Hebbian Learning and Cohen-Grossberg Learning

Sayeda Suaiba Anwar
Lecturer, Department of CSE
East Delta University

Hebbian Learning

Hebb Network was stated by Donald Hebb in 1949. According to Hebb's rule, the weights are found to increase proportionately to the product of input and output. It means that in a Hebb network if two neurons are interconnected then the weights associated with these neurons can be increased by changes in the synaptic gap.

Hebb's Law can be summarized as follows:

As A becomes more efficient at stimulating B during training, A sensitizes B to its stimulus, and the weight on the connection from A to B increases during training as B becomes sensitized to A.

Hebb's Law is too vague.

Hebbian Learning

* Drawbacks

Following Questions may arise:

- How much should a weight increase?
- How active does B need to be for training to occur?

Furthermore, there is no way for the weights to decrease. In theory, they can increase to infinity.

Corrections to Hebb's Law involve normalizing the weights to force them to stay within limited bounds and forcing them to both increase and decrease to retain the normalization.

Hebbian Learning

Neo Hebbian Learning put forward by Steven Grossberg, who developed an explicit mathematical statement for the weight change law of the form

$$w_{AB}^{new} = w_{AB}^{old}(1 - \alpha) + \beta x_B x_A \quad \text{.........(9.1.1)}$$

forgetting constant

Learning constant

Range(0-1)

Hebbian Learning

where w_{AB} is the weight on the synapse connecting neuron A with neuron B, α is the "forgetting" term that accounts for the fact that biological systems forget slowly with time, and β is a "learning" constant that accounts for simultaneous firing of neurons A and B. The right-hand term is called the "Hebbian learning term" because it ties the learning rate to the product of the neuron outputs. Hebbian learning is characterized by the product of two neuron activities. Hence, anytime such a product appears in an equation, Hebbian learning is involved. Generally, both α and β are in the range between 0 and 1.

Hebbian Learning

between 0 and 1.

If we rearrange equation (9.1-1) and put it into the form

$$\frac{w_{AB}^{\text{new}} - w_{AB}^{\text{old}}}{\Delta t} = \frac{\Delta w_{AB}}{\Delta t} = \frac{dw_{AB}}{dt} = -\alpha w_{AB}^{\text{old}} + \beta x_B x_A \tag{9.1-2}$$

and consider only the terms involving w_{AB}, it is clear that the forgetting term involves a slow exponential decay with time constant α. Even so, neo-Hebbian learning does not permit the weights to decrease when the neuron outputs decrease.

To overcome this problem, Grossberg introduced differential Hebbian learning. It has the same mathematical form as Hebbian learning of equation (9.1-1) except that it uses the product of rates of change in the outputs for the neurons A and B, as given

$$w_{AB}^{\text{new}} = w_{AB}^{\text{old}}(1 - \alpha) + \beta \frac{dx_B}{dt} \frac{dx_A}{dt} \tag{9.1-3}$$

Cohen Grossberg Learning
- Pavlov's Experiment

1. **Observational Conditioning**:
   - Involves the copying the actions of others.
   - "Monkey see, Monkey do"
2. **Operational Conditioning:**
   - Involves an action and a response.
   - "Push button, get food"
3. **Classical Conditioning**:
   - Involves a stimulus and a response
   - Pavlov's experiment

Cohen-Grossberg Learning comes from an attempt to mathematically explain the observations from psychological conditioning experiments carried out by **Pavlov.**

Cohen Grossberg Learning
- Pavlov's Experiment

Table 9.1 Stimuli and corresponding responses in Pavlov's experiment

Stage	*Stimulus*	*Response*
1.	Unconditional Stimulus ====> (Plate of Food)	Unconditioned Response (Dog Salivates)
2.	Unconditional Stimulus (Plate of Food) *plus* ====> Conditioned Stimulus (Bell Rings)	Conditioned Response (Dog Salivates)
3.	Conditioned Stimulus ====> (Bell Rings)	Conditioned Response (Dog Salivates)

Cohen Grossberg Learning
- Instars and Outstars

Every Neuron receives hundreds or thousands of inputs through its own synapses from the axon collaterals of other neurons. Schematically, this can be represented as a “star” with radially inward paths called the instar. Indeed, every artificial neuron in a neural network is **an instar.**

Every Neuron also sends out hundreds or thousands of collaterals which branch of from the main axon and go to the synapses of other neurons. Schematically, this also can be represented as a “star” with radially outward paths called the outstar. Every neuron is effectively **an outstar.**

Cohen Grossberg Learning
-   Instars and Outstars

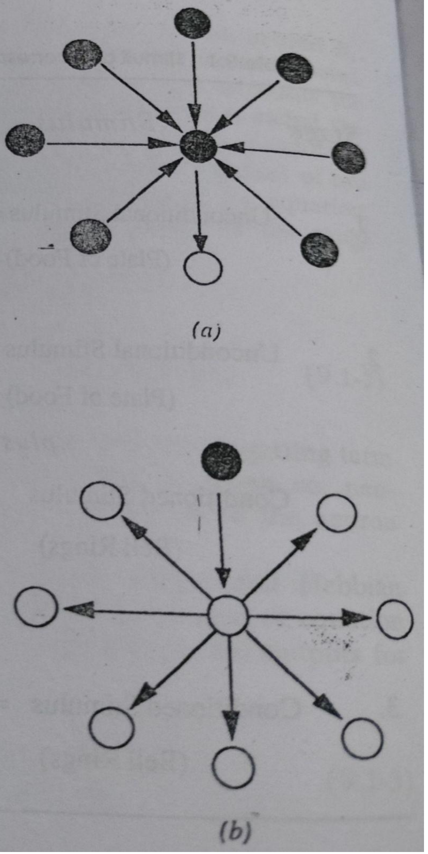

Figure 9.1 Graphical representations of an "instar" (*a*) and an "outstar" (*b*).

Development of Cohen Grossberg Learning Equations - Instar Activity

Let us consider Pavlovian Learning from the perspective of an instar. The activity of the instar processing element or neuron has a number of requirements:

1. The activity must grow when there is an external stimulus.
2. It must rapidly decrease if it is no longer stimulated from outside.
3. It must respond to stimuli from other neurons in the network.

Development of Cohen Grossberg Learning Equations - Instar Activity

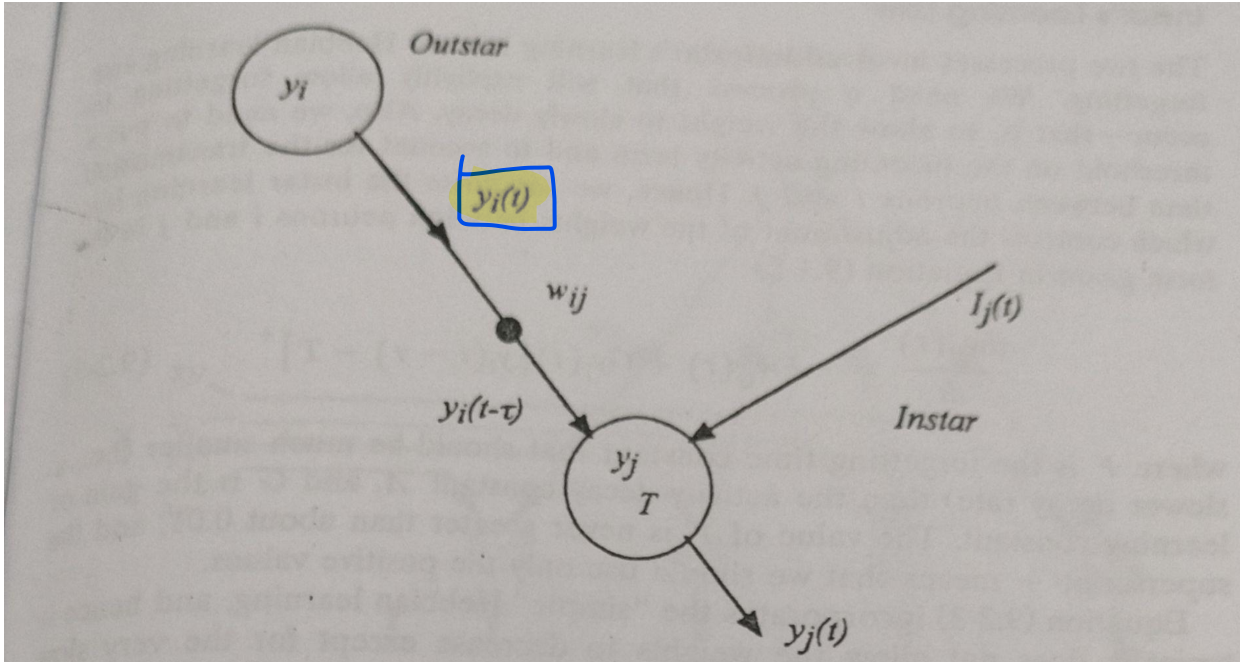

Figure 9.2 An instar y_j receives a signal from the outstar y_i through a weight w_{ij}. (T is the threshold for the incoming signal to node y_j, and τ is the time required for the signal to travel from y_i to y_j.)

Development of Cohen Grossberg Learning Equations - Instar Activity

The activity of instar yj(t) can be represented by a differential equation

$$\frac{dy_j(t)}{dt} = -Ay_j(t) + I_0 + B\sum_{i=1}^{n} w_{ij}y_i(t) \qquad (9.2\text{-}1)$$

forgetting

learning

Development of Cohen Grossberg Learning Equations - Instar Activity

where $y_i(t)$ is the activity of the ith neuron, I_0 is the external stimulus, w_{ij} is the weight between the ith and jth neurons, and A and B are constants. The first term on the right-hand side of equation (9.2-1) allows the activity of the instar to decrease exponentially with a time constant A when it is no longer stimulated by I or inputs from other neurons. The second term I_0 corresponds to an external stimulus, and the third term represents the stimuli from the n neurons in the network. We need to allow for signals received at neuron j that were actually generated in the neuron i at some previous time τ ago and transmitted to neuron j, where τ is the "average" transmission time to from neuron i to neuron j. We also need to put a threshold (T) on the intraneuron inputs so that random noise will not interfere with the network's operation. Hence, we can modify equation (9.2-1) as follows:

$$\frac{dy_j(t)}{dt} = -Ay_j(t) + I_0(t) + B\sum_{i=1}^{n} w_{ij}[y_i(t-\tau) - T]^+ \qquad (9.2\text{-}2)$$

where the superscript + means that only positive values are used.

Development of Cohen Grossberg Learning Equations - Instar's Learning Law

The two processes involved in instar's learning law are Hebbian Learning and Forgetting.

A process is needed that will explicitly allow forgetting to occur (to allow the weight to slowly decay).

We need to put a threshold on the incoming activity term and to account for the transmission time between neurons i and j.

So the instar learning law which controls the adjustment of the weights between neurons i and j can be written as -

$$\frac{dw_{ij}(t)}{dt} = -Fw_{ij}(t) + Gy_j(t)\left[y_i(t-\tau) - T\right]^+ \qquad (9.2\text{-}3)$$

Development of Cohen Grossberg Learning Equations - Instar's Learning Law

Where F is the forgetting time constant that should be much smaller (i.e a slower decay rate) than the activity decay constant A , and G is the gain or learning constant. The value of F is never greater than about 0.01 and the superscript + means that we should use only positive values.