Learning Process-2

By

Mausam Chouksey

Mathematical Models of Hebbian Learning

Consider a synaptic weight w_{kj} of neuron k.

The respective presynaptic (input) signal is denoted by x_j.

The postsynaptic (output) signal is denoted by y_k.

The change (update) of w_{kj} at time step n has the general form

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

where $F(y, x)$ is a function of both postsynaptic and presynaptic sig- nals.

Consider two specific forms of the general Hebbian learning rule.

Standard Hebbian learning rule:

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n)$$

Here η is again the learning rate or parameter.
Repeated application of the input (presynaptic) signal x_j leads to an increase in the output signal y_k.
Finally this leads to an *exponential growth* and saturation of the weight value.

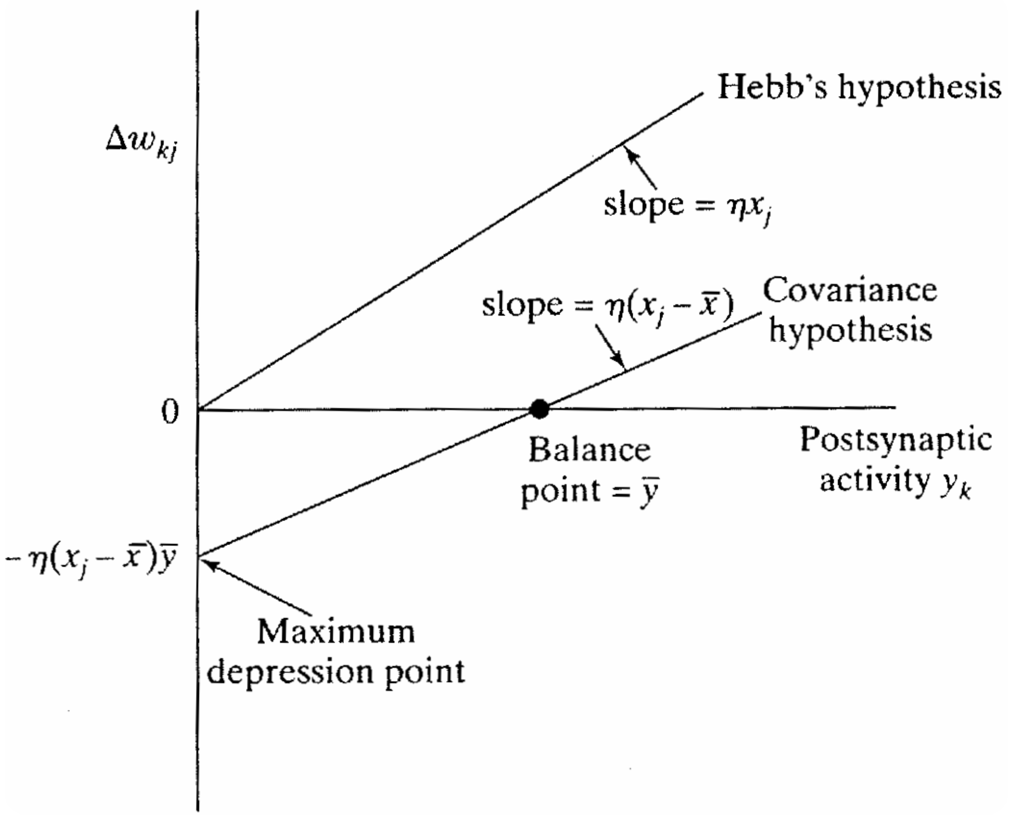
Covariance Hebbian rule:

$$\Delta w_{kj}(n) = \eta[x_j(n) - m_x][y_k(n) - m_y]$$

Here m_x and m_y are time averages of the presynaptic input signal x_j and postsynaptic output signal y_k, respectively.

Covariance rule can converge to a nontrivial state
$x_j = m_x$, $y_k = m_y$
The synaptic strength can both increase and decrease.

Cont'd…



- Standard Hebbian rule and the covariance rule.
- In both cases, the weight update depends on the output signal y_k linearly.
- Hebbian learning has strong physiological evidence.

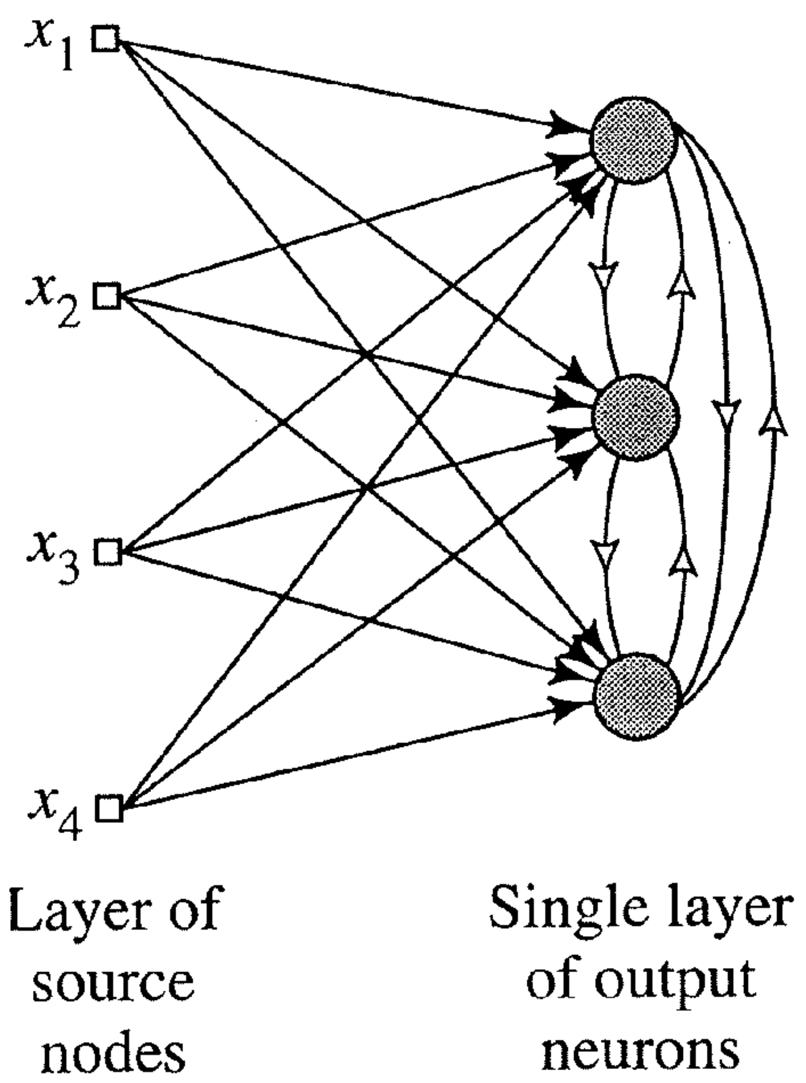
Competitive Learning

- In competitive learning, the neurons in the output layer compete to become active (fired).
- Only a single output neuron is active at any one time.
- In Hebbian learning, several output neurons may be active simulta- neously.
- Competitive learning is highly suitable for finding relevant features for classification tasks.

Cont’d…

- **Three basic elements** of a competitive learning rule:
  1. A set of similar neurons except for some randomly distributed synaptic weights.
     - Therefore the neurons *respond differently* to input signals.
  2. A *limit* imposed on the strength of each neuron.
  3. A competing mechanism for the neurons.
  - Only one output neuron has the right to respond to an input signal.
  - The winner of the competition is called a *winner-takes-all neu-ron*.
- As a result of competition, the neurons become specialized.
- They respond to certain type of inputs, becoming *feature detectors* for different input classes.

Simplest form of a competitive neural network



Cont'd...

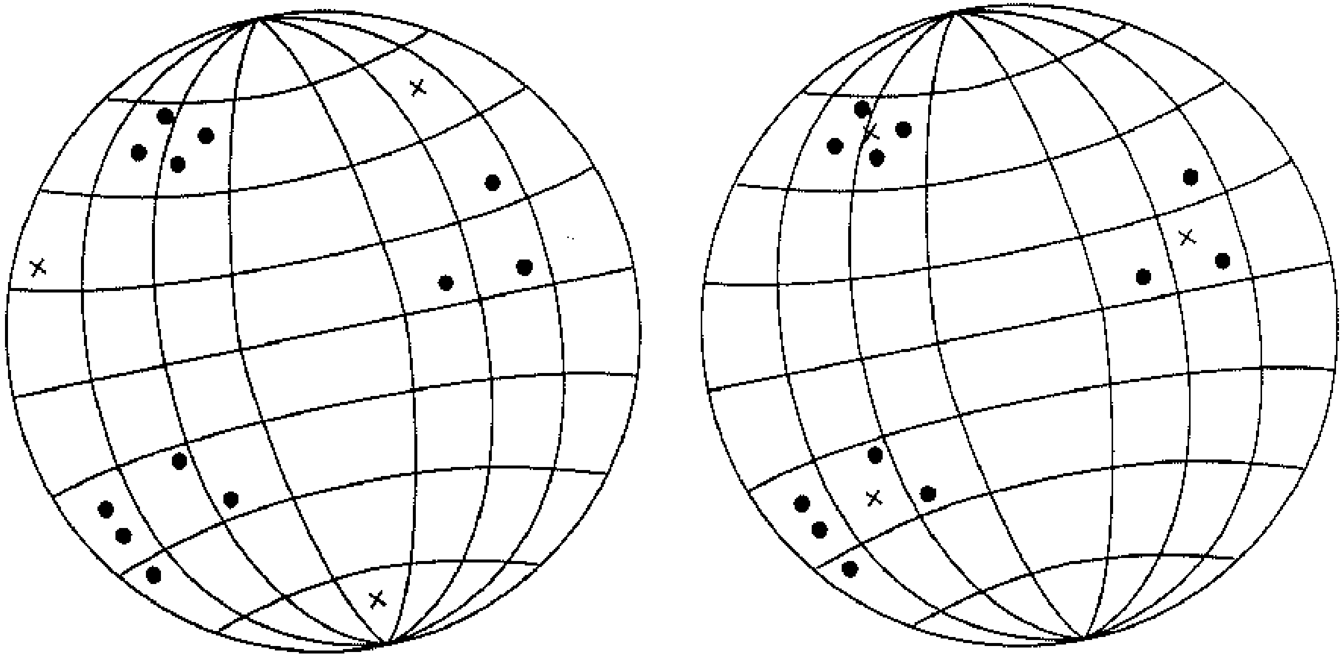
- Feedback connections between the competing output neurons perform
- *lateral inhibition*.
- Each neuron tends to inhibit the neuron to which it is laterally con- nected.
- A neuron k is the winning neuron if its induced local field v_k for a given input pattern $\mathbf{x}$ is the largest one.
- Mathematically, the output signal
- $y_k = 1$, if $v_k > v_j$ for all $j, j \neq k$.
- For other than the winning neuron, the output signal $y_k = 0$.
- The local field v_k represents the combined action of all the forward and feedback inputs to neuron k.
- Typically, all the synaptic weights w_{kj} are positive.

Cont'd...

- Feedback connections between the competing output neurons perform
- *lateral inhibition*.
- Each neuron tends to inhibit the neuron to which it is laterally con- nected.
- A neuron k is the winning neuron if its induced local field v_k for a given input pattern $\mathbf{x}$ is the largest one.
- Mathematically, the output signal
- $y_k = 1$, if $v_k > v_j$ for all j, $j \neq k$.
- For other than the winning neuron, the output signal $y_k = 0$.
- The local field v_k represents the combined action of all the forward and feedback inputs to neuron k.
- Typically, all the synaptic weights w_{kj} are positive.

Cont'd…

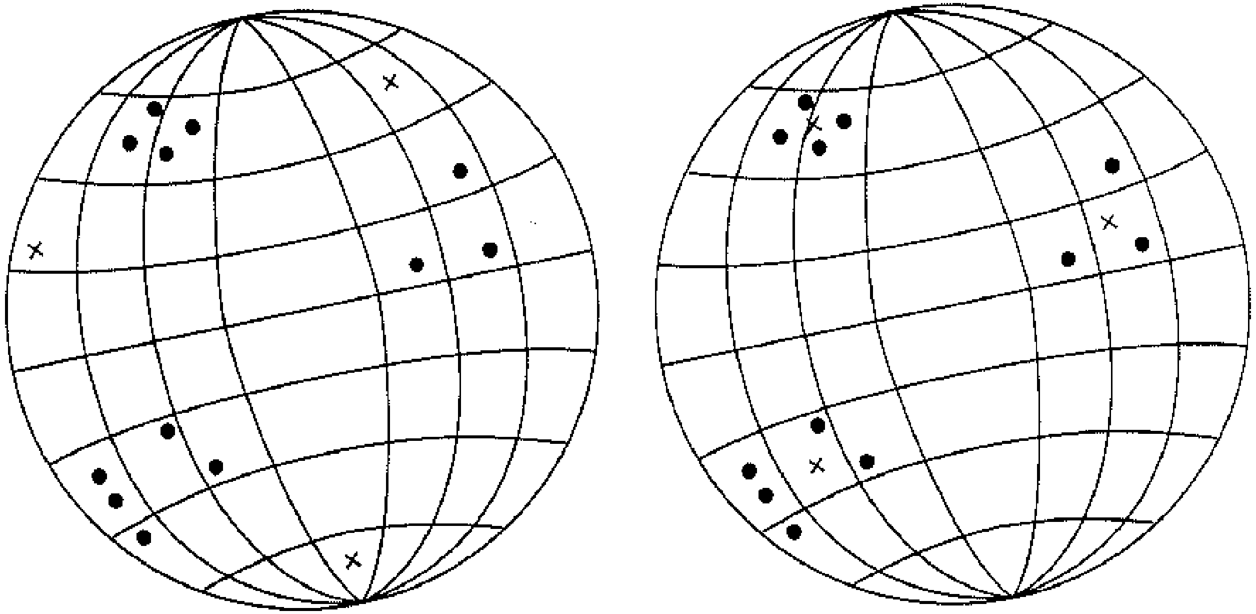
- Here both the input vectors **x** and the weight vectors $\mathbf{w}_k$ are scaled to have unit length (Euclidean norm).
- Then they are points on the surface of an N-dimensional hypersphere (assuming N-dimensional vectors).
- Initial state (Fig. a) shows three clusters of data points (black dots) and initial values of three weight vectors (crosses).



(a) (b)

Cont’d…

- Figure b shows a typical final state of a network resulting from com- petitive learning.
- The weight vectors have moved to the gravity centers of clusters.
- In more difficult cases, competitive learning algorithms may fail to find stable clusters.



(a) (b)

Boltzmann Learning

- In a Boltzmann machine the neurons constitute a recurrent structure, and they operate in a binary manner (either "on" state or in "off" state)
- The machine is characterized by an energy function, E, the value of which is determined by the particular states occupied by the individual neurons of the machine, as shown by

$$E = -\frac{1}{2}\sum_{j}\sum_{\substack{k \\ j \neq k}} w_{kj} x_k x_j$$

- where xj is the state of neuron j and wkj is the synaptic weight connecting neuron j to neuron k and $j \neq k$ means no self-feedback.

Cont'd…

- The machine operates by choosing a neuron k at some step of the learning process, then flipping the state of neuron k from state x_k state to $-x_k$ at some temperature T with probability

$$P\left(x_k \rightarrow -x_k\right) = \frac{1}{1 + exp\left(-\frac{\Delta E_k}{T}\right)}$$

Cont'd…

- Let $\rho_{k\mathrm{j}}^{+}$ denote the *correlation* between the states of neurons j and *k*, with the network in its clamped condition. Let $\rho_{k\mathrm{j}}^{-}$ denote the *correlation* between the states of neurons *j* and *k* with the network in its free-running condition.

- The Δw_{kj} change applied to the synaptic weight w_{kj} from neuron j to neuron *k* is defined by

$$\Delta\omega_{kj} = \eta(\rho_{k_j}^{+} - \rho_{k_j}^{-})$$