Unit-II

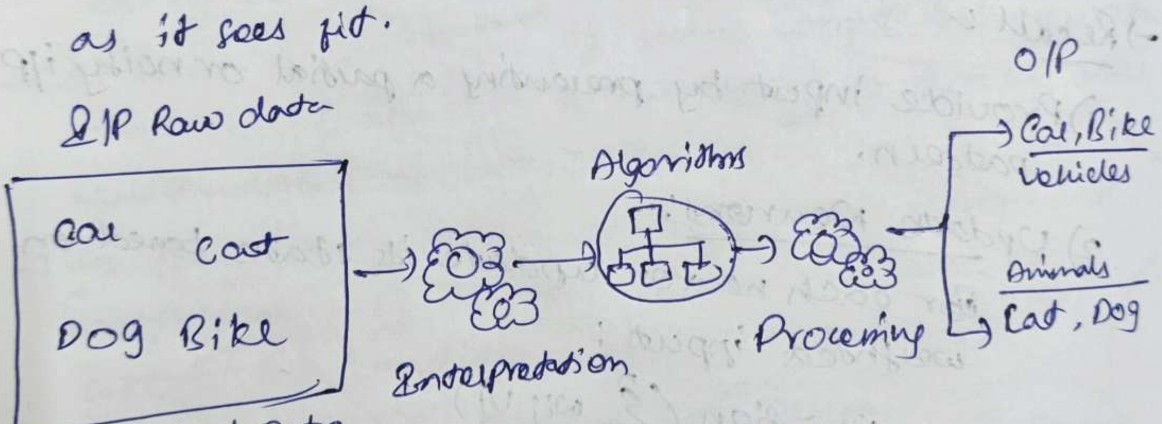
*** Unsupervised Learning Network:-**

→ It is a subfield of ML in which a model is trained on unlabeled data.

→ The main idea for the model is to detect hidden insights & patterns in a given dataset without having to first identify or classify what to look for.

→ Here, we are feeding the model data without specifying what o/p values we want it to produce.

→ This gives freedom to manipulate the data set as it sees fit.



→ Working:-

i) Collection of Necessary Data:-

- Data collected by the unsupervised learning networks are raw & unstructured.
- This data is cheaper to collect as they require no specific labeling or processing for the data to be used.

2) Training the Model:-

- ~~Some~~ The algorithms in unsupervised learning take in unlabeled data & try to make sense of it.
- This can be done by clustering all data points into given clusters or by discovering hidden patterns & trends.

3) Model Evaluation:-

- To ensure the model is giving peak accurate results, we must deliberately test the model's O/P on diff & various i/p variables.
- We can then move on to tuning the model's parameters in order to improve its final result.

→ Types:-

1) Clustering:-

It is the task of classifying unlabeled data into multiple groups (or clusters) based on their similarities & differences. Two of the most well known clustering alg. are K-Mean clustering & heirarchical clustering.

2) Association:-

It is used for discovering relations b/w variables. Association learning is commonly used, basket market analysis, in which the given alg. tries to relate or find a given relationship b/w two products. Ex:- 90% customers who buys A also buys B.

→ Use/App:-

1) Text Sentiment Analysis:-

Semantic Analysis and feeling behind the sentence

2) Speech recognition.

3) AI Chatbots

⊛ Unsupervised learning in Computer Vision:-

→ Computer is a subfield of ML in which computers are capable of extracting useful info from visual data representations such as images & video recording.

→ The goal of computer vision field is to allow computer to view ~~the~~ the world in a matter similar to that of a human's visual eyesight.

→ In unsupervised learning, the model is trained on unlabelled data & images and it is upto the model to detect all anomalies in the image on its own.

→ Use/App:-

1) Cancer Diagnosis:-

Computers can recognize odd anomalies in particular medical scans using unsupervised learning, & differentiating healthy from cancer-positive i/ps (tumors).

2) X-ray Diagnosis:-

Similar to cancer diagnosis, this model is fed a multitude of X-ray scans and any irregularities are detected by the model.

Fixed Weight Competitive Nets :-

- These are a type of unsupervised learning neural networks where neurons compete to respond to a given i/p.
- Only one neuron wins this competition and is activated, while the others are inhibited.
- The key feature is that the weights are usually fixed, and learning occurs by selecting the most suitable o/p neuron.
- It follows Winner - Takes - All (WTA) Principle, where only one neuron is activated for given i/p while others remain inactive.
- No explicit teacher or target output is required for training.
- Working
  1) A data sample is presented to the n/w.
  2) Each neuron calculates its activation value based on dot product of i/p & fixed weights.
  3) The neuron with the highest activation wins.
  4) Only the wining neuron fires, representing the i/p class.
- App:- Data Clustering, Pattern Recognition, Robotics,
- Adv:-
  Simple & efficient, effective for clustering, requires less computational cost.
- Disadv:-
  Not adaptable to changes in data, limited flexibility in learning complex pattern.

MaxNet :-

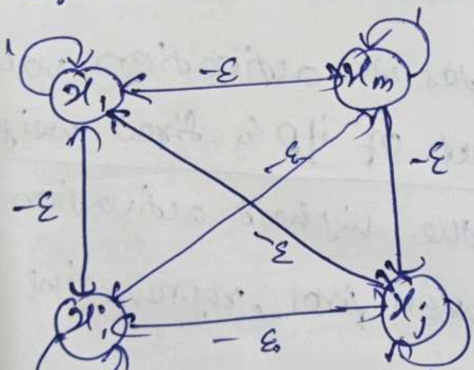
→ It is a type of competitive neural network used to find the neuron with the maximum activation value from a set of i/p neurons.

→ It operates based on WTA principle, where only one neuron remains active, representing the winner.

→ In maxnet, a recurrent network structure is used where neurons are connected to each other.

→ The network iteratively reduces the activations of weaker neurons until only the neuron with the highest i/p remains active.

→ Fixed symmetrical weights are present over the weighted interconnections in the Maxnet.



→ Here, the neurons update their activations based on feedback from other neurons.

→ Applic :-
Pattern Recognition, Data Clustering, Decision Making Systems.

→ Algorithm :-

1) Initialize Neurons :-

- Assign the i/p values to each neuron in the n/w
- Let $x_1, x_2, \ldots x_N$ be initial activation for N neurons.

2) Set Parameters :-

- Choose a small inhibition factor ε (b/w 0 & 1).
- Define a threshold θ for convergence.

3) Iterative Update :-

Repeat following steps until convergence:

- For each neuron x_i, update its activation using:

$$x_i(t+1) = x_i(t) \times \left(1 - \epsilon \times \sum_{j \neq i} x_j(t)\right)$$

- Ensure that any negative activations are set to '0'.

$$x_i(t+1) = \max(0, x_i(t+1))$$

4) Convergence check :-

- Stop if only one neuron has a non-zero activation, or if the change in activations is less than the threshold θ.

5) O/P :-

- The neuron with the highest remaining activation is the winner.

→ Adv :-

- Efficiently finds largest value among i/ps.
- Simple structure & straightforward update rule.

→ Disadv :-

- Sensitive to choice of inhibition factor (ε).
- Performance may degrade with noise in i/ps.
- Slower convergence for large networks.

→ Ex :- Automate Job Candidate Selection in Recruitment.

(4) Hamming Network:-

→ It is a type of neural n/w used for pattern recognition tasks.

→ It is based on the ~~distance~~ concept of Hamming distance, which measures the no. of bit positions where two binary strings differ.

→ This network is designed to classify i/p patterns by finding the closest matching pattern from a set of stored patterns.

→ Operates by computing Hamming distance b/w i/p & stored pattern.

→ Consists of 2 layers:-

1) Correlation Layer:-
   computes similarities b/w i/p & stored pattern.

2) Competition layer:-
   Identifies best match by inhibiting weaker responses.

→ Architecture & Working:-

1) Input Layer:-
   Recieves binary input vector.

2) Correlation Layer:-
   Each neuron computes the Hamming distance b/w i/p vector & a stored reference pattern.

3) Competition Layer:-
   Selects the neuron with smallest Hamming distance by suppressing other neurons.

→ Alg:-

1) Input Pattern Preparation:-

- Provide the binary i/p pattern
  $X = [x_1, x_2, \dots x_n]$
- Ensure the pattern length matches the stored reference patterns.

2) Compute Correlation Layer O/Ps:-

- For each pattern reference P_i in the stored set, calculate Hamming distance b/w the i/p pattern & each reference pattern

$$D(X, P_i) = \sum_{j=1}^{n} (x_j \oplus P_{ij})$$

$\oplus \rightarrow$ XOR operation

3) Normalize Distance:-

- Normalize distance value by subtracting the distance from max possible length of binary vector n:

$$H_i = n - D(X, P_i)$$

$H_i \rightarrow$ similarity score for each pattern.

4) Competition Layer Application:-

- Identify pattern P_k with max similarity score.

$$P_k = \text{argmax}(H_i)$$

5) Output Decision:-

- Return reference pattern P_k as recognized pattern
- If no clear winner exists, report a failure or ambiguous result.

→ Adv:-

Fast & efficient for binary pattern recognition, has simple architecture

→ Disadv:-

Limited to binary data, less suitable for complex, non-binary patterns.

Kohonen Self-Organising Feature Maps :-

→ These are a type of unsupervised ~~n~~ neural networks introduced by Teuvo Kohonen.

→ They are used for data clustering & dimensionality reduction by mapping high-dimensional i/p data onto a lower dimensional (usually 2D) grid while preserving topological relations.

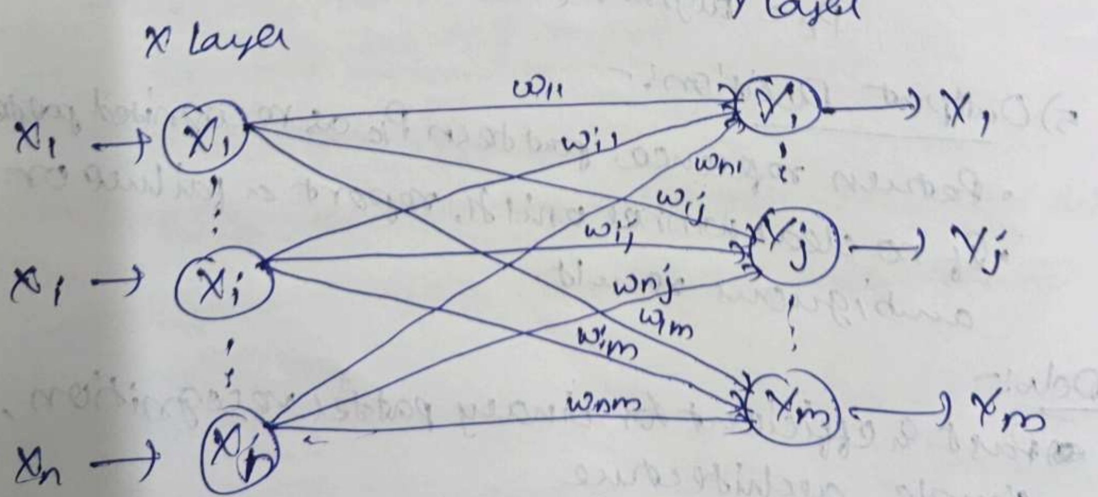
→ Topological preservation:- similar i/p patterns are mapped close to each other on the feature map.

→ It is trained using competitive learning where it suggests that some criteria select a winning processing element.

→ It has only two layers:

I/p layer:- each i/p node represents one feature in the dataset

O/p layer (Map layer):- consists of neurons arranged in a 2D grid.



→ Alg:-

1) Initialize Self-Organizing Feature Map (SOM):-

- Assign small random weights to each neuron in the 2D grid.

2) Present I/P data:-

- Normalize the i/p data to ensure all features have equal importance.

3) Find the Best Matching Unit (BMU):-

- Compute the Euclidean distance b/w i/p vector and weight vector of each neuron.
- Select the neuron with the smallest distance as BMU.

$BMU = \arg\min_j \|x - w_j\|$

$x \rightarrow$ i/p vector, $w_j \rightarrow$ weights of neuron 'j'.

4) Update weights:-

- Update weights of BMU & its neighboring neurons to make them closer to i/p vector.

For:- $w_j(t+1) = w_j(t) + \alpha \times h(j, BMU, t) \times (x - w_j(t))$

where, $w_j(t) \rightarrow$ weights of j at t

$\alpha \rightarrow$ learning rate

$h(j, BMU, t) \rightarrow$ neighborhood function.

5) Reduce Learning rate & neighborhood size:-

- Gradually decrease learning rate & neighborhood radius over time.

6) Repeat:-

steps 2 to 5 for a predefined no. of iterations or until convergence.

→ Applications:-

Pattern Recognition, Data visualization, Image compression.

→ Ex:- Shopping patterns of people young, middle aged & elderly.

Learning Vector Quantization:- (LVQ)

→ LVQ is a type of supervised ML alg. used ~~type of supervised ML~~ for classification tasks.

→ It belongs to the family of competitive learning methods & is the extension of Kohonen's SOM.

→ LVQ works by learning prototypes (codebook vectors) that represent different classes in the data.

→ New data points are then assigned the class of the prototype that is nearest to them and in order for "nearest" to make sense, a distance measure has to be defined.

→ LVQ also applies Winner-Takes-All Hebbian learning-based approach.

→ Alg:-

1) Initialization:-
   - Select the no. of prototype vectors for each class (m prototypes per class).
   - Initialize these prototype vector (w_i) randomly or by sampling from i/p data.
   - Set the learning rate α (initially a small +ve value).

2) Repeat for each training epoch:-
   - For each i/p vector x_i in the training dataset:

a) Find Best Matching Unit (BMU):

- compute Euclidean distance b/w i/p vector & each prototype :

$$d_j = ||x_i - w_j||$$

- select prototype w_k with smallest distance to x_i

$$w_k = \text{argmin}_j ||x_i - w_j||$$

b) Update Prototype vector w_k

- If the class of i/p vector matches the class of BMU:

$$w_k = w_k + \alpha (x_i - w_k)$$

- If the class of i/p vector doesn't match

$$w_k = w_k - \alpha (x_i - w_k)$$

c) Reduce Learning Rate :-

- Gradually decrease learning rate over time :

$$\alpha = \alpha_0 (1 - t/T)$$

where, $t \rightarrow$ current iteration

$T \rightarrow$ max no. of iterations

3) Stopping Criteria :-

- stop training when:

a) max no. of epochs reached.

b) the change in prototype vector become negligible.

→ Adv:-

Simple & easy to implement, can handle multi-class classification problems,

→ Disadv:-

sensitive to initialization of prototypes, may get stuck in local minima.

→ Ex: Finding the fruit apple, banana, orange
apple - small, red, smooth.
banana - long, yellow, smooth
orange - medium, orange, rough.

(2) Counter Propagation Network:- (CPN)

→ A CPN is a type of ANN that combines the features of two networks:
a kohonen SOM and Grossberg Layer (also known as outstar)

→ It is a multilayer network and is generally constructed from an instar-outstar model.

→ It is used for tasks like pattern classification, data compression & function approximation.

→ A CPN has 3 layers:-

1) I/P Layer:-
Recieves i/p data & passes it to the next layer

2) kohonen layer:-
A competitive layer that clusters similar i/p patterns.
- performs unsupervised learning.
- neurons in this layer compete to respond to the i/p and only winning neuron gets activated (WTA Rule)

3) Grossberg Layer:- (O/P Layer):
Maps o/p from kohonen to desired O/P.
- performs supervised learning
- associates patterns from kohonen layer with target O/Ps.

→ Alg :-

1) Kohonen Layer :-

- Input selection :- Choose a training vector x.
- Find the winner neuron by calculating Euclidian distance b/w i/p & each neuron's weight.

$$D_j = \| x - w_j \|$$

- Weight Update : Update weight of winner & optionally, its neighbors :-

$$w_j(new) = w_j(old) + \alpha \cdot (x - w_j)$$

$\alpha \rightarrow$ learning rate

- Repeat until convergence.

2) Grossberg Layer :-

- I/p selection :- Identify winning neuron in kohonen layer.
- Error Calculation : Calculate error b/w target o/p O_i & the Grossberg o/p y.
- Weight Update : Adjust the Grossberg layer's weights to reduce the error :

$$w_i(new) = w_i(old) + \alpha' \cdot (O_i - y_i) \cdot O_i$$

$\alpha' \rightarrow$ new learning rate

$O_i \rightarrow$ o/p from kohonen layer.

- Repeat until convergence

→ Adv :-

- Combines benefits of both supervised & unsupervised learning.
- Efficient for classification tasks.
- Can handle noisy & incomplete data.

→ App :-

- Pattern classification
- Data compression
- Image recognition

→ There are 2 types of CPNs:

1) Full CPN :-

- It has complete connections b/w all layers.
- The o/ps from kohonen layer & direct connections from i/p layer are to the used as i/p to the Grossberg layer.
- I/p layer connects to both the layers.
- kohonen layer performs unsupervised learning to cluster i/p patterns.
- Grossberg layer performs supervised learning & maps the combined input to the desired o/p.

Adv

- Better accuracy due to combined i/p info.
- Faster convergence

Disadv:

- Higher computational complexity.
- Here in the Grossberg layer, before calculating the error, we ~~apply summation function~~ present the i/p vector to the layer along with o/p. ~~both inputs & outputs~~

2) Forward-Only Counter Propogation N/w :-

- In this, the i/p layer is directly only connected to kohonen layer.
- There are no connection from i/p layer to Grossberg layer.
- Basically normal CPN.

Adaptive Resonance Theory :- (ART):

- It is a type of neural n/w model developed by Stephen Grossberg in 1970s.
- It is designed to solve problems related to pattern recognition, particularly when the data given as i/p is noisy, variable or redundant.
- The core idea behind ART is that the n/w can learn & recognize patterns in real-time while maintaining stability & avoiding catastrophic forgetting (forgetting previous learned patterns)
- One of the key challenges ART address is the "stability - plasticity dilemma", which is the problem of balancing b/w preserving old learned knowledge (stability) & the ability to learn new information (plasticity).
- ART maintains this balance by allowing new patterns to be learned without forgetting previous ones.
- Two main components of ART :-

1) F1 Layer (feature layer):
   - This layer is responsible for recieving the i/p patterns & performing feature extraction.
   - It is also where the i/p features are matched to a codebook (set of learned patterns)

2) F2 Layer (category layer):

- This layer holds the learned categories or prototypes where each is associated with a memory that stores characteristics of patterns that belong to it.
- When a new i/p is processed, it is compared to the categories in this layer.

→ ART uses a process called match tracking, where the network checks the similarity b/w the i/p & the stored patterns (or prototypes).

→ ART's learning process involves two stages:

- Bottom-up Activation:

  The i/p pattern activates the F1 layer, which propogates the activation to the F2 layer.

- Top-Down Inhibition:

  Once a category in the F2 layer is activated, it sends a top-down inhibitory signal to the F1 layer, suppressing activation in other categories that don't match the input.

→ Types of ART Models:-

Several variants of ART, including:

- ART 1:-

  This is the original version of ART & is used for binary i/p patterns.

- ART 2:-

  This version is used for continuous i/p patterns.

• ART 3:
An extension of ART2, designed for the spatiotemporal (space-time) patterns.

→ Adv:-
• It exhibits stability & is not distributed by a wide variety of i/ps.
• It can be integrated & used with various other techniques.

→ App:-
• It can be used for various fields like mobile robot control, face recognition, land cover classification, target recognition, etc.

④ Special N/ws:-
① Feedforward NNs
③ CNNs
② RNNs
④ Generative Adversarial N/ws:- (GANs)
→ They are a type of deep learning model used for generating new data instances that resemble training data.
→ Consists of 2 main neural n/ws:
a) Generator (G): Generates fake data samples
b) Discriminator (D): Evaluates authenticity of the generated data & distinguishes b/w real & fake data.
→ Both networks compete & the generator improves overtime to create more realistic data.

→ Appl:-
• Image Generation, Style Transfer,
• Data Augmentation.