USN | | | | | | | | | | |

Internal Assessment Test 2 – October 2019

Sub:	Machine Learning					Sub Code:	15CS73	Branch:	CSE
Date:	12/10/2019	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7 th - A,B & C		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	Explain issues in decision tree learning	[5]	CO3	L2
(b)	Explain inductive bias in decision tree learning.	[5]	CO3	L2
2 (a)	Draw the perceptron network with the notation. Derive an equation of gradient descent rule to minimize the error	[3+7]	CO3	L3
3 (a)	Explain the terms i. Hidden layer ii. Generalization iii. Overfitting iv. Stopping Criterion v. Convergence and local minima	[2+2+2+2+2]	CO3	L2
4 (a)	Write back propagation algorithm which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network	[10]	CO3	L2
5 (a)	Consider two perceptrons defined by the threshold expression $w_0+w_1x_1+w_2x_2>0$. Perceptron A has weight values $w_0=1$, $w_1=2$, $w_2=1$, and Perceptron B has weight values $w_0=0$, $w_1=2$, $w_2=1$ True or False? Perceptron A is more general than perceptron B.	[5]	CO1, 2	L3
(b)	Explain appropriate problems for Neural Network Learning with its characteristics.	[5]	CO3	L2
6 (a)	Prove that posterior probability of hypothesis H(H is consistent with D) is inversely proportionate to version space of H with respect to D by using bayes theorem.	[10]	CO2	L3

7 (a)	A patient takes a lab test and the result comes back positive. It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result in only 97% of the cases. Furthermore, only 0.008 of the entire population has this disease. 1. What is the probability that this patient has cancer? 2. What is the probability that he does not have cancer? 3. What is the diagnosis?	[5]	CO2, 3	L3
(b)	Consider a football game between two rival teams: Team 0 and Team 1. Suppose Team 0 wins 95% of the time and Team 1 wins the remaining matches. Among the games won by Team 0 only 30% of them come from playing on Team 1's football field. On the other hand, 75% of the victories for Team 1 are obtained while playing at home. If Team 1 hosts the next match between the two teams, which team will most likely emerge as the winner?	[5]	CO2, 3	L3
8 (a)	Design a two input perceptron that implements the Boolean function A ^ ~B. Design a two layer network of perceptrons that implements A XOR B.	[5+5]	CO3	L3

15CS73- Machine Learning

Answer key for IAT-2 - Oct-2019

1.a) Explain issues in decision tree learning (any 5 points with explanation for 5 marks)

1. Avoiding Overfitting the Data Reduced error pruning Rule post-pruning
2. Incorporating Continuous-Valued Attributes
3. Alternative Measures for Selecting Attributes
4. Handling Training Examples with Missing Attribute Values
5. Handling Attributes with Differing Costs

1.b) Explain inductive bias in decision tree learning.(5 marks)

Inductive bias is the set of assumptions that, together with the training data, deductively justify the classifications assigned by the learner to future instances . Givena collection of training examples,there are typically many decision trees consistent with these examples. Which of these decision trees does ID3 choose?

ID3 search strategy

(a) selects in favour of shorter trees over longer ones
(b) selects trees that place the attributes with highest information gain closest to the root.

Approximate inductive bias of ID3: Shorter trees are preferred over larger trees

- Consider an algorithm that begins with the empty tree and searches ***breadth first*** through progressively more complex trees.
- First considering all trees of depth 1, then all trees of depth 2, etc.
- Once it finds a decision tree consistent with the training data, it returns the smallest consistent tree at that search depth (e.g., the tree with the fewest nodes).
- Let us call this breadth-first search algorithm BFS-ID3.
- BFS-ID3 finds a shortest decision tree and thus exhibits the bias "shorter trees are preferred over longer trees.

A closer approximation to the inductive bias of ID3: Shorter trees are preferred over longer trees. Trees that place high information gain attributes close to the root are preferred over those that do not.

- ID3 can be viewed as an efficient approximation to BFS-ID3, using a greedy heuristic search to attempt to find the shortest tree without conducting the entire breadth-first search through the hypothesis space.
- Because ID3 uses the information gain heuristic and a hill climbing strategy, it exhibits a more complex bias than BFS-ID3.
- In particular, it does not always find the shortest consistent tree, and it is biased to favour trees that place attributes with high information gain closest to the root.

Restriction Biases and Preference Biases

Difference between the types of inductive bias exhibited by ID3 and by the CANDIDATE-ELIMINATION Algorithm.

ID3

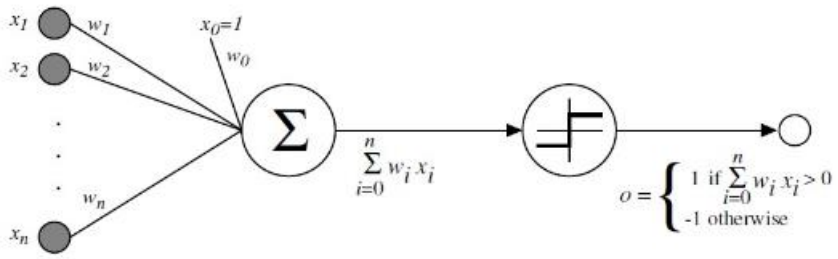
- ID3 searches a complete hypothesis space
- It searches incompletely through this space, from simple to complex hypotheses, until its termination condition is met
- Its inductive bias is solely a consequence of the ordering of hypotheses by its search strategy. Its hypothesis space introduces no additional bias

CANDIDATE-ELIMINATION Algorithm

- The version space CANDIDATE-ELIMINATION Algorithm searches an incomplete hypothesis space
- It searches this space completely, finding every hypothesis consistent with the training data.
- Its inductive bias is solely a consequence of the expressive power of its hypothesis representation. Its search strategy introduces no additional bias
- Occam's razor: is the problem-solving principle that the simplest solution tends to be the right one. When presented with competing hypotheses to solve a problem, one should select the solution with the fewest assumptions.

- Occam's razor: "***Prefer the simplest hypothesis that fits the data***".

2. Draw the perceptron network with the notation(3 marks). Derive an equation of gradient descent rule to minimize the error. (7 marks)



$$o(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$

Sometimes we'll use simpler vector notation:

$$o(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} > 0 \\ -1 & \text{otherwise.} \end{cases}$$

Equation of gradient descent rule to minimize the error

The direction of steepest can be found by computing the derivative of E with respect to each component of the vector $\overrightarrow{w}$. This vector derivative is called the gradient of E with respect to $\overrightarrow{w}$, written as

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right] \qquad \text{equ. (3)}$$

Notice $\nabla E[\vec{w}]$ is itself a vector, whose components are the partial derivatives of $\boldsymbol{E}$ with respect to each of the $\boldsymbol{w_i}$

• The gradient specifies the direction of steepest increase of E, the training rule for gradient descent is

$$\vec{w} \leftarrow \vec{w} + \Delta\vec{w}$$

Where,

$$\Delta\vec{w} = -\eta\nabla E(\vec{w}) \qquad \text{equ. (4)}$$

• Here η is a positive constant called the learning rate, which determines the step
size in the gradient descent search.

• The negative sign is present because we want to move the weight vector in the direction that decreases E

• This training rule can : $w_i \leftarrow w_i + \Delta w_i$ n

Where,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \qquad \text{equ. (5)}$$

Calculate the gradient at each step. The vector of $\frac{\partial E}{\partial wi}$ derivatives that form the gradient can be obtained by differentiating E from Equation (2), as

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2}\sum_d (t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d \frac{\partial}{\partial w_i}(t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d 2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) \\
&= \sum_d (t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w}\cdot\vec{x_d}) \\
\frac{\partial E}{\partial w_i} &= \sum_d (t_d - o_d)(-x_{i,d}) \qquad \text{equ. (6)}
\end{aligned}$$

Substituting Equation (6) into Equation (5) yields the weight update rule for gradient descent

$$\Delta w_i = \eta \sum_{d\in D} (t_d - o_d)\, x_{id} \qquad \text{equ. (7)}$$

3.Explain the terms(explanation of each carries 2 marks each)

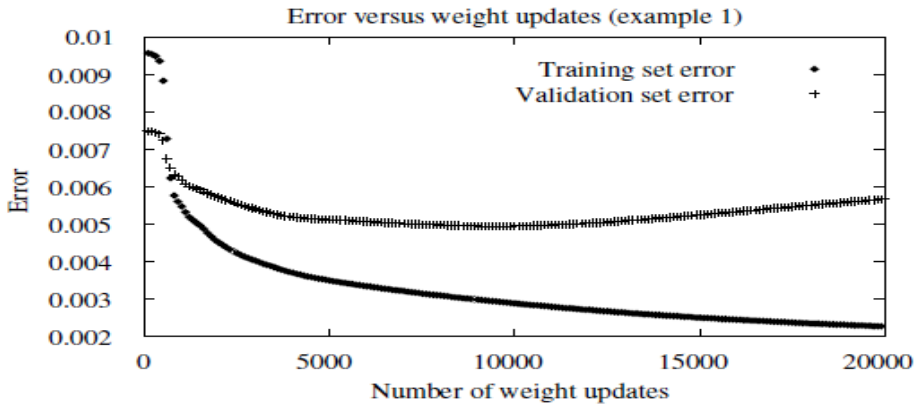
i. **Hidden layer**
ii. **Generalization**
iii. **Overfitting**
iv. **Stopping Criterion**
v. **Convergence and local minima**

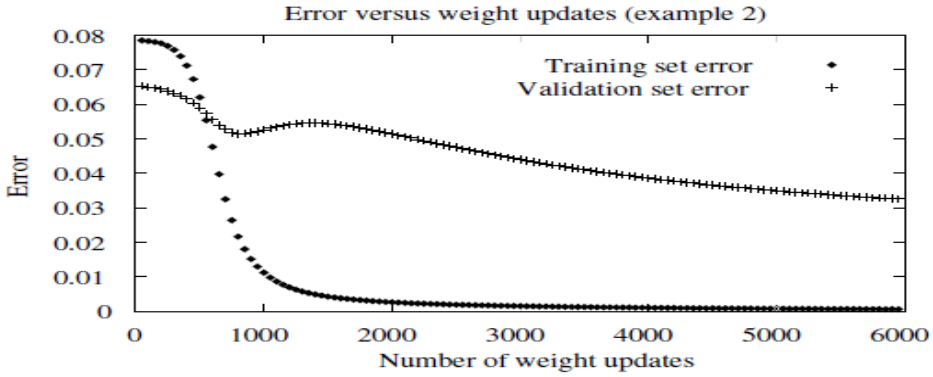
i)Hidden Layer:

This is layer used in between input and output layer to discover intermediate representations at the hidden unit layers inside the network. Because training examples constrain only the network inputs and outputs, the weight-tuning procedure is free to set weights that define whatever hidden unit representation is most effective at minimizing the squared error E. This can lead BACKPROPAGATION to define new hidden layer features that are not explicit in the input representation, but which capture properties of the input instances that are most relevant to learning the target function.

ii)Generalization

In all learning algorithms, generalization is needed to build a model for predicting the unseen examples(future values). For building a model, training is needed until the error E on the training examples falls below some predetermined threshold.
To see the dangers of minimizing the error over the training data, consider how the error E varies with the number of weight iterations





- Consider first the top plot in this figure. The lower of the two lines shows the monotonically decreasing error E over the training set, as the number of gradient descent iterations grows. The upper line shows the error E measured over a

different validation set of examples, distinct from the training examples. This line measures the generalization accuracy of the network-the accuracy with which it fits examples beyond the training data.

iii ,iv) Overfitting and Stopping Criterion

Training the model is continued until the error E on the training examples falls below some predetermined threshold. In fact, this is a poor strategy because the model is susceptible to overfitting the training examples at the cost of decreasing generalization accuracy over other unseen examples.

V. Convergence and local minima

the BACKPROPAGATION algorithm implements a gradient descent search through the space of possible network weights, iteratively reducing the error E between the training example target values and the network outputs. Because the error surface for multilayer networks may contain many different local minima, gradient descent can become trapped in any of these. As a result, BACKPROPAGATION over multilayer networks is only guaranteed to converge toward some local minimum in E and not necessarily to the global minimum error.

4. Write back propagation algorithm which uses stochastic gradient descent method.(8 Marks)

Comment on the effect of adding momentum to the network(2 Marks)

BACKPROPAGATION (*training_example*, η, n_{in}, n_{out}, n_{hidden})

Each training example is a pair of the form $(\vec{x}, \vec{t})$, *where* $(\vec{x})$ *is the vector of network input values,* $(\vec{t})$ *and is the vector of target network output values.*
η *is the learning rate (e.g., .05).* n_i *is the number of network inputs,* n_{hidden} *the number of units in the hidden layer, and* n_{out} *the number of output units.*
The input from unit i into unit j is denoted x_{ji}, *and the weight from unit i to unit j is denoted* w_{ji}

- Create a feed-forward network with n_i inputs, n_{hidden} hidden units, and n_{out} output units.
- Initialize all network weights to small random numbers
- Until the termination condition is met, Do

  - For each $(\vec{x}, \vec{t})$, in training examples, Do

    Propagate the input forward through the network:
    1. Input the instance $\vec{x}$, to the network and compute the output o_u of every unit u in the network.

Propagate the errors backward through the network:

2. For each network output unit k, calculate its error term δ_k

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

3. For each hidden unit h, calculate its error term δ_h

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{h,k}\delta_k$$

4. Update each network weight w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

Where

$$\Delta w_{ji} = \eta \delta_j x_{i,j}$$

Adding momentum:

In BPN the weight update on the n^{th} iteration depend partially on the update that occurred during the $(n-1)^{th}$ iteration can be done as follows:

$$\Delta w_{ji}(n) = \eta\, \delta_j\, x_{ji} + \alpha \Delta w_{ji}(n-1)$$

$0 \le \alpha \le 1$ is a constant called momentum. This speed up the convergence of gradient.

5. a) Consider two perceptrons defined by the threshold expression $w_0+w_1x_1+w_2x_2>0$.

Perceptron A has weight values w_0 =1, w_1 =2, w_2 =1, and

Perceptron B has weight values w_0 =0, w_1 =2, w_2 =1

True or False? Perceptron A is more general than perceptron B. (5 Marks)
Solution:

We will say that $h_j$is (strictly) more-general than h_k(written $h_j >_g h_k$) if and only if $(h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$. Finally, we will sometimes find the inverse useful and will say that ***hj***is ***more specific than*** $h_k$when $h_k$is ***more_general-than*** h_j.

X_1	X_2	$w_0+w_1x_1+w_2x_2$ **Perceptron A**	$w_0+w_1x_1+w_2x_2$ **Perceptron B**	**A more general than B** ($A \geq_g B$)
0	0	1+2*0+1*0=1	0+2*0+1*0=0	1
0	1	1+2*0+1*1=2	0+2*0+1*1=1	1
1	0	1+2*1+1*0=3	0+2*1+1*0=2	1
1	1	1+2*1+1*1=4	0+2*1+1*1=3	1

B(<x1,x2>) = 1→ 2x1+x2 > 0 → 1+2x1+x2 > 0 → A(<x1,x2>)) = 1

Hence **Perceptron A is more general than perceptron B - True.**

5.b) Explain appropriate problems for Neural Network Learning with its characteristics. (5 Marks)

1.*Instances are represented by many attribute-value pairs.*
The target function to be learned is defined over instances that can be described by a vector of
predefined features. These input attributes may be highly correlated or independent of one another. Input values can be any real values.

2.The target function output may be discrete-valued, real-valued, or a vector of several real- or discrete-valued attributes.

3.The training examples may contain errors.
ANN learning methods are quite robust to noise in the training data.

4.Long training times are acceptable.
Network training algorithms typically require longer t
raining times than, say, decision tree learning algorithms. Training times can range from a few seconds to many hours, depending
on factors such as the number of weights in the network, the number of training examples considered, and the settings of various learning algorithm parameters.

5.Fast evaluation of the learned target function may be required.

Though ANN learning times are relatively long, evaluating the learned network, in order to apply it to a subsequent instance, is typically very fast.

6. ***The ability of humans to understand the learned target function is not important.***

The weights learned by neural networks are often difficult for humans to interpret. Learned neural networks are less easily communicated to humans than learned rules.

Examples:

a. Speech recognition
b. Image Classification
c. Financial Predictions

6. Prove that posterior probability of hypothesis H(H is consistent with D) is inversely proportionate to version space of H with respect to D by using bayes theorem.(10 marks)

P(D|h) is the probability of observing the target values D = (d_1 . . .d_m) for the fixed set of instances (x_1 . . . x_m), given a world in which hypothesis h holds (i.e., given a world in which h is the correct description of the target concept c)

Recalling Bayes theorem, we have

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

First consider the case where h is inconsistent with the training data D. Since Equation (6.4) defines $P(D|h)$ to be 0 when h is inconsistent with D, we have

$$P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0 \text{ if } h \text{ is inconsistent with } D$$

The posterior probability of a hypothesis inconsistent with D is zero.

Now consider the case where h is consistent with D. Since Equation (6.4) defines $P(D|h)$ to be 1 when h is consistent with D, we have

$$\begin{aligned} P(h|D) &= \frac{1 \cdot \frac{1}{|H|}}{P(D)} \\ &= \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} \\ &= \frac{1}{|VS_{H,D}|} \text{ if } h \text{ is consistent with } D \end{aligned}$$

$$\begin{aligned} P(D) &= \sum_{h_i \in H} P(D|h_i)P(h_i) \\ &= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \cdot \frac{1}{|H|} \\ &= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} \\ &= \frac{|VS_{H,D}|}{|H|} \end{aligned}$$

To summarize, Bayes theorem implies that the posterior probability $P(h|D)$ under our assumed $P(h)$ and $P(D|h)$ is

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases} \quad (6.5)$$

7.a) A patient takes a lab test and the result comes back positive. It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result in only 97% of the cases. Furthermore, only 0.008 of the entire population has this disease. (5 Marks)

1. **What is the probability that this patient has cancer?**
2. **What is the probability that he does not have cancer?**
3. **What is the diagnosis?**

$$\begin{aligned} P(cancer) &= .008, & P(\neg cancer) &= .992 \\ P(\oplus|cancer) &= .98, & P(\ominus|cancer) &= .02 \\ P(\oplus|\neg cancer) &= .03, & P(\ominus|\neg cancer) &= .97 \end{aligned}$$

$$P(\oplus|cancer)\,P(cancer) = (.98).008 = .0078$$
$$P(\oplus|\neg cancer)\,P(\neg cancer) = (.03).992 = .0298$$

Diagnosis: $h_{MAP} = \neg cancer.$

7.b) Consider a football game between two rival teams: Team 0 and Team 1. Suppose Team 0 wins 95% of the time and Team 1 wins the remaining matches. Among the games won by Team 0 only 30% of them come from playing on Team 1's football field. On the other hand, 75% of the victories for Team 1 are obtained while playing at home. If Team 1 hosts the next match between the two teams, which team will most likely emerge as the winner? (5 Marks)

Let X be the the team hosting the match

Y be the winner of the match

Prob.of Team 0 wins is(P(Y=0) = 0.95

Prob.of Team 1 wins is(P(Y=1) = 0.05 (1-0.95)

Prob.of Team 1 hosted the match it won is P(X=1/Y=1)=0.75

Prob.of Team 1 hosted the match won by Team 0 is P(X=1/Y=0)=0.3

$$P(Y=1|X=1) = \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1)}$$

$$= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1,Y=1)+P(X=1,Y=0)}$$

$$= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1|Y=1)P(Y=1)+(X=1|Y=0)P(Y=0)}$$

$$= \frac{0.75\text{x } 0.05}{0.75x0.05+0.3x\ 0.95}$$

$$= \frac{0.0375}{0.0375+0.285}$$

= 0.1162

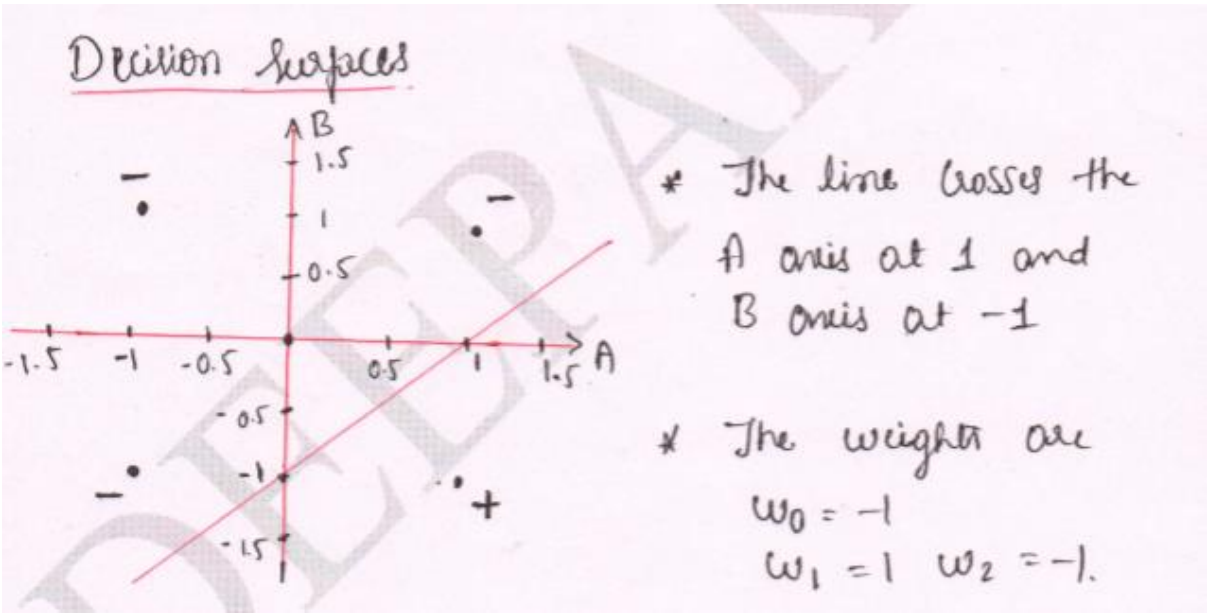
P(Y=0 | X=1) =1- 0.1162

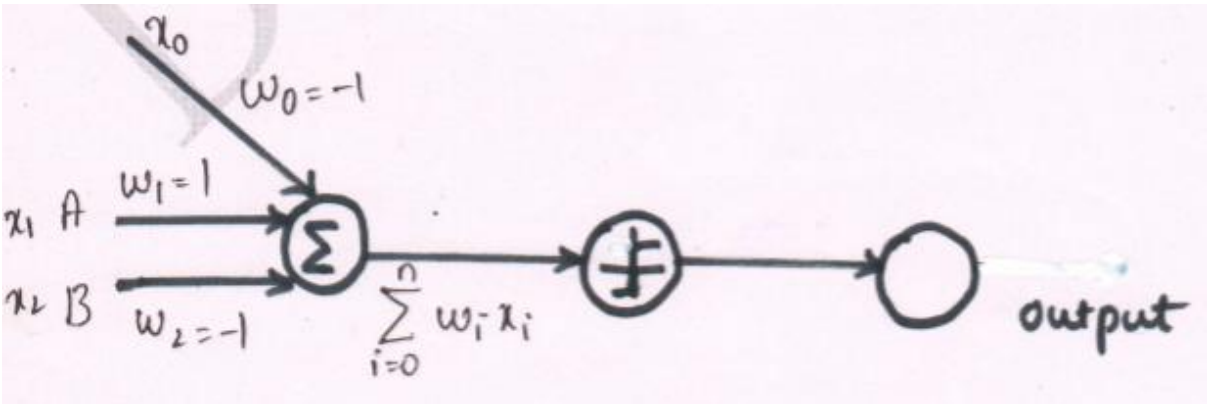
= 0.8838

Since P(Y=0 | X=1) > P(Y=1 | X=1), **Team 0 has a better chance than Team-1 of winning the next match.**

8.a)Design a two input perceptron that implements the Boolean function A ^ ~B. (5 Marks)

A	B	Not B	A AND (NOT B)
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0





b)Design a two layer network of perceptrons that implements A XOR B. (5 Marks)

* Express A XOR B in terms of other logical connectives

$$A \text{ XOR } B = (A \wedge \neg B) \vee (\neg A \wedge B)$$

* Define the perceptron P_1 and P_2 for $(A \wedge \neg B)$ & $(\neg A \wedge B)$

* Composing the outputs of P_1 & P_2 into a Perceptron P_3 that implements $O(P_1) \vee O(P_2)$