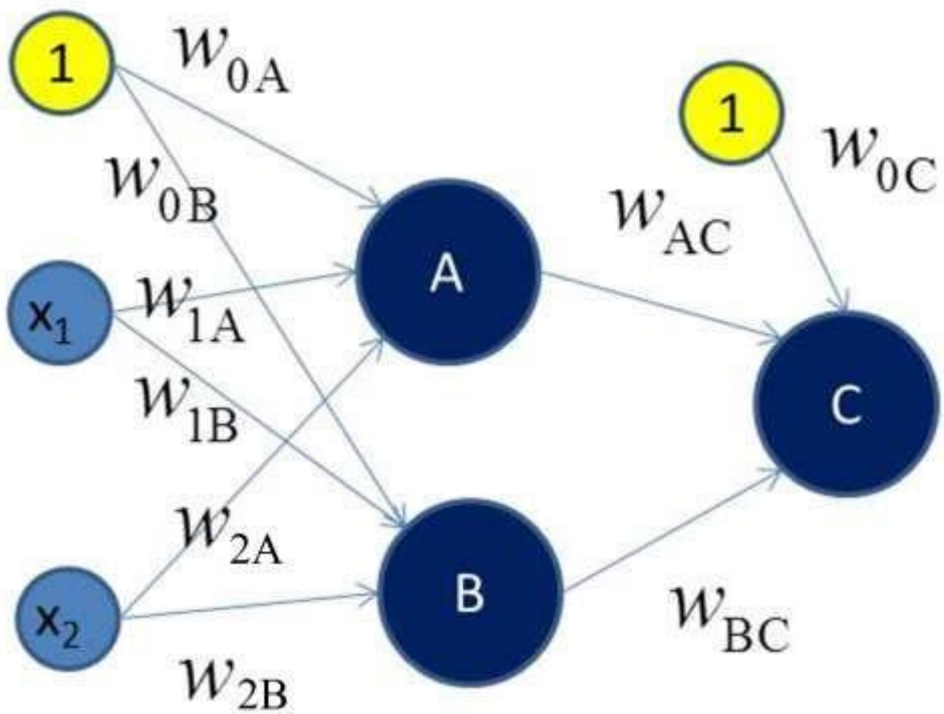
Neural Network Assignment

(this is a very simple assignment ☺)

1. Feed forward the following MLP neural network given the following:

Weights: all weights are 0.1 Activation: both the hidden layer and the output layer use a sigmoid activation Default value of the bias inputs (in yellow): 1 Input data (x1=0.5, x2=0.8)



In feed forward, the layer values and weights are as shown in the table

Input	hidden	hidden	output
X0 = 1	W0A = 0.1	W0B = 0.1	W0C = 0.1
X1 = 0.5	W1A = 0.1	W1B = 0.1	WAC = 0.1
X2 = 0.8	W2A = 0.1	W2B = 0.1	WBC = 0.1

The scalar value for hidden layer Node A, netA

netA = X0*W0A + X1*W1A + X2*W2A = 1*0.1 + 0.5*0.1 + 0.8*0.1 = 0.23

The activation value on netA, aA

aA = sigmoid(netA) = 1/(1+e^(-netA)) = 1/(1+e^(-0.23)) = 0.56

Similarly, scalar value for hidden layer Node B, netB

netB = X0*W0B + X1*W1B + X2*W2B = 1*0.1 + 0.5*0.1 + 0.8*0.1 = 0.23

Activation value over netB, aB

aB = sigmoid (netB) = 1/(1+e^(-netB)) = 1/(1+e^(-0.23)) = 0.56

netZ by aA and aB is computed by,

netZ = 1*W0C + aA*WAC + aB*WBC = 1*0.1 + 0.56*0.1 + 0.56*0.1 = 0.212

Activation value over netZ, yZ

yZ = sigmoid(netZ) = 1/(1+e^(-netZ)) = 1/(1+e^(-0.212)) = 0.55

0.55 is the output from MLP ANN in first iteration.

2. Should we prefer a large hidden layer (many nodes) or a small one? Describe the benefits and drawbacks of each.

The number of nodes in the hidden layer is constructed by the user on their choice. However, increasing the hidden layer nodes increases the power and flexibility of the model. Large number of the hidden layers are usually assumed to deal with identifying complex patterns but this can lead to overfitting the validation set. On the contrary, of there is underfitting or if training accuracy is low, the model may be reconstructed by increasing the number of nodes in the hidden layer. Hence an optimal number of nodes are suggested in the hidden layer without making it large or too small.