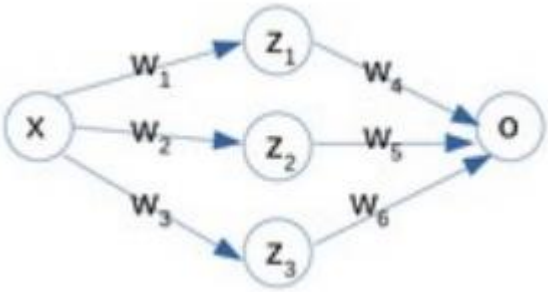
Javier-EchavarrenSuarez-HW3

1.1 (2 points) Consider the neural network shown in the figure below.

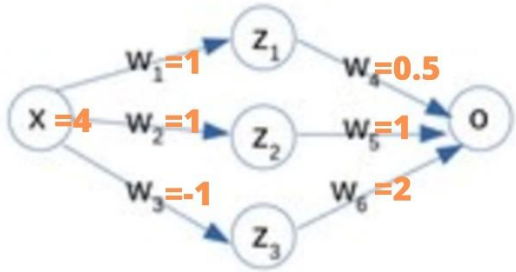
The weight matrix, W, is: [1, 1, -1, 0.5, 1, 2]T . Assume that the hidden layer uses RelU and the output layer uses Sigmoid activation function. Assume squared error. The input x = 4, and the output y = 0.

Recall that RelU is defined as a function, f(x) = max(0, x). Its derivative is Squared error is defined as E(y, ^y) = (y - ^y) 2 .

The partial derivative of E with respect to ^y is -2(y - ^y). Using this information, answer the following questions: (Show all work, and all answers should be rounded to 3 decimal places OR POINTS WILL BE TAKEN OFF!)



(a) Use forward propagation to compute the predicted output.



$$\boldsymbol{ReLU}(z_1) = max(0\,, 4 \cdot 1) = 4$$

$$\boldsymbol{ReLU}(z_2) = max(0\,, 4 \cdot 1) = 4$$

$$\boldsymbol{ReLU}(z_3) = max(0\,, 4 \cdot (-1)) = 0$$

$$Output = \sigma(4 \cdot 0.5 + 4 \cdot 1 + 0 \cdot 2) = \sigma(6) = \frac{1}{1 + exp(-6)} = 0.998$$

(b) What is the loss or error value?

The squared error is defined as: $E(y, \hat{y}) = (y - \hat{y})^2$

$$E(y, \hat{y}) = (0 - 0.9975)^2 = 0.995$$

(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.

$$\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial w_6}} = \frac{\boldsymbol{\partial E}}{\boldsymbol{\partial \sigma(o)}} \cdot \frac{\boldsymbol{\partial \sigma(o)}}{\boldsymbol{\partial o}} \cdot \frac{\boldsymbol{\partial o}}{\boldsymbol{\partial w_6}} = \boldsymbol{-2}(y - \hat{y}) \cdot \boldsymbol{\sigma(o)}(\boldsymbol{1} - \boldsymbol{\sigma(o)}) \cdot z_3$$

$$= \boldsymbol{-2 \cdot -0.9975 \cdot 0.9975(1 - 0.9975)} \cdot 0 = 0$$

$$\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial w_5}} = \frac{\boldsymbol{\partial E}}{\boldsymbol{\partial \sigma(o)}} \cdot \frac{\boldsymbol{\partial \sigma(o)}}{\boldsymbol{\partial o}} \cdot \frac{\boldsymbol{\partial o}}{\boldsymbol{\partial w_5}} = \boldsymbol{-2}(y - \hat{y}) \cdot \boldsymbol{\sigma(o)}(\boldsymbol{1} - \boldsymbol{\sigma(o)}) \cdot z_2$$

$$= \boldsymbol{-2 \cdot -0.9975 \cdot 0.9975(1 - 0.9975)} \cdot 4 = 0.020$$

$$\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial w_4}} = \frac{\boldsymbol{\partial E}}{\boldsymbol{\partial \sigma(o)}} \cdot \frac{\boldsymbol{\partial \sigma(o)}}{\boldsymbol{\partial o}} \cdot \frac{\boldsymbol{\partial o}}{\boldsymbol{\partial w_4}} = \boldsymbol{-2}(y - \hat{y}) \cdot \boldsymbol{\sigma(o)}(\boldsymbol{1} - \boldsymbol{\sigma(o)}) \cdot z_1$$

$$= \boldsymbol{-2 \cdot -0.9975 \cdot 0.9975(1 - 0.9975)} \cdot 4 = 0.020$$

$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial ReLU(z_3)} \cdot \frac{\partial ReLU(z_3)}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_3} = [-2(y - \hat{y}) \cdot \sigma(o)(1 - \sigma(o)) \cdot w_6] \cdot 0 \cdot x$$
$$= 4.97 \cdot 10^{-3} \cdot 0 \cdot 4 = 0$$
$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial ReLU(z_2)} \cdot \frac{\partial ReLU(z_2)}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_2} = [-2(y - \hat{y}) \cdot \sigma(o)(1 - \sigma(o)) \cdot w_5] \cdot 1 \cdot x$$
$$= 4.97 \cdot 10^{-3} \cdot 1 \cdot 4 = 0.020$$
$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial ReLU(z_1)} \cdot \frac{\partial ReLU(z_1)}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1} = [-2(y - \hat{y}) \cdot \sigma(o)(1 - \sigma(o)) \cdot w_4] \cdot 1 \cdot x$$
$$= 2.4875 \cdot 10^{-3} \cdot 1 \cdot 4 = 0.020$$

(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).

$$w_{new} = w_{old} - LR \cdot \nabla E$$

$$w_{6,new} = w_{6,old} - LR \cdot \nabla E = 2 - 1 \cdot 0 \cdot 0.9975$$
$$w_{5,new} = w_{5,old} - LR \cdot \nabla E = 1 - 1 \cdot 0.0199 \cdot 0.9975 = 0.9801$$
$$w_{4,new} = w_{4,old} - LR \cdot \nabla E = 0.5 - 1 \cdot 0.0199 \cdot 0.9975 = 0.4801$$
$$w_{3,new} = w_{3,old} - LR \cdot \nabla E = -1 - 1 \cdot 0 \cdot 0.9975 = -1$$
$$w_{2,new} = w_{2,old} - LR \cdot \nabla E = 1 - 1 \cdot 0.0199 \cdot 0.9975 = 0.9801$$
$$w_{1,new} = w_{1,old} - LR \cdot \nabla E = 1 - 1 \cdot 0.01 \cdot 0.9975 = 0.9900$$

$$ReLU(z_1) = max(0, 4 \cdot 0.99) = 3.960$$

$$ReLU(z_2) = max(0, 4 \cdot 9801) = 3.920$$

$$ReLU(z_3) = max(0, 4 \cdot (-1)) = 0$$

$$Output = \sigma(3.96 \cdot 0.5 + 3.92 \cdot 1 + 0 \cdot 2) = \sigma(5.9) = \frac{1}{1 + exp(-5.9)} = 0.997$$

$$E(y, \hat{y}) = (0 - 0.9973)^2 = 0.995$$

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

The error has decreased once we have updated the weights, as expected, but the improvement is really slow because the slope of the output function with this output is low ($2.49 \cdot 10^{-3}$)

1.2 [1 point] Tan Chapter 4, questions 14, 15.

14. For each of the Boolean functions given below, state whether the problem is linearly separable.

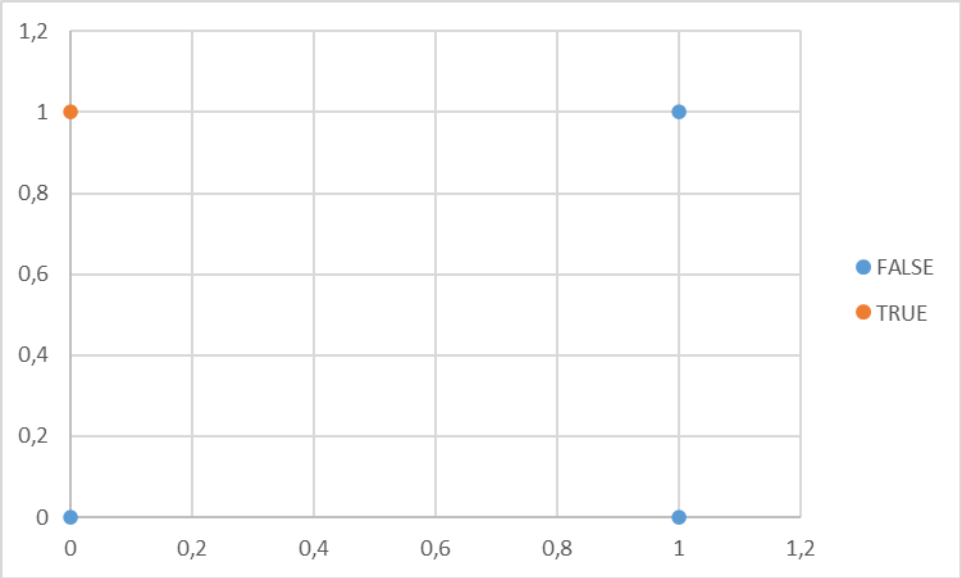
a. A AND B AND C

Label	A	B	C
FALSE	0	0	0
FALSE	0	0	1
FALSE	0	1	0
FALSE	0	1	1
FALSE	1	0	0
FALSE	1	0	1
FALSE	1	1	0

TRUE	1	1	1

We can see in this table that this problem is linearly separable, for example with the function f(A,B,C)=sign(0.75(A+B+C)-2)

b. NOT A AND B



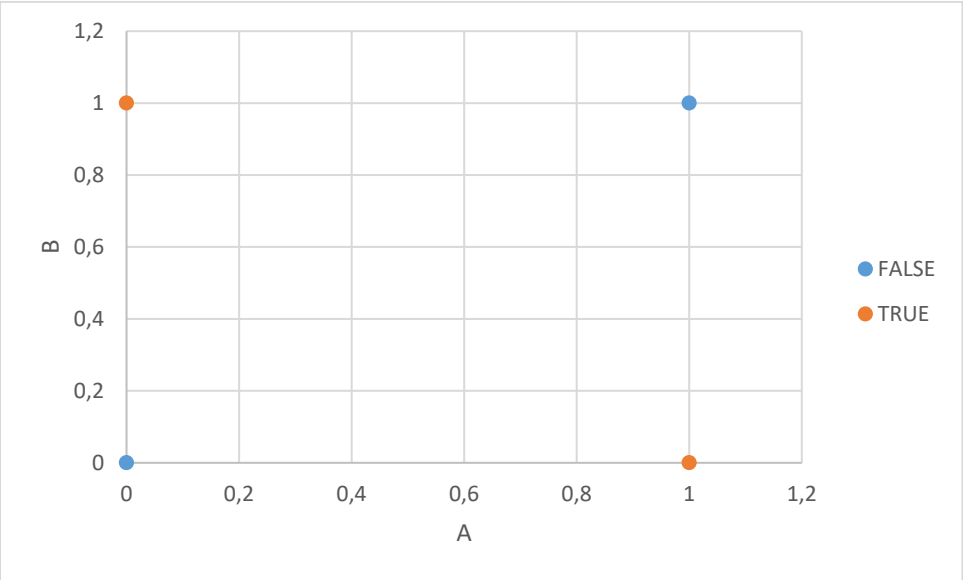
As we can see in the graph, this problem is linearly separable.

c. (A OR B) AND (A OR C)

Label	A	B	C
FALSE	0	0	0
FALSE	0	0	1
FALSE	0	1	0
TRUE	0	1	1
TRUE	1	0	0
TRUE	1	0	1
TRUE	1	1	0
TRUE	1	1	1

We can see in this table that this problem is linearly separable, for example with the function f(A,B,C)=sign(2A+B+C-1,5)

d. (A XOR B) AND (A OR B)

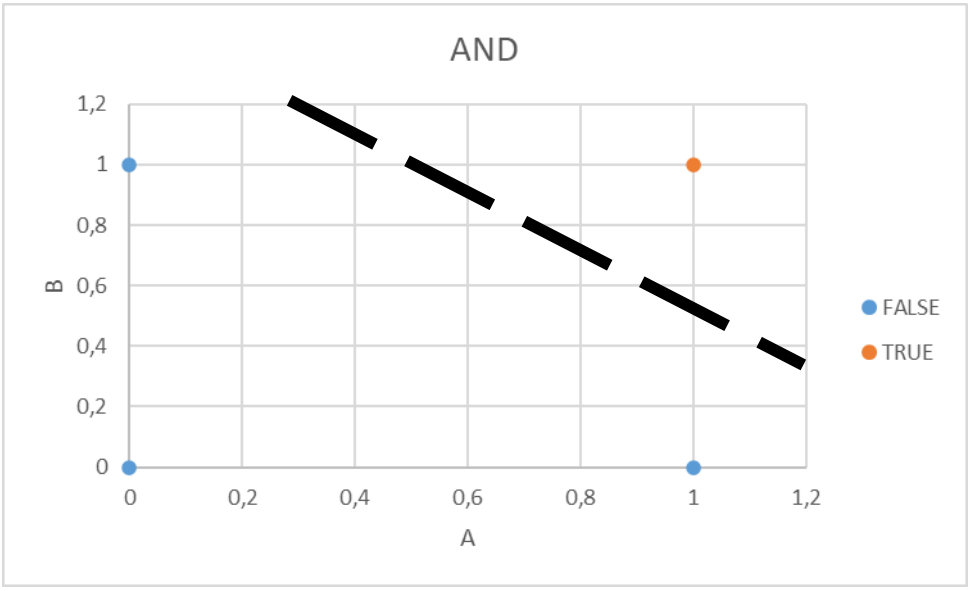


As we can see in the graph, this problem is not linearly separable.

15.

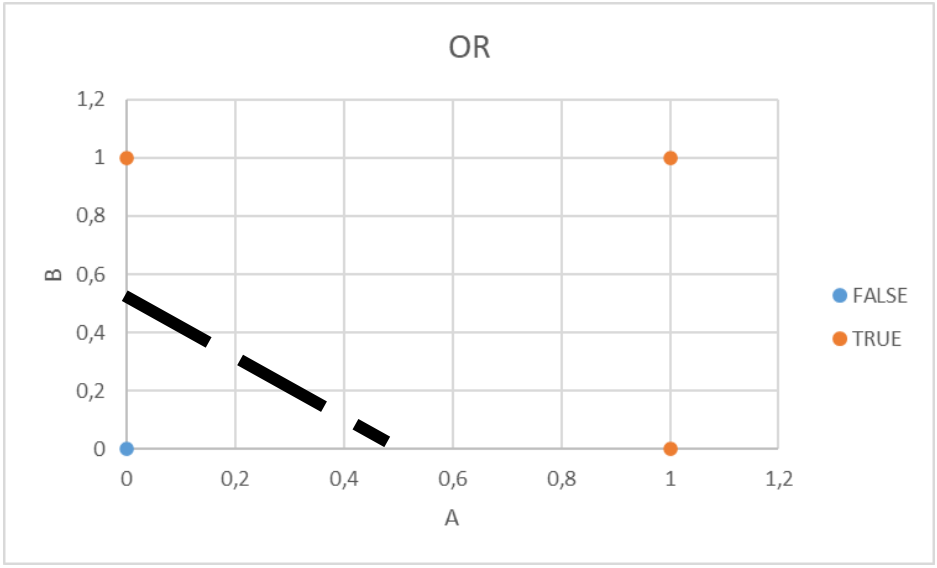
a. Demonstrate how the perceptron model can be used to represent the AND and OR functions between a pair of Boolean variables.

For the AND function:



It is easy to see that a perceptron representing the following function (f(A,B)=sign(A+B-1,5)) can be used to represent AND function.

For the OR function:



It is easy to see that a perceptron representing the following function (f(A,B)=sign(A+B-0.5)) can be used to represent AND function.

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

They are very unstable, as they can be negative or positive and can increase or decrease the weights infinitely when applying backpropagation, as there is no limit, as opposed to the RelU function, that is 0 when the input is negative.

1.1 Tan, Ch. 5 (Association Analysis)

Questions 15

15. Answer the following questions using the data sets shown in Figure 5.34. Note that each data set contains 1000 items and 10,000 transactions. Dark cells indicate the presence of items and white cells indicate the absence of items. We will apply the *Apriori* algorithm to extract frequent itemsets with *minsup* = 10% (i.e., itemsets must be contained in at least 1000 transactions).

(a) Which data set(s) will produce the most number of frequent itemsets?

To solve this problem, we start knowing that an itemset with k items can generate $2^k - 1$ itemsets, and the number of frequent itemsets in a dataset where m number of items are simultaneously present in n number of rows, (provided m/n ≥ minsup) is $2^l - 1$.

That's why we are looking for the dataset(s) that have the most number of items present in the same share of the data.

The dataset that clearly has the most number of items present in the same share of the data is the (e) of the figure 4.34, as most of the items from 0 to 800 are present in the 2000-4000 transaction range (20% of the data), having approximately $2^{800} - 1$ frequent itemsets.

(b) Which data set(s) will produce the fewest number of frequent itemsets?

The dataset with the fewest number of frequent itemsets is the (d), because it doesn't have items present in 10% of the samples (minsup), and that's why it has no itemsets apart from {null}.

(c) Which data set(s) will produce the longest frequent itemset?

The longest frequent itemset will be produced in the (e) dataset, because in the transactions 2000-4000 there are approximately 800 items present in the same transactions, and these transactions are approximately 20% of the dataset. This doesn't happen in the other datasets.

(d) Which data set(s) will produce frequent itemsets with highest maximum support?

The (b) dataset will produce frequent itemsets with highest maximum support, as it has the longest vertical line of all charts.

(e) Which data set(s) will produce frequent itemsets containing items with wide-varying support levels (i.e., items with mixed support, ranging from less than 20% to more than 70%)?

The (e) dataset will produce frequent itemsets containing items with wide-varying support levels, as it contains items with a lot of support (the ones where there are two vertical lines or a big vertical line) and also has other items that are present in less transactions, such as the ones of one little vertical line.

1.2 Zaki, Chapter 8 (Frequent Pattern Mining)
Questions 1(a), 4

1(a) Using minsup = 3/8, show how the Apriori algorithm enumerates all frequent patterns from this dataset.

tid	itemset
t_1	$ABCD$
t_2	$ACDF$
t_3	$ACDEG$
t_4	$ABDF$
t_5	BCG
t_6	DFG
t_7	ABG
t_8	$CDFG$

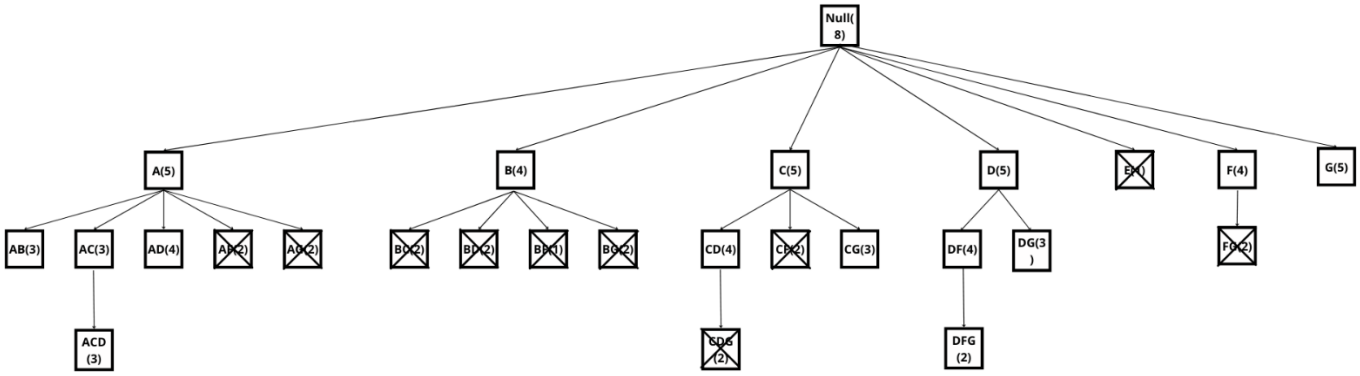


Figure 1: Apriori Tree

The frequent patterns of this dataset (minsup=3/8) are:

1 Null
2 A
3 B
4 C
5 D
6 F
7 G
8 AB
9 AC
10 AD
11 CD
12 CG
13 DF
14 DG
15 ACD
16 DFG

4. Given the database in Table 8.4. Show all rules that one can generate from the set ABE

Table 8.4. Dataset for Q4

tid	itemset
t_1	ACD
t_2	BCE
t_3	$ABCE$
t_4	BDE
t_5	$ABCE$
t_6	$ABCD$

The rules that one can generate from the database, the support of its subsets and rules confidences are shown in the following table.

Itemset	Support	Rule	Confidence
		A->B	0,75
AB	3	B->A	0,6
		A->E	0,5
AE	2	E->A	0,5
		B->E	0,8
BE	4	E->B	0,8
		AB->E	0,666
		AE->B	1
		A->BE	0,5
		B->AE	0,4
ABE	2	E->AB	0,5

1.3 [1 point] Consider a dataset that has 8 predictors. You train a neural network with 3 hidden layers and an output layer that predicts a continuous value (a regression problem). The first hidden layer has 16 neurons, the second has 8 neurons, and the third has 4 neurons. In this network, how many total parameters will you have?

We know that the number of parameters in each layer is the number of neurons in the layer times one plus the number of neurons in its precedent layer , the one being to take into account the bias.

This way, we have:
Hidden layer 1:

$$Parameters = 16 \cdot (8 + 1) = 144\, parameters$$

Hidden layer 2:

$$Parameters = 8 \cdot (16 + 1) = 136\, parameters$$

Hidden layer 3:

$$Parameters = 4 \cdot (8 + 1) = 36\, parameters$$

Output layer:

$$Parameters = 1 \cdot (4 + 1) = 5\, parameters$$

Adding them up, we get the total: 321 parameters.