Mid-Sem Examination, September 2021

Name - Hari Kesh Rai
Year - 3rd
Stream - CSI
Section - 3I
Roll No. - 23
Enrollment No. - 12019009001418
Paper Name - Deep Learning
Paper Code - PEC-CS501
Signature - Hari Kesh
Date - 07/09/2021

1.i)

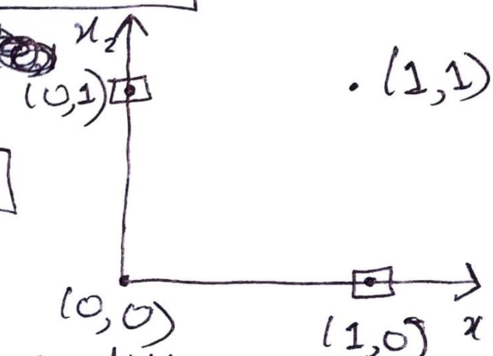
x_1	x_2	XOR	
0	0	0	$w_0 + \sum_{i=1}^{2} w_i x_i < 0$
1	0	1	$w_0 + \sum_{i=1}^{2} w_i x_i \geq 0$
0	1	1	$w_0 + \sum_{i=1}^{2} w_i x_i \geq 0$
1	1	0	$w_0 + \sum_{i=1}^{2} w_i x_i < 0$

$w_0 + w_i 0 + w_2 0 < 0 \Rightarrow \boxed{w_0 < 0}$

$w_0 + w_1 0 + w_2 0 \geq 0 \Rightarrow \boxed{w_2 \geq -w_1}$

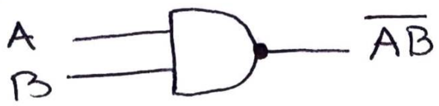
$w_0 + w_1 1 + w_2 0 \geq 0 \Rightarrow \boxed{w_1 \geq -w_0}$

$w_0 + w_1 1 + w_2 1 < 0 \Rightarrow \boxed{w_1 + w_2 < -w_0}$



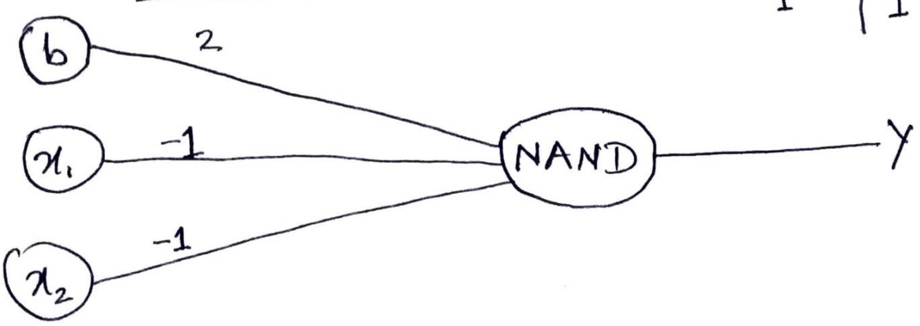
The fourth condition contradicts condition 2 & 3. Hence we cannot have a solution to this set of inequalities. In the graph we can see that it is impossible to draw a line which separates dot points from square points.

iv) 2 - input NAND Gate

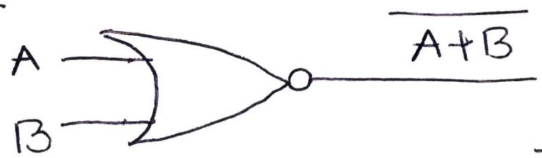


A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

$-x_1 + 2 - x_2$



v) NOR Gate



A	B	output
0	0	1
0	1	0
1	0	0
1	1	0

From the diagram, the NOR Gate is 1 only if both input are 0.

$w_1 x_1 + w_2 x_2 + b$.

initializing w_1 and w_2 as 1 and b as -1.

↳ $x_1(1) + x_2(1) - 1$.

Passing Value $(x_1 = 0, x_2 = 0)$.

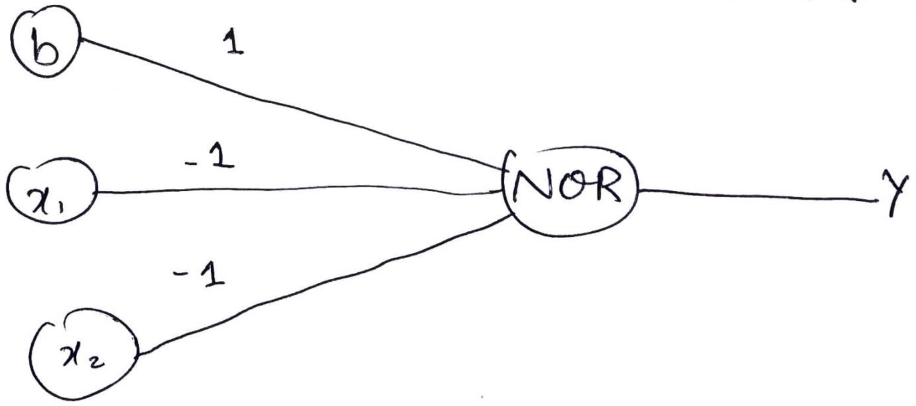
we get, $0 + 0 - 1 = -1$

So, From the perceptron rule, if $wx + b \leq 0$ then $y' = 0$.

This row is incorrect as the output is 1 for the NOR Gate.

So, we want values that will make input $x_1 = 0$ & $x_2 = 0$ to give y' a value of 1

if we change b to 1 we have $0 + 0 + 1 = 1$



vi) When we are saying that the characteristics of the units such as tanh, sigmoid, ReLU are differentiable it is important because of back propagation. The calculations for back propagation are effectively just the chain rule applied back up the neuron. If we cannot differentiate a single unit then we certainly cannot apply the chain value. Non-differentiable units can be trained but it is difficult to do so because there is no back propagation hence the model's errors when training to optimize the weights is more.

2.i) Given $J(W) = w_1^2 + w_2^2 + 3$

Learning Rate $= 0.5$

$J(w) = 2w_1 + 2w_2$

Initializing the parameter,

$$w = [2, 3]^T$$

$$\nabla wJ(w) = \begin{bmatrix} 2 & w_1 \\ 2 & w_2 \end{bmatrix}$$

$$J(w) = w_1^2 + w_2^2 + 3$$

$$\nabla w J(w) = \begin{bmatrix} 2 & w_1 \\ 2 & w_2 \end{bmatrix} \quad \alpha = 2$$

K	w^K	$J(w)$	$\alpha \cdot \nabla w J(w)$	w^{K+1}
0	[2 3]	16	[8 12]	[-6 -9]
1	[-6 -9]	120	[-24 -36]	[18 27]

First Iteration:

$$w^0 = [2, 3]$$

$$J(w) = 2^2 + 3^2 + 3 = 4 + 9 + 3 = 16$$

$$\alpha \cdot \nabla wJ(w) = 2\begin{bmatrix} 2 \times 2 \\ 2 \times 3 \end{bmatrix} = 2\begin{bmatrix} 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \end{bmatrix}$$

$$w^{K+1} = w^K - \alpha \cdot \nabla w.J(w)$$
$$= w^0 - \alpha \cdot \nabla w.J(w)$$
$$= [2, 3] - [8, 12]$$
$$= [-6 \quad -9]$$

Similarly, second iteration: $w = [-6 \quad -9]$

$$J(w) = (-6)^2 + (-9)^2 + 3 = 36 + 81 + 3$$

$$\alpha \cdot \nabla w.J(w) = 2\begin{bmatrix} 2w_1 \\ 2w_2 \end{bmatrix} = 2\begin{pmatrix} 2 \times -6 \\ 2 \times -9 \end{pmatrix} = \begin{bmatrix} -24 \\ -36 \end{bmatrix}$$

$\alpha = 2$,

$$w^2 = w^1 - \alpha \cdot \nabla_w J(w)$$
$$= [-6, -9] - [-24, -36]$$
$$= [18, 27]$$

After performing two iterations

K	w^K	$J(w)$	$\alpha \cdot \nabla_w J(w)$	w^{K+1}
0	[2, 3]	16	[8 , 12]	[-6, -9]
1	[-6, -9]	120	[-24, -36]	[18, 27]

7.b.i) Bias term is an adjustable, numerical term added to a perceptron's weighted sum of inputs & weights that can inc. classification model accuracy.

- weighted sum $= (x_1 w_1) + (x_2 w_2) + (x_3 w_3) + (1 w_{bias})$

The addition of bias term is helpful because it serves as another model parameter (in addition to weights) that can be tuned to make model's performance on training data as good as possible.

Default input value for the bias weight is 1 & weight value is adjustable.

ii) In short no, as the bias units are characterized by the text '+1'. As we can see bias unit is just appended to the start/end of the input and each hidden layer isn't influenced by the values in the previous layer. In other words, these neurons don't have any incoming connections. So, the question comes

3. i) Here the normalized features are $\{8.83, 0.20, 2.56, 1\}$ where only the last feature IsRedAvailable is inhibiting & is binary in nature $\{0,1\}$.
In the weights, there is also the same parameter with 1 value.
But the Θ value is 8.
Hence, the buyer cannot be able to buy the bike based on the McCullah pitts neuron as $g(n) < \Theta$, here $1 < 8$.

ii) A decision boundary is the region of a problem space in which the output label of classifier is ambiguous.
If the decision surface is a hyperplane, then ~~classifier~~ Classification problem is linear & the classes are linearly separable.

5) A) i) Deep feed forward networks, also often called feedforward neural networks or (MLPs) multilayer perceptrons, are the quintessential deep learning models. The goal of a feed forward network is to approximate some funtion f^*.

ii) Sealers are single numbers and are an example of a 0th order tensor. It is necessary to describe the set of values to which a sealer belongs.

vector are ordered

arrays of single numbers and are ~~ordered~~ an example of 1st order tensor.
Matrices are rectangular arrays consisting of numbers and are an example of 2nd order tensor.
The more general entity of a tensor encapsulation the scalar, vector and the matrix.
It is sometime necessary both in the physical science and machine learning.

6. A) i) We need an activation function to map input to output. It help neural network to learn complex relationships and pattern in data. Another use is to add non-linearity to data.

Different Types of activations are-

1) Linear
2) Sigmoid
3) Tanh
4) ReLU
5) Softmax
6) LeakyReLU

ii) One hot encoding is essentially the representation of categorical variables as binary reactors. These categorical values are mapped to integer values. Each Integer value is then represented as binary vectors that is all 0's except index of the integer which marked as 1.

8.i)

a) Deep Learning out perform other techniques if the data ~~base~~ size is large. But with small size, traditional machine learning algorithms are preferrable. Deep Learning really shines when it comes to complex Problems such as image classification, natural language processing, and speech recognition.

The increased processing powered affored by graphical processing unit (GPUS), the enermous amount available data, and development of more advanced algorithm leds the rise of deep learning over machine learning.

b) Broadly, there are 3 types of machine learning algorithms-

i) Supervised Learning ⇒ This algorithm consist of a target/outcome variable (or dependent variable) which is to be predicted by given set of Predictors (independent variable).

ii) Unsupervised Learning → In this algorithm, we dont have any target to outcome variable to predict / estimate. It is used for clustering population in different groups.

iii) Re-inforcement Learning ⇒ Using this algorithms the machine is trained to make specific decessions.

- commonly used machine learning algorithms are-

1) Lenear Regression 2) Logistic Regression 3) Decision Tree

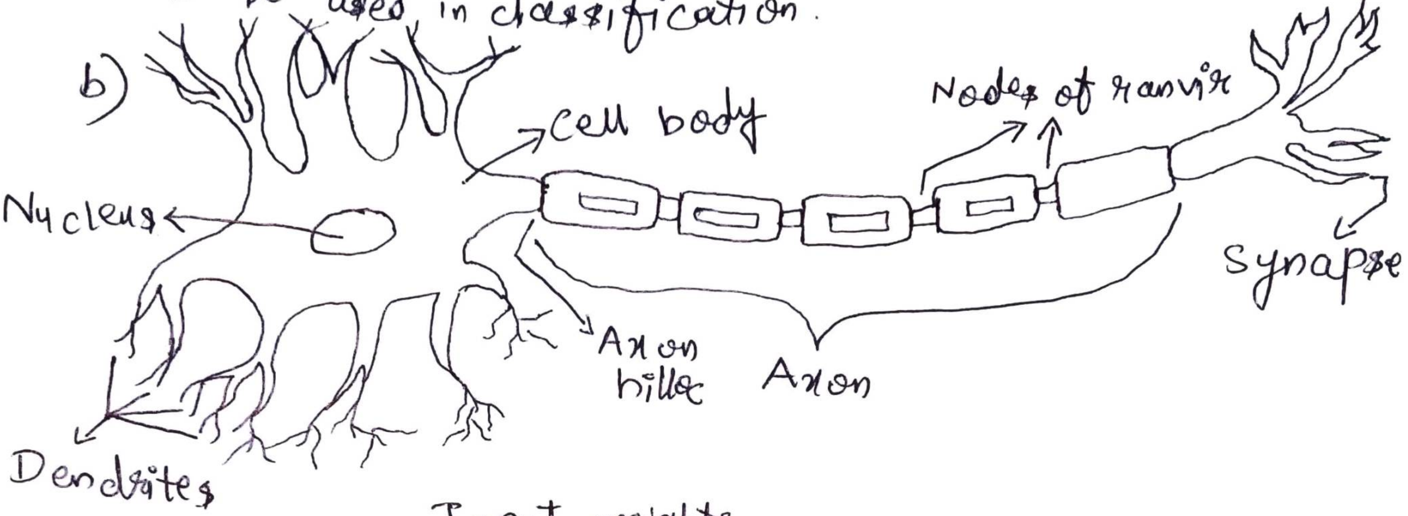
4) SVM 5) Naive Bayss 6) KNN

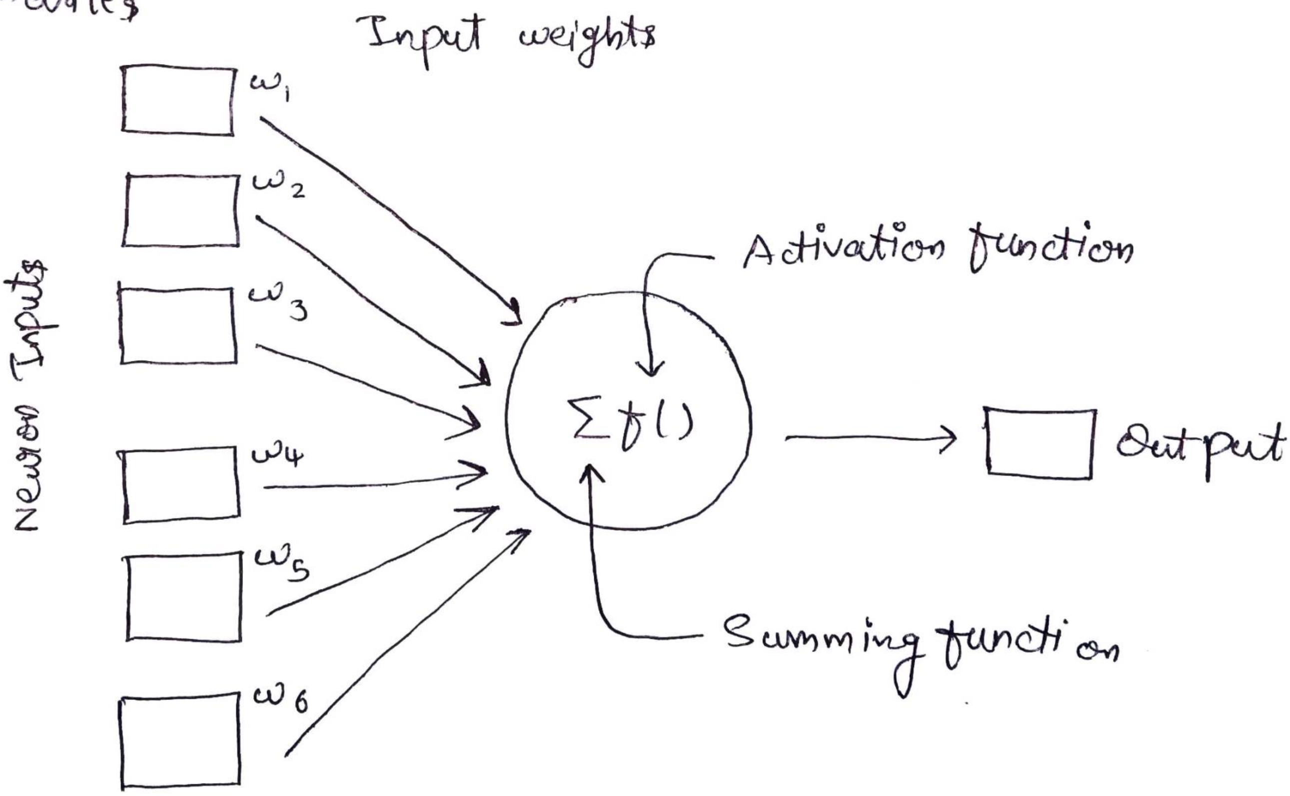
7) K Means 8) Random forest etc.

8.

ii) a) With Bayer's Rule, we want to find the probability an email is spam, given it contains certain words. We do this by finding the probability that each word in the email is spam, and then multiply these probabilities together to get the overall email spam metric to be used in classification.

b)





9. i) Gradient descent is a method for finding the minimum of a function of multiple variables. So, we can use gradient descent as a tool to minimize our cost function. suppose we have a function with variables, then the gradient is the length-n vectors that defines the direction in which the cost is increasing most rapidly.

ii) Can veer off in the wrong direction due to frequent updates.

Lose the benefits of vectorization since we process one observation per time.

Frequent updates are computationally expensive due to using all resources for processing one training sample at a time.

11. A) i) a) The " exclusive-or" problem. It is how to make a neural network produce an identical output when the input conditions don't have anything in common.

b) 3 types of neural network activation func are -

1) Binary Step Function.

2) Linear Step Function

3) Non-linear Activation Function.

11.) a) The probabibility some event in the presence of all outcomes of the other random variable is called the marginal probability / distribution. This is because if all outcomes and possibilities for the 2 variables were laid out together in a table then the marginal probability of one variable would be the sum of probabilities for the other variable on the margin of the table.

$$P(X=A) = SUM\ P(X=A, Y=y_i) \text{ for all } y$$

b) The universal Approximation Theorem states that a neural network with 1 hidden layer can approximate any continous function for inputs within a specific range of the func. jumps around or has large gaps, we won't be able to approximate it.

12.B) i) The loss function used by the perceptron algorithm is called 0-1 loss. 0-1 loss simply means that for each mistaken prediction you incur a penalty of 1 and for each correct prediction incur no penalty. The problem with this loss function is given a linear classifier its hard to move towards a local optimum.

Mean Squared Error is the sum of the squared differences between the prediction and true value. And the output is a single number representing the cost. So the line with the minimum cost function or MSE represents the relationship between x and Y in the best possible manner.

Mid-Sem Exam, Sept. 2021

Name - Hari Kesh Rai
Year - 3rd
Stream - CSE
Section - 3I
Roll No. - 23
Enrollment No. - 12019009001418
Paper Name - It workshop
Paper Code - PCC-CS502
Signature - Hari Kesh Rai
Date - 06/09/2021