений скрытого слоя:

Где под понимается значения весовых коэффициентов и смещений на предыдущем шаге.

Ошибка нейрона выходного слоя:

, где – эталонное значение, а - вычисленное

Взвешенная сумма нейрона выходного слоя, где h – количество нейронов скрытого слоя, в данном случае h = 10:

Взвешенная сумма j-го нейрона скрытого слоя, где n – количество нейронов входного слоя, в данном случае n = 6:

Ошибка i-го нейрона скрытого слоя:

Изменение весовых коэффициентов и смещений нейронной сети происходит по следующим формулам:

Где под понимается значения весовых коэффициентов и смещений на предыдущем шаге.

Адаптивный шаг обучения для скрытого слоя:

Адаптивный шаг обучения для выходного слоя:

В качестве входных значений сначала выбираются элементы временного ряда с 0 по 5, затем с 1 по 6 и т. д. последнее от 988 до 993.

Таким образом, получается 989 обучающих примеров, в первом примере выходное значение равно 6-му элементу ряда, в последнем – 994-му. Последовательность от 994 до 999 используется для проверки в качестве входных данных, а вычисленное 1000-ое значение сравнивается с 1000-ым значением временного ряда для доказательства обученности сети. Так как все входные значения должны лежать в промежутке от 0 до 1, каждое входное значение делится на одно и то же число. В нашем случае на 21, так как < 21 для любого t от 0 до 100.

Блок-схема создания нейросети в MATLAB приведена на рисунке 1.

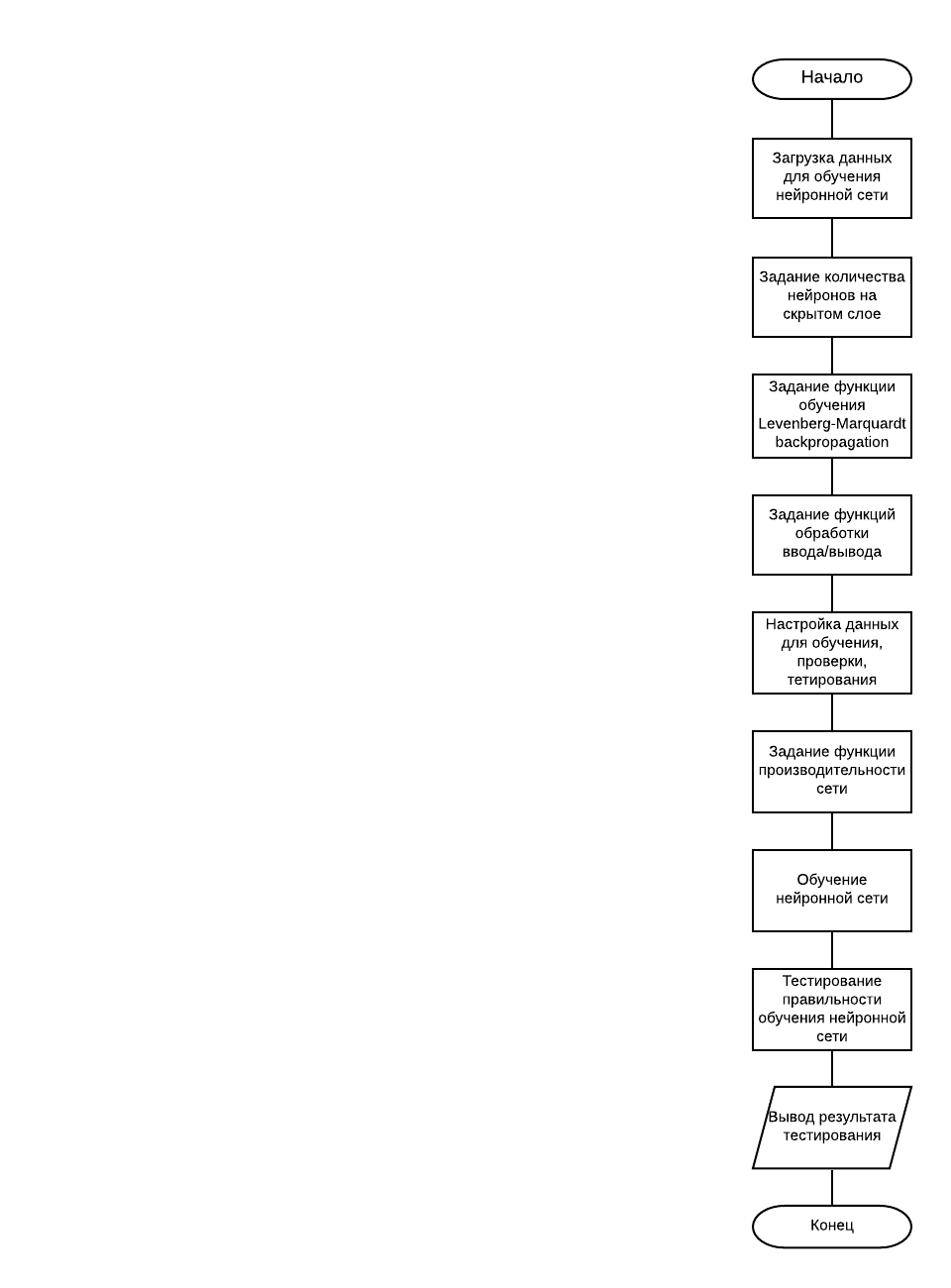


Рисунок 1 - Блок-схема создания нейросети в MATLAB

Обучение нейронной сети с 6 входными нейронами, 10 нейронами на скрытом слое и 1 выходным нейроном представлено на рисунке 2.

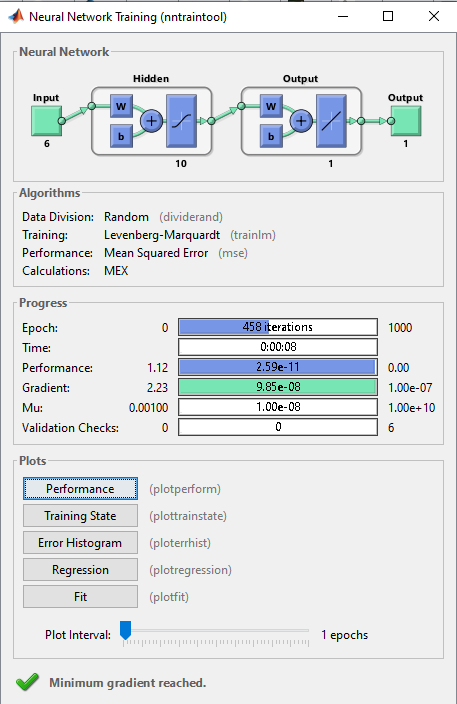


Рисунок 4 – Обучение нейронной сети

Листинг программы

% Solve an Input-Output Fitting problem with a Neural Network

% Script generated by Neural Fitting app

% Created 23-Apr-2020 06:45:02

%

% This script assumes these variables are defined:

%

% inputs - input data.

% outputs - target data.

x = inputs';

t = outputs';

% Choose a Training Function

% For a list of all training functions type: help nntrain

% 'trainlm' is usually fastest.

% 'trainbr' takes longer but may be better for challenging problems.

% 'trainscg' uses less memory. Suitable in low memory situations.

trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network

hiddenLayerSize = 10;

net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions

% For a list of all processing functions type: help nnprocess

net.input.processFcns = {'removeconstantrows','mapminmax'};

net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing

% For a list of all data division functions type: help nndivision

net.divideFcn = 'dividerand'; % Divide data randomly

net.divideMode = 'sample'; % Divide up every sample

net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;

% Choose a Performance Function

% For a list of all performance functions type: help nnperformance

net.performFcn = 'mse'; % Mean Squared Error

% Choose Plot Functions

% For a list of all plot functions type: help nnplot

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...

'plotregression', 'plotfit'};

% Train the Network

[net,tr] = train(net,x,t);

% Test the Network

y = net(x);

e = gsubtract(t,y);

performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance

trainTargets = t .\* tr.trainMask{1};

valTargets = t .\* tr.valMask{1};

testTargets = t .\* tr.testMask{1};

trainPerformance = perform(net,trainTargets,y)

valPerformance = perform(net,valTargets,y)

testPerformance = perform(net,testTargets,y)

% View the Network

view(net)

% Plots

% Uncomment these lines to enable various plots.

%figure, plotperform(tr)

%figure, plottrainstate(tr)

%figure, ploterrhist(e)

%figure, plotregression(t,y)

%figure, plotfit(net,x,t)

% Deployment

% Change the (false) values to (true) to enable the following code blocks.

% See the help for each generation function for more information.

if (false)

% Generate MATLAB function for neural network for application

% deployment in MATLAB scripts or with MATLAB Compiler and Builder

% tools, or simply to examine the calculations your trained neural

% network performs.

genFunction(net,'myNeuralNetworkFunction');

y = myNeuralNetworkFunction(x);

end

if (false)

% Generate a matrix-only MATLAB function for neural network code

% generation with MATLAB Coder tools.

genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');

y = myNeuralNetworkFunction(x);

end

if (false)

% Generate a Simulink diagram for simulation or deployment with.

% Simulink Coder tools.

gensim(net);

end

Тестирование созданной нейронной сети приведено на рисунке 3.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3 – Тестирование созданной нейронной сети

Вывод: По результатам тестирования можно сделать вывод, что значение, вычисленное нейронной сетью 20.9947, отличается от эталонного значения 20.9948 не более чем на 0.0001, что говорит о правильности обучения нейронной сети.