Unit – II

Learning and Training

Learning

- Learning is a process by which free parameters of NN are adapted through stimulation from environment
- Sequence of Events
  - stimulated by an environment
  - undergoes changes in its free parameters
  - responds in a new way to the environment
- Learning Algorithm
  - prescribed steps of process to make a system learn
    - o ways to adjust synaptic weight of a neuron
  - No unique learning algorithms - kit of tools

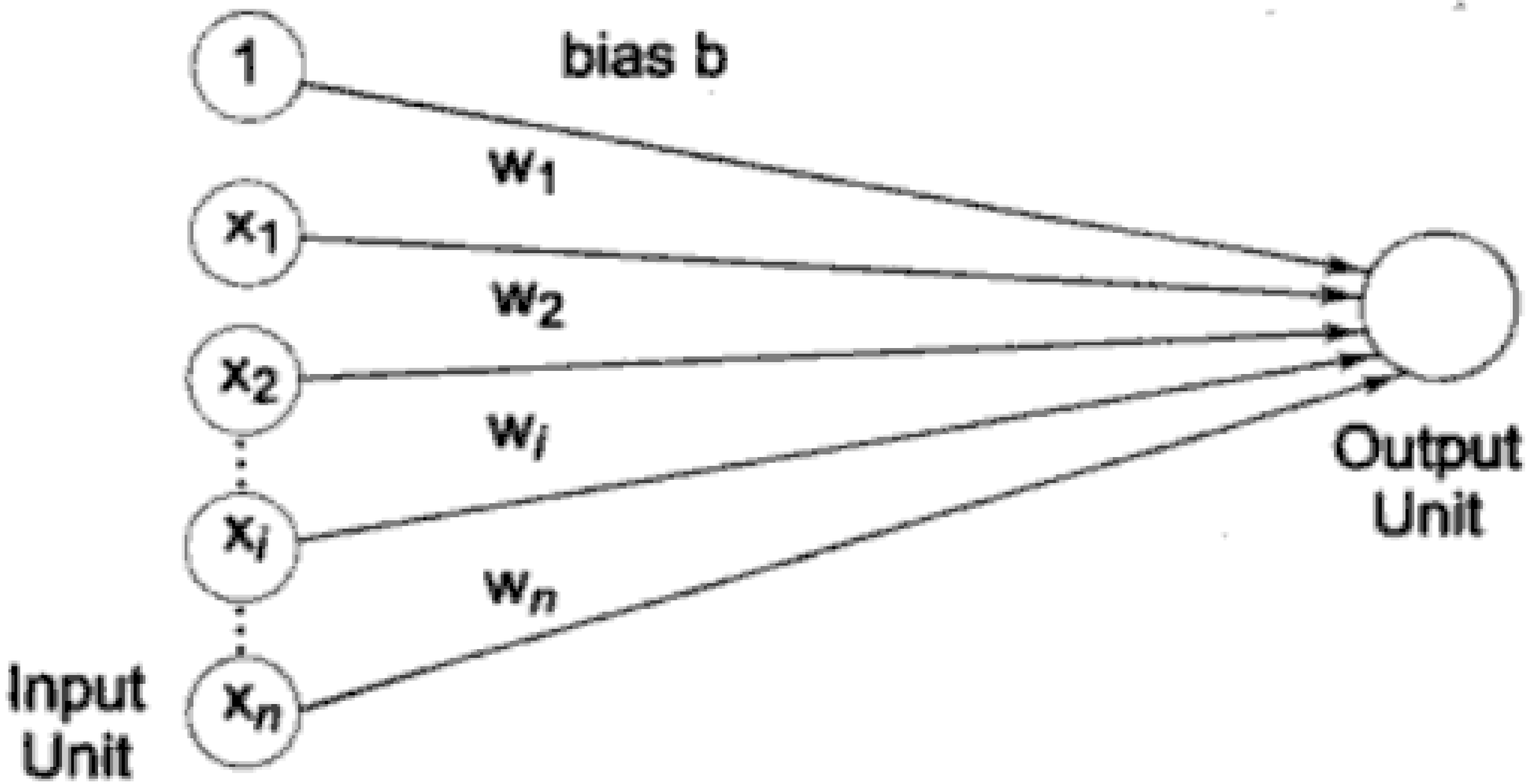
Hebbian Learning Rule

- If two neurons of a connection are activated
  - simultaneously (synchronously), then its strength is increased
  - asynchronously, then the strength is weakened or eliminated
- Hebbian synapse
  - time dependent
    - depend on exact time of occurrence of two signals
  - local
    - locally available information is used
  - interactive mechanism
    - learning is done by two signal interaction
  - conjunctional or correlation mechanism
    - Co-occurrence of two signals

Hebb Net

- The first learning law for artificial neural network was designed by Donald Hebb in 1949.
- The Hebbian rule adopted to formed the Hebb net.
- For Hebb net the input and output data should be in bipolar form.
- Hebb net can’t learn if data is in binary form.

Architecture of Hebb Net



Algorithm

- Step 1: Initialize all weights and bias to Zero.
  - w_i = 0 for i =1 to n where n is number of input neurons and b =0.
- Step 2: For each input training vector and target output set pair (S, t) perform Steps 3-6.
- Step 3: Set activation for input units with the input vectors.
  - $x_i = S_i$ (i= 1to n)
- Step 4: Set activation for output unit with the output neuron
  - y=t
- Step 5: Adjust the weight by apply Hebb rule
  - $w_i(new) = w_i\ (old) + x_i y$ for i = 1 to n.
- Step 6: Adjust the bias
  - b(new) = b(old) + y

Problems - 1

- Apply the Hebb Net to the training the pattern that define AND function with bipolar inputs and targets.
- Solution: The training patterns are given in table:
- The weight change is calculated using
- $\Delta w_i = x_i\, y$ and $\Delta b = y$

X1	X2	Y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Problem 1…

Input			Target	Weight Changes			Weights		
(x_1	x_2	b)	y	Δw_1	Δw_2	Δb	w_1	w_2	B
							Initial (0	0	0)
1	1	1	1	1	1	1	1	1	1
1	−1	1	−1	−1	1	−1	0	2	0
−1	1	1	−1	1	−1	−1	1	1	−1
−1	−1	1	−1	1	1	−1	2	2	−2

Problem

1. Classify the two dimensional input patterns (representing letters) using Hebb rule (The T-C Problem).
2. Classify the two dimensional input patterns (representing letters) using Hebb rule (The I- J Problem).
3. Apply the Hebb net to the training patterns that define XOR function input and target.

Problem

a) Using the Hebb rule, find the weights required to perform the following classifications: Vectors (1 1 1 1) and (-1 1 -1 -1) are the member of class (with target value 1); vectors (1 1 1 -1) and (1 -1 -1 1) are not member of class (with target value -1).

b) Using each of training x vectors as input, test the response of the net.

Memory Based Learning

- Past experiences are stored in memory of correctly classified input-output examples
  - retrieve and analyze “local neighborhood”
  - $(xi, ti)^{N}{}_{i}$ where x_i is the input vector and t_j is the desired response.
- Essential Ingredient
  - Criterion used for defining local neighbor
  - Learning rule applied to the training examples

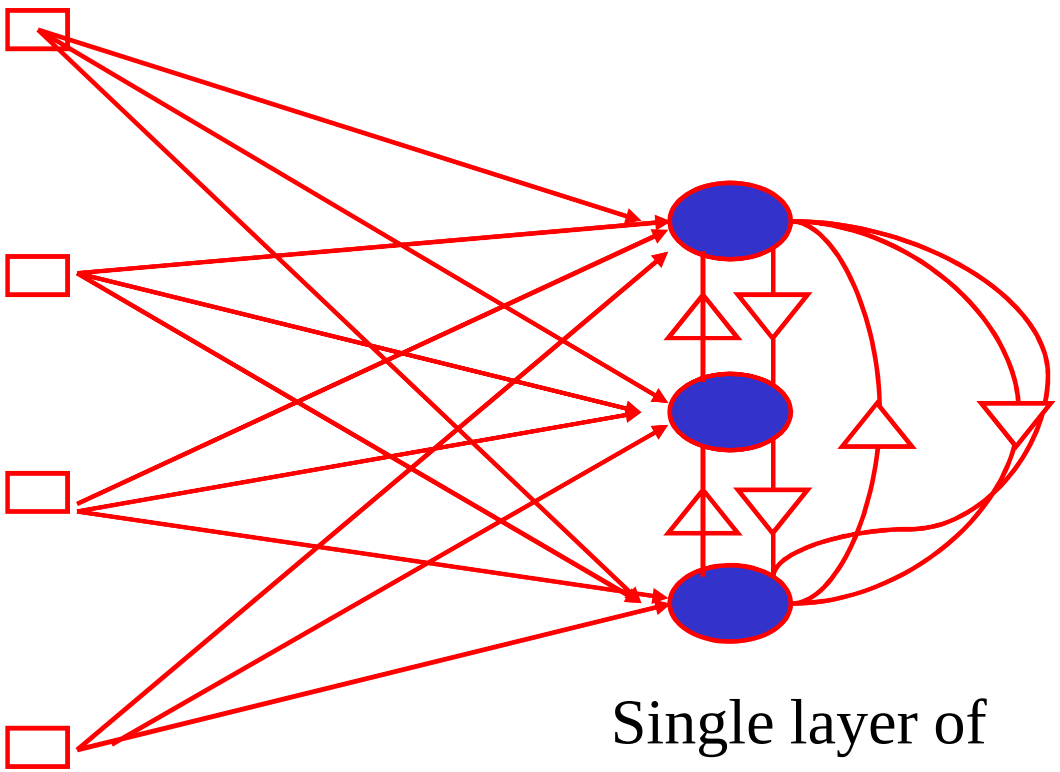
Memory Based Learning…

- Most widely used memory based learning is the Nearest Neibhbor Rule.
- Where the local neighborhood is defined as the training example that lies in the immediate neighborhood of the test vector x
- Nearest Neighbor Rule (NNR)
  - the vector $X'_n \in \{ X_1, X_2, \ldots, X_N \}$ is the nearest neighbor of X_{test} if
  - X'_n is the class of X_{test}

Competitive Learning

- Output neurons of NN compete to became active
- Only single neuron is active at any one time
  - salient feature for pattern classification
- Basic Elements
  - A set of neurons that are all same except synaptic weight distribution
    - responds differently to a given set of input pattern
  - A Limits on the strength of each neuron
  - A mechanism to compete to respond to a given input
    - winner-takes-all
- Neurons learn to respond specialized conditions
  - become feature detectors

Competitive NN…



Feedforward connection is excitatory

Lateral connection (▽) is inhibitory - lateral inhibition

Single layer of output neurons

layer of source Input

Competitive Learning…

Output signal

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

- W_{kj} denotes the synaptic weight

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

- Competitive Learning Rule

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

- If neuron does not respond to a particular input, no learning takes place

Competitive Learning…

- *x* has some constant Euclidean length and

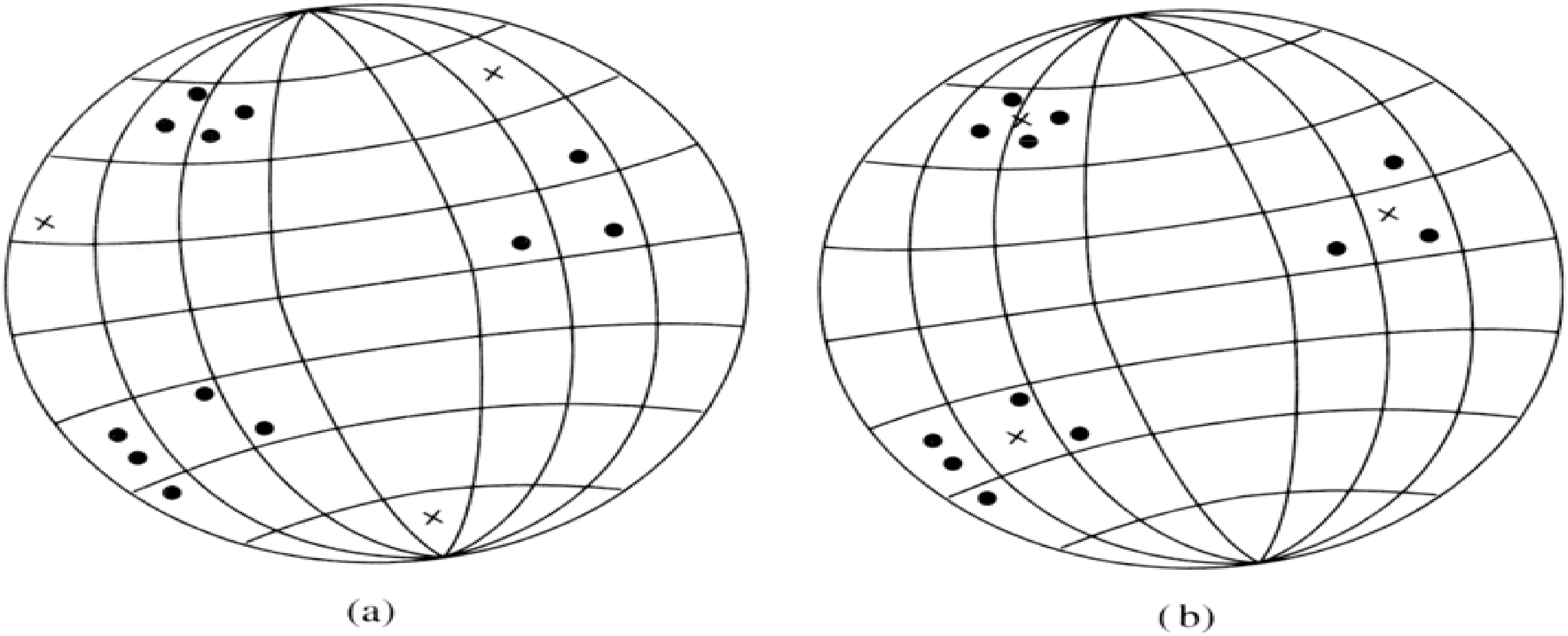


FIGURE 2.5 Geometric interpretation of the competitive learning process. The dots represent the input vectors, and the crosses represent the synaptic weight vectors of three output neurons. (a) Initial state of the network. (b) Final state of the network.

- perform *clustering* thru competitive learning

Credit-Assignment Problem

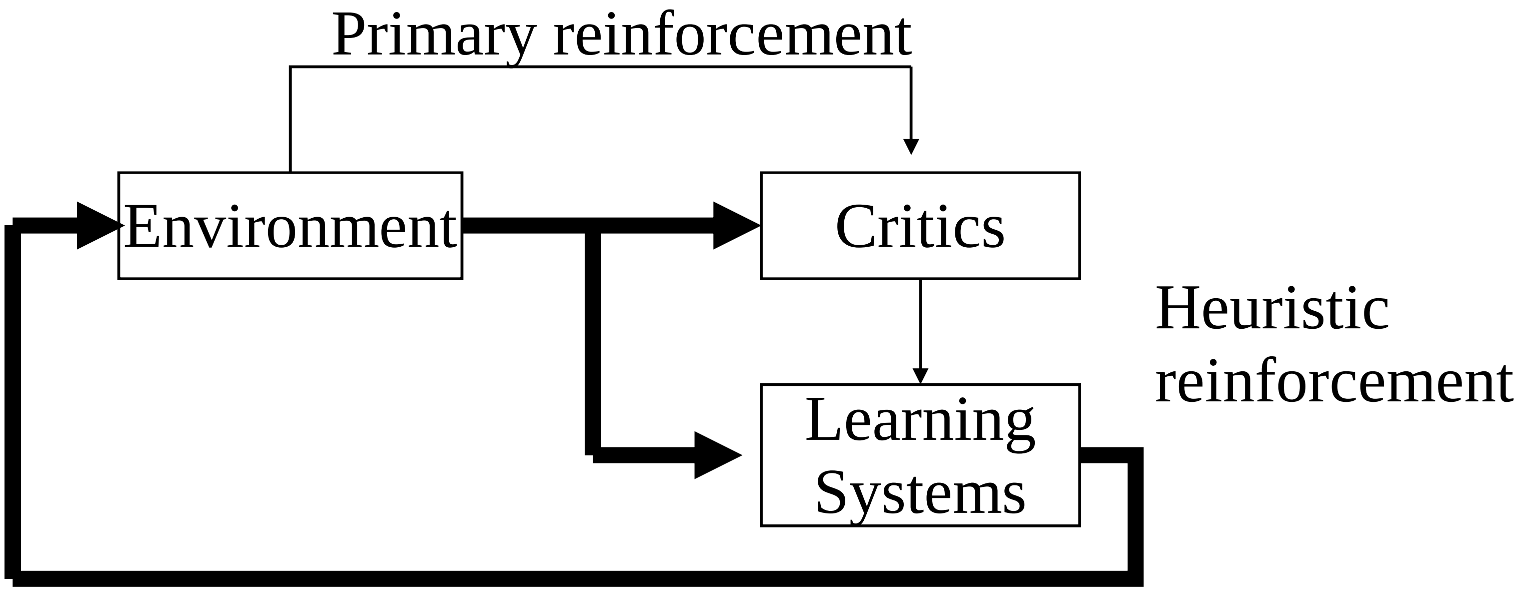
- Problems of assigning credit or blame for overall outcomes to each of internal decisions
- Two sub-problems
  - assignment of credit to actions
    - o temporal credit assignment problem : time instance of action
    - o many actions are taken; which action is responsible the outcome
  - assignment of credit to internal decisions
    - o structural credit assignment problem : internal structure of action
    - o multiple components are contributed; which one do best
- CAP occurs in Error-correction learning
  - in multiple layer feed-forward NN
  - How do we credit or blame for the actions of hidden neurons ?

Learning with Teacher

- Supervised learning
- Teacher has knowledge of environment to learn
- input and desired output pairs are given as a training set
- Parameters are adjusted based on error signal step-by-step
- System performance measure
  - mean-square-error
  - sum of squared errors over the training sample
  - visualized as error surface with parameter as coordinates
- Move toward to a minimum point of error surface
  - may not be a global minimum
  - use gradient of error surface - direction of steepest descent
- Good for pattern recognition and function approximation

Learning without Teacher

- Reinforcement learning
  - No teacher to provide direct (desired) response at each step
    - example : good/bad, win/loose
  - must solve temporal credit assignment problem
    - since critics may be given at the time of final output



Unsupervised Learning

- Self-organized learning
  - No external teacher or critics
  - Task-independent measure of quality is required to learn
  - Network parameters are optimized with respect to the measure
  - competitive learning rule is a case of unsupervised learning

Learning Tasks

- Pattern Association
- Pattern Recognition
- Function Approximation
- Control
- Filtering
- Beamforming

Stability and Convergence

- Stability refers to the equilibrium behavior of the activation state of a neural network.
- Convergence refers to the adjustment behavior of the weights during learning, which will eventually lead to minimization of error between the desired and actual outputs.

Recall and Adaption

- The weights are adjusted to store the information in a given pattern or a pattern pair.
- During performance, the weight changes are suppressed, and the input to the network determines the output activation x_i or the signal value s_i
- This operation is called recall of the stored information.
- The recall techniques can be different for feed forward and feed back.