



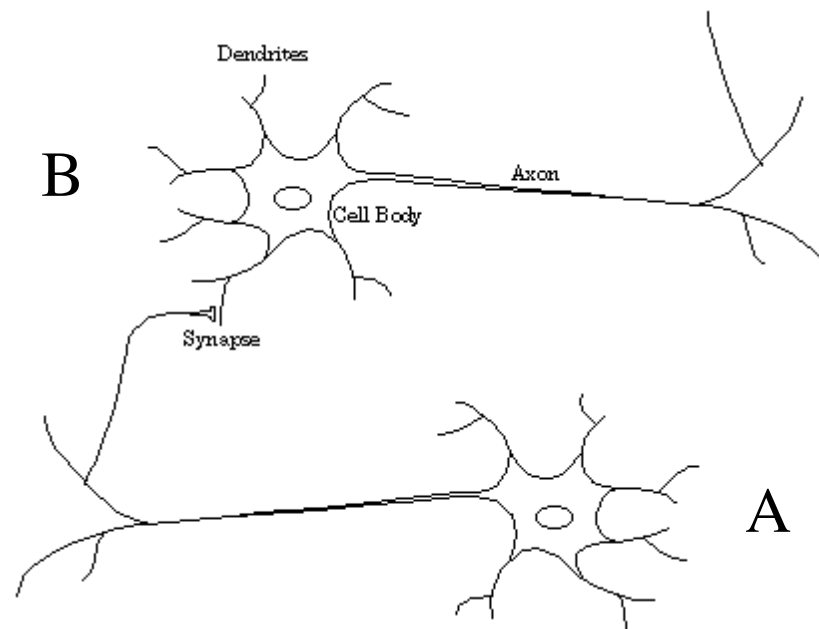
Associative Learning (Supervised Hebbian Learning)

Hebb's Postulate

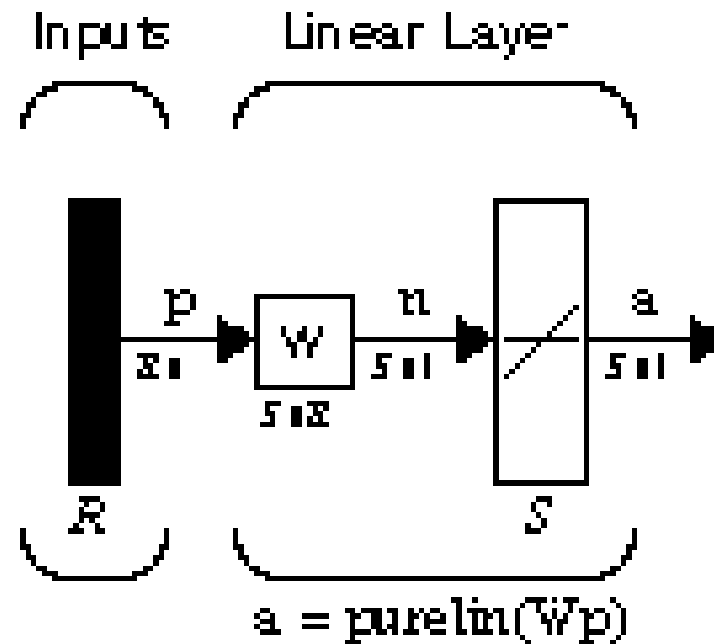


“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.”

D. O. Hebb, 1949



Linear Associator



$$a = Wp \quad a_i = \sum_{j=1}^R w_{ij} p_j$$

Training Set:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$$

Hebb Rule



$$w_{ij}^{new} = w_{ij}^{old} + \Delta f_i(a_{iq})g_j(p_{jq})$$

↑ ↑
 Postsynaptic Signal Presynaptic Signal

Simplified Form:

$$w_{ij}^{new} = w_{ij}^{old} + \Delta a_{iq}p_{jq}$$

Supervised Form:

$$w_{ij}^{new} = w_{ij}^{old} + t_{iq}p_{jq}$$

Matrix Form:

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$$

Batch Operation



$$\mathbf{W} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \square + \mathbf{t}_Q \mathbf{p}_Q^T = \sum_{q=1}^Q \mathbf{t}_q \mathbf{p}_q^T \quad (\text{Zero Initial Weights})$$

Matrix Form:

$$\mathbf{W} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \square & \mathbf{t}_Q \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \square \\ \mathbf{p}_Q^T \end{bmatrix} = \mathbf{T} \mathbf{P}^T$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \square & \mathbf{p}_Q \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \square & \mathbf{t}_Q \end{bmatrix}$$

Performance Analysis



$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \sum_{q=1}^Q \mathbf{t}_q \mathbf{p}_q^T \mathbf{p}_k = \sum_{q=1}^Q \mathbf{t}_q (\mathbf{p}_q^T \mathbf{p}_k)$$

Case I, input patterns are orthogonal.

$$\begin{aligned} (\mathbf{p}_q^T \mathbf{p}_k) &= 1 & q = k \\ &= 0 & q \neq k \end{aligned}$$

Therefore the network output equals the target:

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k$$

Case II, input patterns are normalized, but not orthogonal.

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k + \sum_{q \neq k} \mathbf{t}_q (\mathbf{p}_q^T \mathbf{p}_k)$$

Error



Banana	Apple	Normalized Prototype Patterns						
$\mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$	$\mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$	$\mathbf{p}_1 = \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}$	$\mathbf{t}_1 = \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$	$\begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$	$\mathbf{p}_2 = \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}$	$\mathbf{t}_2 = \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$	$\begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$

Weight Matrix (Hebb Rule):

$$\mathbf{W} = \mathbf{TP}^T = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} -0.5774 & 0.5774 & -0.5774 \\ 0.5774 & 0.5774 & -0.5774 \end{bmatrix} = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix}$$

Tests:

Banana $\mathbf{Wp}_1 = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} -0.6668 \end{bmatrix}$

Apple $\mathbf{Wp}_2 = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix} \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} 0.6668 \end{bmatrix}$

Pseudoinverse Rule - (1)



Performance Index: $\mathbf{W}\mathbf{p}_q = \mathbf{t}_q \quad q = 1, 2, \dots, Q$

$$F(\mathbf{W}) = \sum_{q=1}^Q \|\mathbf{t}_q - \mathbf{W}\mathbf{p}_q\|^2$$

Matrix Form:

$$\mathbf{W}\mathbf{P} = \mathbf{T}$$

$$\mathbf{T} = [\mathbf{t}_1 \ \mathbf{t}_2 \ \dots \ \mathbf{t}_Q] \quad \mathbf{P} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \dots \ \mathbf{p}_Q]$$

$$F(\mathbf{W}) = \|\mathbf{T} - \mathbf{W}\mathbf{P}\|^2 = \|\mathbf{E}\|^2$$

$$\|\mathbf{E}\|^2 = \sum_i \sum_j e_{ij}^2$$



$$\mathbf{WP} = \mathbf{T}$$

Minimize:

$$F(\mathbf{W}) = \|\mathbf{T} - \mathbf{WP}\|^2 = \|\mathbf{E}\|^2$$

If an inverse exists for \mathbf{P} , $F(\mathbf{W})$ can be made zero:

$$\mathbf{W} = \mathbf{TP}^{-1}$$

When an inverse does not exist $F(\mathbf{W})$ can be minimized using the pseudoinverse:

$$\mathbf{W} = \mathbf{TP}^+$$

$$\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$$



Hebb Rule

$$\mathbf{W} = \mathbf{TP}^T$$

Pseudoinverse Rule

$$\mathbf{W} = \mathbf{TP}^+$$

$$\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$$

If the prototype patterns are orthonormal:

$$\mathbf{P}^T \mathbf{P} = \mathbf{I}$$

$$\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T = \mathbf{P}^T$$



$$\begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} \mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_1 = [-1] \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} \quad \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_2 = [1] \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} \quad \mathbf{W} = \mathbf{TP}^+ = [-1 \ 1] \begin{bmatrix} \square & \square \\ \square & \square \\ \square & \square \end{bmatrix}^+$$

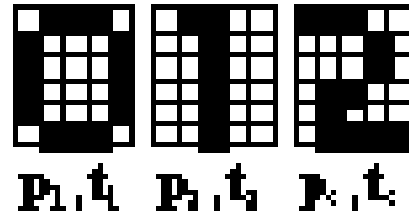
$$\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}^{-1} \begin{bmatrix} -1 & 1 & -1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -0.5 & 0.25 & -0.25 \\ 0.5 & 0.25 & -0.25 \end{bmatrix}$$

$$\mathbf{W} = \mathbf{TP}^+ = [-1 \ 1] \begin{bmatrix} -0.5 & 0.25 & -0.25 \\ 0.5 & 0.25 & -0.25 \end{bmatrix} = [1 \ 0 \ 0]$$

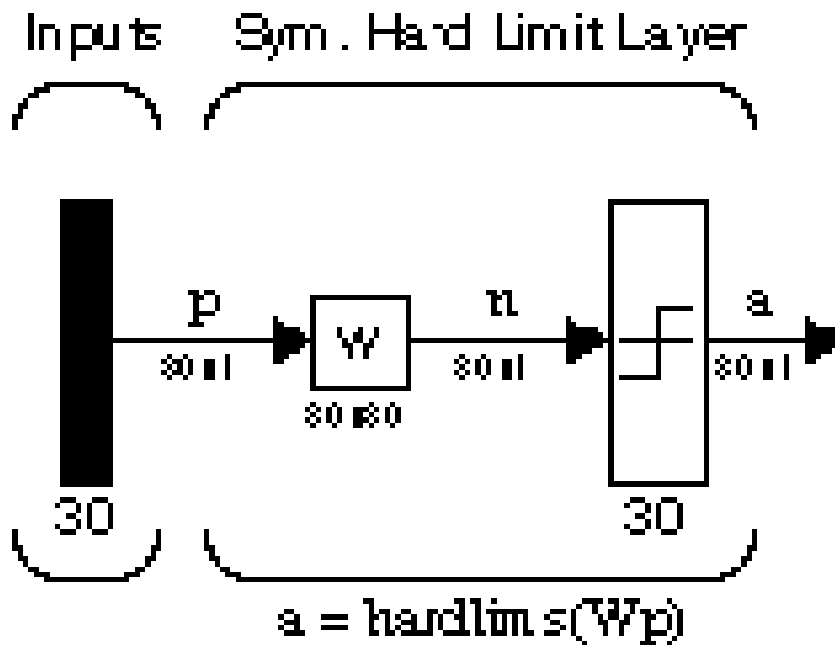
$$\mathbf{Wp}_1 = [1 \ 0 \ 0] \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = [-1]$$

$$\mathbf{Wp}_2 = [1 \ 0 \ 0] \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = [1]$$

Autoassociative Memory



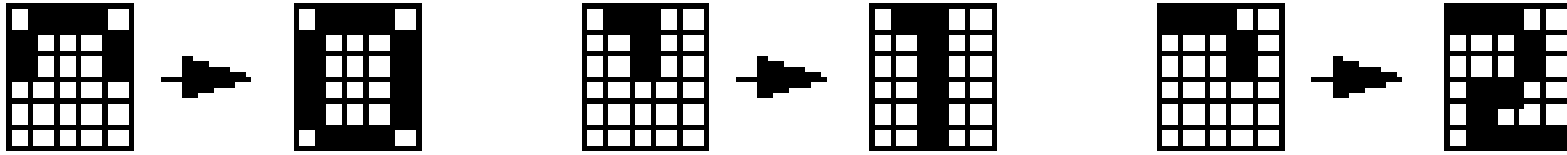
$$\mathbf{p}_1 = [-1 \ 1 \ 1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 0 \ 1 \ -1]^T$$



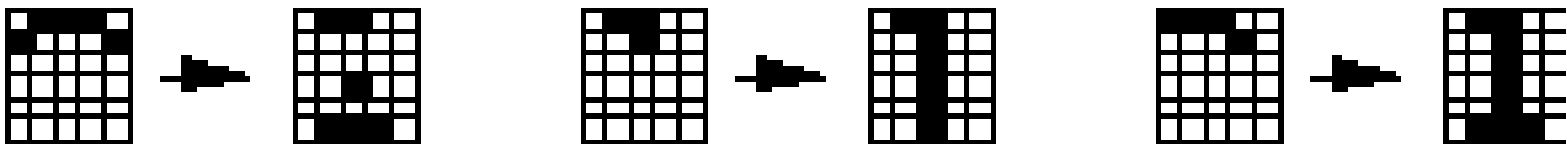
$$\mathbf{W} = \mathbf{p}_1\mathbf{p}_1^T + \mathbf{p}_2\mathbf{p}_2^T + \mathbf{p}_3\mathbf{p}_3^T$$



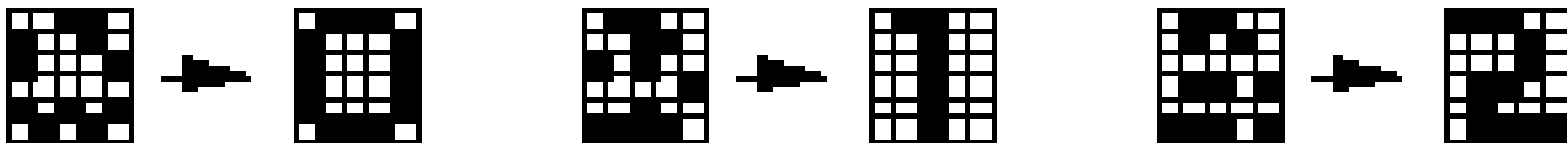
50% Occluded



67% Occluded



Noisy Patterns (7 pixels)



7

Variations of Hebbian Learning



Basic Rule: $\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$

Learning Rate: $\mathbf{W}^{new} = \mathbf{W}^{old} + \eta \mathbf{t}_q \mathbf{p}_q^T$

Smoothing: $\mathbf{W}^{new} = \mathbf{W}^{old} + \eta \mathbf{t}_q \mathbf{p}_q^T - \eta \mathbf{W}^{old} = (1 - \eta) \mathbf{W}^{old} + \eta \mathbf{t}_q \mathbf{p}_q^T$

Delta Rule: $\mathbf{W}^{new} = \mathbf{W}^{old} + \eta (\mathbf{t}_q - \mathbf{a}_q) \mathbf{p}_q^T$

Unsupervised: $\mathbf{W}^{new} = \mathbf{W}^{old} + \eta \mathbf{a}_q \mathbf{p}_q^T$