Date: 15th April, 2014 M.S. Raghavendra Sriram

1000854840

EE 5322 Intelligent Control Systems

Assignment no 4

# Neural Networks

Question 1:

1- Consider the following training data set

x = [5 4 6 5 2 1 2 1

0 -1 0 -1 -1 -2 -2 -3]; % inputs

y = [0 0 0 0 1 1 1 1 ]; % Target values

a- Train a perceptron to classify the data set into two classes. Plot points and

decision boundaries.

b- Use a single neuron with sigmoid activation function to do this classification

problem. Plot the points and decision boundary and compare the results to

part a.

Solution 1 a :

Code :

close all,

clear all,

clc,

format compact

% number of samples of each class

N = 8;

% define inputs and outputs

x = [5 4 6 5 2 1 2 1

0 -1 0 -1 -1 -2 -2 -3]; % inputs

y = [0 0 0 0 1 1 1 1 ];

% outputs

% Plot input samples with PLOTPV (Plot perceptron input/target vectors)

figure(1)

plotpv(x,y);

net = perceptron;

net = train(net,x,y);

view(net);

figure(1)

plotpc(net.IW{1},net.b{1});

y = net(x)

OutPut :

y =

0 0 0 0 1 1 1 1

Plots :

A description...

A description...

Solution 1 b :A description...

Code :

close all,

clear all,

clc,

format compact

% number of samples of each class

% define inputs and outputs

x = [5 4 6 5 2 1 2 1

0 -1 0 -1 -1 -2 -2 -3]; % inputs

y = [0 0 0 0 1 1 1 1 ];

% outputs

% Plot input samples with PLOTPV (Plot perceptron input/target vectors)

figure(1)

plotpv(x,y);

net = perceptron;

net.layers{1}.transferFcn = 'tansig';

net = train(net,x,y);

view(net);

figure(1)

plotpc(net.IW{1},net.b{1});

y = net(x)

Output :y =

Columns 1 through 6

-0.9889 0.9969 -0.9946 0.9936 0.9993 1.0000

Columns 7 through 8

1.0000 1.0000

A description...

Plots :

A description...

Question 2:A description...

Consider the dynamical system

a)Approximate the function f(x) using an MLP neural network and plot the function and the estimation on the same graph.

b)Simulate the system response for exact f(x) and the approximation. Use different initial conditions. Compare the results.

A description...A description...

A description...

A description...

Output :

ans =Neural Network

name: 'Custom Neural Network'

efficiency: .cacheDelayedInputs, .flattenTime,

.memoryReduction

userdata: (your custom info)

dimensions:

numInputs: 1

numLayers: 4

numOutputs: 1

numInputDelays: 0

numLayerDelays: 0

numFeedbackDelays: 0

numWeightElements: 76

sampleTime: 1

connections:

biasConnect: [1; 1; 1; 1]

inputConnect: [1; 0; 0; 0]

layerConnect: [4x4 boolean]

outputConnect: [0 0 0 1]

subobjects:

inputs: {1x1 cell array of 1 input}

layers: {4x1 cell array of 4 layers}

outputs: {1x4 cell array of 1 output}

biases: {4x1 cell array of 4 biases}

inputWeights: {4x1 cell array of 1 weight}

layerWeights: {4x4 cell array of 3 weights}

Question 3:

Use a multilayer perceptron to classify the data set into two classes and Compute the false-positive rate and false-negative rate of your classification results.

Code:

clc;

clear all;

close all;

%data classification using MLP

data=open('breastcancer.mat');

Input=data.data(:,2:10)'; %Input matrix

Target=data.data(:,11)'; %Target values

benign=0;

mali=0;

for i=1:length(Target)

if Target(i)==2

Target(i)=1; %Benign cell

benign=benign+1;

else

Target(i)=-1; %Malignant cell

malignant=malignant+1;

end

end

nnet=feedforwardnet(10);

nnet.divideParam.trainRatio = 70/100;

nnet.divideParam.valRatio = 15/100;

nnet.divideParam.testRatio = 15/100;

nnet=train(nnet,Input,Target);

output=hardlims(sim(nnet,Input));

wrongMalign=0;

wrongBenign=0;

for i=1:length(Target)

if Target(i)==1 && output(i)==-1

wrongMalign=wrongMalign+1;

end

if Target(i)==-1 && output(i)==1

wrongBenign=wrongBenign+1;

end

end

FalsePositive=wrongMalign/benign

FalseNegative=wrongBenign/malignant

Output :

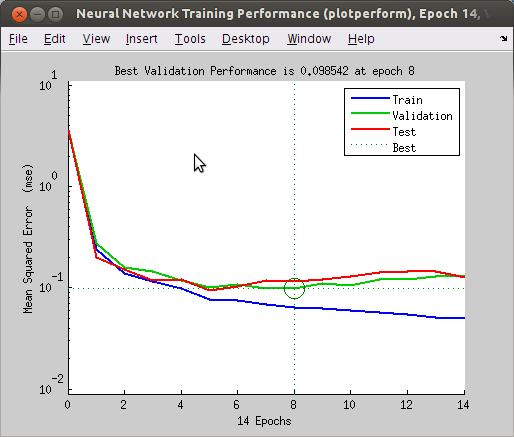
FalsePositive =

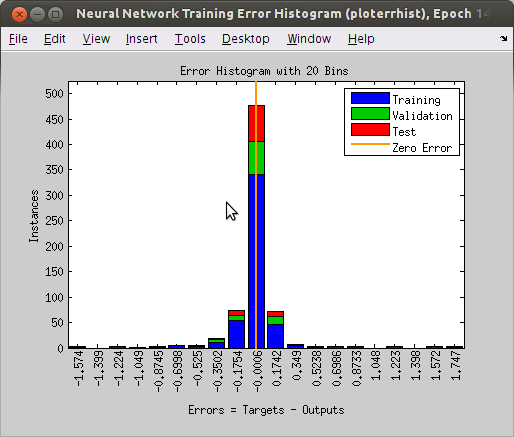
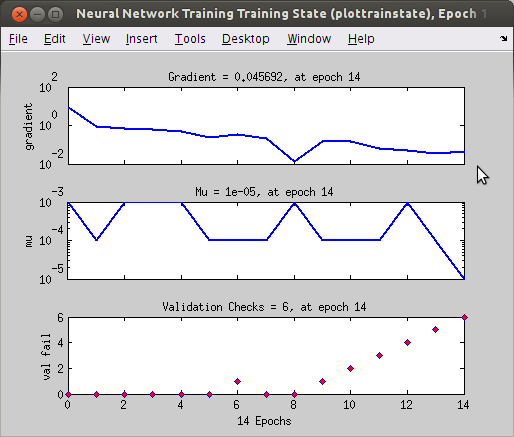
0.0203

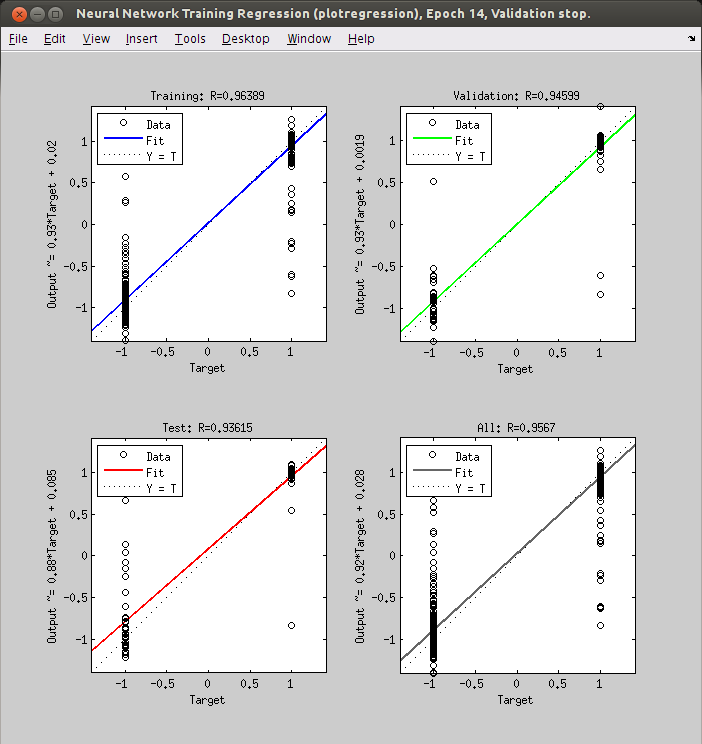
FalseNegative =

0.0293

Plot:







A description...

Solution 3b:

Code :

clc;

clear all;

close all;

%Classification using Self-Organizing Map

data=open('breastcancer.mat');

P=data.data(:,2:10)'; %Input matrix

T=data.data(:,11)'; %Target values

benign=0;

malignant=0;

for i=1:length(T)

if T(i)==2

%T(i)=1; %Benign cell

benign=benign+1;

else

%T(i)=0; %Malignant cell

malignant=malignant+1;

end

end

SOM=selforgmap([8 8],478);

SOM=train(SOM,P);

output=SOM(P);

classes=vec2ind(output);

Output:

