(2) Ch:18 Connectionist Models. (27) (1) 107

K-1

↳ Connectionism: (Connectionism is a set of approaches in the field of AI, in which multiple connections between nodes (brain cell) from a ~~massive~~ large interactive network in which many processes take place simultaneously) and certain processes, operating in parallel, are grouped together in hierarchies that bring about results such as thought or action.

↳ There are many forms of Connectionism but the most common forms use Neural Network models.

↳ **Neural Network** :- A neural network is an interconnected group of neurons. (Biological neural networks, especially the human brain.

↳ **Artificial Neural Network** :-

Artificial Neural Network try to (copy/learn) mimics the human brain.

algorithm.

Learning in Neural Network

Perceptron

- The perceptron an invention of (1962) Rosenblatt was one of the earliest neural network models.
- It models a neuron by taking a weighted sum of its inputs and sending the output 1 if the sum is greater than some adjustable threshold value (otherwise it sends 0).

$g(x) = \sum_{k=0}^{n} WkXk$	Output(x) = $\begin{cases} 1 & \text{if } g(x) > 0 \\ 0 & \text{if } g(x) > 0 \end{cases}$

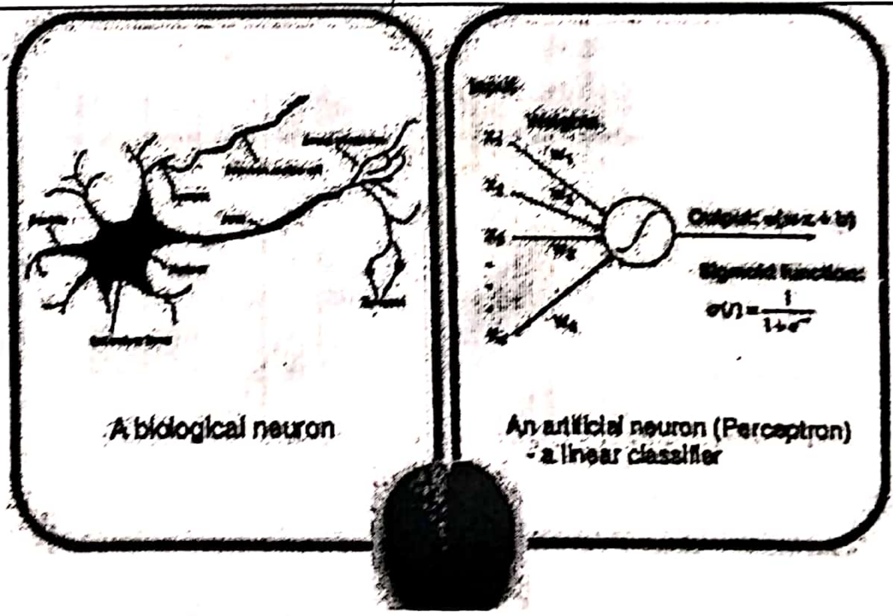


Figure 13.1 A neuron & a Perceptron

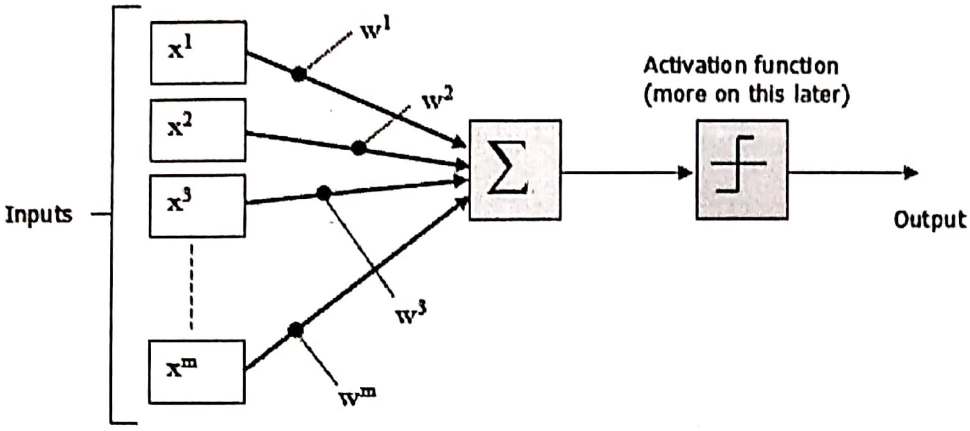


Figure 13.2 Perceptron with adjustable threshold

- In case of zero with two inputs

$$g(x) = w0 + w1x1 + w2x2 = 0$$

$$x2 = -(w1/w2)x1 - (w0/w2) \rightarrow \text{equation for a line}$$

- the location of the line is determine by the weight w0 w1 and w2
- if an input vector lies on one side of the line, the perceptron will output 1
- if it lies on the other side, the perceptron will output 0
- Decision surface: a line that correctly separates the training instances corresponds to a perfectly function perceptron.

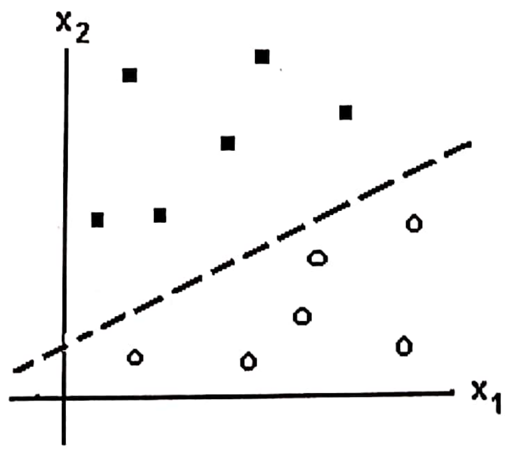


Figure 13.3 Linearly Separable Pattern Classification Problem

Perceptron Learning Algorithm

Given: A classification problem with n input feature (x_1, x_2, ..., x_n) and two output classes.

Compute: A set of weights (w_0, w_1, w_2,....,w_n) that will cause a perceptron to fire whenever the input falls into the first output class.

1. Create a perceptron with n+ 1 input and n+ 1 weight, where the x_0 is always set to 1.
2. Initialize the weights (w_0, w_1,...., w_n) to random real values.
3. Iterate through the training set, collecting all examples *misclassified* by the current set of

weights.

4. If all examples are classified correctly, output the weights and quit.
5. Otherwise, compute the vector sum S of the misclassified input vectors where each vector has the form $(x_0, x_1, \ldots, x_n)$. In creating the sum, add to S a vector $\vec{x}$ if $\vec{x}$ is an input for which the perceptron incorrectly fails to fire, but $-\vec{x}$ if $\vec{x}$ is an input for which the perceptron incorrectly fires. Multiply sum by a scale factor η.
6. Modify the weights $(w_0, w_1, \ldots, w_n)$ by adding the elements of the vector S to them. Go to step 3.

- The perceptron learning algorithm is a search algorithm. It begins in a random initial state and finds a solution state. The search space is simply all possible assignments of real values to the weights of the perceptron, and the search strategy is gradient descent.
- The perceptron learning rule is guaranteed to converge to a solution in a finite number of steps, so long as a solution exists.
- This brings us to an important question. What problems can a perceptron solve? Recall that a single-neuron perceptron is able to divide the input space into two regions.
- The perceptron can be used to classify input vectors that can be separated by a linear boundary. We call such vectors linearly separable.
- Unfortunately, many problems are not linearly separable. The classic example is the XOR gate. **It was the inability of the basic perceptron to solve such simple problems that are not linearly separable or non-linear.**

Artificial neural network

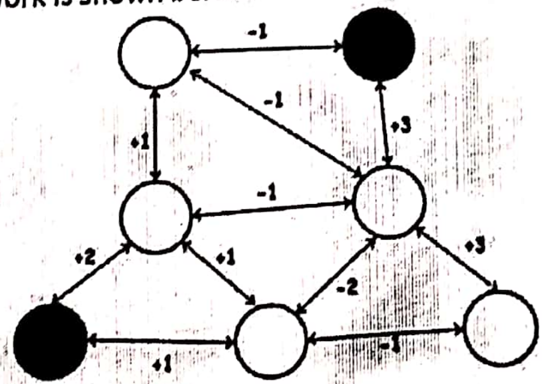
- An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain process information.
- Typically, neurons are five to six orders of magnitude slower than silicon logic gates; events in a silicon chip happen in the nanosecond (10-9 s) range, whereas neural events happen in the millisecond (10-3 s) range.
- The brain is a highly complex, nonlinear, and parallel information-processing system.
- It has the capability of organizing neurons so as to perform certain computations (e.g. pattern recognition, perception, and motor control) many times faster than the fastest digital computer.
- A brain has great structure and the ability to build up its own rules through what we usually refer to as experience.
- During this early stage of development, about one million synapses are formed per second.
- Synapses are elementary structural and functional units that mediate the interactions between neurons.
- A developing neuron is synonymous with a plastic brain: Plasticity permits the developing nervous system to adapt to its surrounding environment.
- Axons act as transmission lines, and dendrites represent receptive zones. Neurons come in a

Introduction

- Neural network architectures have been called connectionist architecture.
- They are characterized by having:
    - A very large number of simple neuron-like processing elements.
    - A large number of weighted connections between the elements. The weights on the connections encode the knowledge of a network.
    - Highly parallel, distributed control.
    - An emphasis on learning internal representations automatically.

Hopfield Networks

- Hopfield [1982]: introduced a neural network as a theory of memory: Model of content addressable memory.
- Features of a Hopfield Network:
    - Distributed representation
    - A memory is stored as a pattern of activation across a set of processing elements.
    - Distributed, asynchronous control
    - Content-addressable memory
    - A number of patterns can be stored in a network. To retrieve a pattern, a specific portion of it is specified and the network automatically finds the closest match.
    - Fault tolerance: If a few processing elements misbehave or fail completely, the network will still function properly.
    - Each processing element makes decisions based only on its own local situation.
- A simple Hopfield Network is shown below:



Black → active

White → inactive

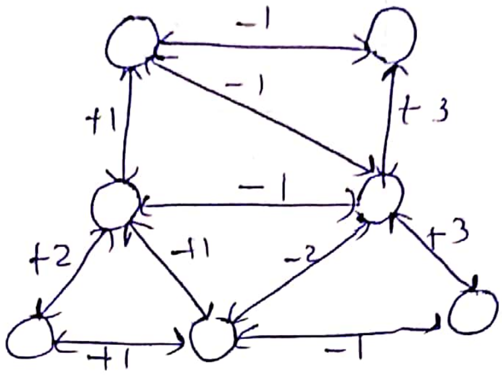
- Processing elements or units are always in one of two states, active or inactive.
- Units are connected to each other with weighted symmetric connection a positive weighted connection indicates that the two units tend to activate each other.
- A negative weighted connection allows an active unit to deactivate a neighboring unit.

Four stable state of Hopfield network :

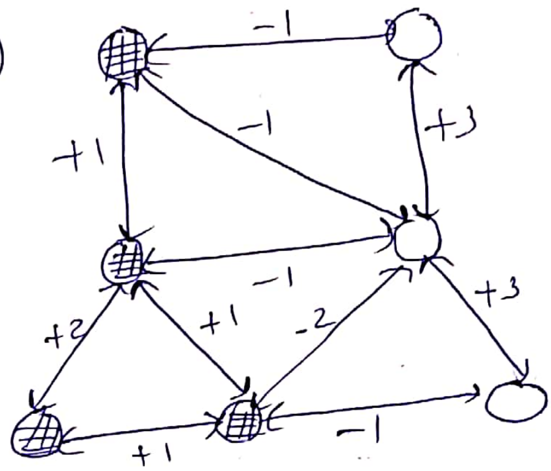
○ → Inactive

▦ → active

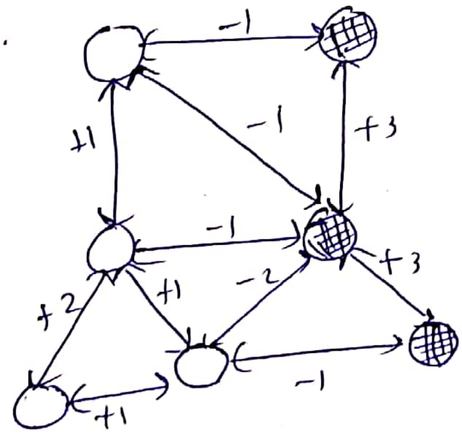
1)



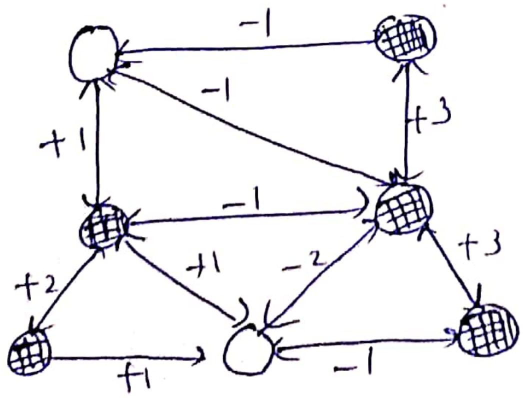
2)



3)



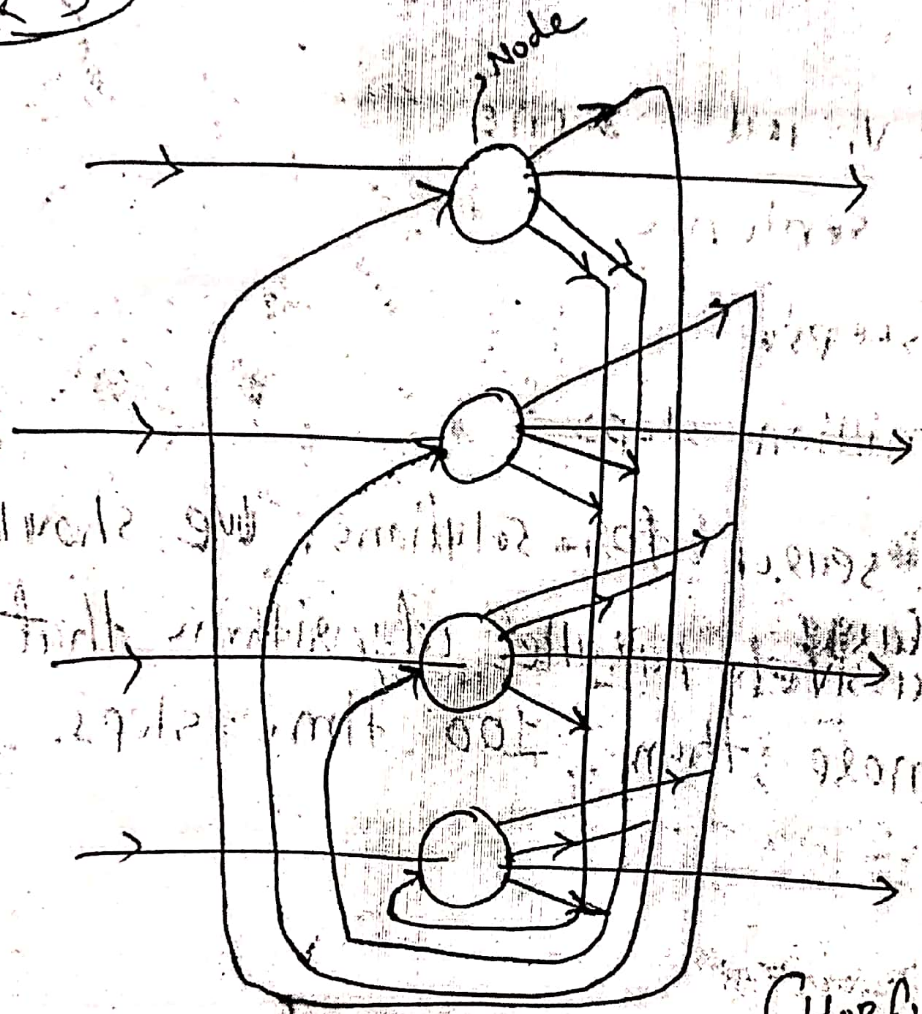
4)



Recurrent Networks

- Recurrent Networks are used in temporal AI task such as planning, natural language processing, etc...
- A recurrent neural network (RNN) is a class of neural network where connections between units form a directed cycle.
- This creates an internal state of the network which allows it to exhibit dynamic temporal behavior.
- Unlike feed-forward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs.
- This makes them applicable to tasks such as un-segmented connected handwriting recognition, where they have achieved the best known results.
- This is the basic architecture developed in the 1980s: a network of neuron-like units, each with a directed connection to every other unit.
- Each unit has a time-varying real-valued activation. Each connection has a modifiable real-valued weight. Some of the nodes are called input nodes, some output nodes, the rest hidden nodes.

K-3



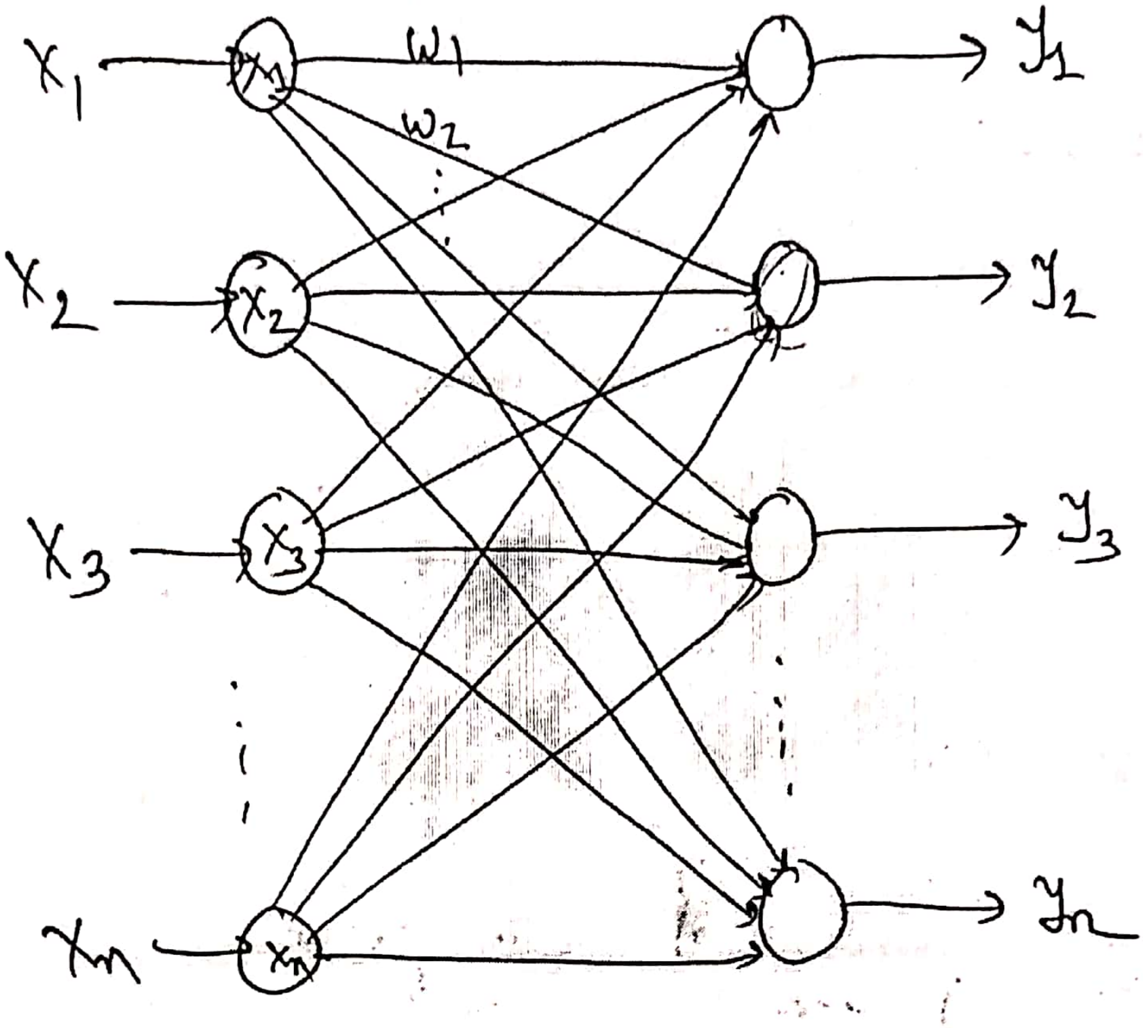
(HOPfield Recurrent network)

(*) Single layer Network (K-7)

(6)

(18.1.1)

Type-I



Input = x_1, x_2, x_3 --- x_n

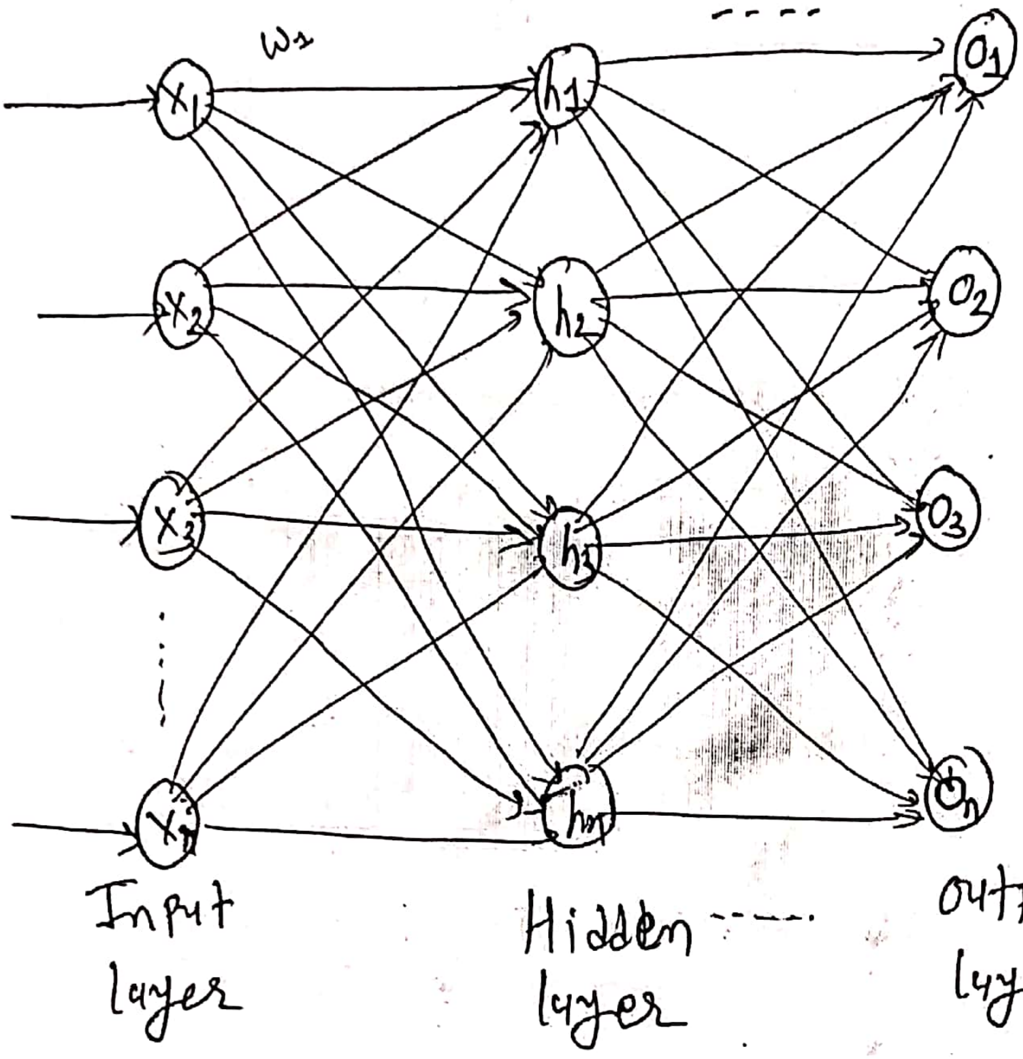
output = y_1, y_2, y_3 ---- y_n

weight = ω_1, ω_2 -----

Scanned by CamScanner

7 * Multilayer Network (K-8)

Type-2



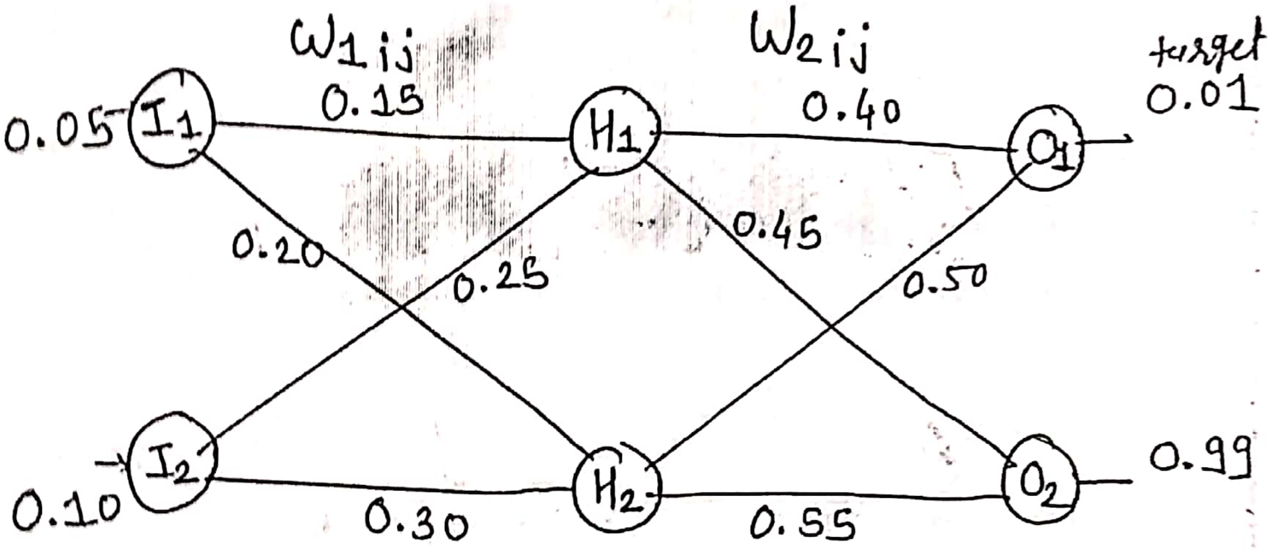
Input = $x_1, x_2, x_3 \ldots x_n$

output = $O_1, O_2, O_3 \ldots O_n$

Hidden layer = $h_1, h_2, h_3, \ldots h_n$ (More possible hidden layer)

Backpropagation Networks (K-9)

18.2.2



↳ Three layers
- Input (x_i or I_i)
- hidden (h_i)
- output (O_i)

↳ Weights on Connection between Input to hidden layer is $W1_{ij}$ and weights on Connection between hidden to output layer is $W2_{ij}$.

⇒ Training a Network by backpropagation involves 3 stages :-

1. Feedforward of the input training Pattern
2. Backpropagation of the associated error
3. Adjustment of weights.

Scanned by CamScanner

(K10)

→ During feedforward, each input unit receives an input signal & broadcasts the signal to each of the hidden units. Each hidden unit then computes its activation and sends its signal to each output unit. Each output unit computes its activation to form the response of the net for the given input pattern.

→ During Training, each output unit compares its computed activation with its target value to determine the associated error for that pattern with that unit.

→ ~~Er~~ Difference (error) is used to update the weights between the output and hidden unit.

$$E_{total} = \sum \frac{1}{2} (target - output)^2$$

$$E_{O1} = \frac{1}{2} (target - output\ O_1)^2$$

$$E_{O2} = \frac{1}{2} (target - output\ O_2)^2$$

$$E_{total} = E_{O1} + E_{O2}$$

∂

δ

$$W_5 \rightarrow \frac{\partial E_{total}}{\partial W_5}$$

Scanned by CamScanner

(18)

Connectionist AI & Symbolic AI

⇒ **Symbolic AI** :-

↳ Symbolic AI represents information through symbols and their relationships.

→ Represented by propositions.

Ex:

LECTURE_BORING ∧ TIME_LATE $\vec{\ominus}$ SLEEP.

→ If the features LECTURE_BORING and TIME_LATE are both true, the feature SLEEP is also true.

→ Connectives :-

→ ∨ "or"

→ ∧ "and"

→ $\vec{\ominus}$ implies or if then

→ ¬ "not"

Scanned by CamScanner

Example (K-14)

"Anil is intelligent or he is a good actor. If Anil is intelligent, then he can count from 1 to 10. Anil can only count from 1 to 2. Therefore, Anil is good actor."

↳ **Prepositions:-**

I: "Anil is Intelligent".
A: "Anil is good Actor"
C: "Anil can count from 1 to 10".

↳ Step 1 : $\neg C$

Step 2 : $I \rightarrow C$

Step 3 : $\neg I$ (from 1 & 2)

Step 4 : $A \vee I$

Step 5 : A (from 3 & 4).

Conclusion: A ("Anil is good actor").

⟹ Symbolic AI fail to solve following Prob.

① Common Sense Problem :
- In symbolic Required rule → Not possible rule for Common sense
- fail to give info.

② Expert System : Expert system needs large amount of ~~data~~ knowledge from expertise & Require large data to process, where symbolic AI fails.

③ Game Playing :- In game playing when more than one player are involved symbolic AI fails to work.

[illegible] methodology [illegible] થાય નહીં

5-20

⟹ Connectionist AI

↳ Connectionist models focus more on learning than representation.

– Use Backpropagation Algo.

1. supervised learning
2. unsupervised learning

Supervised and Unsupervised Learning

Supervised learning

- An essential ingredient of supervised learning is the availability of an external teacher, which is able to provide the neural network with a desired or target response.
- The network parameters are adjusted under the combined influence of the training vector and the error signal.
- This adjustment is carried out iteratively in a step-by-step fashion with the aim of eventually making the neural network emulate the teacher.
- This form of supervised learning is in fact an error-correction learning, which was already described.

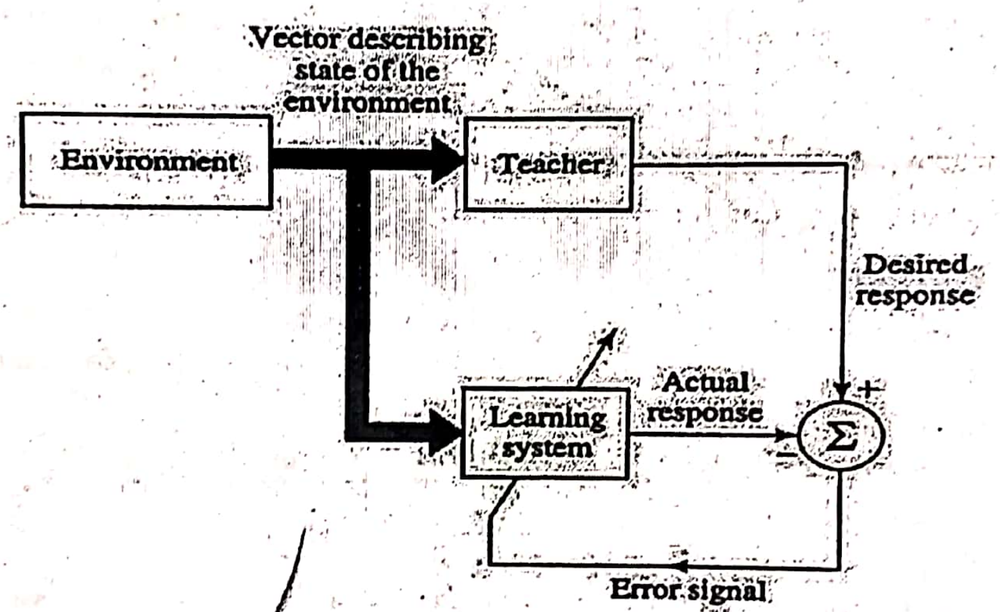


Figure 13.6 Supervised learning

Unsupervised learning

- In unsupervised or self-organized learning there is no external teacher to oversee the learning process.
- In other words, there are no specific samples of the function to be learned by the network.
- Rather, provision is made for a task-independent measure of the quality of

representation that the network is required to learn and the free parameters of the network are optimized with respect to that measure.

- Once the network has become tuned to the statistical regularities of the input data, it develops the ability to form internal representations for encoding features of the input and thereby creates new classes automatically.

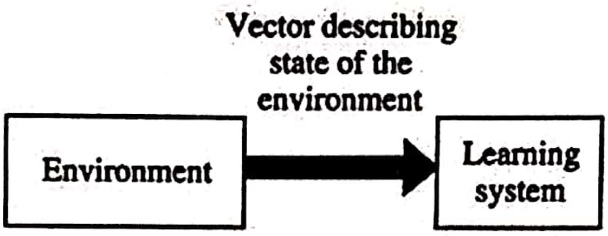


Figure 13.7 Unsupervised learning

Applications of neural networks

1. **Neural Networks in Practice**
    - Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:
        - Sales forecasting , industrial process control, customer research, data validation, risk management, target marketing.
    - ANN are also used in the following specific paradigms: recognition of speakers in communications; diagnosis of hepatitis; recovery of telecommunications from faulty software; interpretation of multi-meaning Chinese words; undersea mine detection; texture analysis; three-dimensional object recognition; hand-written word recognition; and facial recognition.
2. **Neural networks in medicine**
    - Artificial Neural Networks (ANN) are currently a promising research area in medicine and it is believed that they will receive extensive application to biomedical systems in the next few years.
    - At the moment, the research is mostly on modeling parts of the human body and recognizing diseases from various scans.
    - Modeling and Diagnosing the Cardiovascular System
        - At the moment, the research is mostly on modeling parts of the human body and recognizing diseases from various scans
        - Diagnosis can be achieved by building a model of the cardiovascular system of an individual and comparing it with the real time physiological measurements taken from the patient.
    - Electronic noses
        - ANNs are used experimentally to implement electronic noses

- Electronic noses have several potential applications in telemedicine.
- The electronic nose would identify odors in the remote surgical environment.
- These identified odors would then be electronically transmitted to another site where a door generation system would recreate them.

- Instant Physician
  - An application developed in the mid-1980s called the "instant physician" trained an auto associative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis, and treatment for a particular case.

3. Neural Networks in business

- Business is a diverted field with several general areas of specialization such as accounting or financial analysis.
- Almost any neural network application would fit into one business area or financial analysis.
- There is some potential for using neural networks for business purposes, including resource allocation and scheduling.
- There is also a strong potential for using neural networks for database mining, which is, searching for patterns implicit within the explicitly stored information in databases.
- Marketing
  - There is a marketing application which has been integrated with a neural network system.
  - The Airline Marketing Tactician (a trademark abbreviated as AMT) is a computer system made of various intelligent technologies including expert systems.
  - A feedforward neural network is integrated with the AMT and was trained using back-propagation to assist the marketing control of airline seat allocations.
  - The adaptive neural approach was amenable to rule expression.
  - Additionally, the application's environment changed rapidly and constantly, which required a continuously adaptive solution.
- Credit Evaluation
  - The HNC Company, founded by Robert Hecht-Nielsen, has developed several neural network applications.
  - One of them is the Credit Scoring system which increases the profitability of the existing model up to 27%.
  - The HNC neural systems were also applied to mortgage screening.
  - A neural network automated mortgage insurance underwriting system was developed by the Nestor Company.

- Apart from that ANN is also used for,
  - **Classification**
    - ✓ In marketing: consumer spending pattern classification
    - ✓ In defense: radar and sonar image classification

- In agriculture & fishing: fruit and catch grading
- In medicine: ultrasound and electrocardiogram image classification, EEGs, medical diagnosis

- **Recognition and identification**
  - In general computing and telecommunications: speech, vision and handwriting recognition.
  - In finance: signature verification and bank note verification.
- **Assessment**
  - In engineering: product inspection monitoring and control
  - In defense: target tracking
  - In security: motion detection, surveillance image analysis and fingerprint matching
- **Forecasting and prediction**
  - In finance: foreign exchange rate and stock market forecasting
  - In agriculture: crop yield forecasting
  - In marketing: sales forecasting
  - In meteorology: weather prediction