SAGNIK BASU 113EC0199

**QUESTION**

### Application of Perceptron/ Adaline in function approximamtion

Design a perceptron to approximate the function z=sin(pi\*x).Cos(pi\*y) where -0.5 <= x, y <= 0.5. The perceptron consist of 2 inputs and 2 weights with only one threshold. Train the weights and threshold using perceptron learning rule. Consider a suitable activation function based on the application. Generate a training data table consisting of 100 samples spread across the input space. After training, determine the mean square error (MSE) of the network for the output for these 100 samples which are different from the input training data set.

**SOLUTION**

**ARCHITECTURE**

****

**MATLAB CODE**

**clc;**

clear all;

close all;

% Function Approximation using perceptron

%

% Function = sin(x)\*cos(y)

%

constant=500;

x=rand(1,constant)-0.5; %training table

y=rand(1,constant)-0.5;

inp=zeros(constant,2);

for i=1:length(x)

inp(i,1)=x(i);

inp(i,2)=y(i);

end

thr=0.000001;

for j=1:20

var1=2\*rand(1,1)-1;

var2=2\*rand(1,1)-1;

w=[var1 var2];

z\_des=sin(pi\*x).\*cos(pi\*y); %desired output

bias=rand(1,1);

e=100000000;

weight1=zeros(1,constant);

weight2=zeros(1,constant);

bias1=zeros(1,constant);

for i=1: length(x)

e=100000000;

while(e>thr)

f=w(1)\*x(i)+w(2)\*y(i)+ bias;

g=(exp(f)-exp(-f))/(exp(f)+exp(-f)); %tanh activation func

e=z\_des(i)-g; %error

w=w+e\*.040\*inp(i,:); %updating weight values

bias=bias+e;

end

weight1(i)=w(1);

weight2(i)=w(2);

bias1(i)=bias;

end

plot(weight2,bias1);

hold on;

%plot(weight1,bias1);

%hold on;

end;

hold off;

percp=zeros(1,length(x));

for i=1:length(x)

percp(i)=w(1)\*x(i)+w(2)\*y(i)+ bias;

end;

%nstant\_test=500;

%no for testing our perceptron

for k=1:500

error=0;

constant\_test=500;

x\_test=rand(1,constant\_test)-0.5; %training table

y\_test=rand(1,constant\_test)-0.5;

for i=1:length(x\_test)

check1=sin(pi\*x\_test(i).\*cos(pi\*y\_test(i)));

check2=w(1)\*x\_test(i)+w(2)\*y\_test(i)+ bias;

final\_err=check1-check2;

error=error+(final\_err\*final\_err);

end

MSE(k)=error/length(x);

z=zeros(constant,constant);

plot(MSE);

hold on;

end

for i=1:length(x)

real(i,j)=sin(pi\*x(i)).\*cos(pi\*y(i));

estimated(i,j)=w(1)\*x(i)+w(2)\*y(i)+ bias;

end

%hold off;

%subplot(2,2,1);

%plot(real);

%subplot(2,2,2);

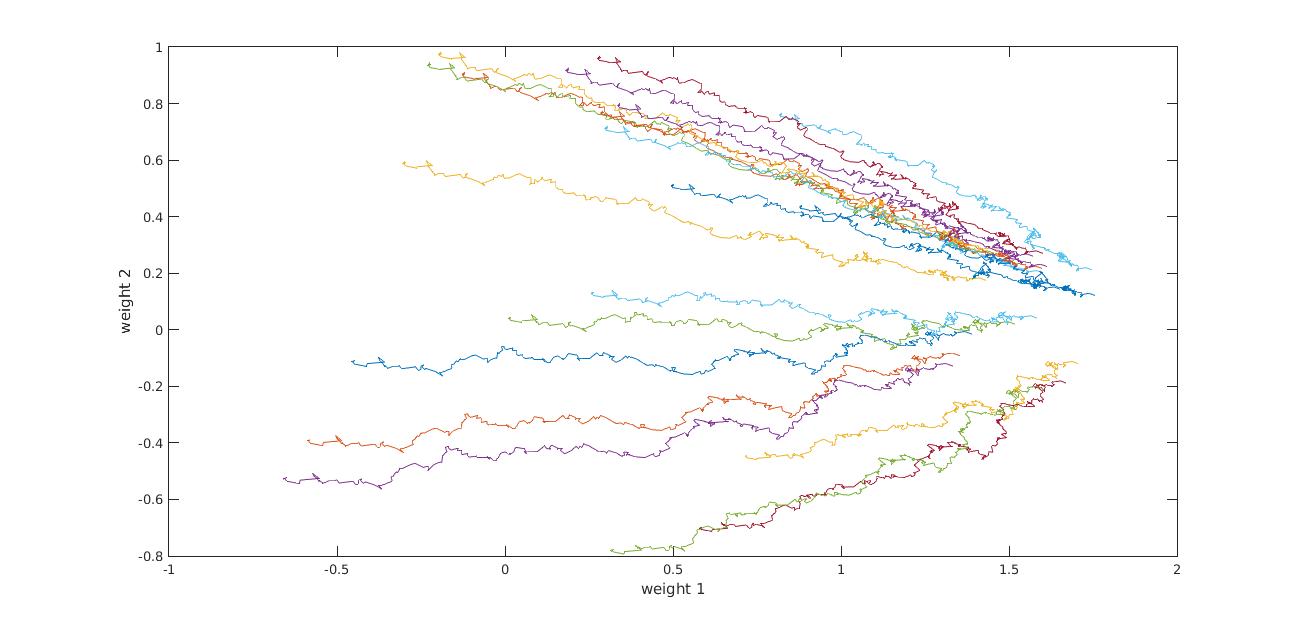
%plot(estimated);

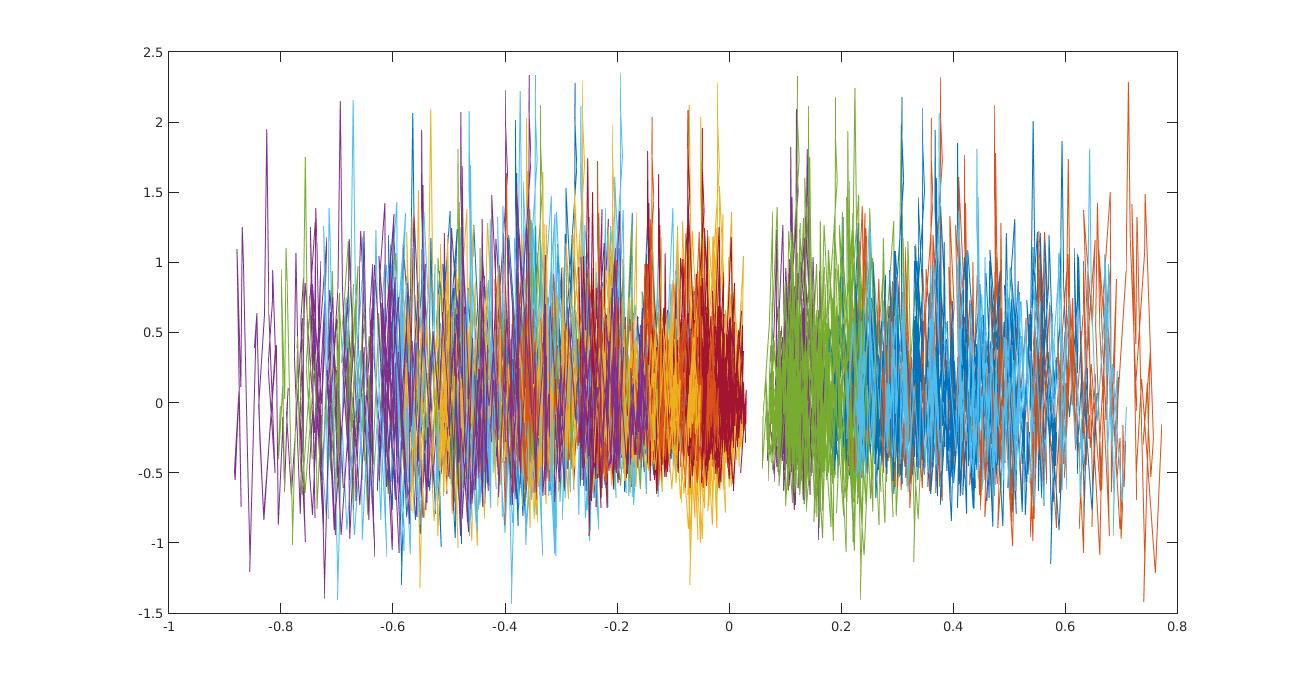
%axis([1 100 1 100]);

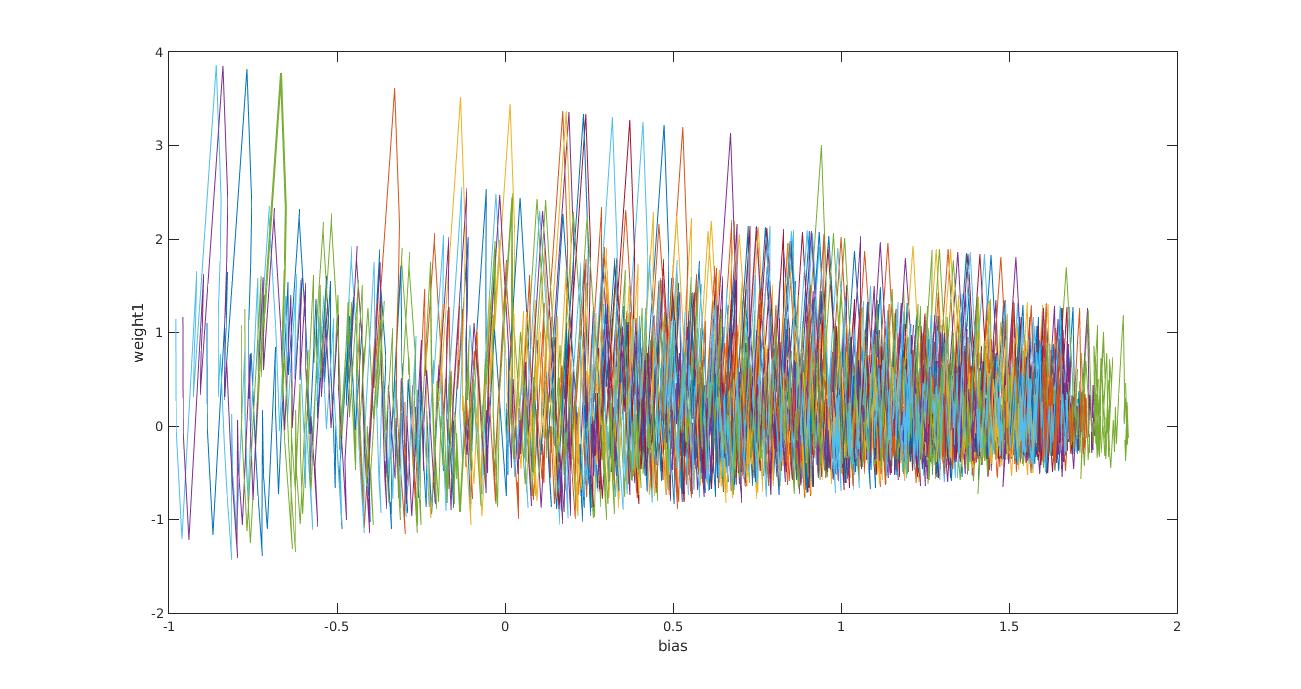
%plot(z\_des);

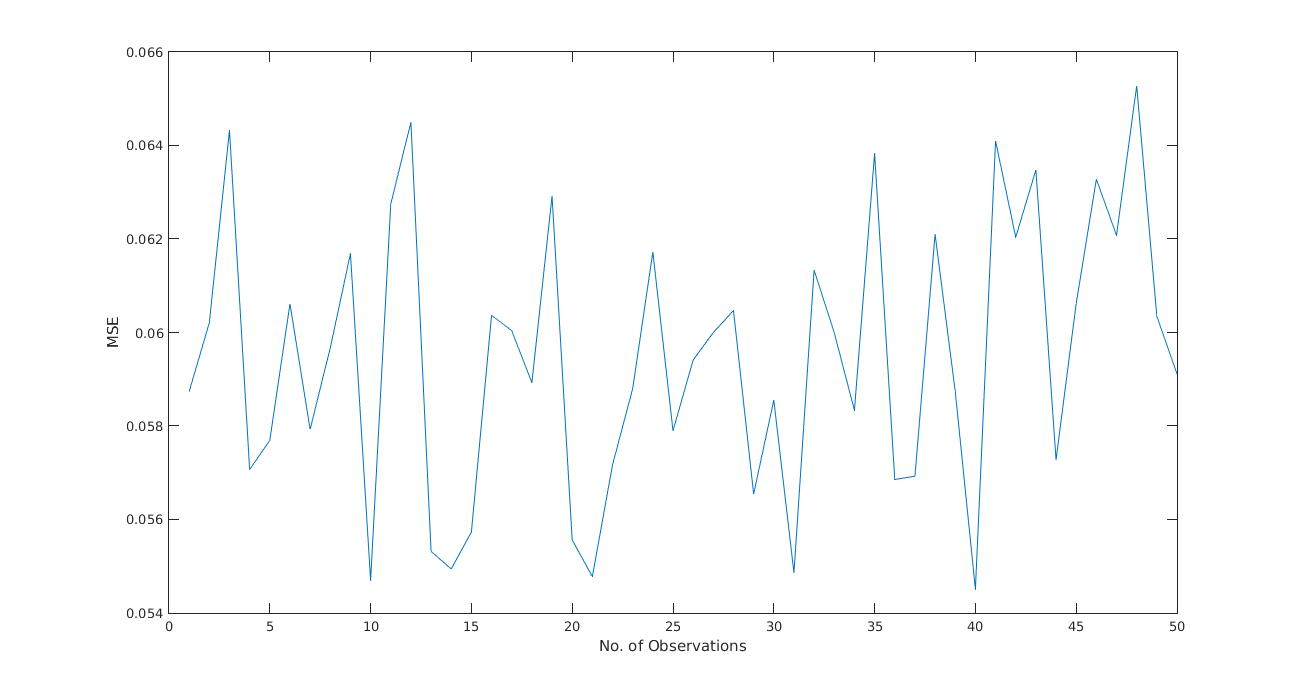
%plot(weight1,weight2);

**FIGURES**

****

****

****

****