FREEMIND

Date

Page

Q1 Compute y cordinate of line with m = 0.5 C = -2 of following x-cordinate

$y = mx + c$

$x = [0, 1.5, 3, 4, 5, 7, 9, 10]$

Codding: lab-1.m

```
x= [0, 1.5, 3, 4, 5, 7, 9, 10];
m= 0.5 ; c =-2;
y= m*x + c;
disp(y);
```

Q2 Create a vector t with 10 elements them Compute the following

$x = t \sin(t)$, $z = \frac{\sin(t)}{t^2}$, $t = 1:10$

$y = \frac{t-1}{t+1}$

Codding: lab-2.m

```
t=[1,2,3,4, 5, 6, 7, 8, 9, 10];
x= t.* sin(t);
y= (t-1) /(t+1);
z= sin(t)./t.^2
disp(x)
disp(y)
disp(z)
```

Output >> lab-1.m

[-2 , -1.25, -0.50, 0, 0.5, 1.5, 2.5, 3]

Output >> lab-2.m

x is 0.84147, 1.818 0.423 -3.0272 -4.7946
-1.6765 4.5989 7.9149

Y is 0.74257

Z is 0.84147 0.22 0.015 -0.0473 -0.03835
-0.0054
-0.000776 0.0134 0.0154 0.005

FREEMIND

Date

Page

lab_3.m

Q)3 $x = r\cos\theta$

$y = r\sin\theta$

$r = 2$

Create a collumn vector for with values 0, $\pi/4$, $\pi/2$, $3\pi/4$, π and $5\pi/4$ Compute vector x and y

Codding lab_3.m

```
theta =[0, pi/4, pi/2 ,3*pi/4 ,pi, 5*pi/4 ];
r= 2;
theta = theta';
x= r* sin (theta);
y = r* cos ( theta);
disp (x); disp(y);
```

FREEMIND

Date

Page

Create a vector of n-elements from 0 to 10 when $r = 0.5$ Create another vector $X = [r^0, r^1, r^2, \ldots r^n]$

Sum of this vector Repeate the procedure taking n from 0 to 30 and 0 to 100

Codding lab-4.m

```
n = 0:10; r = 0.5;
x = r.^n;
Summation = Sum(x)
disp(['Summation 0:10 is', num2str(Summation)]);

n = 0:30; x = r.^n;
Summation = Sum(x);
disp(['Summation 0:30 is', num2str(Summation)]);

n = 0:100; x = r.^n;
Summation = Sum(x);
disp(['Summation 0:100 is', num2str(Summation)]);
```

Output >> lab_3.m

$$\begin{bmatrix} 0 \\ 1.414 \\ 2.000 \\ 1.414 \\ 0.000 \\ -1.414 \end{bmatrix}$$

$$\begin{bmatrix} 2.000 \\ 1.414 \\ 0.000 \\ -1.414 \\ -2.000 \\ -1.414 \end{bmatrix}$$

Output >> lab_4.m

Summation of 0:10 is 1.999

Summation of 0:30 is 2

Summation 0:100 is 2

FREEMIND

Date

Page

Q5 Create a vector and a matrix by using the Command

V = 0 : 0.2 : 12;

M = [sin(V); cos(V)];

Find the size of V and M extract 10-element of each element and display them as collumn vector.

Codding lab-5.m

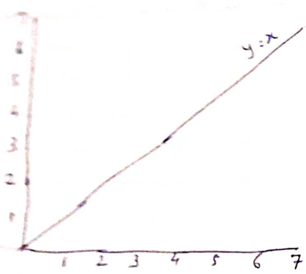
```
V = 0: 0.2 : 12;
M = [ sin(V); cos(V) ];
M = M( :, 1:10);
disp( [' size of V', num2str ( size(V)) ]);
disp ( [' size of M, num2str( size ([sin(V) : cos(V)]))]
disp (V (:, 1 : 10)')
disp( M');
```

Ramani
9/01/19

Output lab-5.m

Size of V 1 61
Size of M 2 61

$$
\begin{bmatrix} 0 \\ 0.200 \\ 0.400 \\ 0.600 \\ 0.800 \\ 1.000 \\ 1.200 \\ 1.400 \\ 1.600 \\ 1.800 \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1.000 \\ 0.1987 & 0.9801 \\ 0.3894 & 0.9211 \\ 0.5646 & 0.8253 \\ 0.7174 & 0.6967 \\ 0.8415 & 0.5403 \\ 0.9320 & 0.3624 \\ 0.9854 & 0.1700 \\ 0.9996 & -0.0292 \\ 0.9738 & -0.[illegible] \end{bmatrix}
$$



FREEMIND

Date

Page

Q6 Boxplot

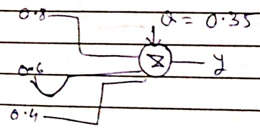
Codding lab-6.m

```
X = [40,20,80,60,100,120,140,180,160];
boxplot(X);
```

Q7 Single layer Neuron with 3-input

Codding lab-7.m

```
A = [0.8, 0.6, 0.4];
B = [0.1, 0.3, -0.21];
Q = 0.35;
y = A*B'+Q;
disp(y)
```



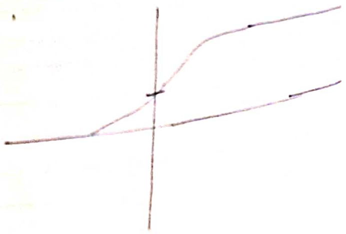
Activation functions

Q8 Linear, $f(x)=x$

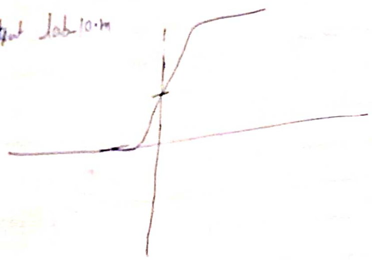
Codding lab-8.m

```
x=[1,2,3,4,5,6,7];
y=x;
plot(x,y);
```

Output lab9.m



Output lab10.m



Q9 Binal Sigmoid function

$$F(x) = \frac{1}{1-e^{-\lambda}}$$

Codding lab-9.m

```
X= -3:0.2:20;
Y= 1./(1+exp(-5*X));
plot(X,Y)
```

Q10 Bipolar Sigmod

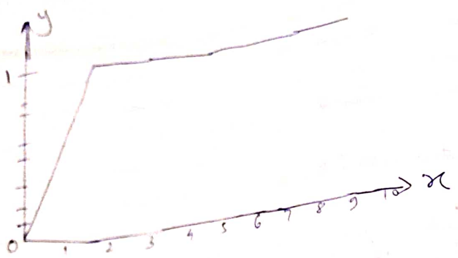
$$F(x) = \frac{1-e^{-\lambda x}}{1+e^{-\lambda x}}$$

Codding lab-10.m

~~Q11~~

```
X= -3.0:0.2:10;
Y= 1-exp(-5*X)./(1+exp(-5*X));
plot(Y,X);
```

Output lab-11.m



FREEMIND

Date
Page

Q 11 Ramp function

Codding lab-11.m

```
x= 0:0.5:10;
y = [];
For i = 1: length (x)
  if (x(i) >1)
      y(i)=1
else
    if (x(i)<=1 && x(i) >=0)
         y(i) = x(i);
    else
        y(i)=0
    end
end
plot(x,y).
```

Output lab12.m

~~y = [0,0,0,1]~~

y = 0 1 1 1
W = 0.5 0.5
B = -0.2

Implementation of gates using perceptron Algorithm.

Q12 OR Gate Implementation

Codding lab_12.m

```
x = [0,0;0,1;1,0;1,1];
Y = [0,0,0,1];
W = [0.5, 0.5]; B = 0.8; eta = 0.5;
y_e = [];
y = [0,0,0,0];
For i = 1:5
  For j = 1:4
    y(1,j) = x(j,:)*W' + B;
    if (y>=0)
        y(1,j) =1
    else
        y(1,j) =0;
    end
    e(1,j) = Y(1,j)-y(1,j);
    W = W+eta*e(i,j)*x(j,:);
    B = B+eta*e(i,j);
  end
end
disp(y)
disp(W)
disp(B)
```

Output lab-13.m

0 0 0 1

0 2 1
-3

Date

Page

Q 13 Implementation of And Gate

```
X = [0,0; 0,1; 1,0; 1,1];
Y = [0,0,0,1];
W = [1,1]; B = 0; eta = 1
y = [0,0,0,0];

For i = 1:5
  For j = 1:4
    y(1,j) = x(j, :) * W' + B;
    if ( y >= 0)
        y(1,j) = 1;
    else
        y(1,j) = 0;
    end
    e = Y(1,j) - y(1,j)
    W = W + eta * e * X(j,1);
    B = B + eta * e;
  end
end
disp(y)
disp(W)
disp(B)
```

Output lab-14.m

1 0
3 1
-2

Date
Page

Q14 Implementation of NOR Gate

```
X = [0 ; 1];
Y = [1,0];
W = 0.5; B = 0.8; eta = 0.5;
y_e = []
y = [0,0];

for i=1:4
 for j = 1:2
   y(1,j) = X(j,:)*W + B
   if y>=0
     y(1,j) =1
   else
     y(1,j) = 0
   end
   e(i,j) = Y(1,j)-y(1,j);
   W = W+ eta*e(i,j)*X(j,:);
   B = B+ eta*e(i,j);
 end
end
disp(y)
disp(W)
disp(B)
```

Output Lab-15.m

1, 1, 1, 0

3 2

-1

Q15 Implementation of NAND Gate

```
X= [0,0; 0,1; 1,0; 1,1];
Y= [1,1,1,0];
W= [1,1]; B=0; eta =1;
y_e = [];
y= [0,0,0,0];
For i = 1:8
 For j =1:4
    y(1,j)= x(j,:)*W'+B;
 if (y>=0)
    y(1,j)=1
 else
    y(1,j)=0
 end
 e= Y(1,j)-y(1,j);
 W= W+eta*e*X(j,:);
 B= B+eta*e;
end
end
disp(Y)
disp(W)
disp(B)
```

1, 0 0 0

-3

3 2

Q16 Implementation of NorGate

```
X = [0,0; 0,1; 1,0; 1,1];
Y = [1,0,0,0];
y_e = [];
y = [];
for i = 1:5
 for j = 1:4
   y(1,j) = X(j,:)*W'+B;
   if (y >= 0)
       y(1,j) = 1;
   else
       y(1,j) = 0;
  e(i,j) = Y(1,j) - y(1,j)
  W = W + eta * e(i,j) * X(j,:);
  B = B + eta * e(i,j);
end
end
disp(y)
disp(B)
disp(W)
```

Date

Page

Q17 Implementation of XOR Gate

$x = [0,0; 0,1; 1,0; 1,1]$