**compiler :-** [**MATLAB Login | MATLAB & Simulink (mathworks.com)**](https://matlab.mathworks.com/)

Program

%ANDNOT function using Mcculloch-Pitts neuron

clear;

clc;

%Getting weights and threshold value

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);

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('Weights of Neuron');

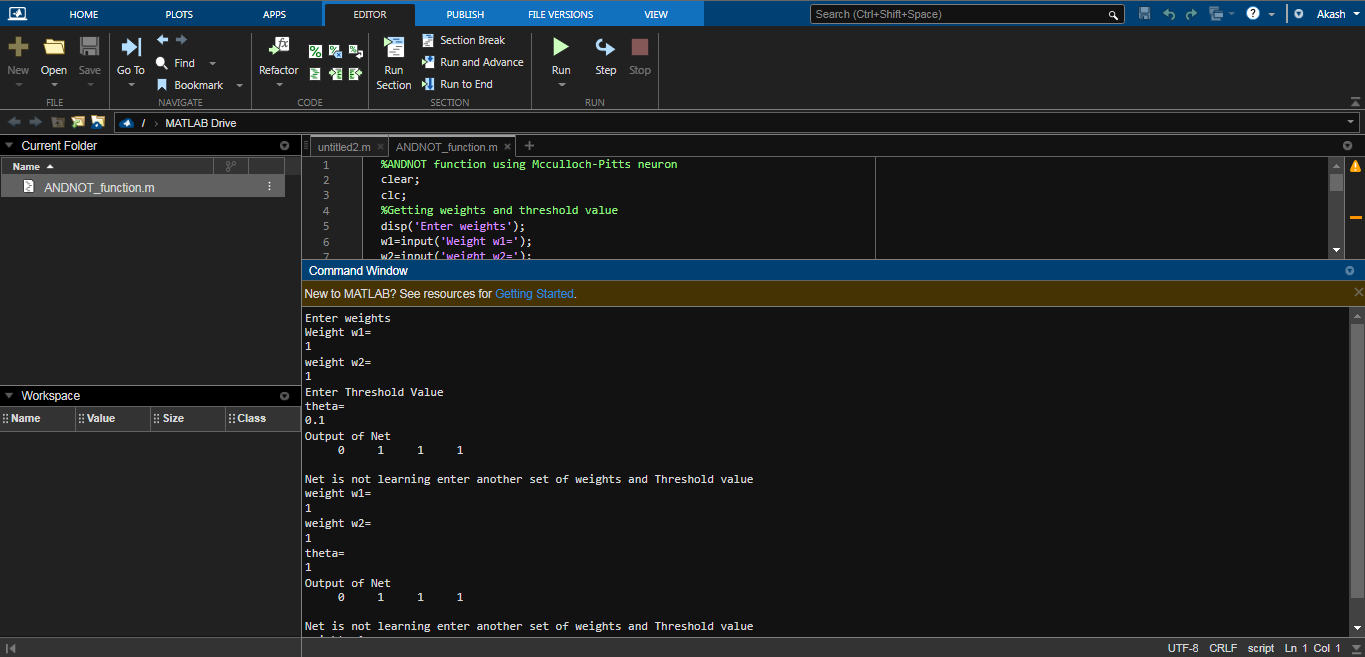
disp(w1);

disp(w2);

disp('Threshold value');

disp(theta);

output :-



2nd code but different terms

%Adaptive Resonance Theory (ART)

clear ;

clc ;

b =[0.57 0.0 0.3;0.0 0.0 0.3;0.0 0.57 0.3;0.0 0.47 0.3];

t =[1 1 0 0;1 0 0 1;1 1 1 1];

vp =0.4;

L =2;

x =[1 0 1 1];

s = x ;

ns =sum( s ) ;

y = x \* b ;

con =1;

while con

for i =1:3

if y ( i ) == max( y )

J = i ;

end

end

x = s .\* t(J ,:) ;

nx =sum( x ) ;

if nx / ns >= vp

b (: , J) = L \* x (:) /( L -1+ nx ) ;

t (J ,:) = x (1 ,:) ;

con =0;

else

y(J)=-1

con=1

end

if y+1==0

con=0

end

end

disp('top down weights');

disp(t);

disp('bottom up weights');

disp(b);

output

