PCA

Define PCA. Briefley describe the steps of PCA calculation.

Principle component analysis, or PCA, is a dimensionality reduction method that is often used to reduce the dimensionality of a large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

The steps of PCA calculation:

(1) Standardization:

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contribute equally to the analysis.

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}$$

2. Covariance matrix computation :

The aim of this step is to understand how the variables of the input data are varying from the mean with respect to each other, or in other words, to see if there is relationship between them.

$$\mathrm{Cov}(x,y) = \sum_{i=1}^{n} \frac{(x_i-\bar{x})(y_i-\bar{y})}{n-1}$$

$$C = \begin{array}{cc} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} x \\ \\ y \end{array} & \begin{bmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) \\ & \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) \end{bmatrix} \end{array}$$

3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components.

Eigen vectors and eignvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.

Step 4: Feature vector:

Computing the eigenvectors and ordering them by their eigenvalues in descending order, allows us to find the principal components in order of significance. In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance, and form ~~ah~~ with the remaining ones a matrix of vectors that we call Features vector.

5. Recast the data along the principal component axis:

$$\text{Final\_data\_set} = \text{FeatureVector}^{T} * \text{StandardizeOriginalDataset}^{T}$$

T → Transpose.

4) Explain the dimension reduction mechanism of PCA?

Let, $\lambda_1, \lambda_2, \lambda_3$ denote the largest l eigenvalues of the correlation matrix R. We may then approximate the data vector x by turncating by expansion of equation $x = Qa = \sum_{j=1}^{m} a_j q_j$ —(1) after l terms are get,

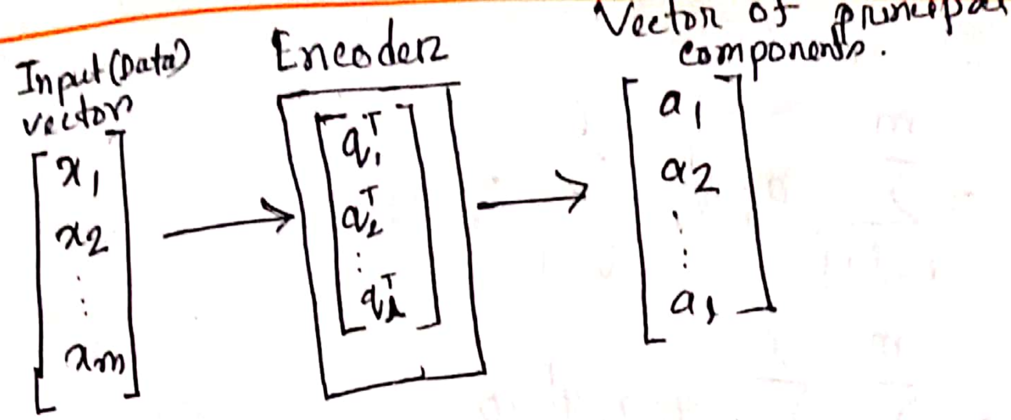
$$x = \sum_{j=1}^{l} a_j q_j \quad \text{—(1)}$$

$$= [q_1, q_2 \dots q_l] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix}, \quad l \leq m \quad \text{—(2)}$$

Given the original data vector x we may use to compute the set of principal components retained in equation (2) as follows.

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_l^T \end{bmatrix} x, \quad l \leq m \quad \text{—(3)}$$

The linear projection of equation (3) from R^m to R^l represent on encoder for the approximate representation of the data vector x correspondingly linear projection of eqn (2)



~~fig~~ fig : Encoding

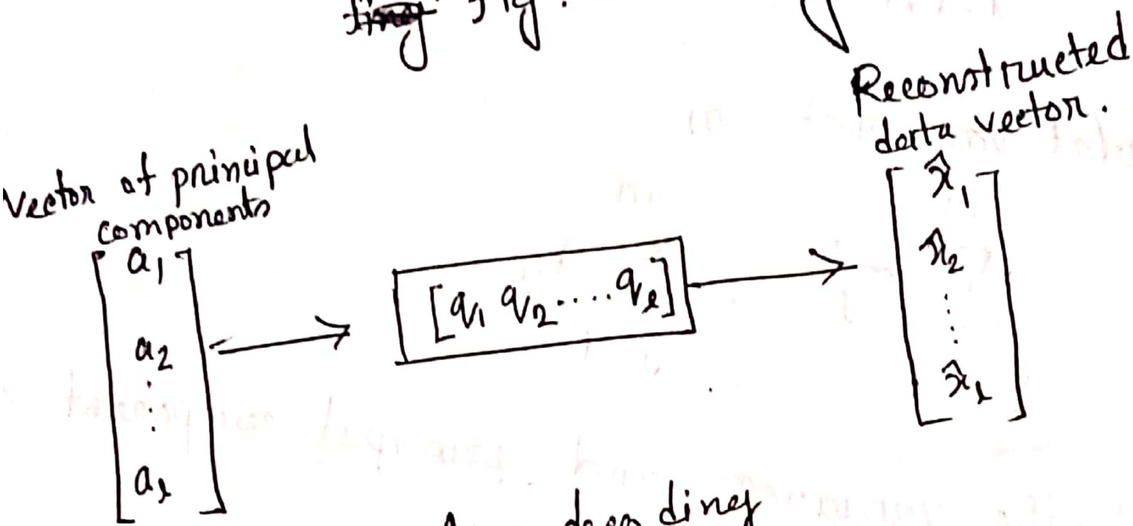


fig: decoding

~~from $\mathbb{R}^l$ to $\mathbb{R}^m$ represents a decoder~~

The approximation error vector e equals the difference bet^n the original data vector x and the approximating data vector $\hat{x}$, as shown by

$$e = x - \hat{x} \qquad \text{——— (4)}$$

From eq^n (1), (2) and (4),

$$e = \sum_{J=l+1}^{m} a_j q_J \qquad \text{——— (5)}$$

Using (2) and (5)

$$e^T \hat{x} = \sum_{i=l+1}^{m} a_i q_i^T \sum_{J=1}^{l} a_J q_J$$

$$= \sum_{i=l+1}^{m} \sum_{J=1}^{l} a_i a_j q_i^T q_j \qquad \text{—— (6)}$$

total variance m

$$\sum_{J=1}^{m} \sigma_J^2 = \sum_{J=1}^{m} \lambda_J$$

σ_J^2 is the variance and principal component a_j

$$\sum_{J=1}^{l} \sigma_J^2 = \sum_{J=1}^{l} \lambda_J$$

The total variance of the $(l-m)$ elements approximation error vector $x-\hat{x}$

$$\sum_{J=l+1}^{m} \sigma_J^2 = \sum_{J=l+1}^{m} \lambda_j$$

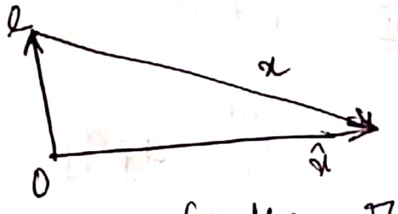


fig: Illustration of the relationship between vector x, its reconstructed version $\hat{x}$, and error vector

~~Briefly describe the models of PCA.~~

What is confusion matrix? Calculate P-value, accuracy, sensitivity, specificity and f-score from the following confusion matrix

		Predicted class		
		Yes	No	Total
Actual class	Yes	523	63	586
	No	137	112	249
	Total	660	175	835

Confusion matrix: A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

		Predicted class		
		Yes	No	Total
Actual class	Yes	523 TP	63 FN	586
	No	137 FP	112 TN	249
	Total	666	175	

$\therefore$ Accuracy $= \dfrac{TP+TN}{TP+TN+FP+FN} = \dfrac{523+112}{835}$

precision/p-value $= \dfrac{Tp}{Tp+Fp} = \dfrac{523}{523+137}$

Recall/sensitivity $= \dfrac{Tp}{Tp+FN} = \dfrac{523}{523+63}$

specificity $= \dfrac{TN}{TN+FP} = \dfrac{112}{112+137}$

f-score $= 2\times \dfrac{\text{Precision} \times \text{Recall}}{\text{precision} + \text{Recall}}$

Fuzzy Vs crisp

Fuzzy	crisp
Prescribe by vague or ambiguous properties.	Defined by precise and certain characteristics
Elements are allowed to be partially included in the set.	Elements is either the member of a set or not
Used in fuzzy controllers	Digital design
Infinite valued	bi-valued

[iii] Fuzzy vs probability

Fuzzy	Probability
Fuzzy logic is a concept of fuzzy set membership	Probability is the concept of subjective probability
Fuzzy logic determines how much a variable is in a set	Probality determines how probable that a variable is in a set
Fuzz logic is concerned with the undecidability in the outcome of clearly defined and randomly occuring	Probability is concerned with the undecidability in the outcome of the set

[4] Define α-cut and strong α-cut with proper eqn.

α-cut:

(i) It is a deffuzziification method.

(ii) In this method, a fuzzy set A is transformed into a crisp set.

A_α for a given value of α $(0 \leq \alpha \leq 1)$

(iii) $A_\alpha = \{\mu_A(a) \geq \alpha, \forall x \in X\}$.

Strong α cut :

A set A_α is called strong α cut set if it consists of all the elements of a fuzzy set whose membership function have value strictly greater than a specified value.

$$\cancel{A_\alpha = \{x \mid \mu_A(x) > \alpha\}}$$

$$A_\alpha = \{ \mu_A(x) > \alpha \ \ \forall x \in X \}$$

Q) Briefly explained the types of defuzzification.

Defuzzification is the process of transforming a fuzzy output of a fuzzy inference system into a crisp output.

crisp input ⟹ $\boxed{\text{Fuzzy System}}$ ⟹ crisp output.

Types of defuzzification:

(i) Lambda cut
(ii) Maxima
(iii) Weighted sum
(iv) centroid.

Ⓠ What are the types of Fuzzy rule based model?

(i) Linguistic fuzzy Model
(ii) ~~Max-M~~ min-max fuzzy model. (পড়তে হবে)
(iii) Product sum fuzzy model
(iv) Functional fuzzy model
(v) Gradual Fuzzy model.

Ⓠ Explain the ~~co~~ core of a fuzzy set.

The core of a fuzzy set A is the crisp subset of x consisting of all elements with membership grades equal to one.

$$core(A) = \{x \mid \mu_A(x) = 1\}.$$

Briefly explain the construction of SVM for pattern recognition and derive the following eqn $\sum_{i=1}^{N} \alpha_i d_i k(x, x_i) = 0$

or,
Explain the building of SVM for pattern recognition using inner product kernel.

Building of SVM for pattern recognition using inner product kernel:

We know that, given a set of non linear transformations, a ~~hyper~~ hyperplane acting as the decision surface is,

$$\sum_{J=1}^{m} w_J \phi_J(x) + b = 0 \quad \text{——— (I)}$$

where, w = weight
b = bias

If it is simplified then,

$$\sum_{J=0}^{m} w_J \phi_J(x) = 0 \quad \text{——— (II)}$$

If we assume $\phi_0(x) = 1$ for all x so that w_0 denotes the bias b. Eq^n (II) defines the decision surface computed in the feature space.

Lets define the vector,

$$\phi(x) = [\phi_0(x), \phi_1(x) \ldots \phi_m(x)]^T \quad \text{——— (III)}$$

where, $\phi_0(x) = 1$

Then the vector $\phi(x)$ will represent the image reduced in the feature space. In terms of image we may define the decision ~~suf~~ surface in the compat form.

$$W^T \phi(x) = 0 \quad \text{——} \quad (IV)$$

So, by linear separability of features we may write

$$W = \sum_{i=1}^{N} \alpha_i d_i \phi(x_i) \quad \text{——} \quad (V)$$

where, $\phi(x_i)$ = feature vector

Substituting eq^n (V) ~~and in~~ ~~we de~~ (IV) we define the design surface.

$$\sum_{i=1}^{N} \alpha_i d_i \phi^T(x_i)\, \phi(x_i) = 0 \quad \text{——} \quad (VI)$$

$\phi^T(x_i)\, \phi(x)$ represents ~~ϕ~~ the inner product of two vectors.

We may therefore introduce the inner product kernel denote by $k(x, x_i)$

$$\therefore\ k(x, x_i) = \phi^T(x)\, \phi(x_i)$$

$$= \sum_{J=0}^{m} \phi_J(x)\, \phi_J(x_i) \quad \text{—} \quad (VII) \quad \text{for } i = 1, 2 \ldots N$$

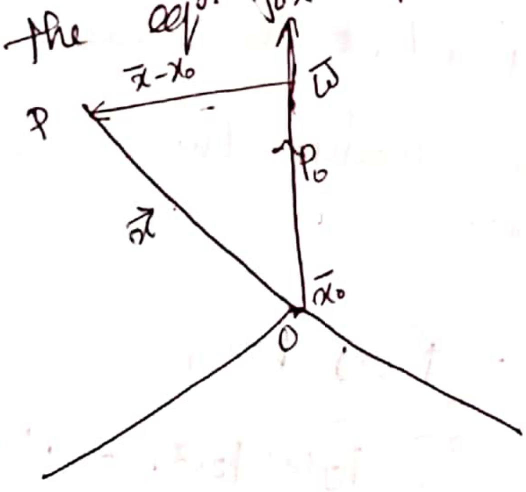
From this definition we immediately see that the inner-product kernel is a symmetric function of its arguments, as shown by

$$K(x, x_i) = K(x_i, x) \text{ for all } i.$$

Most importantly, we may use the inner-product ~~optional~~ optimal kernel $K(x, x_i)$ to construct the optimal hyperplane in the feature space without having to consider the feature space itself in explicit form. This is readily seen by eqn (vii) in (vi), whereby the optimal hyperplane is now defined by

$$\sum_{i=1}^{N} \alpha_i d_i K(x, x_i) = 0$$

Derive the eqn for Optimal hyper plane of SVM.



An eq for optimal hyper-plane is defined by a point P_0 and a perpendicular vector to the plane $\overline{\omega}$ at that point,

Define vectors,

$\vec{x_0} = \vec{OP_0}$

$\vec{x} = \vec{OP}$

where P is an arbitary point.

The condition is,

P will on the plane if the $\bar{x}-\bar{x}_0$ is perpendicular to $\bar{\omega}$

$\therefore \overline{\omega} \cdot (\bar{x}-\bar{x}_0) = 0$

or. $\overline{\omega}\bar{x} - \overline{\omega}\bar{x}_0 = 0$

Let define,

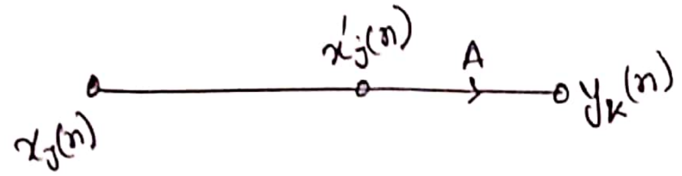
$b = -\overline{\omega} \cdot \bar{x}_0$

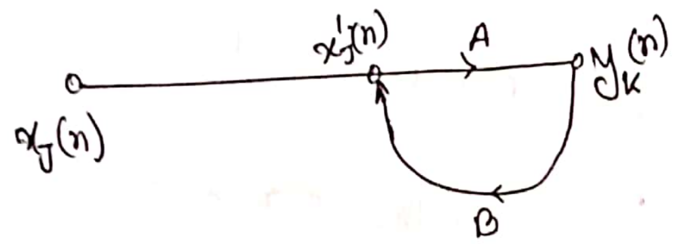
$\therefore \overline{\omega}\vec{x} + b = 0$

this is the optimal hyperplane of SVM's equation.

Derive the eq^n for a single loop-feedback system, or for a recurrent network derive the following eq^n.

$$y_k(n) = \sum_{l=0}^{\infty} w^{l+1} x_j(n-l)$$





$$y_k(n) = A[x'_j(n)] \qquad \because x'_j(n) = x_j(n) + B y_k(n)$$

$$y_k(n) = A[x_j(n) + B y_k(n)]$$

$$y_k(n) = A x_j(n) + AB[y_k(n)]$$

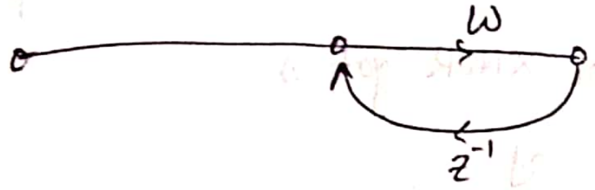
$$A x_j(n) = y_k(n) - AB[y_k(n)]$$

$$= y_k(n)(1-AB)$$

$$y_k(n) = \frac{A}{1-AB} x_j(n) \quad \text{——— (1)}$$

We refer to $A/(1-AB)$ as the closed-loop operator of the system, and to AB as the open-loop operator.

Scanned with CamScanner



$$\therefore \frac{A}{1-AB} = \frac{W}{1-W z^{-1}}$$

$$= w(1-wz^{-1})^{-1}$$

using the binomial expansion we get,

$$\frac{A}{1-AB} = W\sum_{l=0}^{\infty} w^{l} z^{-l} \quad \text{——— (2)}$$

$$\cancel{\therefore y_k(n) = W\sum_{l=0}^{\infty} w^{l} z^{-l}[x_j(n)]}$$

Substituting eq^n (2) in eq^n (1), we get,

$$y_k(n) = w\sum_{l=0}^{\infty} w^{l} \cdot z^{-l}[x_j(n)]$$

From the definition of z^{-1} we have

$$z^{-l}[x_j(n)] = x_j(n-l)$$

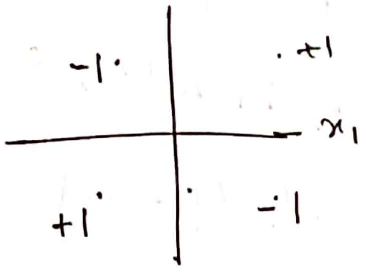
$$\therefore y_k(n) = \sum_{l=0}^{\infty} w^{l+1}\, x_j(n-l)$$

Design a SVM to solve the XNOR problem

Truth table for XNOR gate is

x_1	x_2	y
-1	-1	+1
-1	+1	-1
+1	-1	-1
+1	+1	+1

Plotting the data we get,



This is not linearly separable we can not divide these class by drawing line. we have to apply a kernel to solve this type of ~~pattern~~ problem.

let, the kernel be $\phi(x_1, x_1x_2)$

Now the truth table will be

$-x_1$	$-x_1x_2$	y
+1	+1	+1
+1	-1	-1
-1	-1	-1
-1	+1	+1

Plotted data will be

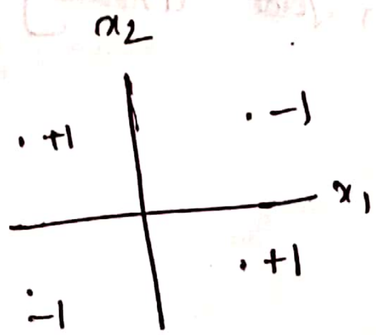
Now this will be separable linearly.

Q) Design a SVM to solve the XOR Problem.

The truth table for XOR

x_1	x_2	y
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	-1

Plotting the Data,



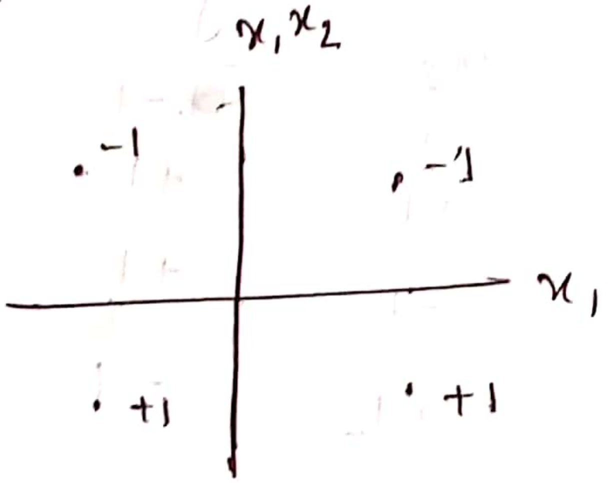
This is not linearly separable.

let the kernel will be $(x_1, x_1 x_2)$

Now, the truth table will be,

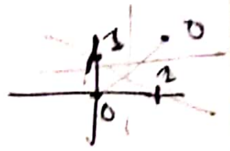
x_1	$x_1 x_2$	y
-1	+1	-1
-1	-1	+1
+1	-1	+1
+1	+1	-1

Plot the data



Now, this is linearly separable.

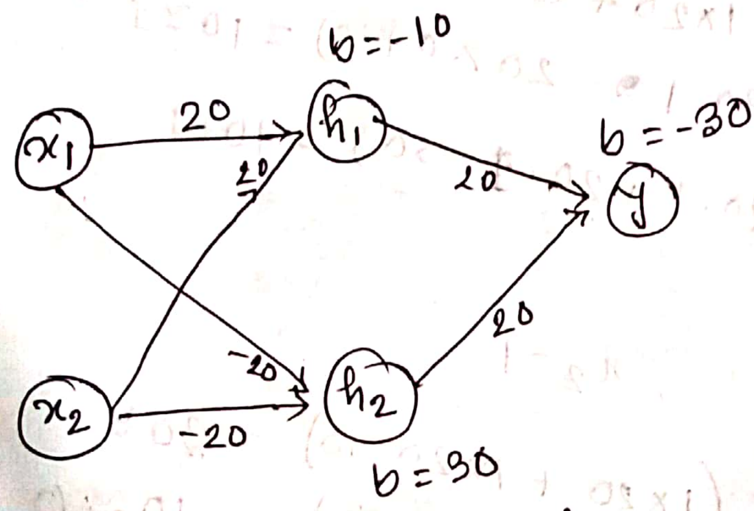
Scanned with CamScanner



Q) Solve the XOR problem using Multilayer perception.

The truth table for XOR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0



Signal flow graph

$$h_1 = \sigma(20x_1 + 20x_2 - 10)$$
$$h_2 = \sigma(-20x_1 - 20x_2 + 30)$$
$$y = \sigma(20h_1 + 20h_2 - 30)$$

for, $x_1 = 0$, $x_2 = 0$

$$h_1 = \sigma(0 + 0 - 10) \approx 0$$
$$h_2 = \sigma(-0 - 0 + 30) \approx 1$$
$$y = \sigma(0 + 1 - 30) \approx 0$$

minus value ≈ 0

Plus value ≈ 1

for, $x_1 = 0$ $x_2 = 1$

$h_1 = \sigma(0\times20 + 1\times20 - 10) = 10 \approx 1$

$h_2 = \sigma(-20.0 + 1\times20 + 30) = 10 \approx 1$

$y = \sigma(20.1 + 20.1 - 30) = 10 \approx 1$

for, $x_1 = 1$ $x_2 = 0$

$h_1 = \sigma(1\times20 + 0\times20 - 10) = 10 \approx 1$

$h_1 = \sigma(-20.1 - 20\times0 + 30) = 10 \approx 1$

$h_2 = \sigma(20.1 + 20.1 - 30) = 10 \approx 1$

$y = \sigma(20.1 + 20.1 - 30)$

for, $x_1 = 1$, $x_2 = 1$

$= \sigma(1\times20 + 1\times20 - 10) = 30 \approx 1$

$h_1 = \sigma(-20.1 - 20.1 + 30) = -10 \approx 0$

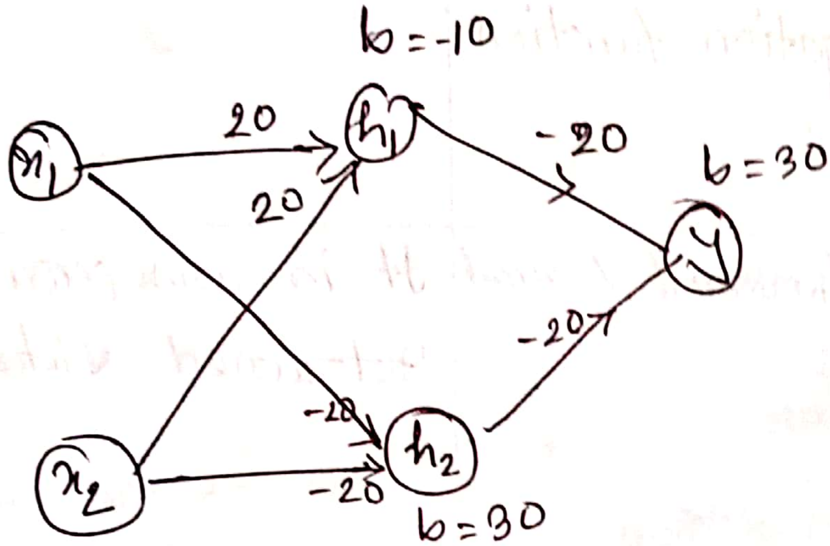
$h_2 = \sigma(20.1 + 20.0 - 30) = -10 \approx 0$

$y = \sigma(20.1 + 20.0 - 30)$

Solve the ~~XOR~~ XNOR problem with multilayer perception.

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

$$y = \begin{cases} 0 & \text{if } wx + b \leq 0 \\ 1 & \text{if } wx + b > 0 \end{cases}$$

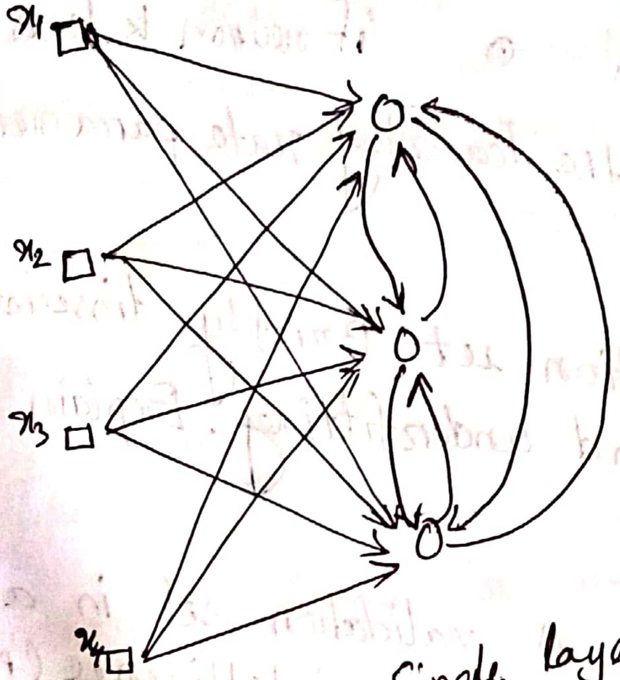


Neural Networks Vs Deep learning

Neural Network	Deep learning
Neural Networks class of machine learning algorithms where the artificial neuron forms the basic computational unit and networks are used to describe the inter connectivity among each other	It is a class of machine learning algorithms which uses non linear processing units multiple layers for feature transformation and extraction
Components of Neural Network are Neurons, connection and weight, propagation function, Learning rule.	Motherboard, processors, RAM, PSU.
It is feed forward Neural Networks	It is unsupervised Pretrained Networks.

Scanned with CamScanner

[Q] Briefly discuss competitive learning with proper eqn and diagrams.



Layer of source Node

Single layer of output Neurons

For a neuron k to be the winning neuron, its induced local field v_k for a specified input pattern x must be the largest among all the neurons in the network.

$$y_k = \begin{cases} 1 & \text{if } v_k > v_j \text{ for all } j,\ j \neq k \\ 0 & \text{otherwise.} \end{cases}$$

Let w_{kj} denote the synaptic weight connecting input Node j to neuron k.

$$\sum_j w_{kj} = 1 \quad \text{for all } k.$$

According to the standard competitive learning rule, the change Δw_{kj} applied to synaptic weight w_{kj} is defined by

Scanned with CamScanner

$$\Delta w_{kj} = \begin{cases} \eta (x_j - w_{kj}) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases}$$

where η is the learning rate parameter.

III Define validation set. Briefly discuss the terms overfitting and underfitting. Explain the ways of solving them.

Validation set : A validation set is a set of data used to train artificial intelligence (AI) with the goal of finding and optimizing the best model to solve a given problem. Validation sets are also known as dev sets.

Overfitting in Neural Networks :

Overfitting happens when the neural network is very good at learning its training set, but cannot generalize beyond the training set.

The symptoms of overfitting are :

- Low bias
- High variance

Underfitting in Neural Network.

Underfitting happens when the network is not able to generate accurate predictions of the training set - not to mention the validation set.

The symptoms of underfitting are.

- High bias
- High variance

To prevent overfitting or underfitting:

1. Cross validation
2. Train with more data
3. Data ~~aug~~ augmentation
4. Reduce complexity or Data Simplification
5. Ansembling
6. Early stopping
7. We need to regularization in case of Linear and SVM models.
8. In decision tree models, you can reduce the maximum depth.

4) Suppose you have a basket, which contain different types of unknown fruits that you have never seen before. How you want to classify the fruits using computer intelligence. Which type of learning is suitable for this and why? Give proper justification of your answer.

From this question it is clear that the fruits are unknown to me. For this type of problem we use unsupervised learning.

Let, the ~~fruits contain~~ basket contains Apple, cherry, Banana and Grape. I have to classify this with computer intelligence and this fruits are totally unknown to the machine. Now the unsupervised learning will ~~to~~ follow these steps.

(i) It will take a fruit and will arrange them by by considering physical character of the particular fruit.

Suppose we have considerd color.

(ii) Suppose we have arrange them on considering base

→ Then it will arrange them on considering base condition as color

→ The group will be something like that.

- Red color Group: apples & cherry
- Green color Group: Bananas & Grapes.

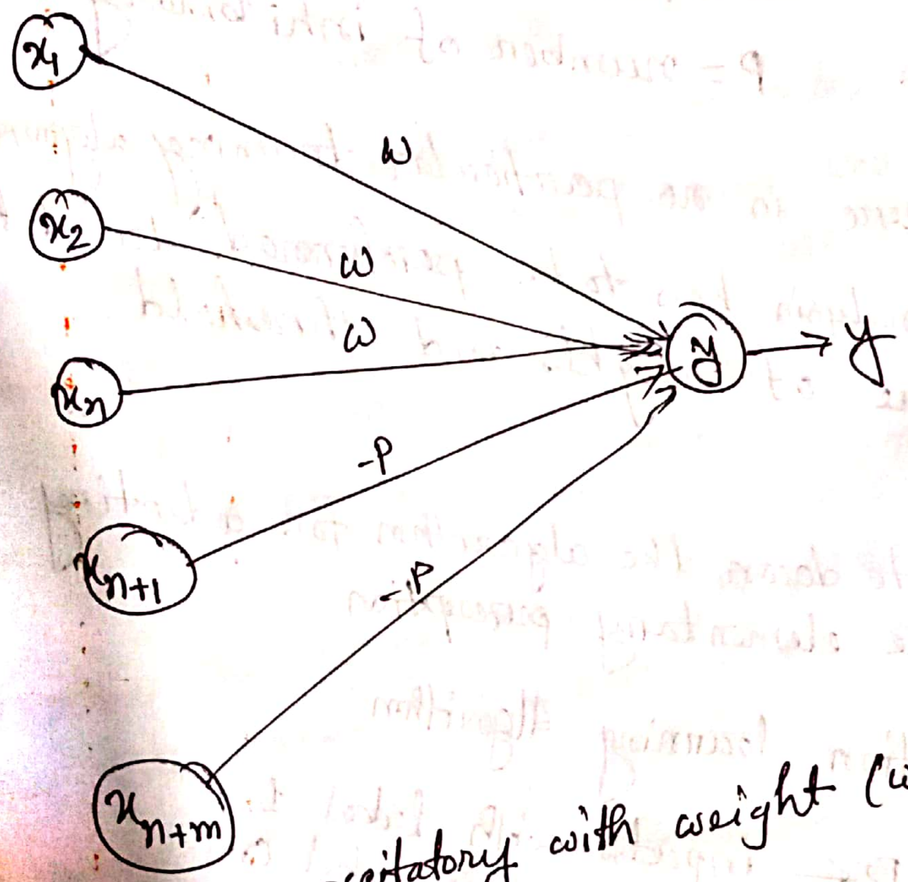
(iii) Now it will take another physical character like size.

→ Red color and big size: apple
→ Red color and small size: cherry
→ Green color and big size: banana
→ Green color and small size: Grapes.
→ Green color and

Briefly discuss on McCulloch and Pitts neural Network model.

1. There is a fixed treshold for each neuron and if the net input to the neuron is greater than threshold then the neuron fires

2. Most widely used in logical functions



Here both excitatory with weight $(\omega > 0)$ enhibitory with weight $-P$ $(P<0)$ connections.

3. Sciene firing of the ~~out~~ output neuron is based on threshold activation function is defined by

$$f(y_n) = \begin{cases} 1, & y_n \geq 0 \\ 0, & y_n < 0 \end{cases}$$

4. If inhibitory weights are used then threshold with activation function would satisfy following condition

$$\theta > nw - p$$

n = number of input vector
w = number of exibitory weights
p = number of inhibitory weights

5. There is no particular training algorithm
6. Analysis has to be performed to determine value of weights and threshold.

(4) Write down the algorithm for adapting weight vector of the elementary perception

Perception learning Algorithm:

$P \leftarrow$ inputs with label 1
$N \leftarrow$ inputs with label 0

Initialize w randomly

while convergence do
  Pick random $x \in P \cup N$;
  if $x \in P$ and $w, x < 0$ then
    $w = w + x$;

Scanned with CamScanner

end

if $x \in N$ and $w \cdot x \geq 0$ then

$w = w - x$

end

end

end

□ Explain the dilemma of learning rate.

A neural network learns or approximates a function to best map inputs to outputs from examples in the training dataset. When the learning rate is too large, gradient descent can inadvertently increase rather than decrease the training errors.

কোনো একটা

□ Explain the reason single layer perception cannot solve XOR problem with proper illustration.

A single layer perception is a function with degree 1.

Let the equation where input is a vector of n dimension be

table

$$y = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

This is a perception with unthreshold output as mentioned in Tom Mitchell text. For classification we need to set a threshold T. i.e

$y = T$

$\therefore \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n = T$

Here $x_0, x_1, \ldots x_n$ are variables. This will be a plane in $n+1$ dimensions.

But as mentioned in other answers, XOR problem cannot be solved by liner plane. Hence XOR problem cannot be solved by single layer perception.

Q. Discuss the advantages of multilayer perception over single layer perception.

Multilayer perception with one hidden layer are capable of approximating any continous function. Multilayer perception. are often applied to supervised learning problems, they train on a set of input - output pairs and learn to model the correlation between those inputs and outputs. MLP are useful in research for their ability to solve problems stochastically, which often allows approximate solution. This is the main advantages MLP over SLP.

Explain the basic rules for the construction signal flow graph.

The flow of signals in the various parts of the graph is directed by three basic rules:

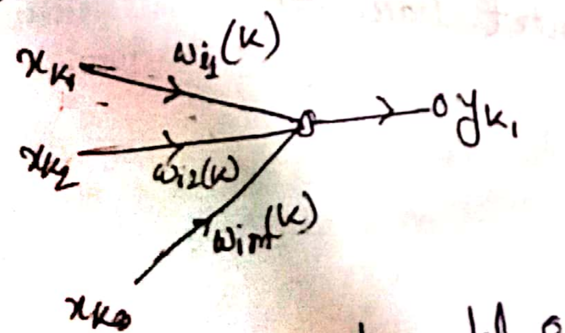
Rule 1: A signal ~~follows~~ flows along a link only in the direction defined by the arrows of the links. There are two different types of links

(i) Synaptic links, where behaviour governed by a Linear input output relation

(ii) Activation links - behaviour governed in general by a non-linear input-output relation

Rule 2: A node signal equals the algebric sum of all signals entering, the pertinent node via the incoming link.

Rule 3: The signal at a node is transmitted to each outgoing link originating from that node with the transmission being entirely independent of the transfer function of the outgoing links.



Signal-flow graph model of a linear neuron labeled i

$y = T$

$\therefore \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n = T$

Here $x_0, x_1, \dots x_n$ are variables. This will be a plane in $n+1$ dimensions.

But as mentioned in other answers, XOR problem cannot be solved by liner plane. Hence XOR problem cannot be solved by single layer perception.

Q4) Discuss the advantages of multilayer perception over single layer perception.

Multilayer perception with one hidden layer are capable of approximating any continous function. Multilayer perception are often applied to supervised learning problems, they train on a set of input-output pairs and learn to model the correlation between those inputs and outputs. MLP are useful in research for their ability to solve problems stochastically, which often allows approximate solution. This is the main advantages MLP over SLP.

Explain the basic rules for the construction signal flow graph.

The flow of signals in the various parts of the graph is directed by three basic rules:

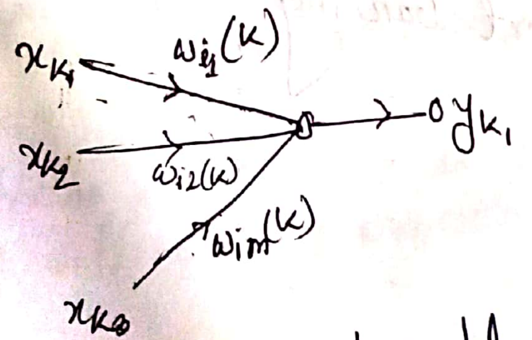
Rule 1: A signal ~~follows~~ flows along a link only in the direction defined by the arrows of the links. There are two different types of links.

(i) Synaptic links, whose behaviour governed by a linear input output relation

(ii) Activation links - behaviour governed in general by a non-linear input-output relation

Rule 2: A node signal equals the algebric sum of all signals entering, the pertinent node via the incoming link.

Rule 3: The signal at a node is transmitted to each outgoing link originating from that node with the transmission being entirely independent of the transfer function of the outgoing links.



Signal-flow graph model of a linear neuron labeled i

Explain the characteristics of directed graph of a neuron.

A directed graph is said to be partially complete. Its characters as follow :

1. Source nodes supply input signals to the graph.
2. Each neuron is represented by a single node called a computation node.
3. The communication links interconnecting the source, and computation nodes of the graph carry no weight, they merely provide direction of signal flow of graph.

Define learning in the contex of neural network.

Neural Network learning rule process is a method mathematically logic or algorithm which is prove the networks performance and training times, usually this rules is applied to over the network.

there are the model of machine learning

(i) Unsupervised Learning
(ii) Supervised Learning
(iii) Reinforcement learning

Why bias input is necessary for training in neural networks?
Bias is just like an additional parameter in the Neural Network which is used to adjust the output along with weighted sum of the inputs to the neuron. Therefor Bias is a constant which helps the model in a way that it can fit best for the given data.

$$\text{output} = \sum \text{weight} * \text{input} + \text{bias}$$

Example : $y = mx + \underbrace{c}_{\text{bias}}$

What do you understand by the term knowledge? Explain the rules of knowledge representation.
knowledge refers to stored information or models used by a person or machine to interpret, predict and appropiately respond to the world.

Rules of knowledge Representation;

Rule 01: Similar inputs from similar classes should usually produce simmilar representations inside the network and should therefore be classified as belonging to the same category.

Rule 02: Items catagorized as seperate classes should be given widely different reprentations in the network.

Rule 03: If a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network.

Rule 04: Prior information and invariances should be built into the design of a neural network, there by ~~simplifing~~ simplifying the network design by not having to learn them.

Q. Briefly explain the error correction learning with proper diagram, graph and equations.

In error correction learning process is done using the error b which is generated by comparing output with desired response or target output. Neuron k is driven by a signal vector $x(n)$ which is produced by one or more layers of hidden neurons, which are themselves driven by an input vector.

By definition,

$$e_k(n) = d_k(n) - y_k(n)$$

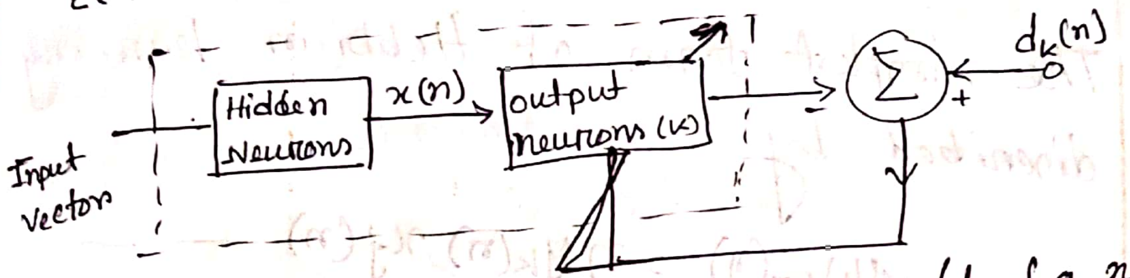
$d_k(n) \rightarrow$ desired output of neuron k.

$y_k(n) \rightarrow$ actual output of neuron k.

Error is removed by continous adjustment in a step-by-step manner for this a cost function is used which is ~~target~~ to minimize.

$$\mathcal{E}(n) = \frac{1}{2} e_k^2(n)$$

$\mathcal{E}(n)$ is instantaneous ~~val~~ value of the error energy.



The adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question.

Briefly describe the two part rule of Hebbian learning.

~~Rule 01: If two neurons on either side of a synapse, then the strength of that synapse is selectively increased.~~

Oldest Rule

unsupervised

Rule 01: If two neurons on either side of a synapse are activated simultaneously (synchronously), then the strength of that synapse is selectively increased

Rule 02: If two neurons on either side of a synapse are activated asynchonusly, then that synapse is selectively weakened or eliminated.

ynchronusly
+1 +1 = +1
synchronusly
-1 = -1
-1 = -1
+1 = -1

4. Briefly describe the Hebbs hypothesis and its limitation. Explain how the covariance hypothesis can overcome that limitation

The simplest form of Hebbian learning is described by

$$\Delta W_{kj}(n) = \eta\, y_k(n)\, x_j(n) \quad \text{———— (1)}$$

η = learning rate

$x_j(n)$ = Input vector

$y_k(n)$ = Output

The limitation of Hebbian hypothesis is that, in eqn (1) the repeated application of the input signal x_j leads to a increase in y_k and therefore exponential growth that finally derives the synaptic connection into saturation. At that point no information will be stored in the synapes and selectivity is lost.

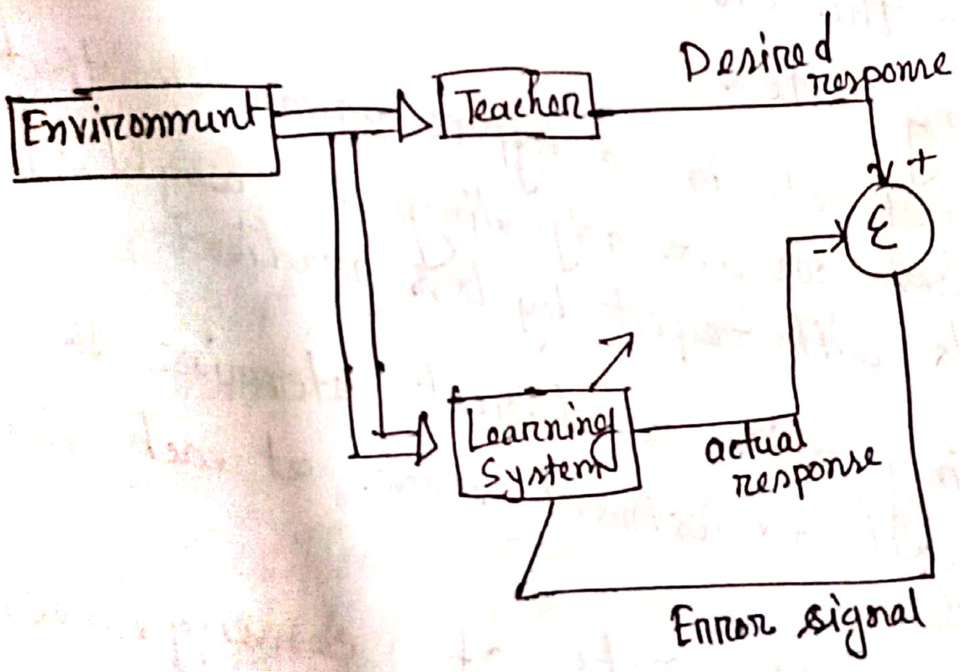
Covariance hypothesis: One way the overcoming the limitation of hebb's hypothesis introduced in Sejnowski.

Scanned with CamScanner

In this ~~hyt~~ hypothesis the presynaptic and post synaptic hypothesis signals are replaced by the departure of pre synaptic and post synaptic signal, from the respective average values over a certain line interval.

$$\Delta w_{kj} = \eta \, (x_j - \bar{x}) \, (y_k - \bar{y})$$

Depict the block diagram of learning with a teacher.



Incremental Vs Batch learning

Batch	Incremental
① The machine learning model must be trained with available data, then it can be launch	Train the machine one by one or by small batch
The model is slow and costly	The model is very fast and cheap
Not for the frequent data update	It is suitable for regular incoming data

Explain the dilemma of deciding the value of learning rate.

Learning rate is a hyper parameter that control how much we are adjusting the weight of ~~our~~ our network with respect by loss gradient.

Learning rate co-efficient determine the size of the weight adjustment made at each iteration.

Influence the rate of convergence.

Five basic learning Rules

(I) Error correction learning
(II) Memory based learning
(III) Hebbian learning
(IV) Competitive learning
(V) Boltzman learning

Scanned with CamScanner

Hetero - association vs auto-association

Hetero	Auto
Input-output space of the network are not same $y_K \neq x_K$	Input-output (data) space of the network have same dimentionality $y_K = x_K$
Involved in supervised learning	Involved in unsupervised learning

(ii) What is the significance of a transfer function? List the different types of transfer functions? If the input to a single input neuron is 2.0, its weights is 1.3 and its bias is 3.0, what is the output of the neuron if it has the following transfer function?

(i) Handlimit

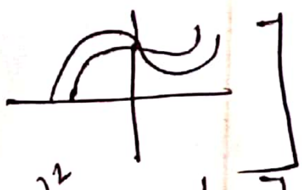
(ii) Linear.

⇒ List of different types of transfer function

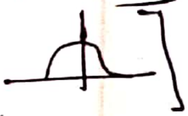
(i) Threshold function $\left[F(x) = \begin{cases} 1 & \text{is } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \right]$

(ii) Piecewise-liner function

(iii) Sigmoid function $\left[F(x) = \frac{1}{1+e^{-x}} \right.$



(iv) Gaussian function $\left[F(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \right.$



Scanned with CamScanner

Given,

$\omega = 1.3$

$b = 3.0$

$x = 2.0$

output $y = \omega x + b$

$= 1.3 \times 2.0 + 3.0$

$= 5.6$

(i) for Hand limit

$$y = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Here $y = 5.6 \geq 0$

Handlimit$(y) = 1$

(ii) For linear,

$y = \omega x + b$

$= 1.3 \times 2 + 3$

$= 5.6$

Scanned with CamScanner

⊞ Properties of backpropagation

1) Feed forward neural network for supervised learning
(ii) Generalization exists
(iii) Efficiency compute the gradient
(iv) Updating weight to minimize loss.

⊞ Limitations of backpropagation:

1. Does not require normalization for input vector.
2. Requires derivatives of activation function.
3. Not guranted to find global minimum of error function.
4. It has double to crossing plate.
5. Sensitive to noise
6. We need to use the matrix based approach instead of mini-batch.

⊞ Define fuzzy if then rule

A simple fuzz IF THEN rule is defined by

If x is A then y is B

where A and B are linguistic variable defined by fuzzy sets on the ranges x and y respectively. The if part 'a is A' is called the antecedent and the THEN part is called consequent.

Modus Ponens vs Modus tollens

Ponens	Tollens
Modus ponens is the name of any argument to the following form, If A then B A is true: Then B is also true	Modus tollens is the ~~...~~ name of any argument in the following form, If A then B B is not the case Therefore A is not the case either
If you buy a car then you will need a driver You bought a car So you will need a driver $\therefore MP \rightarrow \frac{P \rightarrow Q,\ P}{Q}$	If you get kacchi, then you won't need burger You need burger So you don't need Kacchi $\therefore MT \rightarrow \frac{P \rightarrow Q, \neg Q}{\neg P}$

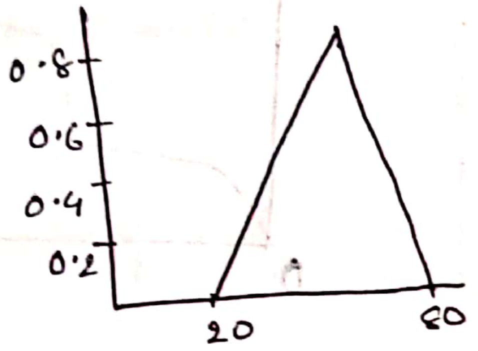
Describe the types of membership function in Fuzzy.

(i) Triangular
(ii) Trapezoidal
(iii) Gaussian
(iv) Sigmoidal
(v) L-R MF

Scanned with CamScanner

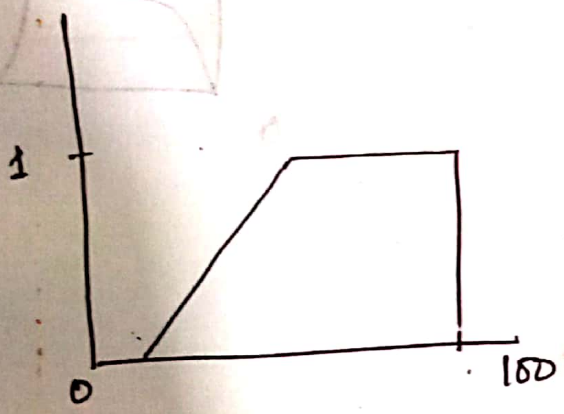
Triangular :

$$\text{triangle}(x;a,b,c) = \begin{cases} 0 , & x \leqslant a \\ \frac{x-a}{b-a} , & a \leqslant x \leqslant b \\ \frac{c-x}{c-b} & b \leqslant x \leqslant c \\ 0 , & c \leqslant b \end{cases}$$



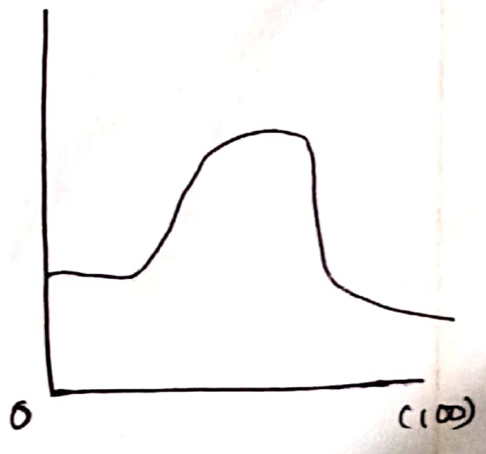
Trapezoidal :

$$\text{trapezoidal}(x;a,b,c,d) = \begin{cases} 0 ; & x \leqslant a \\ \frac{x-a}{x-b} ; & a \leqslant x \leqslant b \\ 1 ; & b \leqslant x \leqslant c \\ \frac{d-x}{d-c} ; & c \leqslant x \leqslant d \\ 0 ; & d \leqslant x \end{cases}$$



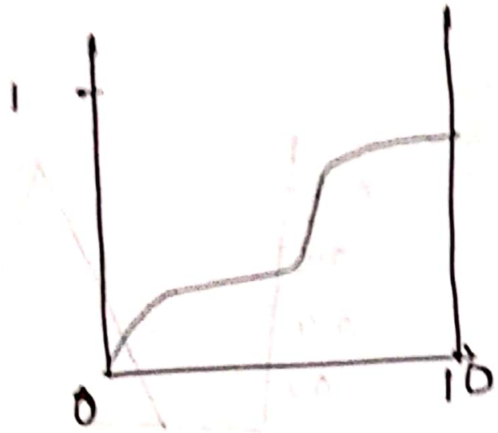
Gaussion :

$$\text{gaussian}(x:c,\sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$$



Sigmoidal:

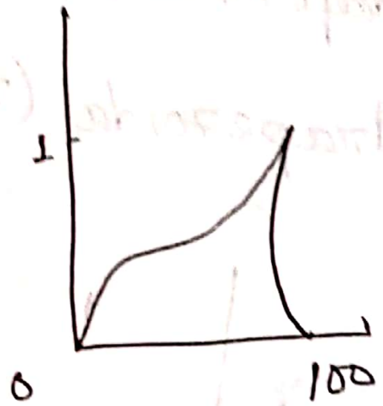
$$\text{sigma}(x; a, c) = \frac{1}{1 + \exp[-a(x-c)]}$$



L-R:

$$F_L(x) = \max\left(0, \sqrt{1-x^2}\right)$$

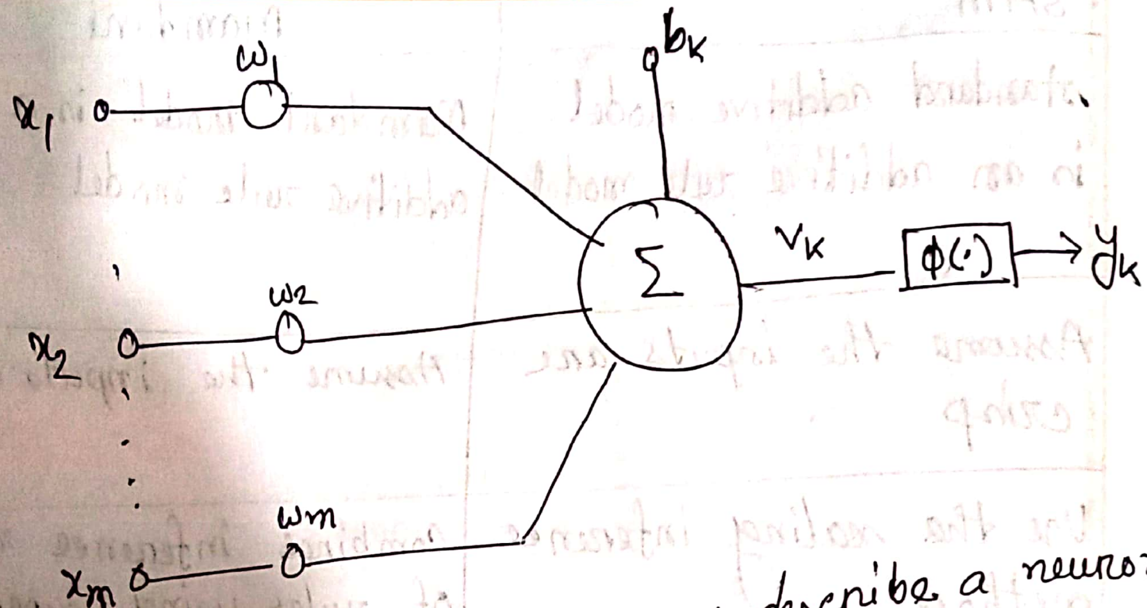
$$F_R(x) = e^{-|x|^3}$$



SAM vs Mamdani

SAM	Mamdani
standard additive model is an additive rule model	Mamdani model is a non additive rule model
Assume the inputs are crisp	Assume the inputs are fuzzy
Use the scaling inference method	Combines inference result of rules using super ~~impotions~~ imposition
Use addition to combine the conclusion of rules	Use superimposition to combine the conclusion of rules

Q) Briefly describe the model of neuron with appropriate figure and eq^n



In mathematical terms, we may describe a neuron k by writing the following pair of eq^n

$$u_K = \sum_{J=1}^{m} w_{KJ} x_J$$

$$y_K = \phi(u_K + b_K)$$

The bias b_K is an external parameter of artificial neuron k.

$$v_K = \sum_{J=0}^{m} w_{KJ} x_J$$

$$y_K = \phi(v_K)$$

Neural network viewed of directed graph:

Rule 1:

(a) Synaptic link

$$x_j \xrightarrow{\;w_{kj}\;} y_k = w_{kj}x_j$$

(b) Activation link

$$x_j \xrightarrow{\;\phi(\cdot)\;} y_k = \phi(x_j)$$

Rule 2:

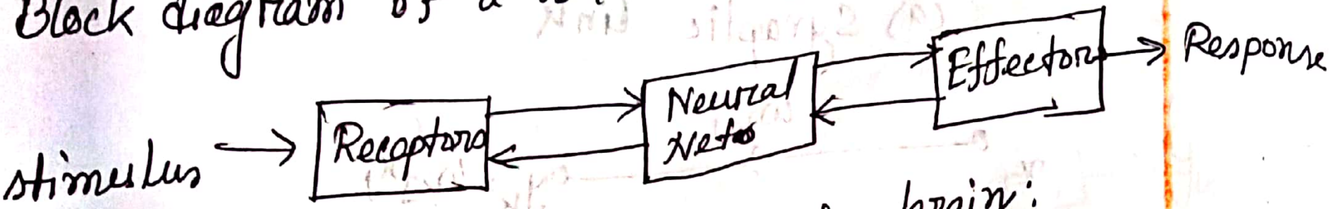
$$\begin{tikzcd}
y_i \arrow[dr] & \\
& y_k = y_i + y_2 \\
y_j \arrow[ur] &
\end{tikzcd}$$

Rule 3:

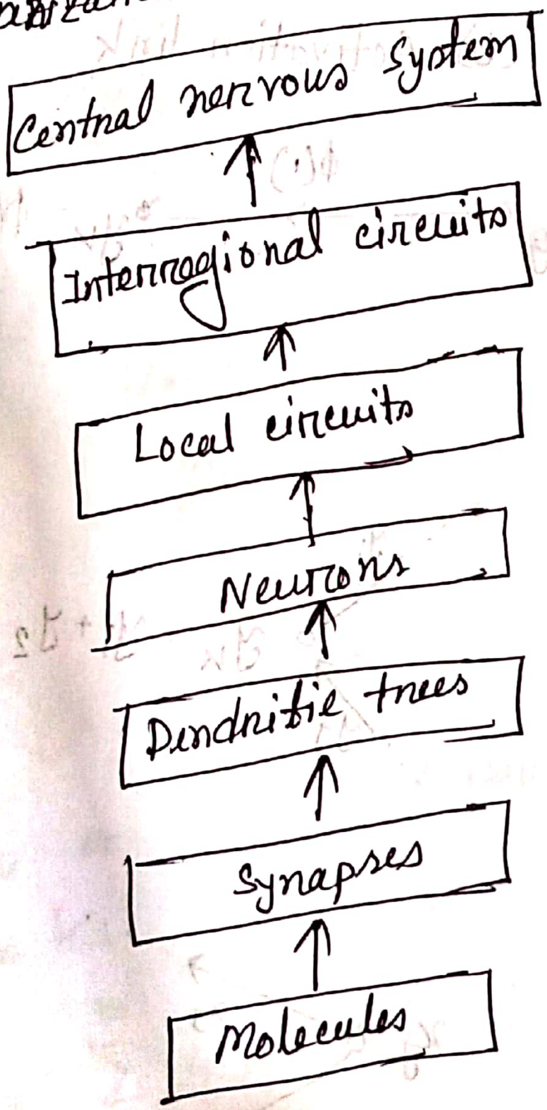
$$\begin{tikzcd}
& x_j \\
x_j \arrow[ur] \arrow[dr] & \\
& x_j
\end{tikzcd}$$

[I] What is neural network and how does it resemble with brain

Block diagram of a nervous system :-



Structural organization of levels in brain:



[3] Define CNN

A convolutional neural network is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

CNN is powerful image processing artificial intellegence that use deep learning to perform both generative and descriptive tasks using machine vision that incluide image and video recognition.

[4] What are the basic steps of CNN

Convolution + ReLU
↓
Pooling
↓
Flattening
↓
Full connection

[4] Write a short-note about Hessian matrix.

The Hessian matrix is a square matrix of second-order partial derivatives of scalar-valued function, or scalar field. It describes the local curvature of a function of many variable. Hence originally used the term "Function determination"

$$(H_f)_{ij} = \frac{\delta^2 f}{\delta x_i \, \delta x_j}$$

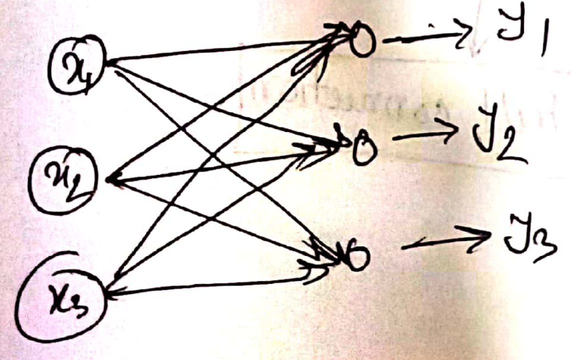
[4] Briefly discuss the model of PCA

(i) Subspace Model

(ii) APEX

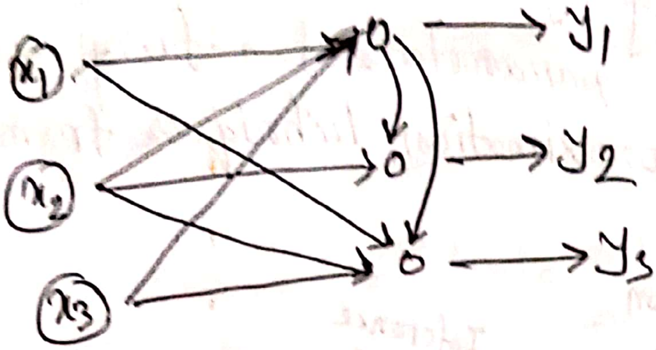
(iii) multilayer auto-associative

Subspace Model: A multicomponent ~~extension~~ extract of ~~oja's rule~~ the eigen vector.

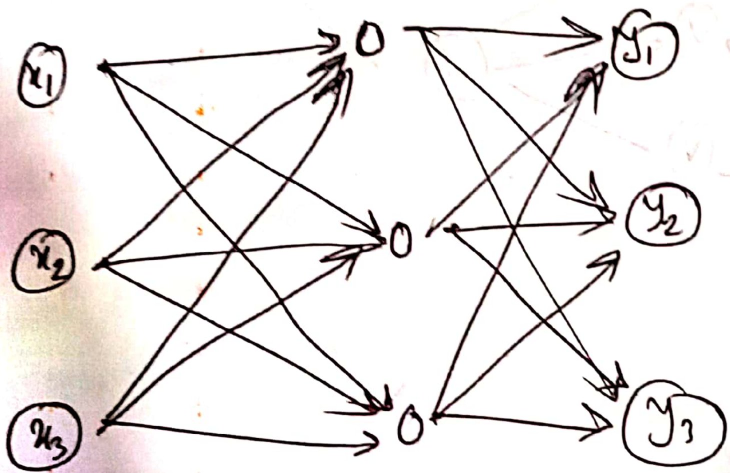


$$\Delta W_K = B_K \left(y_K x_K^T - y_K y_K^T W_K \right)$$

APEX model:



multilayer Networks



train using auto associative output

$$e = x - y_i$$

Q. What is neuro fuzzy system? Explain with example

A neuro-fuzzy system is a learning machine that finds the parameters of a fuzzy system by exploiting approximation techniques from neural networks.