School of Information Sciences (SOIS), MAHE, Manipal 576104

Second Semester Master of Engineering - ME (Big Data and Data Analytics)

Degree Examination April - 2018

BDA 602 Machine Learning

Scheme for Evaluation

1A. Write the steps involved in *choosing the Training Experience* in designing a machine learning systems. (6 marks)

Steps in choosing the Training Experience

- Choose the type of training experience
- Degree to which the learner controls the sequence of training examples
- How well it represents the examples over which the final system performance P must be measured

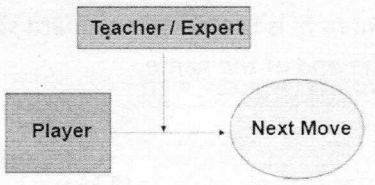
1. Choose the type of training experience from which our system will learn (2 Marks)

- Training Types:
  - Direct Training
  - Indirect Training
- Training type – impact on success/ failure of the learner

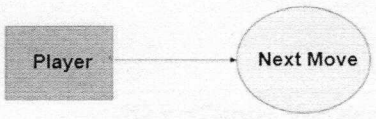
Direct Training - consisting of individual checkers board states and the correct move for each.
Indirect Training - consisting of the move sequences and final outcomes of various games played.

2. Degree to which the learner controls the sequence of training examples (2 Marks)

Degree to which the learner controls the sequence of training examples.



Learner rely on the teacher to select board states and to provide the correct move for each.



Learner itself propose board states (finds confusing and ask the teacher for the correct move).

3. Representation of the distribution of examples over which the final system performance P must be measured

A checkers learning problem: (2 Marks)

Task T: playing checkers
Performance measure ***P:*** percent of games won in the world tournament
Training experience E: games played against itself
In order to complete the design of the learning system, we must now choose
1. Exact type of knowledge to be learned
2. Representation for this target knowledge and 3. Learning mechanism

1B. How will you *choose the Target Function* in designing a machine learning system? Explain with example

(4 marks)

- Checkers-playing program generate the ***legal moves***
- The program needs only to learn ***how to choose the best*** move from among these legal moves.
- Let us call this function ***ChooseMove***

$$ChooseMove : B \rightarrow M$$

- Input: Set of legal board states ***B***
- Output move from the set of legal moves ***M***
- Let us call this target function ***V***

$$V : B \rightarrow R$$

- ***Target function V*** maps any legal board state from the set B to some real value (*R denotes* the set of real numbers).

(2 Marks)

- Let us therefore define the target value ***V(b)*** for an arbitrary board state ***b*** in *B*, ***as*** follows:
  - if b is a final board state that is **won**, then ***V(b)*** = 100
  - if b is a final board state that is **lost**, then ***V(b)*** = -100
  - if b is a final board state that is **drawn**, then ***V(b)*** = ***0***
  - if b is a not a final state in the game, then V(b) = V(b'), where b' is the best final board state that can be achieved starting from b and playing optimally until the end of the game.

Note: V(b) for every board state is not efficiently computable.

(2 Marks)

2. Write *FIND-S algorithm* to find a Maximally Specific Hypothesis and obtain the hypothesis space search by Find-S for the given training examples.

(3+7 Marks)

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Find-S Algorithm

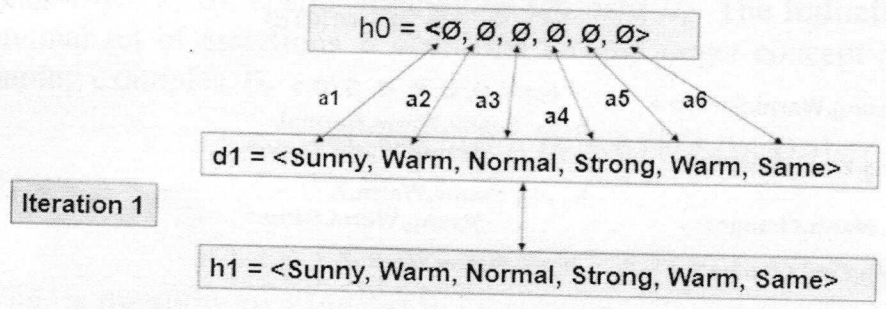
(3 Marks)

1. *Initialize h to the most specific hypothesis in H*
2. *For each positive training instance x*
   - *For each attribute constraint a_i in h*
     - *If the constraint a_i in h is satisfied by x*
     - *Then do nothing*
     - *Else replace a_i in h by the next more general constraint that is satisfied by x*
3. *Output hypothesis h*

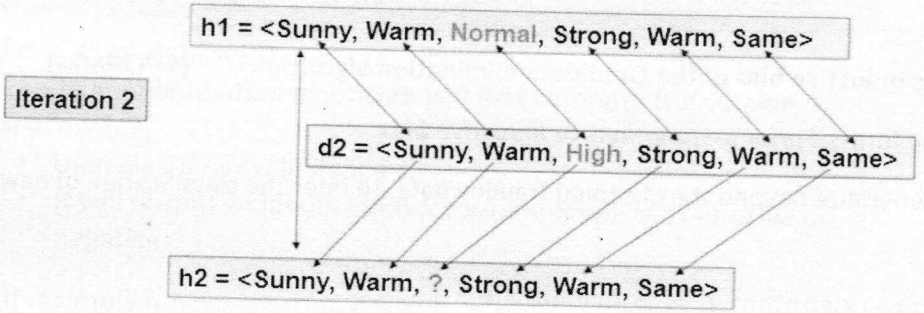
(7 Marks)

Step-1: **h0 = <Ø, Ø, Ø, Ø, Ø, Ø>**

Step-2:



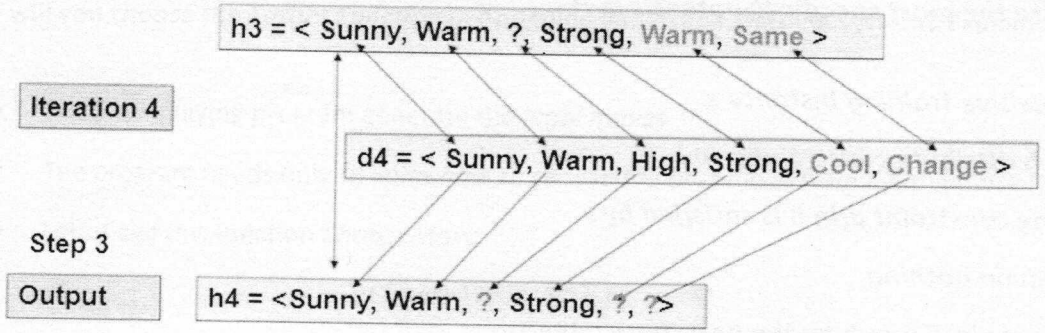
1 Mark



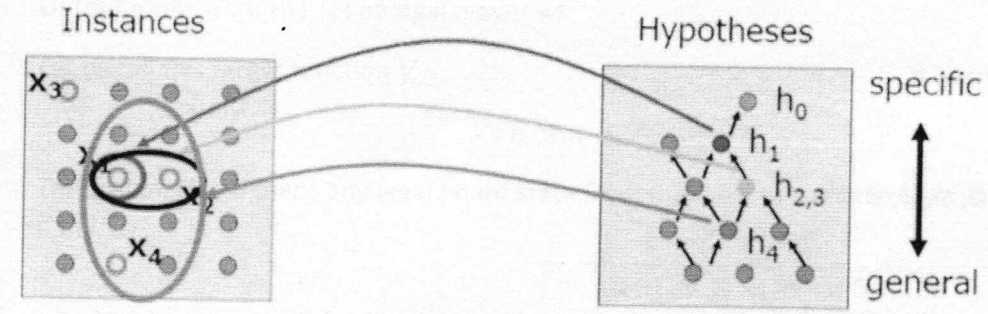
2 Marks

Iteration 3 | Ignore | h3 = <Sunny, Warm, ?, Strong, Warm, Same>

1 Mark



2 Marks



1 Mark

x_1=<Sunny,Warm,Normal,Strong,Warm,Same>+

x_2=<Sunny,Warm,High,Strong,Warm,Same>+

x_3=<Rainy,Cold,High,Strong,Warm,Change> -

x_4=<Sunny,Warm,High,Strong,Cool,Change> +

h_0=< Ø, Ø, Ø, Ø, Ø, Ø,>

h_1=< Sunny,Warm,Normal, Strong,Warm,Same>

$h_{2,3}$=< Sunny,Warm,?, Strong,Warm,Same>

h_4=< Sunny,Warm,?, Strong,?,?>

3. Define inductive bias. Explain the inductive bias of the Candidate-elimination algorithm? (4+4 Marks)

Inductive learning requires some form of ***prior assumptions or inductive bias.***

Key idea: make the learner to generalize beyond the observed training data, to infer the classification of new instances.

consider an arbitrary learning algorithm L is provided an arbitrary set of training data $D_c = \{(x, c(x))\}$ of some arbitrary target concept c.

After training, L is asked to classify a new instance x_i.
Classification $L(x_i, D_c)$ may be positive or negative.

The inductive inference step performed by L as follows

$$(D_c \wedge x_i) \succ L(x_i, D_c)$$

$L(x_i, D_c)$ is inductively inferred from $(D_c \wedge x_i)$

(2 Marks)

- Inductive inference for the instances C and D unknown (+ve or – ve).

- What additional assumptions could be added to $D_c \wedge x_i$ so that $L(x_i, D_c)$ would follow deductively

Definition:
Inductive bias of L is the set of assumptions (assertions) B such that for all new instances x_i.

$$(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)$$

Set of additional assumptions B sufficient to justify its inductive inferences as deductive inferences.

(2 Marks)

Definition: Consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and let $D_c = \{\langle x, c(x)\rangle\}$ be an arbitrary set of training examples of c. Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c. The **inductive bias** of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

(2 Marks)

What is the inductive bias of the Candidate-elimination algorithm?

- For a given data set D_c
- Candidate-elimination algorithm will first compute the version space $VS_{H,Dc}$
  - Then classify new instance x_i
  - It will output a classification for x_i unanimously as positive or negative
- It is simply the assumption $c \in H$.

Given this assumption, each inductive inference performed by the Candidate-elimination algorithm can be justified deductively.

Inductive bias of Candidate-elimination algorithm: The target concept c is contained in the given hypothesis space H.

(2 Marks)

4. Obtain the entropy and information gain to find the root node of the decision tree for the following training Examples. **(12 Marks)**

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification (yes/no) is

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

where $p_{\oplus}$ is the proportion of positive examples in **S** and $p\Theta$, is the proportion of negative examples in S. In all calculations involving entropy we define 0 log 0 to be 0.

(2 Marks)

There are 14 examples. 9 positive and 5 negative examples [9+, 5-].

The entropy of S relative to this boolean (yes/no) classification is

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

(2 Marks)

Information gain, is simply the expected reduction in entropy caused by partitioning the examples according to this attribute.

More precisely, the information gain, ***Gain(S, A)*** of ***an*** attribute **A**, relative to a collection of examples ***S,*** is defined as

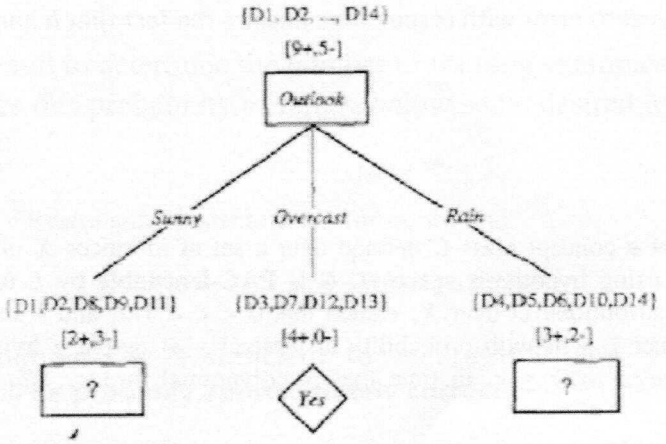
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where ***Values(A)*** is the set of all possible values for attribute **A**, and ***Sv,*** is the subset of **S** for which attribute **A** has value **v, i.e.,**

$$S_v = \{s \in S | A(s) = v\}$$

(2 Marks)

$$Gain(S, Outlook) = 0.246$$
$$Gain(S, Humidity) = 0.151$$
$$Gain(S, Wind) = 0.048$$
$$Gain(S, Temperature) = 0.029$$



(6 Marks)

5A. Defining the true error of a hypothesis h with respect to target concept c and instance distribution D. With suitable diagram explain the error of a hypothesis h with respect to target concept **(7 Marks)**

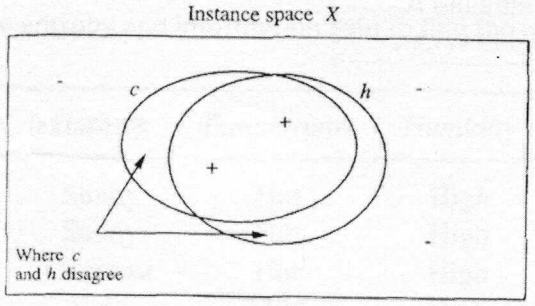
- Let us begin by defining the ***true error*** of a hypothesis ***h*** with respect to target concept c and instance distribution *D.*

Definition: The **true error** (denoted $error_{\mathcal{D}}(h)$) of hypothesis h with respect to target concept c and distribution $\mathcal{D}$ is the probability that h will misclassify an instance drawn at random according to $\mathcal{D}$.

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[c(x) \neq h(x)]$$

$\Pr_{x \in \mathcal{D}}$ - probability is taken over the instance distribution D.

(3 Marks)



(2 Marks)

- The **error** of h with respect to *c* is the probability that a randomly drawn instance will fall into the region where h and *c* disagree on its classification.
- The + and - points indicate positive and negative training examples.

Note h has a non-zero error with respect to *c* despite the fact that *h* and *c* agree on all five training examples observed.

(2 Marks)

5B. Define PAC Learning **(3 Marks)**

Definition: Consider a concept class C defined over a set of instances X of length n and a learner L using hypothesis space H. C is **PAC-learnable** by L using H if for all $c \in C$, distributions $\mathcal{D}$ over X, ϵ such that $0 < \epsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, n, and $size(c)$.

6. State and prove the ε-Exhausted Version Space theorem to determine the number of training examples required to reduce this probability of failure below some desired level δ. **(10 marks)**

ε-Exhausted Version Space

Definition: The version space $VS_{H,D}$ is said to be **ϵ-exhausted** with respect to c and $\mathcal{D}$, if every hypothesis h in $VS_{H,D}$ has error less than ϵ with respect to c and $\mathcal{D}$.

$$(\forall h \in VS_{H,D})\ error_{\mathcal{D}}(h) < \epsilon$$

(2 Marks)

Theorem: [Haussler, 1988].

If the hypothesis space H is finite, and D is a sequence of $m \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to H and D is not ϵ-exhausted (with respect to c) is less than

$$|H|e^{-\epsilon m}$$

(2 Marks)

Proof:

- Let $h_1, h_2, \ldots h_k$ be all the hypotheses in H that have true error greater than ε with respect to c. [$H_{bad} = \{h_1, h_2, \ldots h_k\}$]
- We fail to ε-exhaust the version space if and only if at least one of these k hypotheses happens to be consistent with all ***m*** independent random training examples.

- The probability that any single hypothesis having true error greater than ε would be consistent with one randomly drawn example is at most (1 - ε).

- Probability that this hypothesis will be consistent with ***m*** independently drawn examples is at most $(1 - \varepsilon)^m$

- Given that we have k hypotheses with error greater than ε, the probability that at least one of these will be consistent with all ***m*** training examples is at most

$$k(1-\varepsilon)^m$$

since k is ≤ |H|, this is at most $H(1-\varepsilon)^m$

- We use a general inequality stating that if $0 \leq \varepsilon \leq 1$; then then $(1 - \varepsilon) \leq e^{-\varepsilon}$.

$$\boxed{k(1-\epsilon)^m \leq |H|(1-\epsilon)^m \leq |H|e^{-\epsilon m}}$$

(4 Marks)

Let us use this result to determine the number of training examples required to reduce this probability of failure below some desired level δ.

$$|H|e^{-\epsilon m} \leq \delta$$

Rearranging terms to solve for m, we find

$$m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta))$$

- Number of training examples ***m*** is sufficient to assure that any consistent hypothesis will be **probably approximately correct**.

(2 Marks)

7A. Define Bayes theorem. **(3 Marks)**

For two events, A and B, Bayes' theorem allows you to figure out p(A|B) (the probability that event A happened, given that test B was positive) from p(B|A) (the probability that test B happened, given that event A happened).

The formal definition for the rule is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

7B. Find a patient's probability of having liver disease if they are an alcoholic for the following data: The past data tells you that 10% of patients entering your clinic have liver disease and five percent of the clinic's patients are alcoholics. Among those patients diagnosed with liver disease, 7% are alcoholics. If the patient is an alcoholic, what is the chances of having liver disease?

(7 marks)

To find a patient's probability of having liver disease if they are an alcoholic.

"Being an alcoholic" is the test for liver disease.

A : mean the event "Patient has liver disease."

Past data tells you that 10% of patients entering your clinic have liver disease. P(A) = 0.10. (2 Marks)

B : mean the test that "Patient is an alcoholic."
Five percent of the clinic's patients are alcoholics. P(B) = 0.05.

B|A: Among those patients diagnosed with liver disease, 7% are alcoholics.
Probability that a patient is alcoholic, given that they have liver disease, is 7%. (2 Marks)

Bayes' theorem tells you:

P(A|B) = (0.07 * 0.1)/0.05 = 0.14 (2 Marks)

If the **patient is an alcoholic**, their **chances of having liver disease is 0.14 (14%).**
This is a large increase from the 10% suggested by past data.
But it's still unlikely that any particular patient has liver disease. (1 Marks)

8A. Data set given in the table is for a company produce tissues (used by biological labs). Company objective is to predict how well their products are accepted by the clients. They conducted a survey with their clients to find the acceptance levels of the products. As shown in figure, Type- 1 and 2 are not well accepted whereas Type- 3 and 4 are well accepted. **(6 Marks)**

Name	Acid Durability	Strength	Class
Type-1	7	7	Bad
Type-2	7	4	Bad
Type-3	3	4	Good
Type-4	1	4	Good

Test data: Type-5; Acid Durability = 3; Strength = 7 and find the class for this case?

Apply the Euclidian distance measure for the data to find the distances from the new data **Type-5.**

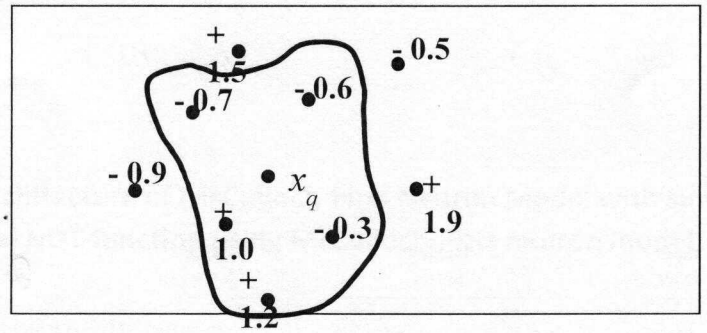
Name	Acid Durability	Strength	Distance	NeighborRank
Type-1	7	7	Sqrt((7-3)²+(7-7)²) =4	3
Type-2	7	4	Sqrt((7-3)²+(4-7)²)=5	4
Type-3	3	4	Sqrt((3-3)²+(4-7)²)=3	1
Type-4	1	4	Sqrt((1-3)²+(4-7)²)=3.6	2

4 Marks

- **If k =1**, ONE immediate neighbor Type 3 Good (new type is = Good)
- **If k =2**, TWO immediate neighbor Type 3, Type 4 = Good; (new type is = Good)
- **If k =3**, THREE immediate neighbor Type 3, Type 4 = Good & Type 1 = Bad (but the probability of Good is high so, consider new type is classified as Good)

2 Marks

8B. In the diagram below the numbers next to the + and - signs refer to the values taken by a real-valued target function. Calculate the value predicted for the target function at the query instance x_q by the 5-Nearest Neighbour Learning. (4 Marks)



$$\tilde{f}_k(x_q) = \frac{1}{k}\sum_{i=1}^{k} f_k(x_i)$$

2 Marks

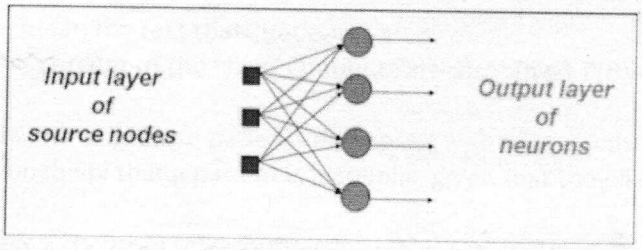
where x_i are the $k-$nearest neighbours

$$\tilde{f}_k(x_q) = \frac{1}{5}\{1.0 + 1.2 - 0.6 - 0.7 - 0.3\} = 0.12$$

2 Marks

9A. Explain the structure of single layer and multilayer artificial neural network models with suitable diagrams (5 Marks)

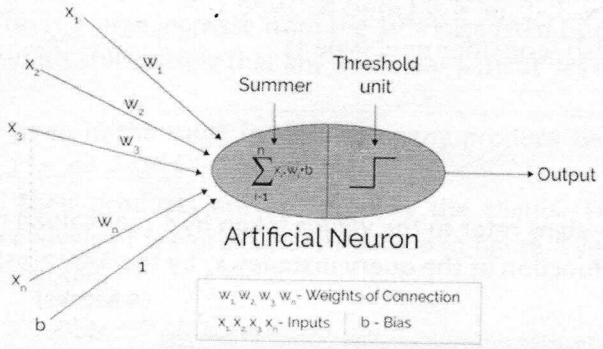
Single Layer ANN



- Only one layer of weighted interconnections
- Weighted input(s) are processed by only one layer and provide output(s)

2 Marks

- Single Layer ANN - Example

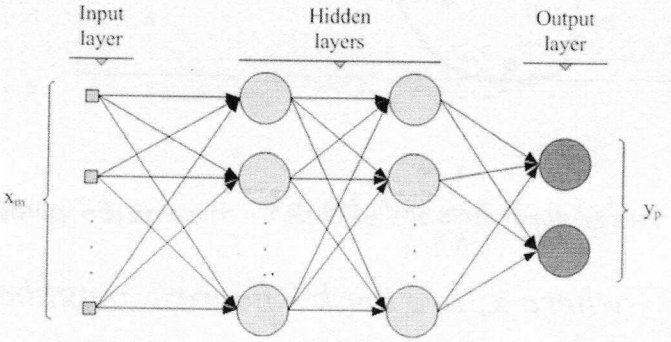


X_1, X_2 ... are input
b – bias weight
w_1, w_2 ... are interconnecting weights

1 Mark

Multilayer ANN

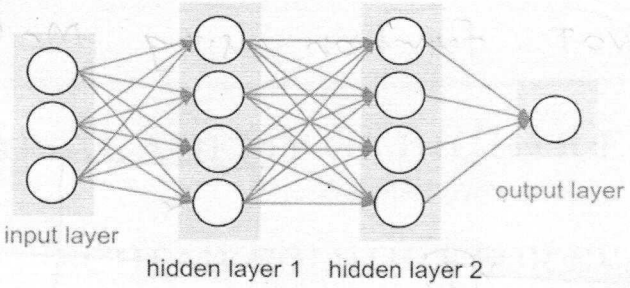
- Multilayer ANN are called layered networks
  - Input layer
  - Hidden layer(s)
  - Output layer



2 Marks

9B. Describe the architecture models of feed forward and Recurrent Neural Network models with suitable diagrams. (5 Marks)

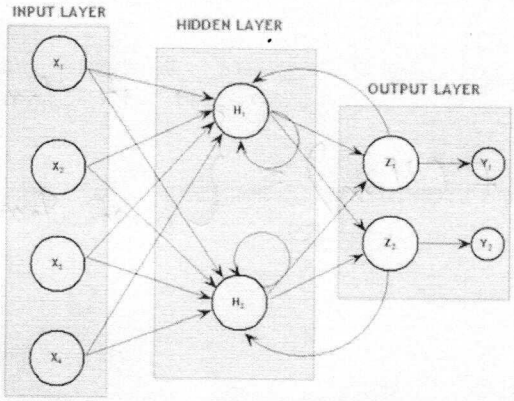
Feed Forward Neural Network Model



2 Marks

- The information is propagated from the inputs to the outputs
- Time has no role (NO cycle between outputs and inputs)

Recurrent Neural Network Model



- All nodes are connected to all other nodes
- Every node is both input/ output node
- Delays are associated
- Training is more difficult
- Performance may be problematic
  - Stable Outputs may be more difficult

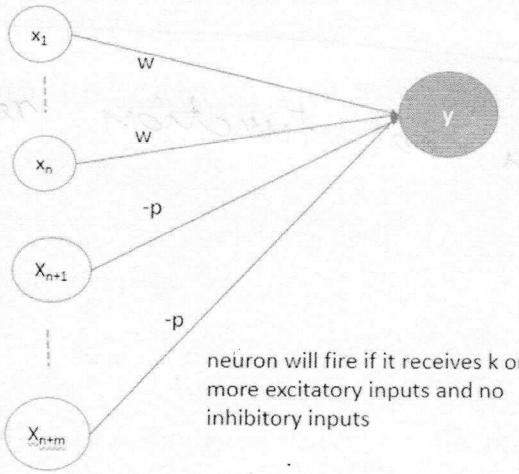
3 Marks

10. Describe the architecture of McCulloch-Pitts Neuron Model with suitable diagram. Also realize the output of logical NOT function using McCulloch-Pitts neuron model. (10 Marks)

- Here, y is McCulloch-Pitts neuron
- Receives signal from any number of neurons
- Connection weights from x_1 .. x_n are excitatory, denoted by w.
- Connection weights from x_{n+1} .. X_{n+m} are inhibitory, denoted by –p.
- McCulloch-Pitts neuron y has the activation function

$$f(y_{in}) = \begin{cases} 1; if\ f(y_{in}) \geq \theta \\ 0; if\ f(y_{in}) < \theta \end{cases}$$

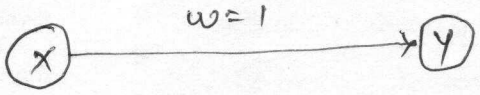
- Where, θ – threshold and y_{in} - net input to y



neuron will fire if it receives k or more excitatory inputs and no inhibitory inputs

4 Marks

Realize logical NOT function using McCulloch-Pitts neuron Model



x	y
0	1
1	0

Threshold on unit $y = 1$

$$y_{in} = x \cdot w = x \cdot 1 = x$$

Activation function $y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} < 1 \\ 0 & \text{if } y_{in} \geq 1 \end{cases}$

Presenting inputs

x	y_{in}	Activation	output (y)
0	0	$0 < 1$	1
1	1	$1 \geq 1$	0

Thus NOT function realized.

(6 Marks)