HW7

1-1.1-a

Given,

W [1,1,-1,0.5,1,2]

X=4 ,y=0

a)Use forward propagation to compute the predicted output

Answer

y1= x*w[1]
y1=4*1 =4

Using RelU activation function in the hidden layer,

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

The y1 is greater than the 0,so no change in value

Outy1=max(0,y1)

Outy1=y1

Y2= x*w[2]

Y2=4*1 =4

Using Relu activation function in the hidden layer

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

The value of y2 is greater than the 0, so no change in the output of activation function

Outy2=max(0,y2)

Outy2=y2

Y3=x*w[3]

Y3=4*-1

Y3=-4

Using Relu activation function in the hidden layer

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

The value of y3 is less than 0, so the value of y3 is 0(y3=0)

Outy3=max(0,y3)

Outy3=0

Yp=w[4]*y1+w[5]*y2+w[6]*y3

Yp=0.5*4 + 1*4 +0*2

Yp=6

Using the sigmoid activation in the output layer $yout = \frac{1}{1+e^{-x}}$

$yout = \frac{1}{1+e^{-6}}$

Yout=0.997

This is the predicted output 0.997

1-1.1-b

What is the loss or error value?

Answer:

Squared error:

E(yp, yout) = (yp - yout) ^{2}

E(yp, yout) = (0 - 0.997) ^{2}

E(yp, yout) = 0.994

The error is 0.974 we need to backpropagate to get the expected output

1-1.1-c

Using backpropagation, compute the gradient of the weight vector, that is, compute the partial derivative of the error with respect to all of the weights

Answer:

Updating the weights w

$$\frac{d(E)}{d(w[6])} = \frac{d(E)}{d(yout)} * \frac{d(yout)}{d(yp)} * \frac{d(yp)}{d(w[6])}$$

partial derivative of E with respect to yout $\frac{d(E)}{d(yout)}$

=-2(yp - yout)

=-2(0-0.997)

=1.994

The derivative of yps with respect to ypred $\frac{d(yout)}{d(yp)}$

=σ(1- σ)

=0.997(1-0.997)

=0.002

Finding the derivative of error with respective to w4

$$\frac{d(E)}{d(w[4])} = -2(yp - yout) * \sigma(1 - \sigma) * \mathrm{x}$$

$\frac{d(E)}{d(w[4])}$=0.023

Finding the derivative of error with respective to w5

$$\frac{d(E)}{d(w[5])} = -2(yp - yout) * \sigma(1 - \sigma) * \mathrm{x}$$

$\frac{d(E)}{d(w[4])}$=0.023

Finding the derivative of error with respective to w6

$$\frac{d(E)}{d(w[6])} = -2(yp - yout) * \sigma(1 - \sigma) * \mathrm{x}$$

$\frac{d(E)}{d(w[6])}$= 0

Finding the derivative of error with respective to w1

$$\frac{d(E)}{d(w1)} = \frac{d(E)}{d(yout)} * \frac{d(yout)}{d(yp)} * \frac{d(yp)}{d(y1out)} * \frac{d(y1out)}{d(y1)} * \frac{d(y1)}{d(w[1])}$$

$$\frac{d(E)}{d(w1)} = -2(yp - yout) * \sigma(1 - \sigma) * \mathrm{w}[4] * 1 * 4$$

$$\frac{d(E)}{d(w1)} = 0.011$$

Finding the derivative of error with respective to w2

$$\frac{d(E)}{d(w2)} = \frac{d(E)}{d(yout)} * \frac{d(yout)}{d(yp)} * \frac{d(yp)}{d(y2out)} * \frac{d(y2out)}{d(y2)} * \frac{d(y2)}{d(w[2])}$$

$$\frac{d(E)}{d(w1)} = -2(yp - yout) * \sigma(1 - \sigma) * \mathrm{w}[5] * 1 * 4$$

$$\frac{d(E)}{d(w1)} = 0.023$$

Finding the derivative of error with respective to w3

$$\frac{d(E)}{d(w3)} = \frac{d(E)}{d(yout)} * \frac{d(yout)}{d(yp)} * \frac{d(yp)}{d(y3out)} * \frac{d(y3out)}{d(y3)} * \frac{d(y3)}{d(w[3])}$$

$$\frac{d(E)}{d(w1)} = -2(yp - yout) * \sigma(1 - \sigma) * \mathrm{w}[6] * 0 * 4$$

$$\frac{d(E)}{d(w1)} = 0$$

1-1.1-d

Using a learning rate of 1.0, compute new weights from the gradient. With the new weights, use forward propagation to compute the new predicted output, and the loss (error).

Answer

The Weights updating:

Learning rate=1.0

W=1 - learning rate*(derivative of error with respective to weight)

W1 = 1- 1*(0.011)

W1 = 0.989

W2 = 1 – 1*(0.023)

W2 = 0.977

W3 = 1 – 1*(0)

W3 = -1

W4 = 0.5-1*(0.023)

W4 = 0.477

W5 = 1-1*(0.023)

W5 = 0.977

W6 = 2-1*(0)

W6 = 2

Forword pass:

Y1 = 4(0.989) = 3.956

Y1out =relu(y1)

Y1out= 3.956

Y2 = 4(0.977) = 3.908

Y1out=relu(y2)

Y1out=3.908

Y3= -1(4) = -4

Y3out=relu(y3)

Y3out=0

Yp = y1out*w4+y2out*w5+y3out*w6

Yp = 3.956 * 0.477 + 3.908*0.977+0*2

Yp= 5.705

E(y, yp)= $(y – yp)^2$

E(y, yp) = $(0- 0.996)^2$

E(y , yp)= 0.992

1-1.1-e

Comment on the difference between the loss values you observe in (b) and (d)

Answer

The difference in the both errors is =0.994 – 0.992

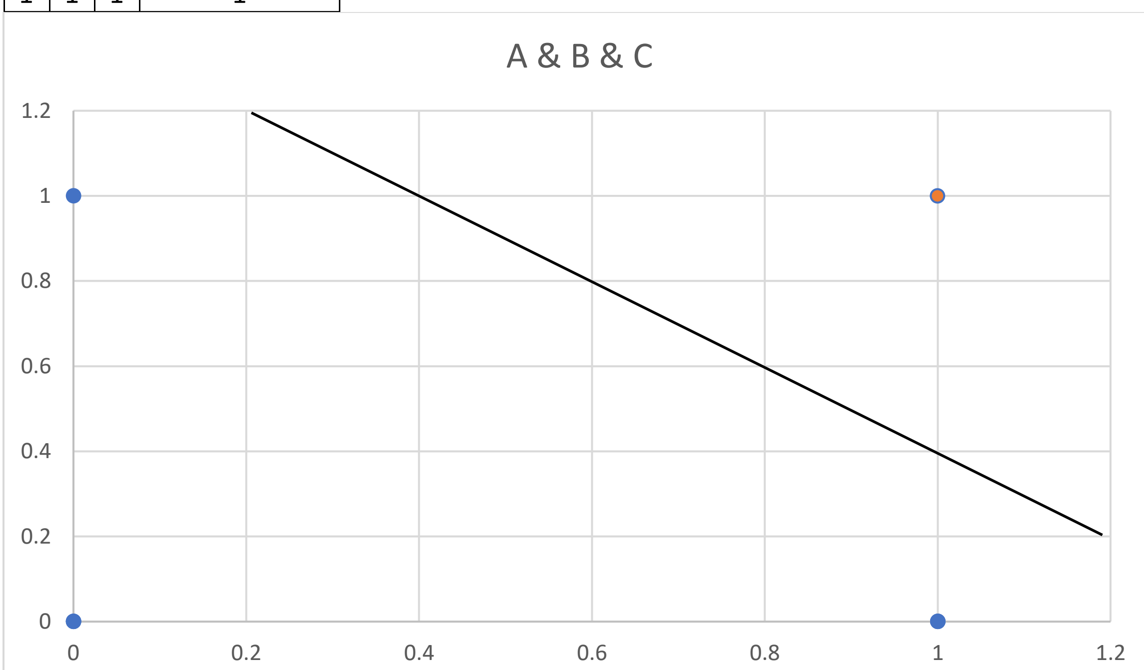
= 0.002

1-1.2-14-a

A AND B AND C

Answer:

A	B	C	A And B And C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



The Orange DOT's are 1 and the Blue dot's are 0's

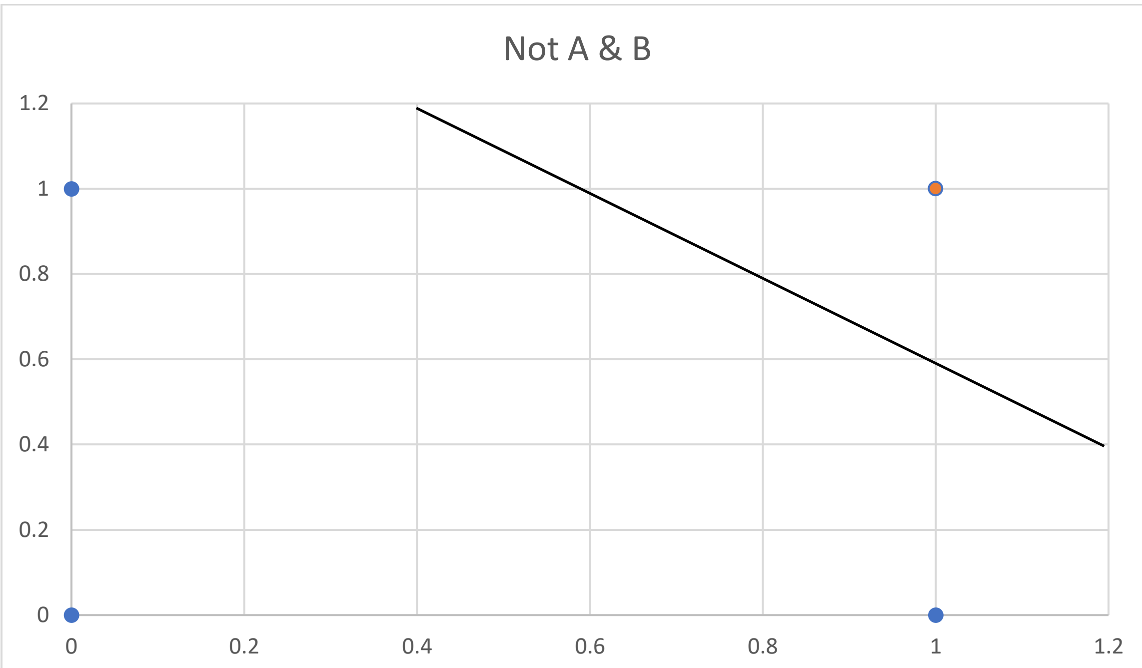
Yes this is Linearly separable

1-1.2-14-b

NOT A AND B

Answer:

A	B	Not A	Not A And B
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	1



The Orange DOT's are 1 and the Blue dot's are 0's

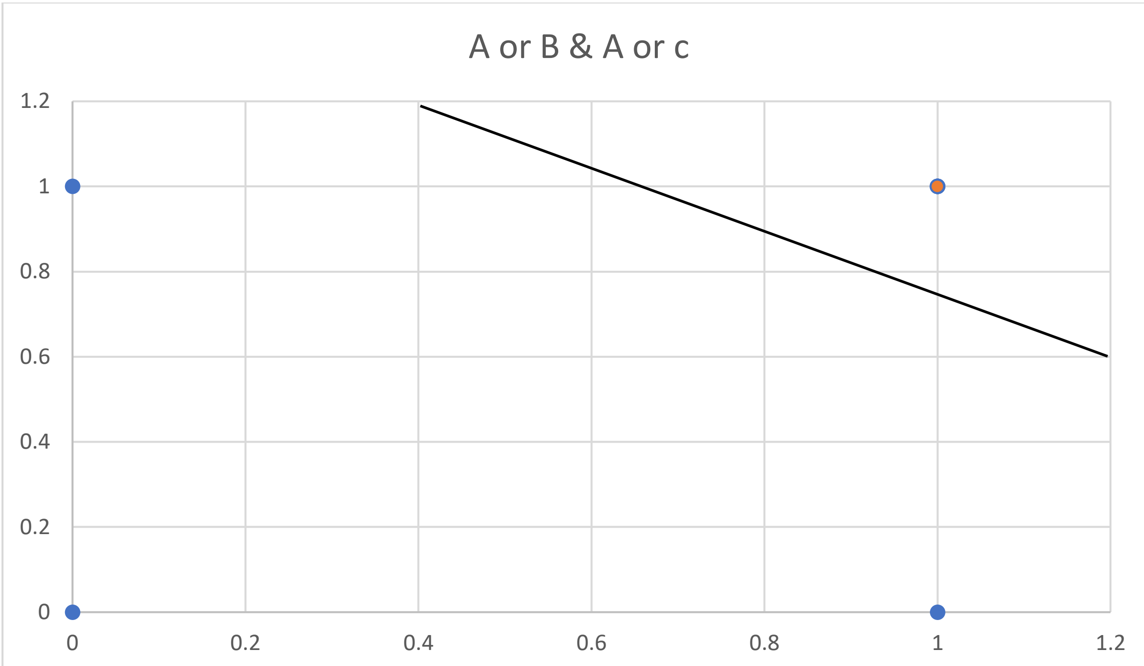
Yes this is Linearly separable

1-1.2-14-c

(A OR B) AND (A OR C)

Answer:

A	B	C	P=A OR B	Q=A OR C	P AND Q
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1



The Orange DOT's are 1 and the Blue dot's are 0's

Yes this is Linearly separable

1-1.2-14-d

(A XOR B) AND (A OR B)

Answer: Is not Linear separable

1-1.2-15-a

a) Demonstrate how the perceptron model can be used to represent the AND and OR functions between a pair of Boolean variables.

Answer

AND function between pair of Boolean

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

W1=1, W2= 1, b= -1.5

Y= A*w1+ B*w2 + b

The threshold is 0.5,

if   y>=0 then 1

Else 0

OR Function between A and B

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

W1=1, W2=1, b= -0.5

Y=A*w1+B*w2+ b

The threshold is 0.5

If y>=0 then 1

Else 0

1-1.2-15-b

Comment on the disadvantage of using linear functions as activation functions for multi-layer neural networks.

Answer:

The linear activation functions are also called as straight line activation functions, the input of this function will be y= mx+b format. We cannot define it in specific range. If we use the linear functions as activation functions, there will no use its still the linear combination of its input attributes then it works same as the perceptron which cannot classify multiple classes and it cannot be used for the multi-layer neural network. The multi layer neural network has the non linear relationships input and outputs. So linear functions as activation functions has no use of using in multi layer neural networks.

1-1.3

Given,

No .of Predictors=8

1st hidden layer=16 neurons

2nd hidden layer =8 neurons

3rd hidden layer has 4 neurons

Output layer has only 1 neuron

X= 8*16 + 16*8 + 8*4 +4*1 + 16*1+8*1+4*1

X= 321

There are total of 321 parameters in the neural network