Important terms of ANN

Basic models of ANN

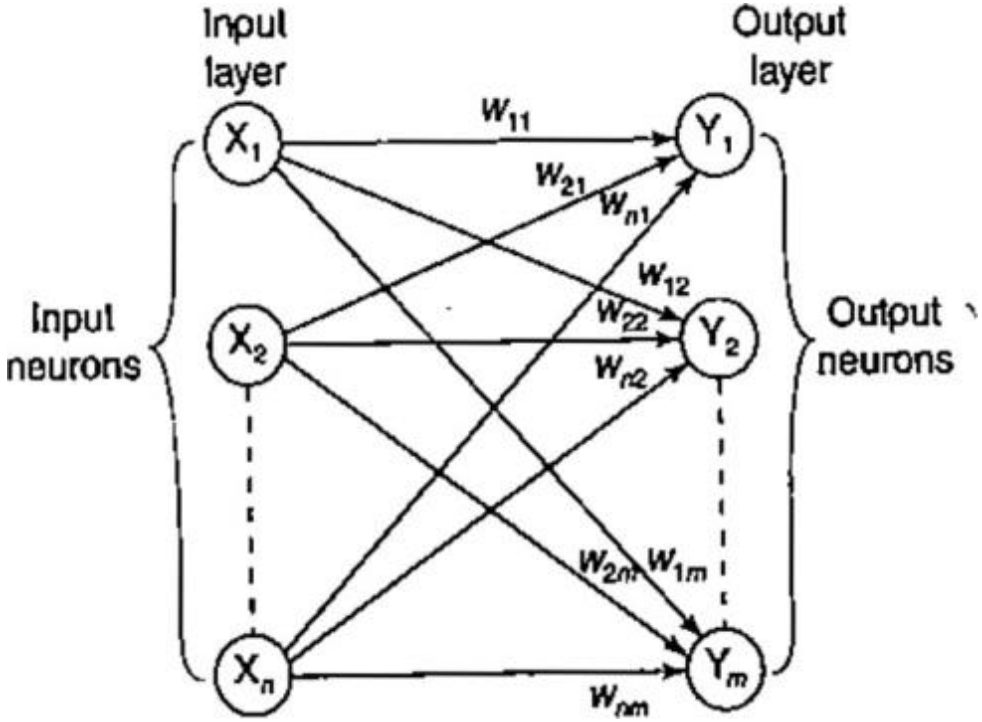
- ANN models are specified by three main entities:
  1. the model's synaptic interconnections
  2. the training or learning rules adopted for updating and adjusting the connection weights
  3. their activation functions

- The arrangement of neurons to form layers and the connection pattern formed within and between layers is the network architecture.
  1. Single-layer feed-forward network
  2. Multilayer feed-forward network
  3. Single node with its own feedback
  4. Single-layer recurrent network
  5. Multilayer recurrent network

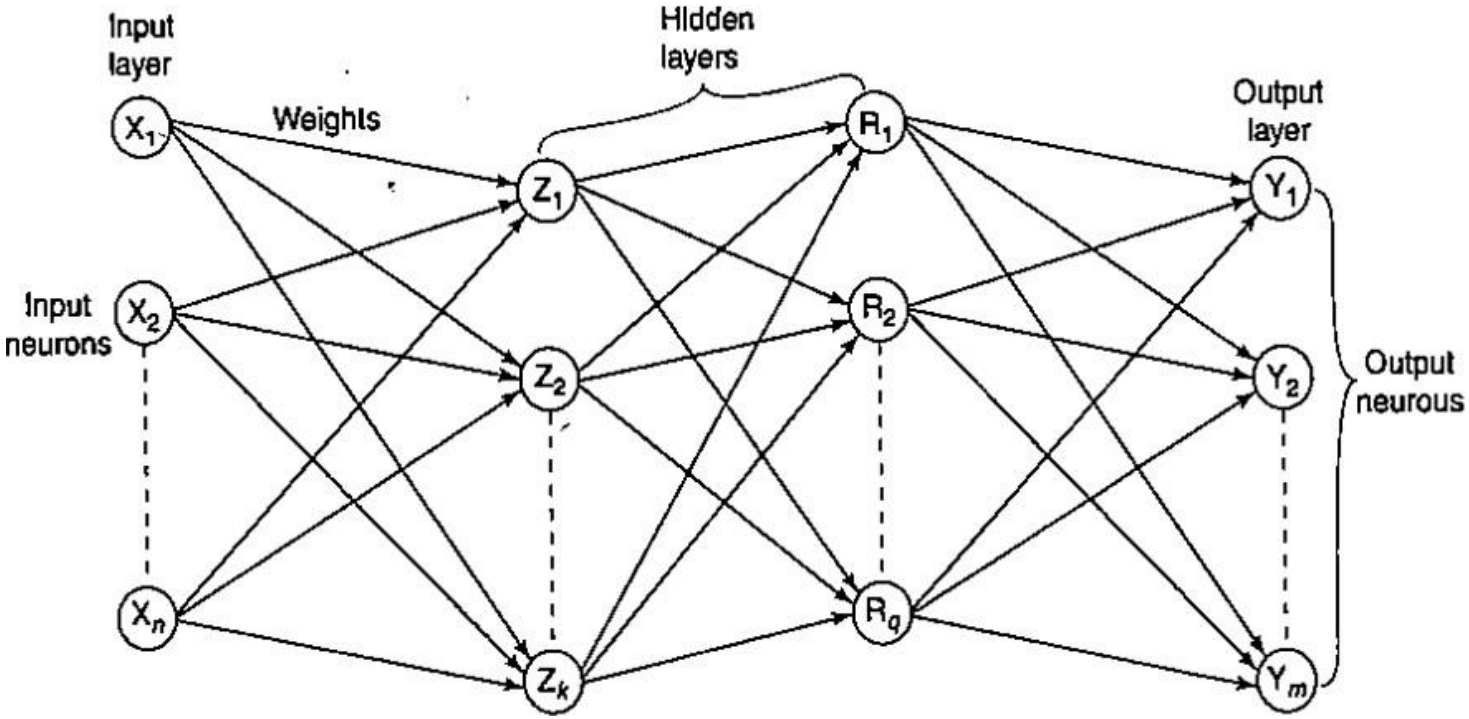
Single-layer feed-forward network

Single-layer feed-forward network

When a layer of the processing nodes is formed, the inputs can be connected to these nodes with various weights, resulting of series of outputs, one per node



Multi-layer feed-forward network



Multi-layer feed-forward network

Multi-layer feed-forward network

- Formed by the interconnection of several layers
- The input layer is that which receives the input and this layer has no function except buffering the input signal. The output layer generates the output of the network
- Any layer that is formed between input and output layers is called hidden layer
- There may be zero to several hidden layers in an ANN
- More the number of the hidden layers, more is the complexity of the network (but provide an efficient output)
- In fully connected network, every output from one layer is connected to each and every node in the next layer

Single-node with its own feedback

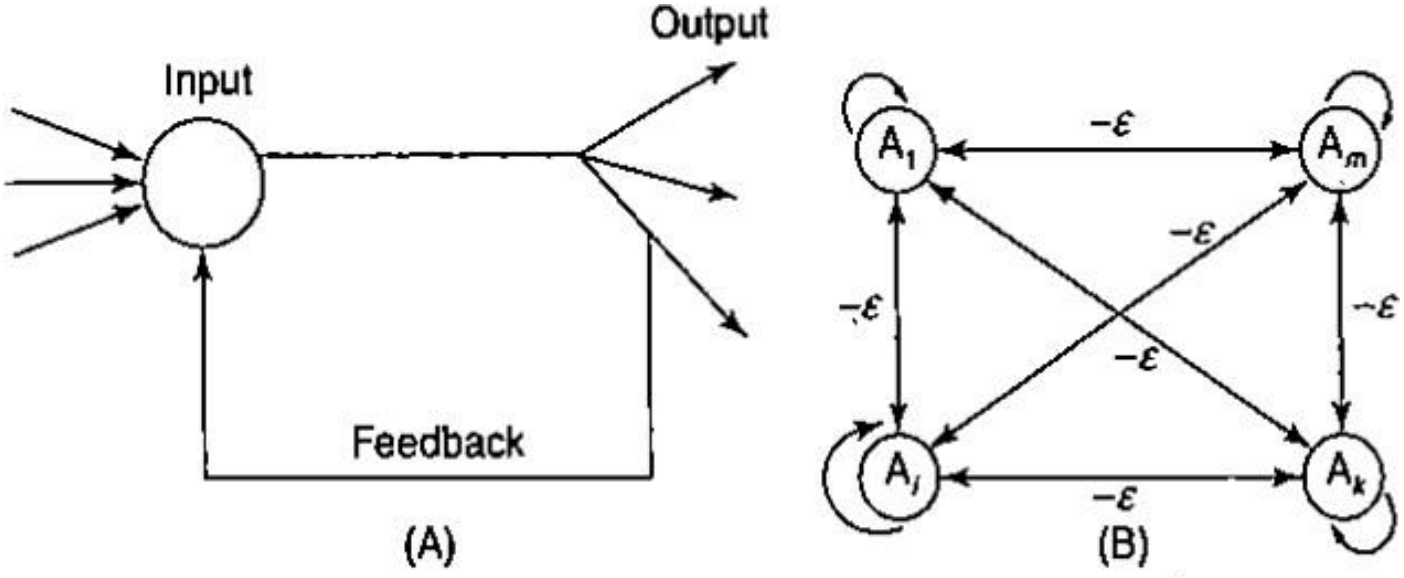


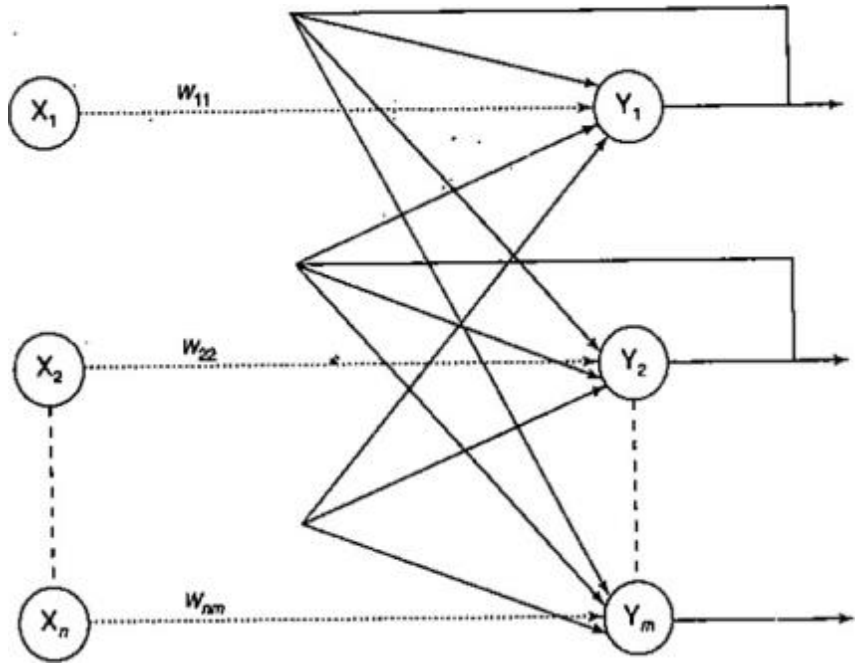
Figure 2-8 (A) Single node with own feedback. (B) Competitive nets.

Single-node with its own feedback

Single-node with its own feedback

- A network is said to be a feed-forward network if no neuron in the output layer is an input to a node in the same layer or in the preceding layer.
- When outputs can be directed back as inputs to same or preceding layer nodes called as feedback network
- If the feedback of the output of the processing elements is directed back as input to the processing elements in the same layer $\Rightarrow$ lateral feedback
- Fig. (A) shows simple recurrent neural network having a single neuron with feedback to itself
- The competitive interconnections has fixed weights of $-\varepsilon$ (Fig.B). This net is called Maxnet

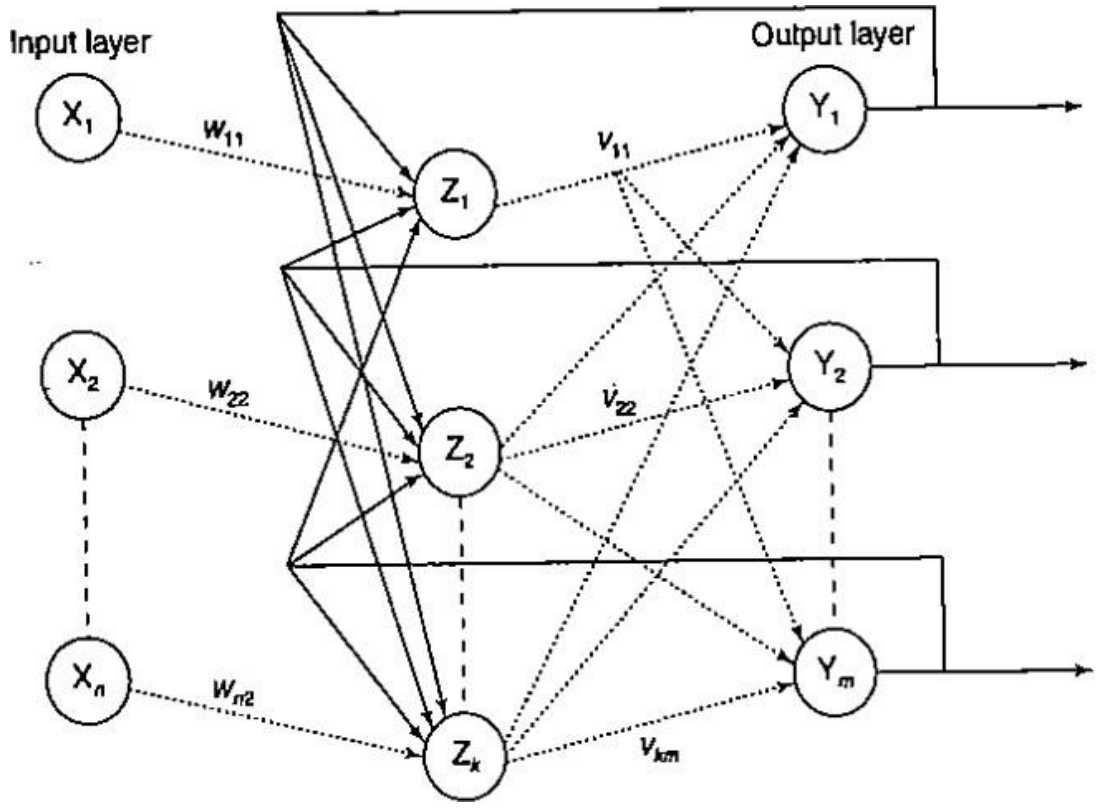
Single-layer recurrent network



Single-layer recurrent network

Single layer network with a feedback connection in which a processing element's output can be directed back to the processing element itself or to the other processing element or to both.

Multilayer recurrent network



A processing element output can be directed back to the nodes in a preceding layer, forming a multilayer recurrent network. Also, in these networks, a processing element output can be directed back to the processing element itself and to other processing elements in the same layer

Important terminologies of ANNs

Weights

- Each communication link between neurons is associated with weights that contain information about input signal
- Weight information is used by the net to solve a problem and weight matrix can also be called connection matrix
- If there are "n" processing elements in an ANN and each processing element has "m" adaptive weights, then weight matrix W is defined as

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n^T \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix}$$

Important terminologies of ANNs

Weights

- where $w_i = [w_{i1}, w_{i2}, \ldots, w_{im}]^T, i = 1, 2, \ldots, n$ is the weight vector of processing element and w_{ij} is the weight from processing element "i^{rr} (source node) to processing element "j^{rr} (destination node).
- The weights encode long-term memory (LTM) and the activation states of neurons encode short-term memory (STM) in a neural network

Important terminologies of ANNs

Bias

- The bias is included in the network has its impact in calculating the net input and is included by adding a component $x_0 = 1$ to the input vector X
- The bias is considered like another weight $w_{0j} = b_j$
- Bias can be of two types: positive bias and negative bias.
- The positive bias helps in increasing the net input of the network and the negative bias helps in decreasing the net input of the network

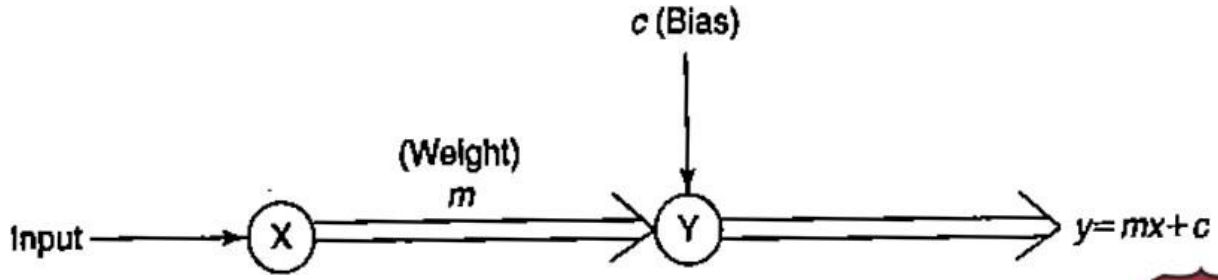
Important terminologies of ANNs

Bias

- Consider an equation of straight line

$$y = mx + c$$

- x is the input, m is the weight, c is the bias and y is the output
- Block schematic shows equation of the straight line
- Thus, bias plays a major role in determining the output of the network



Important terminologies of ANNs

Threshold

- Threshold is a set value based upon which the final output of the network is calculated
- Threshold value is used in the activation function
- A comparison is made between the calculated net input and the threshold to obtain the network output. For each and every application there is a threshold limit
- The activation function using threshold can be defined as

$$f(\text{net}) = \begin{cases} 1 & \text{if net} \geq \theta \\ -1 & \text{if net} < \theta \end{cases}$$

- where θ is the fixed threshold value

Important terminologies of ANNs

Learning Rate

- The learning rate is denoted by α and is used to control the amount of weight adjustment at each step of training
- The learning rate, ranging from 0 to 1 determine the rate of learning at each time step.

Momentum Factor

- Convergence is made faster if a momentum factor is added to the weight updation process.
- Generally done in the back propagation network and if momentum has to be used, the weights from one or more previous training patterns must be saved
- Momentum helps the net in reasonably large weight adjustments

Important terminologies of ANNs

Vigilance Parameter

- Vigilance parameter is denoted by ρ and generally used in Adaptive Resonance Theory (ART)
- Used to control the degree of similarity required for patterns to be assigned to the same cluster unit
- Range of this parameter is from 0.7 to 1 to perform useful work in controlling the number of clusters

Hebb network

Hebb network

Hebb network

- According to Donald Hebb (1949), the learning is performed by the change in the synaptic gap
- In Hebb rule, the weight vector is found to increase proportionately to the product of the input and the learning signal
- The learning signal is equal to the neuron's output
- In Hebb learning, if two interconnected neurons are 'on' simultaneously then the weights associated with these neurons can be increased by the modification made in their synaptic gap (strength)
- The weight update in Hebb rule is given by

Hebb network

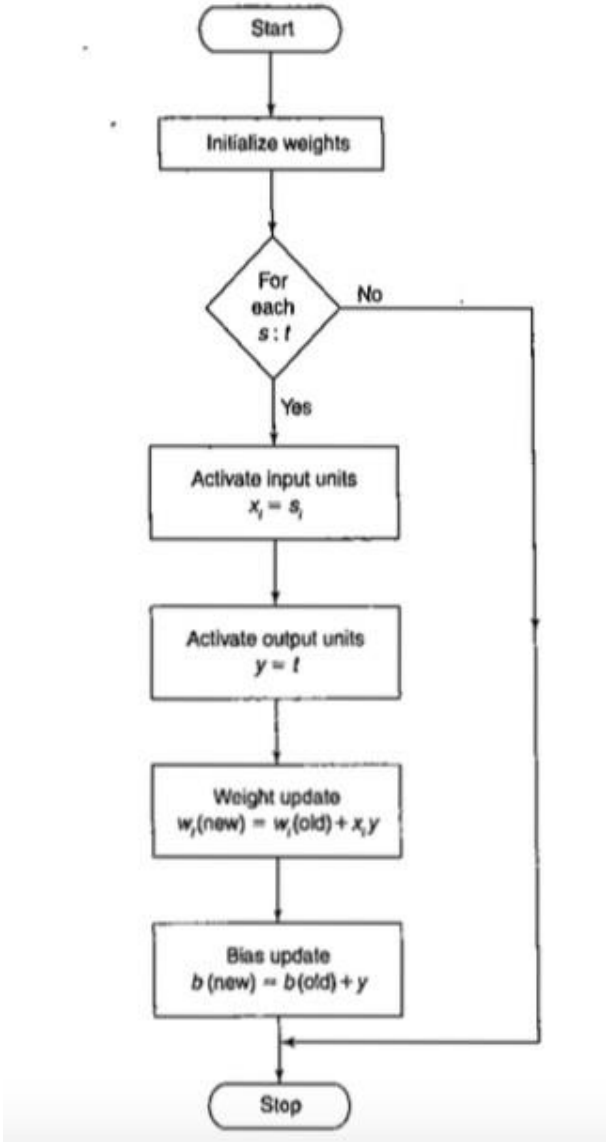
Hebb network

- The weight update in Hebb rule is given by

$$w_i(\text{new}) = w_i(\text{old}) + x_i y \qquad (1)$$

- The Hebb rule is more suited for bipolar data than binary data
- If binary data is used, the above weight updation formula cannot distinguish two conditions namely:
  1. A training pair in which an input unit is "on" and target value is "off"
  2. A training pair in which both the input unit and the target value are "off"

Flowchart of Training Algorithm



Training Algorithm

1. **Step 0**: First initialize the weights. Basically in this network, they may be set to zero i.e. $w_i = 0$ for $i = 1 \ \ to \ \ n$, where n=total number of input neurons
2. **Step 1**: Steps 2-4 have to performed for each input training vector and target output pair, $s : t$
3. **Step 2**: Input units activations are set. Generally, the activation function of input layer is identity function: $x_i = s_i$ for $i = 1$ to n
4. **Step 3**: Output units activations are set: $y_i = t_i$
5. **Step 4**:Weight adjustments and bias adjustments are performed:
   $w_i(\text{new}) = w_i(\text{old}) + x_i y$
   $b(\text{new}) = b(\text{old}) + y$
6. The above five steps complete the algorithmic process. In step 4, the weight updation formula can also be given in vector form as
   $w(\text{new}) = w(\text{old}) + xy$
   Here the change in weight can be expressed as:
   $\Delta w = xy$
   As a result:
   $w(\text{new}) = w(\text{old}) + \Delta w$

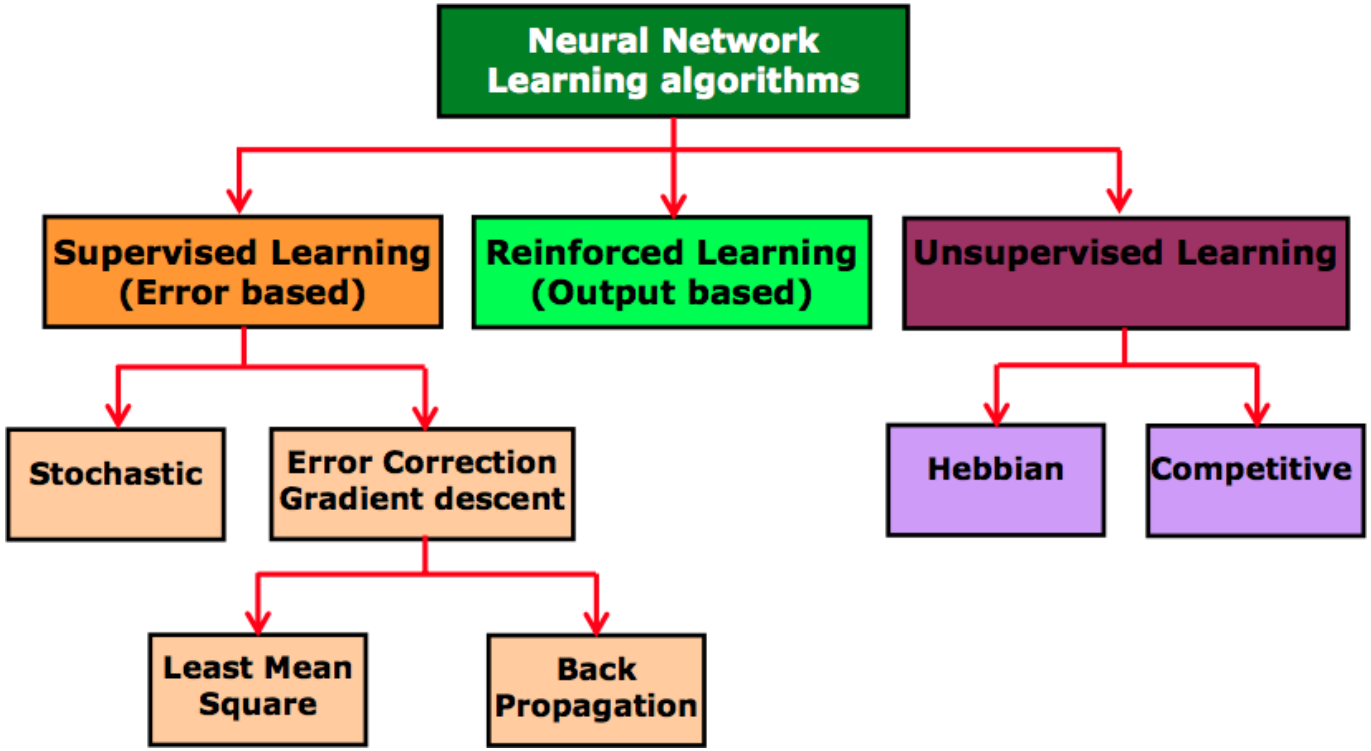
Supervised learning

Learning

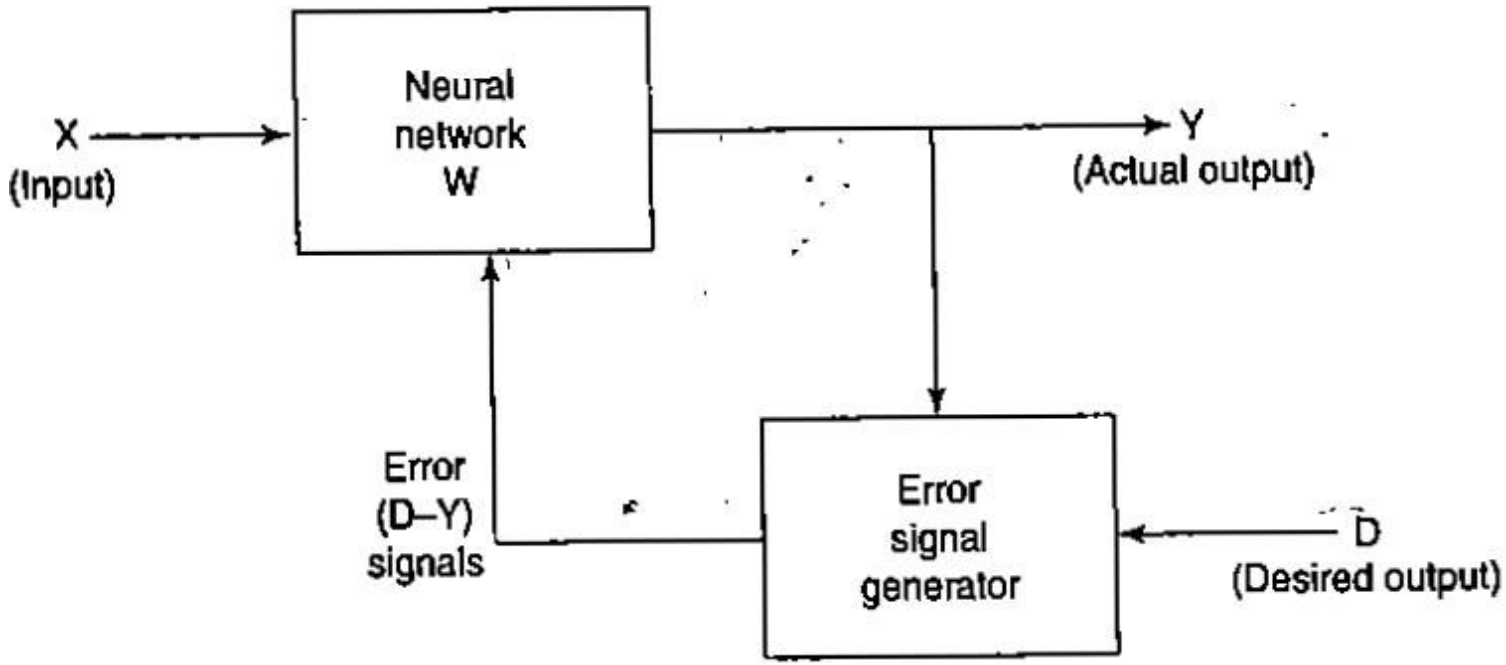
Learning

- Learning or training is a process by means of which a neural network adapts itself to a stimulus by making proper parameter adjustment resulting in desired output.
- Broadly, there are two kinds learning in ANNs
  1. Parameter learning: updates the connecting weights
  2. Structure learning: focuses on the change in network structure (number of processing elements and their connection types)
- ANN learning can be generally classified into three categories:
  1. Supervised learning
  2. Unsupervised learning
  3. Reinforcement learning
- Based on learning: (a) Hebbian (2) Gradient descent (3) Competitive

Classification of learning algorithms



Supervised learning

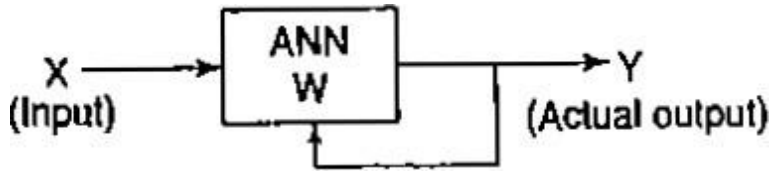


Supervised learning

Supervised learning

- A teacher is present during learning process and provides expected outputs
- Each input vector requires a corresponding target vector, which represents the desired output. The input vector along with the target vector is called training pair
- During training, the input vector is presented to the network, which results in an output vector which is the actual output vector.
- Actual output vector is compared with the desired (target) vector
- If there exists a difference between the two output vectors then an error signal is generated
- This error signal is used for adjustment of weights until the actual output matches the desired (target) output

Unsupervised learning



Unsupervised learning

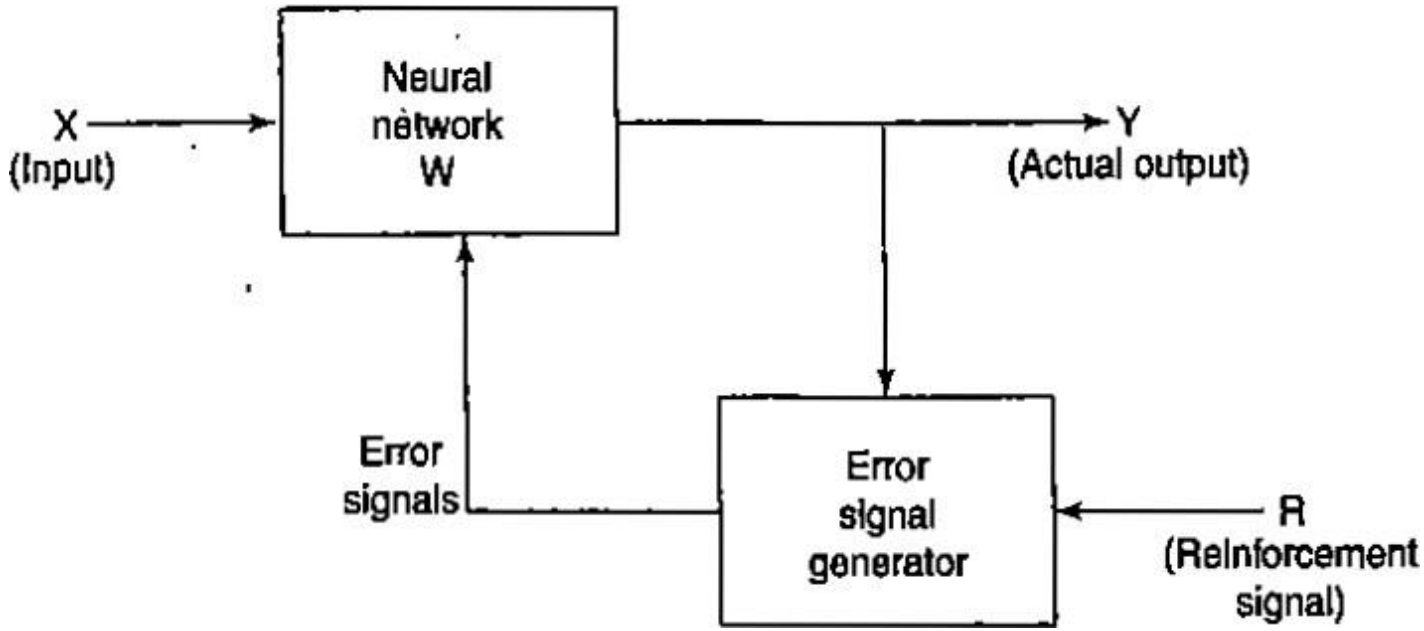
- Learning without the help of a teacher
- Expected (desired) output is not presented during learning
- Learning process is independent and is not supervised by a teacher
- Input vectors of are grouped without the use of training data to specify how a member of each group looks or to which group a number belongs
- In training process, network receives the input patterns and organizes these patterns to form clusters
- When a new input pattern is applied, ANN gives an output response indicating the class to which the input pattern belongs

Unsupervised learning

Unsupervised learning

- If for an input, a pattern class can not be found, then a new class is generated
- The network must itself discover patterns, regularities, features or categories from the input data and relations for the input data over the output
- While discovering all these features, the network undergoes change in its parameters. This process is called self-organizing in which exact clusters will be formed by discovering similarities and dissimilarities among the objects

Reinforcement learning



Reinforcement learning

Reinforcement learning

- A teacher is present but does not present the expected (desired) output but only indicated if the computed output is correct or incorrect
- Only critic information is available, nor the exact information. The learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal
- The feedback obtained is only evaluative and not instructive
- External reinforcement signals are processed in the critic signal generator, and the obtained critic signals are sent to the ANN for adjustment of weights properly so as to get better critic feedback in future
- The reinforcement learning is also called learning with a critic as opposed to learning with a teacher, which indicates supervised learning

Applications of ANN

Applications of ANN

An artificial neural network (ANN) may be defined as an information processing model that is inspired by the way biological nervous systems, such as the brain, process information. An ANN is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

Applications

- Air traffic control could be automated with the location, altitude, direction and speed of each radar blip taken as input to the network
- Animal behavior, predator/prey relationships and population cycles may be suitable for analysis by neural networks
- Appraisal and valuation of property, buildings, automobiles, machinery
- Betting on horse races, stock markets, sporting events (predictions)
- Criminal sentencing (predicted using a large sample of crime details as input and the resulting sentences as output)
- Complex physical and chemical processes that may involve the interaction of numerous (possibly unknown) mathematical formulas could be modeled heur

Applications of ANN

Applications

- Data mining, cleaning and validation could be achieved by determining which records suspiciously diverge from the pattern of their peers.
- Direct mail advertisers could use neural network analysis of their databases to decide which customers should be targeted, and avoid wasting money on unlikely targets
- Echo patterns from sonar, radar, seismic and magnetic instrumems could be used to predict their targets
- Econometric modeling
- Employee hiring to predict which job applicant would show the best job performance
- Expert consultants could package their intuitive expertise into a neural network to automate their services
- Fraud detection regarding credit cards, insurance or taxes using past incidents
- Handwriting and typewriting recognition (Optical Character Recognition)
- Lake water levels could be predicted based upon precipitation patterns and river/dam flows
- Machinery control automation by capturing the actions of experien ators into a neural network

Applications of ANN

Applications

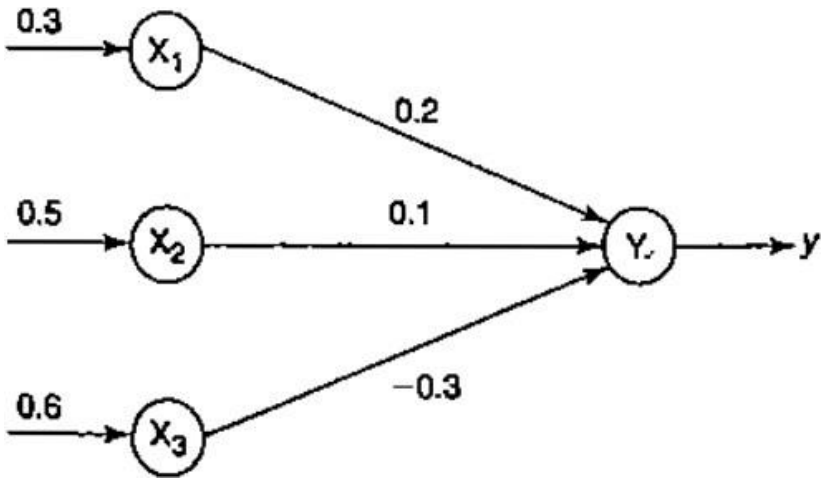
- Medical diagnosis
- Medical research relies heavily on classical statistics to analyze research data
- Music composition - The network is trained to recognize patterns in the pitch and tempo of certain music, and then the network writes its own music
- Photos and fingerprints recognition
- Recipes and chemical formulations could be optimized based on the predicted out-come of a formula change
- Retail inventories could be optimized by predicting demand based on past patterns
- River water levels could be predicted based on upstream reports, and time and location of each report.
- Scheduling of buses, airplanes and elevators could be optimized by predicting demand
- Staff scheduling requirements for restaurants, retail stores, police stations, banks, etc., could be predicted based on the customer flow, day of week, paydays, holidays, weather, season, etc

Applications of ANN

Applications

- Strategies for games, business and war can be captured by analyzing the expert player;s response to given stimuli
- Traffic flows prediction and optimization
- Speech Recognition and Speech Synthesis
- Weather prediction
- Image processing applications
- ECG/EEG/EMG Filtering/Classification
- Machine Control/Robot manipulation
- VLSI placement and routing
- Hospital patient stay length prediction
- Natural gas price prediction

Example 1



For the network shown, calculate the net input to the output neuron.

- The given neural net consists of three input neurons and one output neuron.
- Inputs and weights are

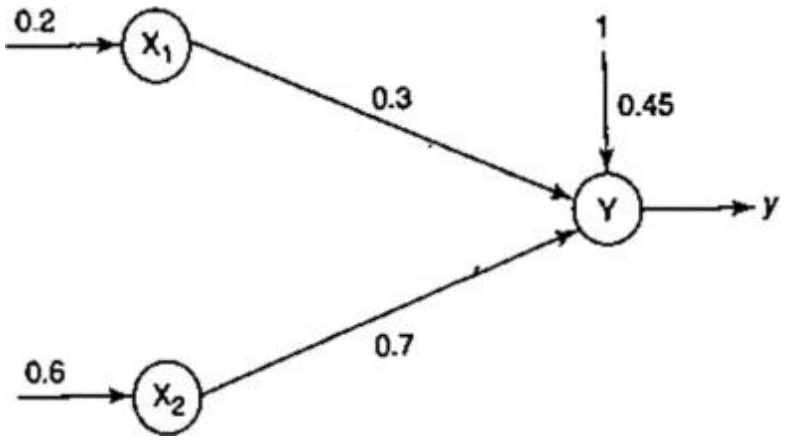
$$[x_1, x_2, x_3] = [0.3, 0.5, 0.6]$$

$$[w_1, w_2, w_3] = [0.2, 0.1, -0.3]$$

- The net input can be calculated as

$$\begin{aligned} y_{in} &= w_1x_1 + w_2x_2 + w_3x_3 \\ &= 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times (-0.3) \\ &= 0.06 + 0.05 - 0.18 \\ &= -0.07 \end{aligned}$$

Example 2



Calculate the net input for the network shown, with bias included in the network

- The given neural net consists of two input neurons, a bias and one output neuron.
- Inputs, weights and bias are

$$[x_1, x_2] = [0.2, 0.6]$$

- $$[w_1, w_2] = [0.3, 0.7]$$
- $b = 0.45$
- The net input can be calculated as

$$\begin{aligned} y_{in} &= b + w_1x_1 + w_2x_2 \\ &= 0.45 + 0.2 \times 0.3 + 0.6 \times 0.7 \\ &= 0.45 + 0.06 + 0.42 \\ &= 0.93 \end{aligned}$$