

Цель работы	2
Содержание	2
Постановка задачи	2
Исходные данные	2
Алгоритм решения	3
Построение сети	3
Обучение сети	5
Результаты обучения	9
Выводы	12

Цель работы

Изучение основных свойств и основ работы с GUI – интерфейсом пакета Neural Networks Toolbox в программной среде MatLab.

Содержание

Постановка задачи

Построить и натренировать нейронную сеть для вычисления функции

$$y = x_1 * \sin(x_2), x_1, x_2 \in [-1; 1]$$

Исходные данные

В качестве исходных данных с помощью написанного скрипта были сгенерированы значения для x_1 и x_2 , а затем по ним вычислены соответствующие значения y :

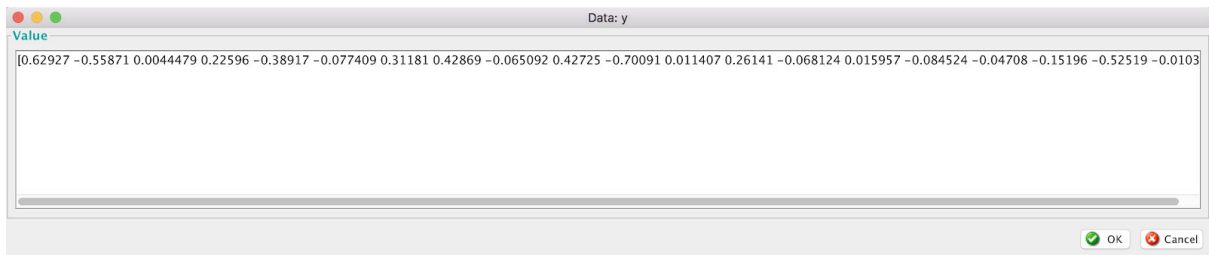
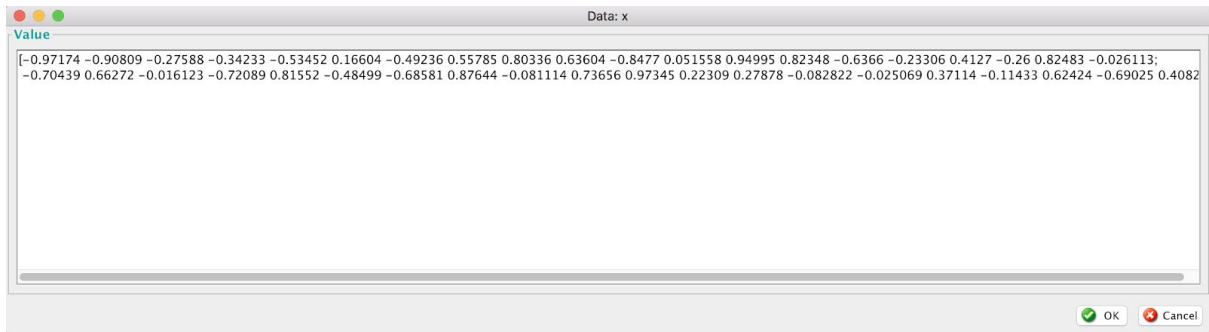
```
x = [-0.971744075369 -0.908085502026 -0.275879929245 -0.34233462519
-0.534516732717 0.166042363216 -0.492364276638 0.557852348271
0.803364729692 0.63603628916 -0.847704356355 0.0515582795791
0.949948818347 0.8234815387 -0.636596416475 -0.233058037265
0.412697908595 -0.259996319331 0.82482797877 -0.026113028086 ;
-0.704394141359 0.662720300264 -0.0161233750095 -0.720889931336
0.815524377274 -0.48498938254 -0.685808227007 0.87643916787
-0.081113665884 0.736559535742 0.973447793537 0.223086935397
0.278783480788 -0.0828220159893 -0.0250693227568 0.371137173967
-0.114327560673 0.624237227564 -0.690246562882 0.408250344609]
```

```
y = [0.629274528104 -0.55871195223 0.00444792283491 0.225959155974
-0.389173696763 -0.0774087781133 0.311813661833 0.428690121478
-0.0650924249649 0.427253084799 -0.700907259645 0.0114068110504
0.261412902368 -0.0681244556635 0.0159573694507 -0.0845244206891
-0.0470800263705 -0.151962264571 -0.52519051128 -0.0103669785159]
```

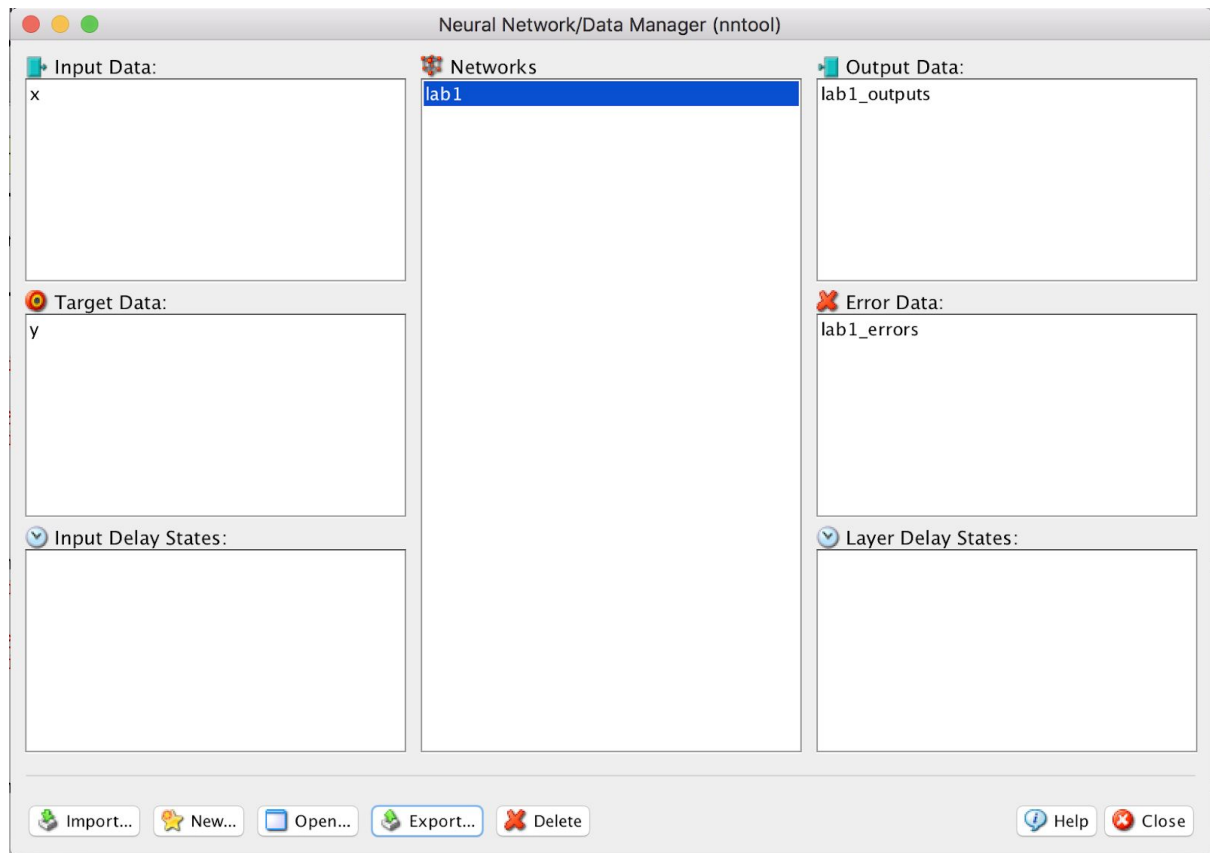
Алгоритм решения

Построение сети

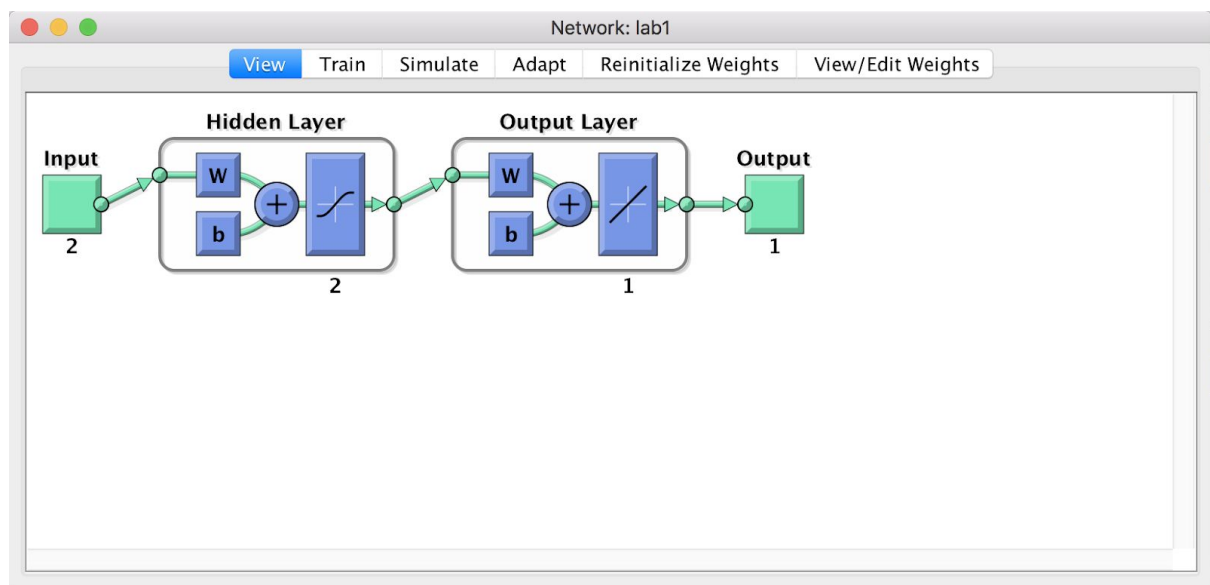
1. С помощью функции **nntool** откроем основное окно интерфейса и сформируем последовательность входов и целей в рабочей области.
2. Внесем исходные данные (*Input Data* и *Target Data*) в окно *Neural Network/Data Manager*.



3. Добавим сеть во вкладке *Networks*

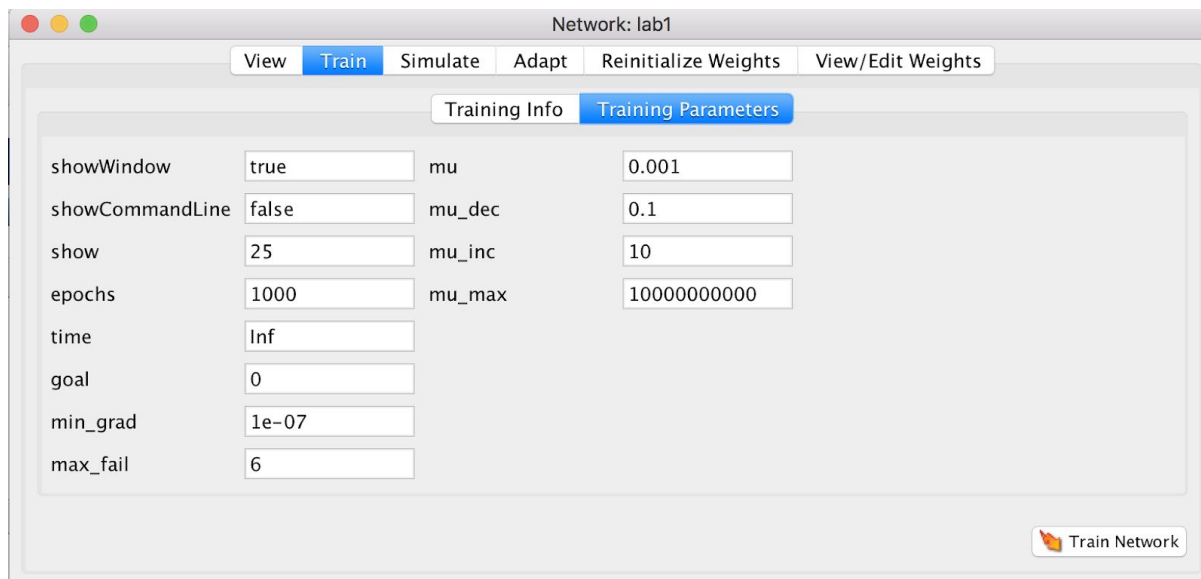
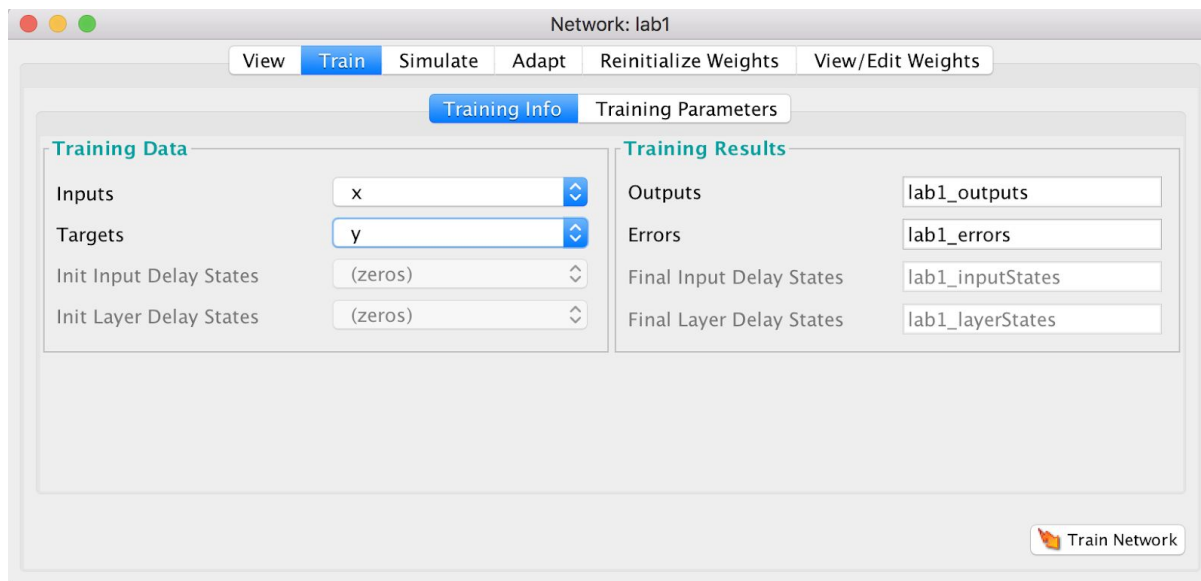


4. Зайдем в настройки сети. Выберем ее тип *Feed-forward backprop* с прямой передачей сигнала и обратным распространением ошибки.
5. Добавим сети *Input Data*, *Target Data*, количество нейронов первого слоя равно 2, функция активации во втором слое - линейная (*PURELIN*).



Обучение сети

Установим значения параметров во вкладке Train, как сказано в условии:



И нажмем *Train Network*. Качество обучения сети на выбранной обучающей последовательности показан на следующих графиках:

Neural Network Training (nntraintool)

Neural Network

The diagram shows a neural network architecture with three layers: an Input layer with 2 nodes, a Hidden Layer with 2 nodes, and an Output Layer with 1 node. Each layer is represented by a box containing a weight matrix (W) and a bias vector (b), which are added together and passed through an activation function (represented by a blue box with a curve). The input layer is connected to the hidden layer, which is connected to the output layer.

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	9 iterations	1000
Time:		0:00:00	
Performance:	0.0392	0.0264	0.00
Gradient:	0.0916	0.000234	1.00e-07
Mu:	0.00100	0.000100	1.00e+10
Validation Checks:	0	6	6

Plots

Performance

(plotperform)

Training State

(plottrainstate)

Regression

(plotregression)

Plot Interval:

1 epochs

Validation stop.

Stop Training

Cancel

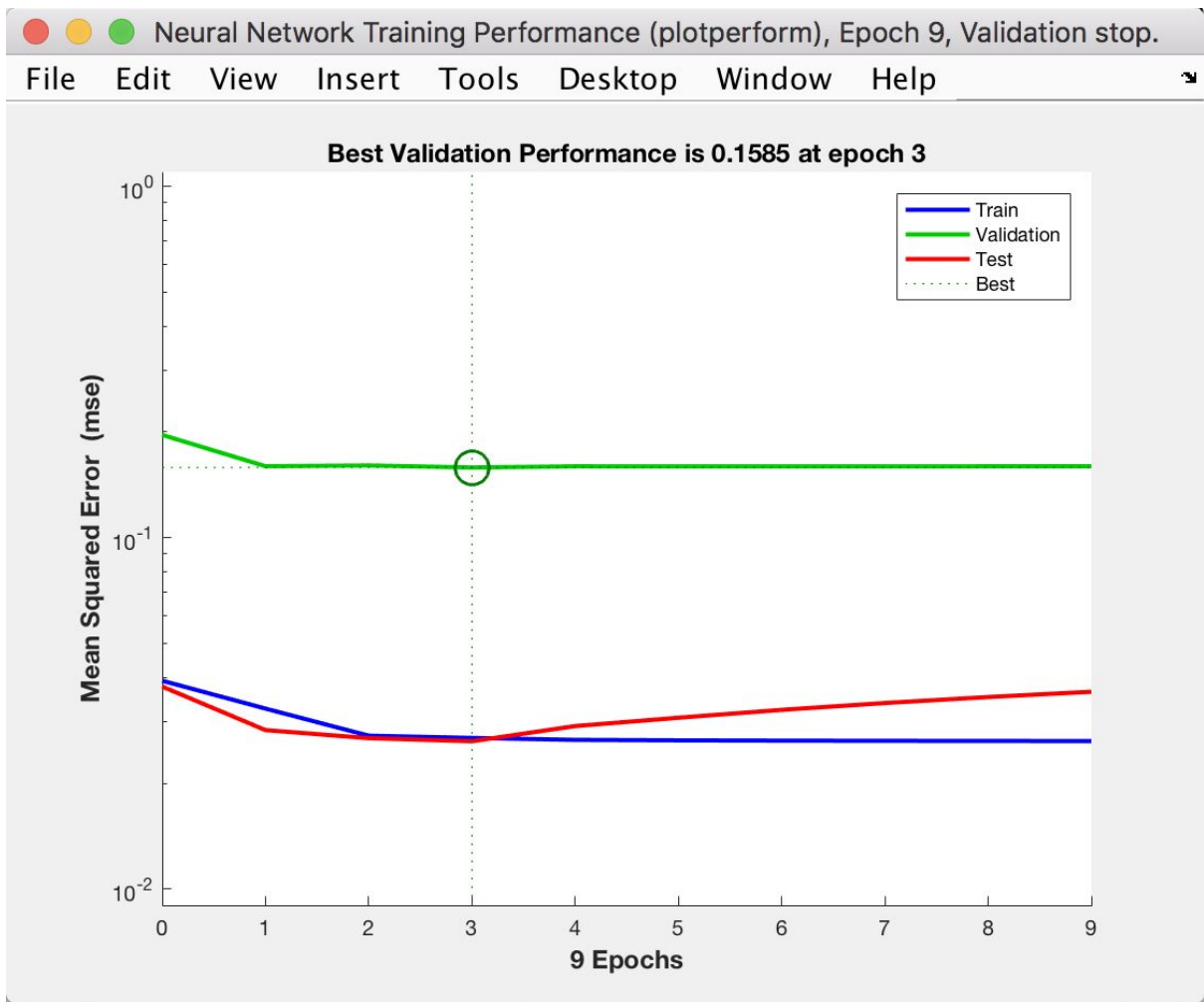
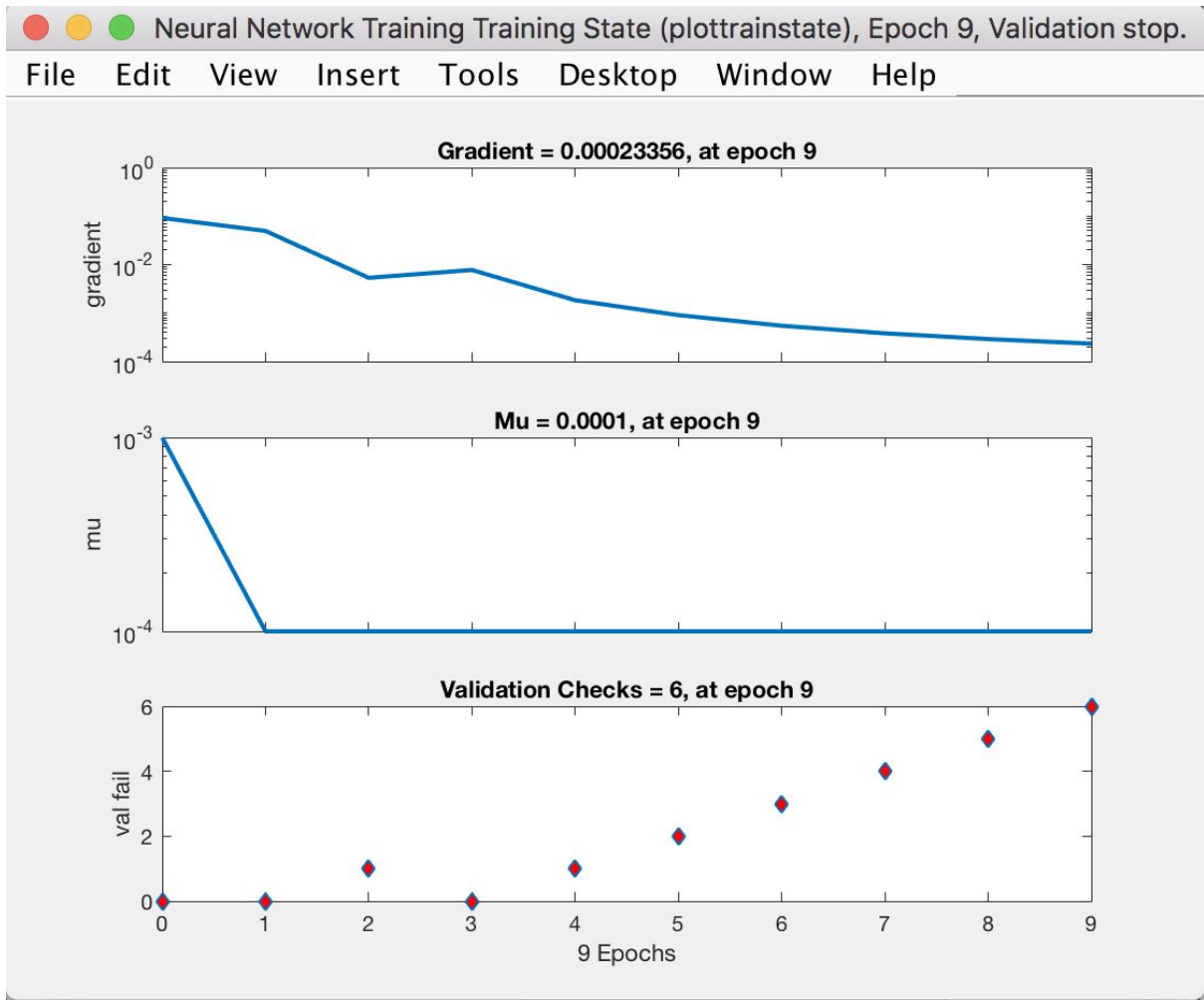


График Performance



Γραφικ Training State

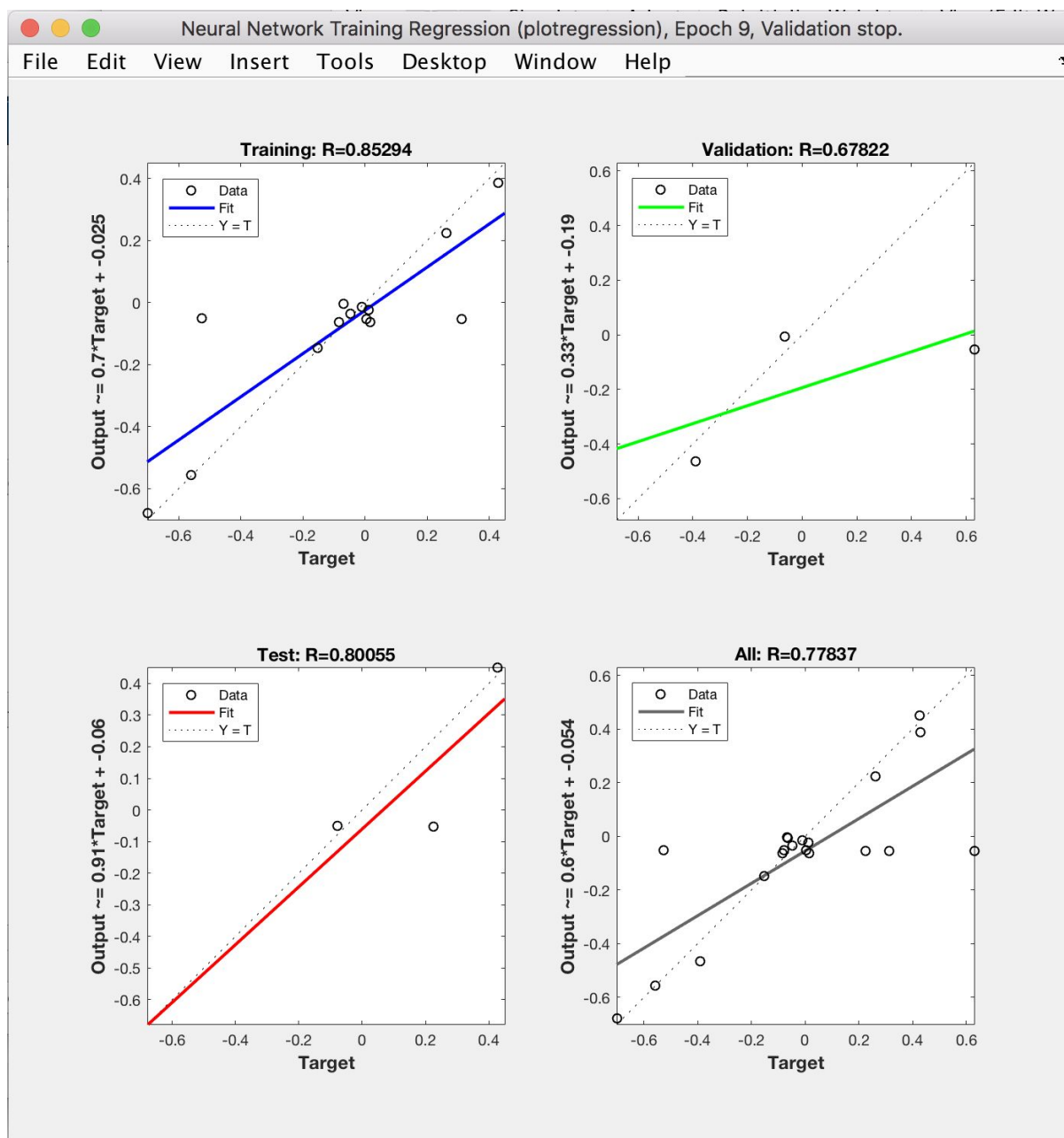


График Regression

Результаты обучения

Результаты обучения сети можно посмотреть в окне *Network/Data Manager*, а именно выбирая параметры *lab1_outputs* и *lab1_errors*:

lab1_outputs = [-0.053503 -0.55667 -0.052302 -0.053126 -0.4651 -0.051539
-0.053213 0.3876 -0.005608 0.45023 -0.67868 -0.022636 0.22348 -0.0040626
-0.061654 -0.062779 -0.035276 -0.1466 -0.050654 -0.014699]

```
lab1_errors = [0.68278 -0.002041 0.05675 0.27909 0.075925 -0.025869
0.36503 0.041095 -0.059484 -0.022976 -0.022227 0.034042 0.037935 -0.064062
0.077612 -0.021746 -0.011804 -0.0053586 -0.47454 0.0043324]
```

Кроме того, мы можем выгрузить полученные результаты в рабочую область, что позволяет печатать/обрабатывать значения в дальнейшем:

```
Workspace | Run
y =
Columns 1 through 13
    0.6293    -0.5587    0.0044    0.2260   -0.3892   -0.0774    0.3118    0.4287   -0.0651    0.4273   -0.7009    0.0114    0.2614
Columns 14 through 20
   -0.0681    0.0160   -0.0845   -0.0471   -0.1520   -0.5252   -0.0104

>> lab1_errors
lab1_errors =
Columns 1 through 13
    0.6828   -0.0020    0.0568    0.2791    0.0759   -0.0259    0.3650    0.0411   -0.0595   -0.0230   -0.0222    0.0340    0.0379
Columns 14 through 20
   -0.0641    0.0776   -0.0217   -0.0118   -0.0054   -0.4745    0.0043

>> lab1_outputs
lab1_outputs =
Columns 1 through 13
   -0.0535   -0.5567   -0.0523   -0.0531   -0.4651   -0.0515   -0.0532    0.3876   -0.0056    0.4502   -0.6787   -0.0226    0.2235
Columns 14 through 20
   -0.0041   -0.0617   -0.0628   -0.0353   -0.1466   -0.0507   -0.0147
```

А также получить сам объект сети:

```
>> lab1

lab1 =

    Neural Network

        name: 'Custom Neural Network'
        userdata: (your custom info)

    dimensions:

        numInputs: 1
        numLayers: 2
        numOutputs: 1
        numInputDelays: 0
        numLayerDelays: 0
        numFeedbackDelays: 0
        numWeightElements: 9
        sampleTime: 1

    connections:

        biasConnect: [1; 1]
        inputConnect: [1; 0]
        layerConnect: [0 0; 1 0]
        outputConnect: [0 1]

    subobjects:

        input: Equivalent to inputs{1}
        output: Equivalent to outputs{2}

        inputs: {1x1 cell array of 1 input}
        layers: {2x1 cell array of 2 layers}
        outputs: {1x2 cell array of 1 output}
        biases: {2x1 cell array of 2 biases}
```

Выводы

При выполнении первой лабораторной работы были получены необходимые навыки в построении базовой модели нейронной сети, умение устанавливать параметры сети, а также возможности ее обработки.