**Program 1: Write a MATLAB program to generate a few activation functions that are being used in neural networks.**

%Illustration of various activation function used in NN's

x=-10:0.1:10

tmp=exp(-x);

y1=1./(1+tmp);

y2=(1-tmp)./(1+tmp);

y3=x;

subplot(231);plot(x,y1);grid on;

axis([min(x) max(x) -2 2]);

title('Logistic Function');

xlabel('(a)');axis('square');

subplot(232);plot(x,y2);grid on;

axis([min(x) max(x) -2 2]);

title('Hyperbolic Tangent Function');

xlabel('(b)');axis('square');

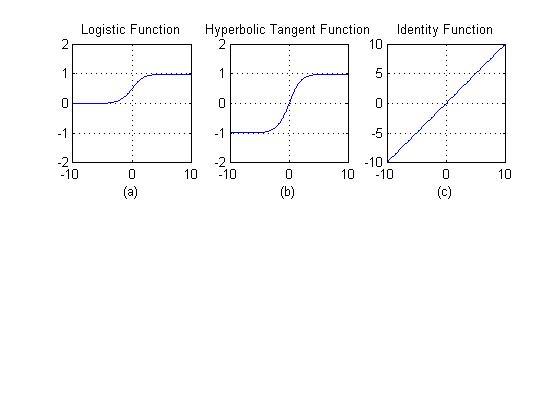
subplot(233);plot(x,y3);grid on;

axis([min(x) max(x) min(x) max(x)]);

title('Identity Function');

xlabel('(c)');axis('square');

**Output:**



**Program 2: Generate ANDNOT function using Mc-Culloch-Pits neural net by a MATLAB program.**

%Generate ANDNOT function using McCulloch-Pits neural net

clc;

clear;

disp('Enter weights: ');

w1=input('weight w1= ');

w2=input('weight w2= ');

disp('Enter Threshold value: ');

theta=input('theta= ');

y=[0 0 0 0];

x1=[0 0 1 1];

x2=[0 1 0 1];

z=[0 0 1 0];

con=1;

while con

zin=x1\*w1+x2\*w2;

for i=1:4

if(zin(i)>=theta)

y(i)=1;

else

y(i)=0;

end

end

disp('output of net');

disp(y);

disp(z);

if y==z

con=0;

else

disp('Net is not learning enter another set of weights and Threshold value');

w1=input('weight w1=');

w2=input('weight w2=');

theta=input('theta=');

end

end

disp('Mcculloch-pitts Net for ANDNOT function');

disp('The weights of neuron');

disp(w1);

disp(w2);

disp(' Threshold value');

disp(theta);

**Output:**

Enter weights:

weight w1= 1

weight w2= 1

Enter Threshold value:

theta= 1

output of net

0 1 1 1

0 0 1 0

Net is not learning enter another set of weights and Threshold value

weight w1=1

weight w2=-1

theta=1

output of net

0 0 1 0

0 0 1 0

Mcculloch-pitts Net for ANDNOT function

The weights of neuron

1

-1

Threshold value

1

**Program 3: Generate XOR function using Mc-Culloch-Pits neuron by writing a M-file.**

clear

clc

%XOR function using McCulloch-Pits

disp('Enter Weights ')

w11=input('Weight W11= ');

w12=input('Weight W12= ');

w21=input('Weight W21= ');

w22=input('Weight W22= ');

v1=input('Weight V1= ');

v2=input('Weight V2= ');

disp('Enter the value of theta');

theta=input('theta= ');

%y1=[0 0 0 0];

%y2=[0 0 0 0];

y=[0 0 0 0];

x1=[0 0 1 1];

x2=[0 1 0 1];

%zin=[0 0 0 0];

z=[0 1 1 0];

con=1;

while con

zin1=x1\*w11+x2\*w21;

zin2=x1\*w12+x2\*w22;

for i=1:4

if zin1(i)>=theta

y1(i)=1;

else

y1(i)=0;

end

if zin2(i)>=theta

y2(i)=1;

else

y2(i)=0;

end

end

yin=y1\*v1+y2\*v2;

for i=1:4

if yin(i)>=theta

y(i)=1;

else

y(i)=0;

end

end

disp('Output of net ');

disp(y);

disp(z);

if y==z

con=0;

else

disp('Net is not learning enter another set of weights and Threshold value');

w11=input('Weight W11= ');

w12=input('Weight W12= ');

w21=input('Weight W21= ');

w22=input('Weight W22= ');

v1=input('Weight V1= ');

v2=input('Weight V2= ');

disp('Enter the value of theta');

theta=input('theta= ');

end

end

disp('Mcculloch-pitts Net for XOR function');

disp('The weights of neuron');

disp(w11);

disp(w12);

disp(w21);

disp(w22);

disp(v1);

disp(v2);

disp('Threshold value');

disp(theta);

**Output:**

Enter Weights

Weight W11= 1

Weight W12= -1

Weight W21= -1

Weight W22= 1

Weight V1= 1

Weight V2= 1

Enter the value of theta

theta= 1

Output of net

0 1 1 0

0 1 1 0



**Program 4: Write a MATLAB program for perceptron net for an AND function with bipolar input and targets.**

clear;

clc;

x=[1 1 -1 -1; 1 -1 1 -1];

t=[1 -1 -1 -1];

w=[0 0];

b=0;

alpha=input('Enter learning rate=');

theta=input('Enter threshold value= ');

con=1;

epoch=0;

while con

con=0;

for i=1:4

yin=b+x(1,i)\*w(1)+x(2,i)\*w(2);

if yin>theta

y=1;

end

if yin<=theta & yin>=-theta

y=0;

end

if yin<-theta

y=-1;

end

if y-t(i)

con=1;

for j=1:2

w(j)=w(j)+alpha\*t(i)\*x(j,i);

end

b=b+alpha\*t(i);

end

end

epoch=epoch+1;

end

disp('Percentron for AND function');

disp('Final Weight matrix');

disp(w);

disp('Final Bais');

disp(b);

**Output:**

Enter learning rate=1

Enter threshold value= 0.5

Percentron for AND function

Final Weight matrix

1 1

Final Bais

-1