Intelligent Control

ANN Learning Methods

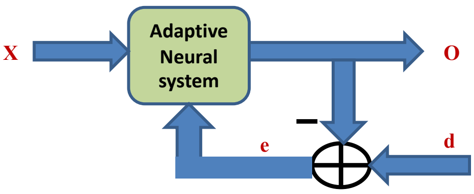
Hassan Bevrani

Professor, University of Kurdistan

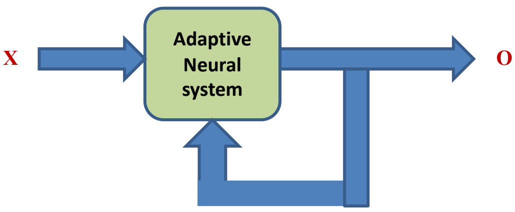
Fall 2023

Learning Methods: Review

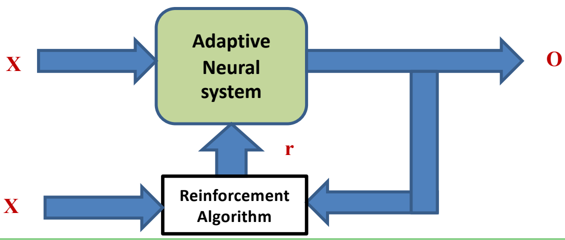
1. **Supervised learning**



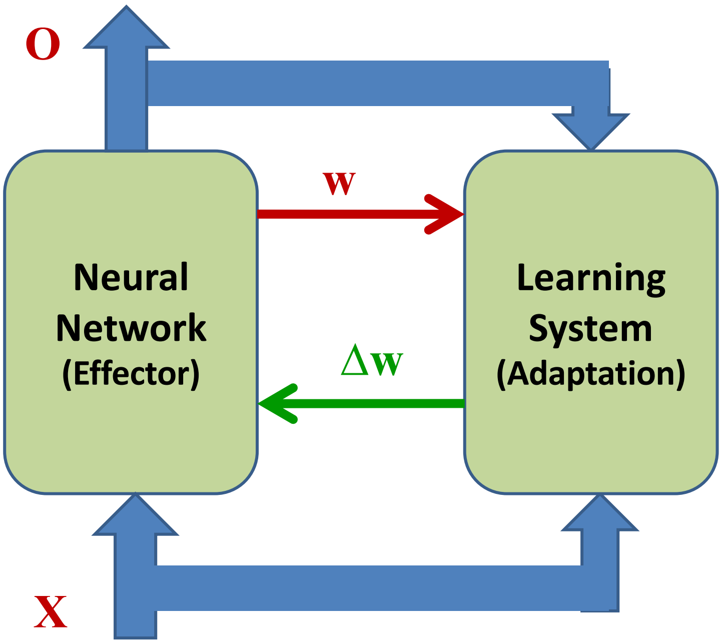
2. **Unsupervised learning**



3. **Reinforcement learning**



Learning Methods: Review

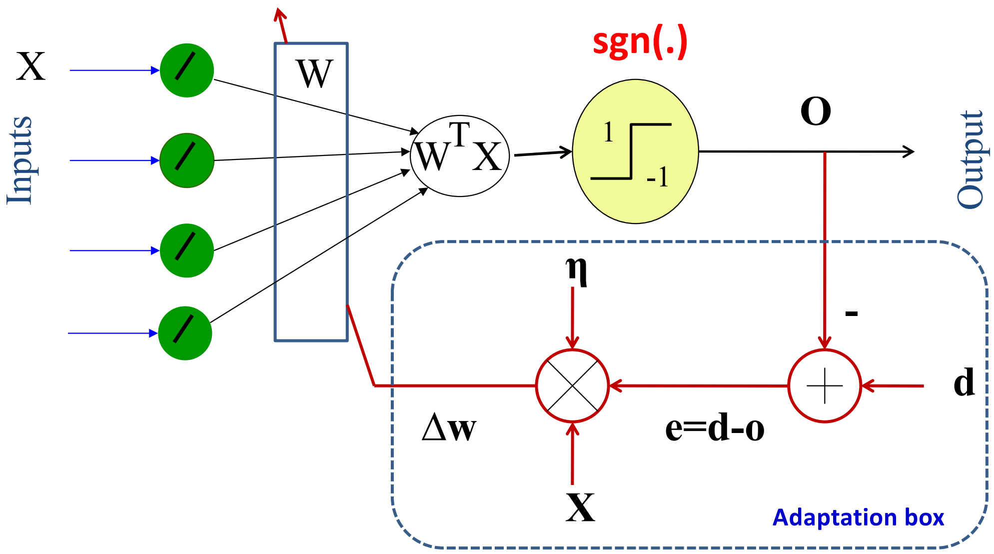


Supervised Learning Method

1. Perceptron learning
2. Widrow-Hoff learning
3. Correlation learning
4. Back-propagation learning
5. Generalized learning
6. Specialized learning

Perceptron (single layer) learning algorithm

Learning rule: Δw=ηeX ➡ W(k+1)=W(k)+ηeX (Rosenblatt, 1958)



Example

Perceptron (single layer) learning algorithm

1. Randomly initialize all the networks weights.
2. Apply inputs and find outputs (feed-forward).
3. Compute the errors.
4. Update each weight as

$$w_{ij}(k+1) = w_{ij}(k) + \eta\, x_i(k)\, e_j(k)$$

5. Repeat steps 2 to 4 until the errors reach the satisfactory level.

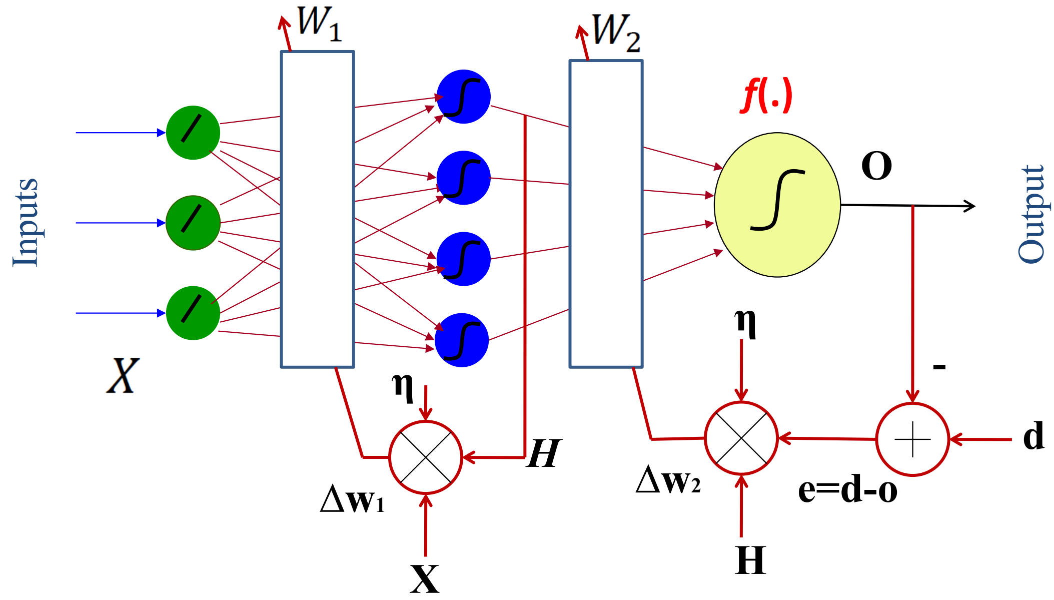
Widrow-Hoff learning

Learning rule: $\Delta w = \eta e X$ ⇨ **$W(k+1) = W(k) + \eta e X$**

By **Widrow and Hoff (~1960)**

Adaptive **lin**ear **e**lements for signal processing (**Adaline** algorithm) The same architecture of perceptrons

Widrow-Hoff learning algorithm



Example

Learning Mechanism

- Try to reduce the mean squared error (**MSE**) between the net input and the desired output.
- The MSE is a performance index for evaluation of learning algorithms

$$J(x) = E[e^T e] = E[(d\text{-}o)^T(d\text{-}o)]$$

or

$$E = \frac{1}{2}\|\mathbf{o} - \mathbf{o}_d\|_2^2$$

Learning Mechanism

- Delta rule
  - Let $i_j = (i_{0,j}, i_{1,j}, \ldots, i_{n,j})$ be an input vector with desired output d_j
  - The squared error

$$E = (d_j - net_j)^2 = (d_j - \sum_l w_l i_{l,j})^2$$

- Its value determined by the weights w_l
  - Modify weights by gradient descent approach

$$\begin{aligned}\frac{\partial E}{\partial w_k} &= 2(d_j - \text{net}_j)\frac{\partial}{\partial w_k}(\text{-net}_j) \\ &= -2(d_j - \text{net}_j)\ i_{k,j}.\end{aligned}$$

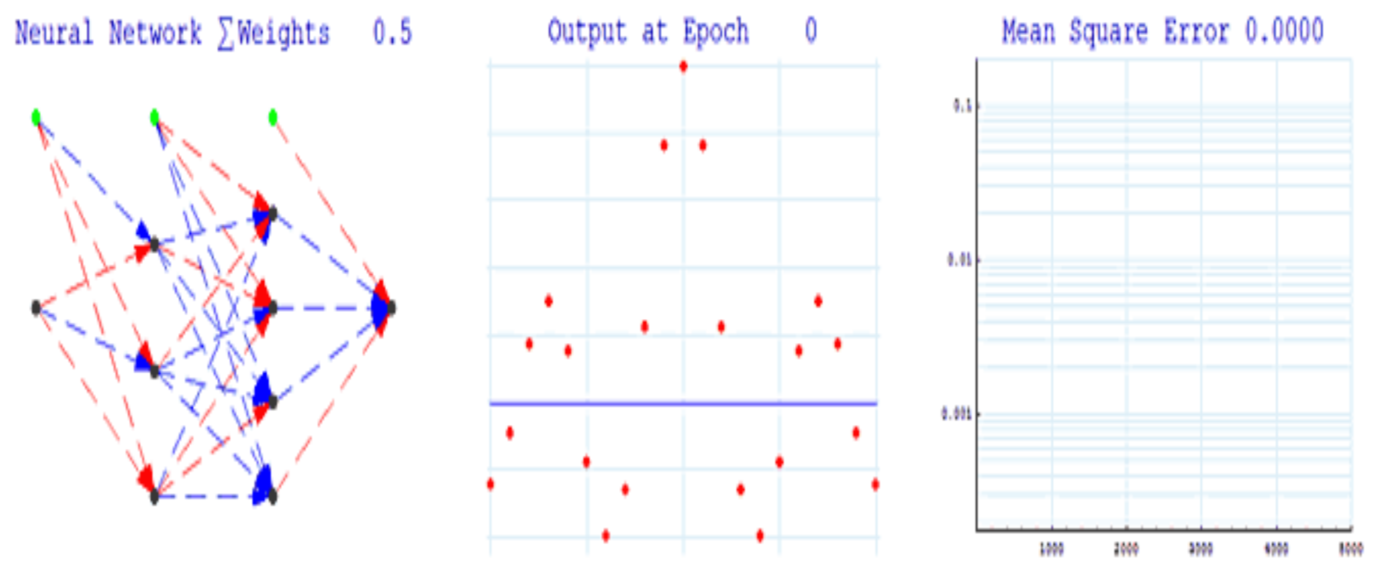
Learning Mechanism

- Change weights in the opposite direction of $\partial E / \partial w_k$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

$$\Delta w_k = \eta (d_j - \sum_l w_l i_{l,j}) \cdot i_{k,j} = \eta (d_j - net_j) \cdot i_{k,j}$$

Learning Mechanism



Learning Mechanism

– Weights will be dynamically changed

– $\boldsymbol{E}$ decreases until the system reaches a state with (local) minimum $\boldsymbol{E}$ (a small change of any w_i will cause $\boldsymbol{E}$ to increase)

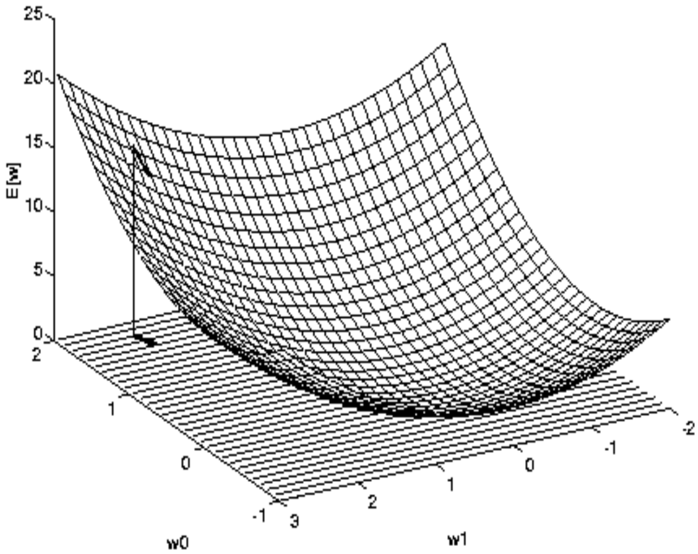
– At a local minimum $\boldsymbol{E}$ state, $\partial \boldsymbol{E} / \partial \boldsymbol{w}_i = 0 \;\; \forall \boldsymbol{i}$, but $\boldsymbol{E}$ is not guaranteed to be zero ($net_j = d_j$)

- This is why Adaline usually uses nonlinear function rather than linear function

The error surface

We calculate the direction of steepest descent along the error surface by computing

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$



Learning Rate

- The learning rate (η) should be sufficiently small

Selection of learning rate

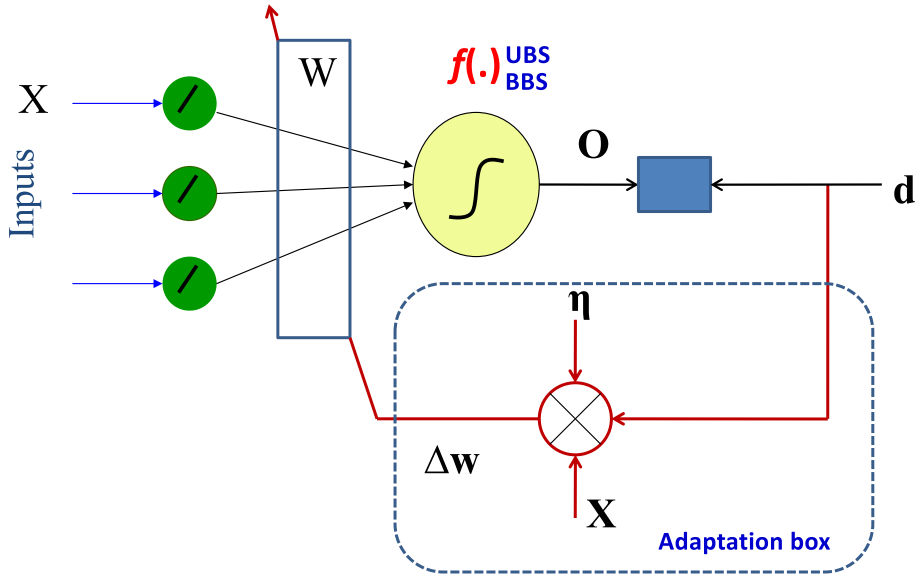
- If η is too large: risk of instability
- If η is too small (≈ 0): very slow to converge
- Common choice: $\eta = 1$.

Supervised Learning Modes

1. Perceptron learning
2. Widrow-Hoff learning
3. Correlation learning
4. Back-propagation learning
5. Generalized learning
6. Specialized learning

Correlation learning algorithm

Learning rule: Δw=ηdX ➡ **W(k+1)=W(k)+ ηdX**



Example

Unsupervised Learning Modes

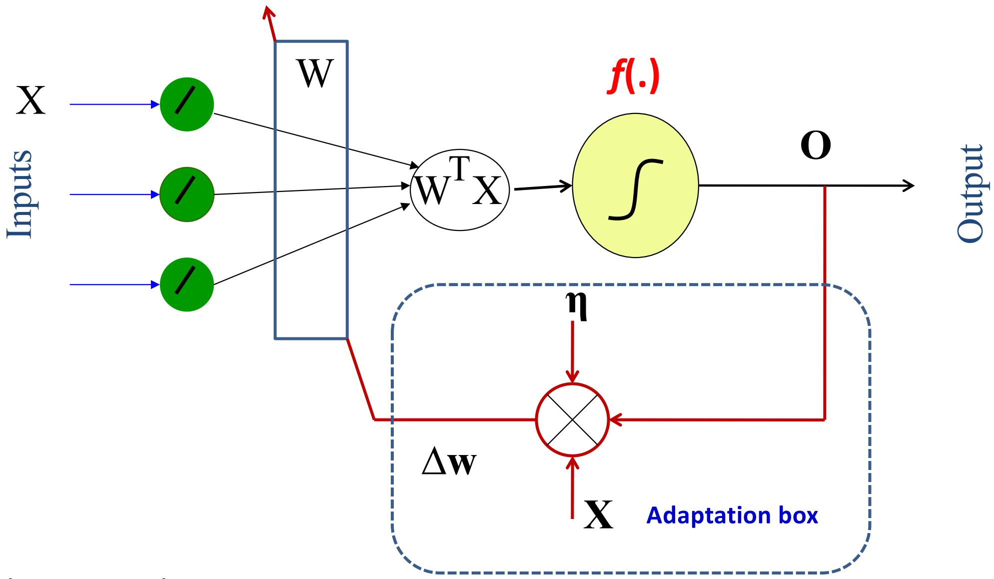
1. Hebbian learning

2. Crossberg & Corpenter learning

3. Kohonen learning

Hebbian learning algorithm

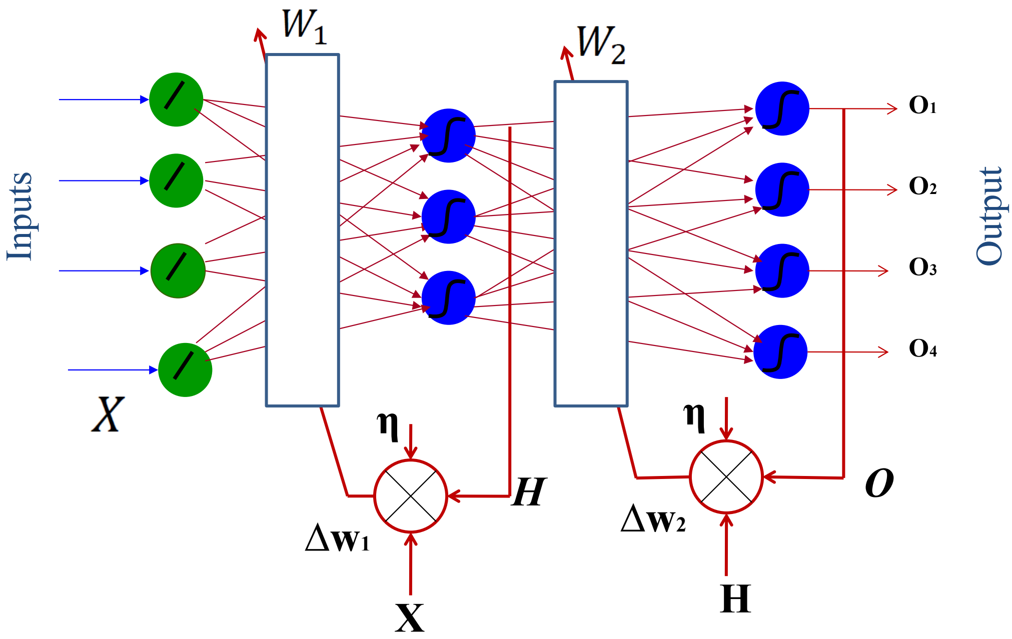
Learning rule: $\Delta w = \eta O X$ ⇨ $W(k+1) = W(k) + \eta O X$ (1949)



η : constant learning = learning rate, 0< η <1

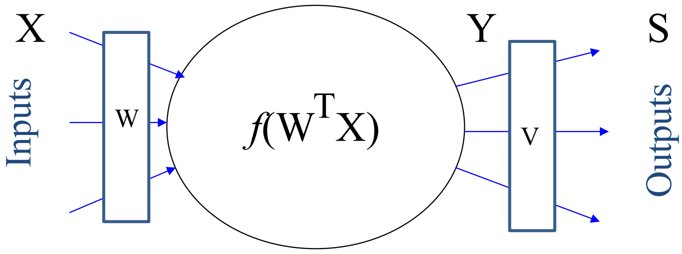
Hebbian learning algorithm

Example:



Instar-Outstar Structure

(Grossberg & Carpenter 1974-82)



$$f(net) = \frac{1}{1+e^{-a.net}}$$

Learning rule: $\Delta w = \eta_1 (X - W)$

$\Delta v = \eta_2 (Y - V)$

Example

Thank you!