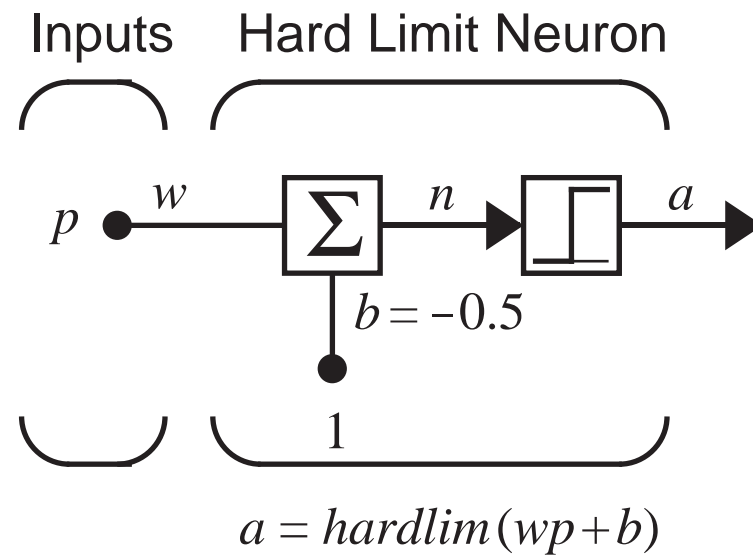




Associative Learning

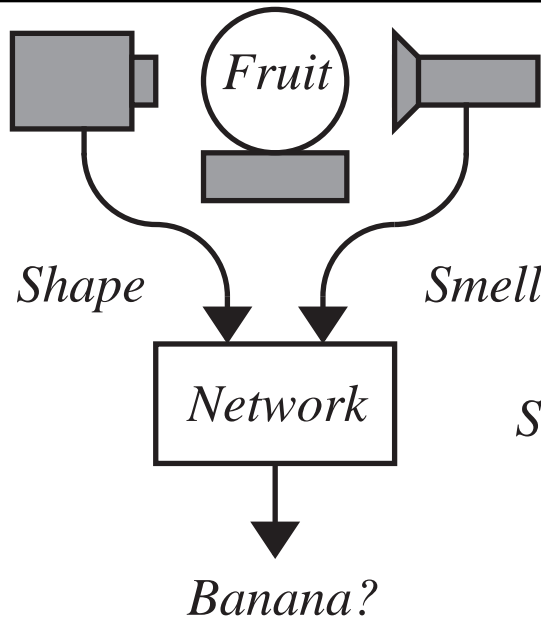
Simple Associative Network



$$a = \text{hardlim}(wp + b) = \text{hardlim}(wp - 0.5)$$

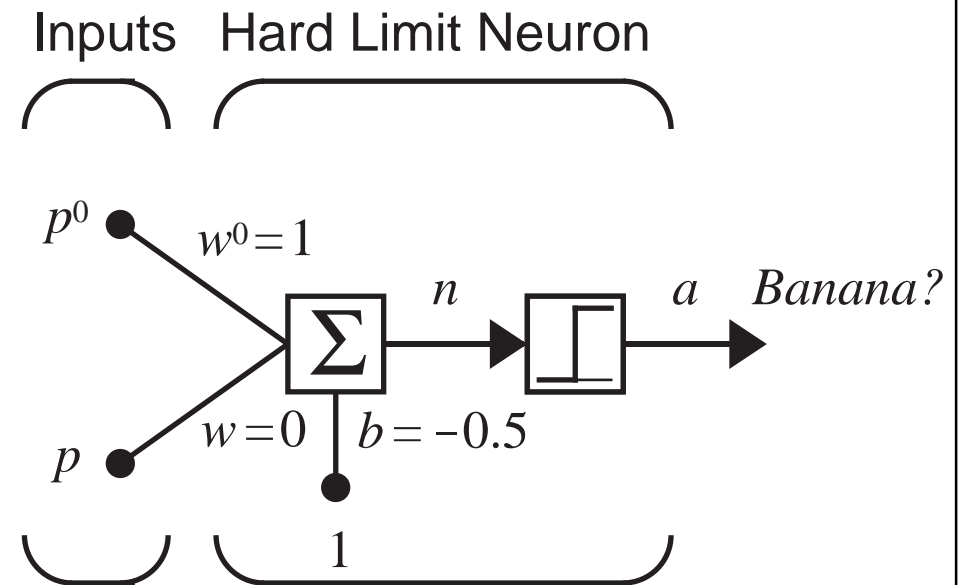
$$p = \begin{cases} 1, & \text{stimulus} \\ 0, & \text{no stimulus} \end{cases}$$

$$a = \begin{cases} 1, & \text{response} \\ 0, & \text{no response} \end{cases}$$



Sight of banana

Smell of banana



$$a = \text{hardlim}(w^0 p^0 + w p + b)$$

Unconditioned Stimulus

$$p^0 = \begin{cases} 1, & \text{shape detected} \\ 0, & \text{shape not detected} \end{cases}$$

Conditioned Stimulus

$$p = \begin{cases} 1, & \text{smell detected} \\ 0, & \text{smell not detected} \end{cases}$$



$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

Vector Form:

$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha \mathbf{a}(q) \mathbf{p}^T(q)$$

Training Sequence:

$$\mathbf{p}(1), \mathbf{p}(2), \dots, \mathbf{p}(Q)$$



Initial Weights:

$$w^0 = 1, w(0) = 0$$

Training Sequence:

$$\{p^0(1) = 0, p(1) = 1\}, \{p^0(2) = 1, p(2) = 1\}, \dots$$

$$\alpha = 1$$

$$w(q) = w(q-1) + a(q)p(q)$$

First Iteration (sight fails):

$$\begin{aligned} a(1) &= \text{hardlim}(w^0 p^0(1) + w(0)p(1) - 0.5) \\ &= \text{hardlim}(1 \cdot 0 + 0 \cdot 1 - 0.5) = 0 \quad (\text{no response}) \end{aligned}$$

$$w(1) = w(0) + a(1)p(1) = 0 + 0 \cdot 1 = 0$$



Second Iteration (sight works):

$$\begin{aligned} a(2) &= \mathit{hardlim}(w^0 p^0(2) + w(1)p(2) - 0.5) \\ &= \mathit{hardlim}(1 \cdot 1 + 0 \cdot 1 - 0.5) = 1 \quad (\text{banana}) \end{aligned}$$

$$w(2) = w(1) + a(2)p(2) = 0 + 1 \cdot 1 = 1$$

Third Iteration (sight fails):

$$\begin{aligned} a(3) &= \mathit{hardlim}(w^0 p^0(3) + w(2)p(3) - 0.5) \\ &= \mathit{hardlim}(1 \cdot 0 + 1 \cdot 1 - 0.5) = 1 \quad (\text{banana}) \end{aligned}$$

$$w(3) = w(2) + a(3)p(3) = 1 + 1 \cdot 1 = 2$$

Banana will now be detected if either sensor works.



- Weights can become arbitrarily large
- There is no mechanism for weights to decrease

Hebb Rule with Decay



$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha \mathbf{a}(q) \mathbf{p}^T(q) - \gamma \mathbf{W}(q-1)$$

$$\mathbf{W}(q) = (1 - \gamma) \mathbf{W}(q-1) + \alpha \mathbf{a}(q) \mathbf{p}^T(q)$$

This keeps the weight matrix from growing without bound, which can be demonstrated by setting both a_i and p_j to 1:

$$w_{ij}^{max} = (1 - \gamma) w_{ij}^{max} + \alpha a_i p_j$$

$$w_{ij}^{max} = (1 - \gamma) w_{ij}^{max} + \alpha$$

$$w_{ij}^{max} = \frac{\alpha}{\gamma}$$



$$\alpha = 1$$

$$\gamma = 0.1$$

First Iteration (sight fails):

$$\begin{aligned} a(1) &= \text{hardlim}(w^0 p^0(1) + w(0)p(1) - 0.5) \\ &= \text{hardlim}(1 \cdot 0 + 0 \cdot 1 - 0.5) = 0 \quad (\text{no response}) \end{aligned}$$

$$w(1) = w(0) + a(1)p(1) - 0.1w(0) = 0 + 0 \cdot 1 - 0.1(0) = 0$$

Second Iteration (sight works):

$$\begin{aligned} a(2) &= \text{hardlim}(w^0 p^0(2) + w(1)p(2) - 0.5) \\ &= \text{hardlim}(1 \cdot 1 + 0 \cdot 1 - 0.5) = 1 \quad (\text{banana}) \end{aligned}$$

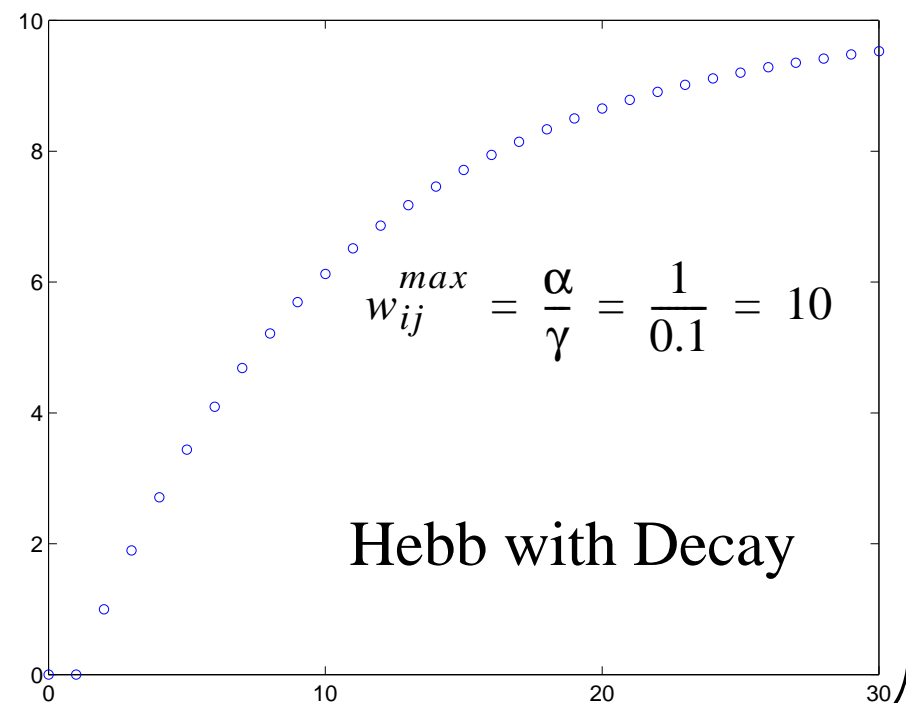
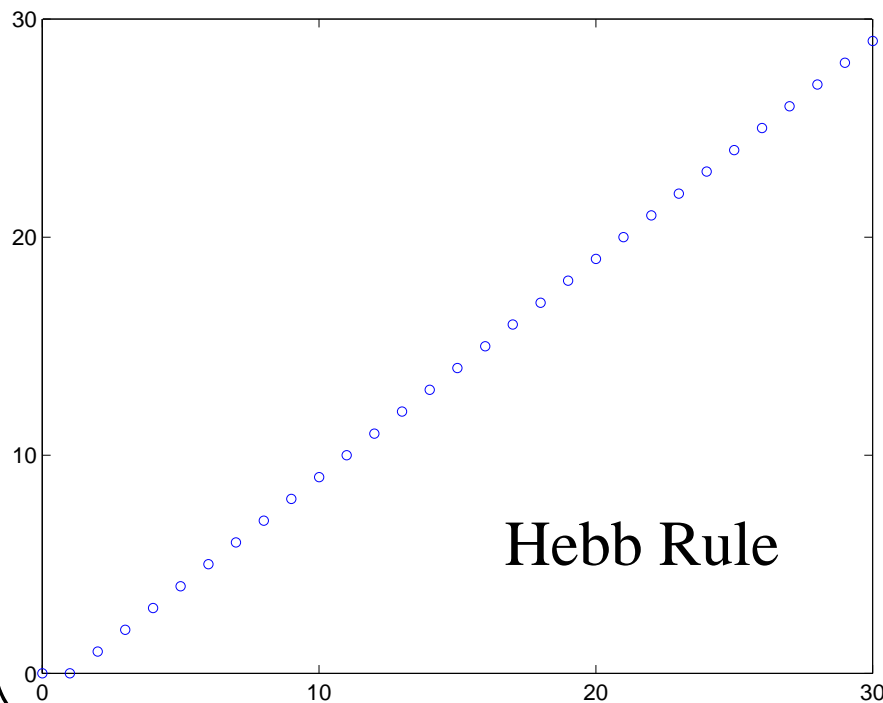
$$w(2) = w(1) + a(2)p(2) - 0.1w(1) = 0 + 1 \cdot 1 - 0.1(0) = 1$$



Third Iteration (sight fails):

$$\begin{aligned} a(3) &= \text{hardlim}(w^0 p^0(3) + w(2)p(3) - 0.5) \\ &= \text{hardlim}(1 \cdot 0 + 1 \cdot 1 - 0.5) = 1 \quad (\text{banana}) \end{aligned}$$

$$w(3) = w(2) + a(3)p(3) - 0.1w(3) = 1 + 1 \cdot 1 - 0.1(1) = 1.9$$





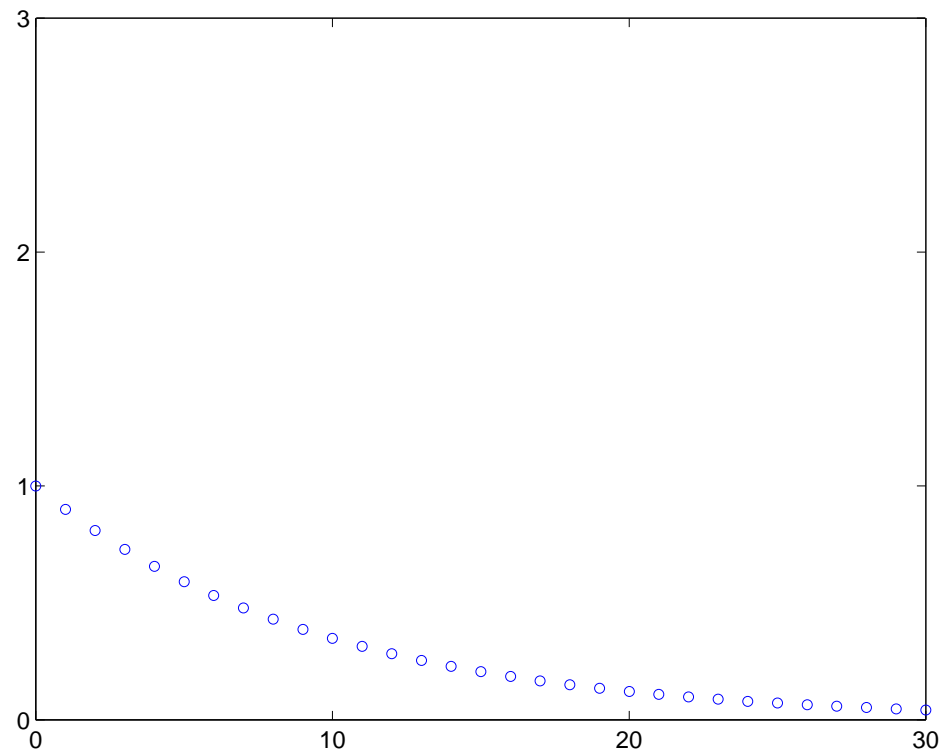
- Associations will decay away if stimuli are not occasionally presented.

If $a_i = 0$, then

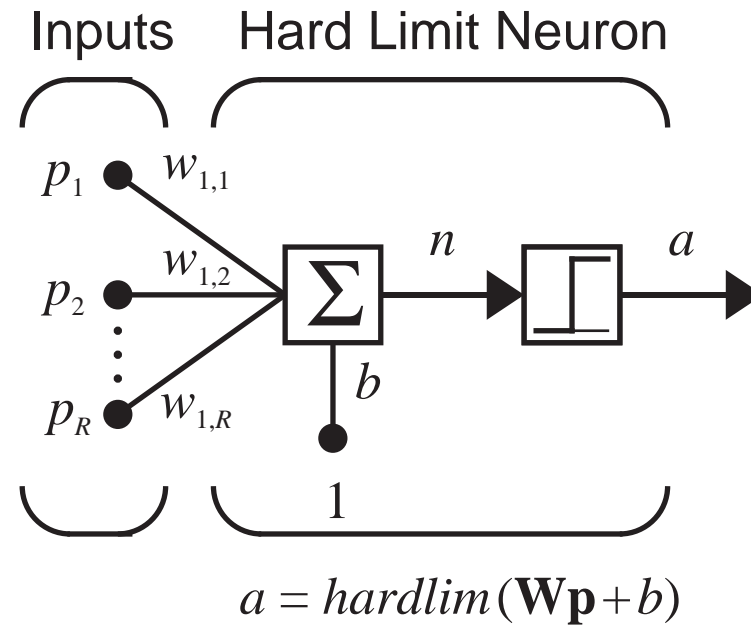
$$w_{ij}(q) = (1 - \gamma)w_{ij}(q - 1)$$

If $\gamma = 0.1$, this becomes

$$w_{ij}(q) = (0.9)w_{ij}(q - 1)$$



Therefore the weight decays by 10% at each iteration where there is no stimulus.



Instar Operation



$$a = \text{hardlim}(\mathbf{W}\mathbf{p} + b) = \text{hardlim}({}_1\mathbf{w}^T\mathbf{p} + b)$$

The instar will be active when

$${}_1\mathbf{w}^T\mathbf{p} \geq -b$$

or

$${}_1\mathbf{w}^T\mathbf{p} = \|{}_1\mathbf{w}\| \|\mathbf{p}\| \cos\theta \geq -b$$

For normalized vectors, the largest inner product occurs when the angle between the weight vector and the input vector is zero -- the input vector is equal to the weight vector.

The rows of a weight matrix represent patterns to be recognized.



If we set

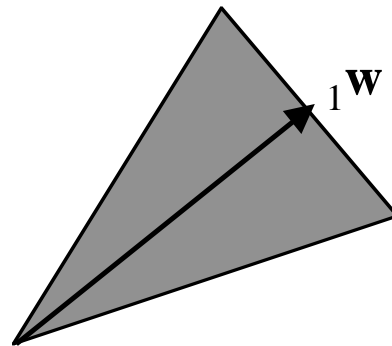
$$b = -\|_1 \mathbf{w}\| \|\mathbf{p}\|$$

the instar will only be active when $\theta = 0$.

If we set

$$b > -\|_1 \mathbf{w}\| \|\mathbf{p}\|$$

the instar will be active for a range of angles.



As b is increased, the more patterns there will be (over a wider range of θ) which will activate the instar.



Hebb with Decay

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

Modify so that learning and forgetting will only occur when the neuron is active - Instar Rule:

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q) - \gamma a_i(q) w_{ij}(q-1)$$

or

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) (p_j(q) - w_{ij}(q-1))$$

Vector Form:

$${}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha a_i(q) (\mathbf{p}(q) - {}_i\mathbf{w}(q-1))$$

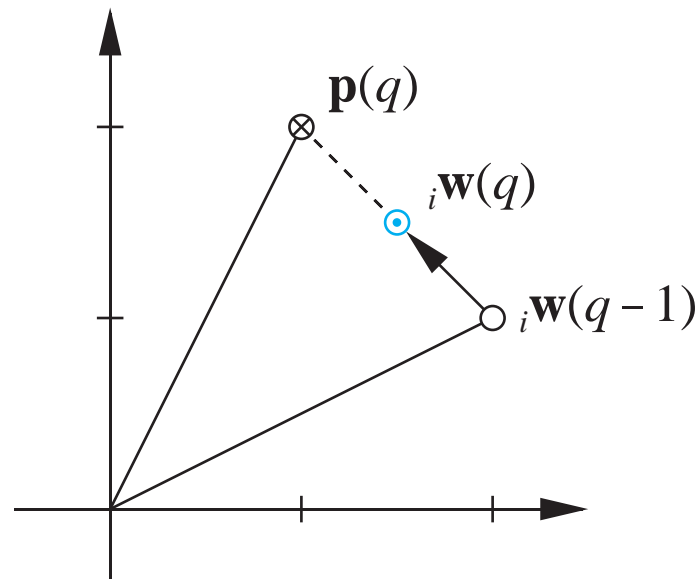


For the case where the instar is active ($a_i = 1$):

$${}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1))$$

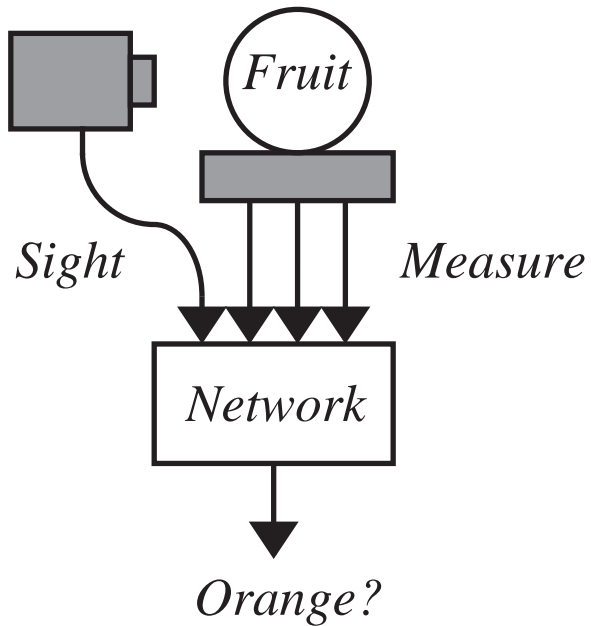
or

$${}_i\mathbf{w}(q) = (1 - \alpha){}_i\mathbf{w}(q-1) + \alpha\mathbf{p}(q)$$



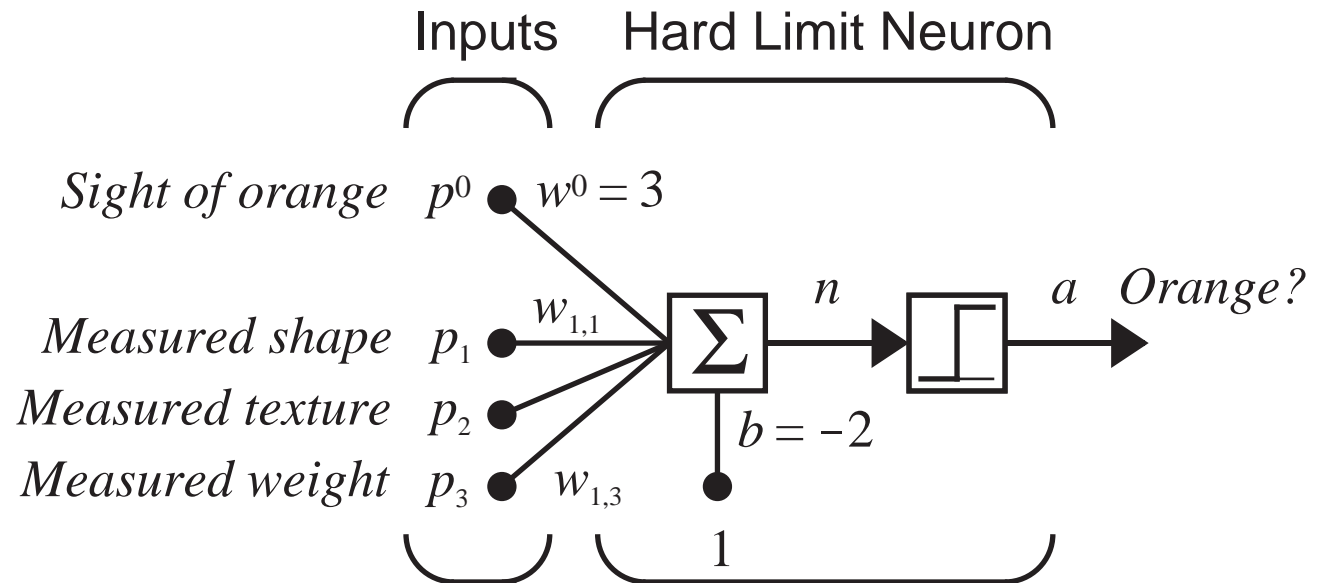
For the case where the instar is inactive ($a_i = 0$):

$${}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1)$$



$$p^0 = \begin{cases} 1, & \text{orange detected visually} \\ 0, & \text{orange not detected} \end{cases}$$

$$\mathbf{p} = \begin{bmatrix} \text{shape} \\ \text{texture} \\ \text{weight} \end{bmatrix}$$



$$a = \text{hardlim}(w^0 p^0 + \mathbf{W} \mathbf{p} + b)$$



$$\mathbf{W}(0) = {}_1\mathbf{w}^T(0) = [0 \ 0 \ 0]$$

$$\left\{ p^0(1) = 0, \mathbf{p}(1) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}, \left\{ p^0(2) = 1, \mathbf{p}(2) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}, \dots$$

First Iteration ($\alpha=1$):

$$a(1) = \text{hardlim}(w^0 p^0(1) + \mathbf{W}\mathbf{p}(1) - 2)$$

$$a(1) = \text{hardlim}\left(3 \cdot 0 + [0 \ 0 \ 0] \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2\right) = 0 \quad (\text{no response})$$

$${}_1\mathbf{w}(1) = {}_1\mathbf{w}(0) + a(1)(\mathbf{p}(1) - {}_1\mathbf{w}(0)) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 0 \left(\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



$$a(2) = \text{hardlim}(w^0 p^0(2) + \mathbf{W}\mathbf{p}(2) - 2) = \text{hardlim}\left(3 \cdot 1 + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2\right) = 1 \quad (\text{orange})$$

$${}_1\mathbf{w}(2) = {}_1\mathbf{w}(1) + a(2)(\mathbf{p}(2) - {}_1\mathbf{w}(1)) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 1 \left(\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

$$a(3) = \text{hardlim}(w^0 p^0(3) + \mathbf{W}\mathbf{p}(3) - 2) = \text{hardlim}\left(3 \cdot 0 + \begin{bmatrix} 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2\right) = 1 \quad (\text{orange})$$

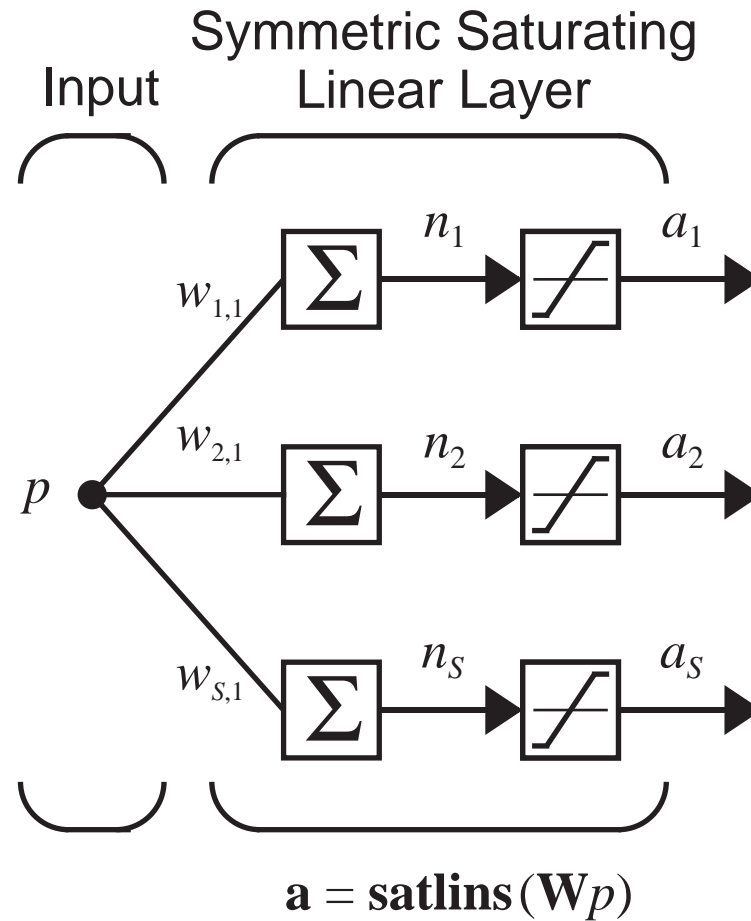
$${}_1\mathbf{w}(3) = {}_1\mathbf{w}(2) + a(3)(\mathbf{p}(3) - {}_1\mathbf{w}(2)) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 1 \left(\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

Orange will now be detected if either set of sensors works.



$${}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)), \quad \text{for } i \in X(q)$$

Learning occurs when the neuron's index i is a member of the set $X(q)$. We will see in Chapter 14 that this can be used to train all neurons in a given neighborhood.





Suppose we want the outstar to recall a certain pattern \mathbf{a}^* whenever the input $p=1$ is presented to the network. Let

$$\mathbf{W} = \mathbf{a}^*$$

Then, when $p=1$

$$\mathbf{a} = \text{satlins}(\mathbf{W}_p) = \text{satlins}(\mathbf{a}^* \cdot 1) = \mathbf{a}^*$$

and the pattern is correctly recalled.

The columns of a weight matrix represent patterns to be recalled.



For the instar rule we made the weight decay term of the Hebb rule proportional to the output of the network. For the outstar rule we make the weight decay term proportional to the input of the network.

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q)p_j(q) - \gamma p_j(q)w_{ij}(q-1)$$

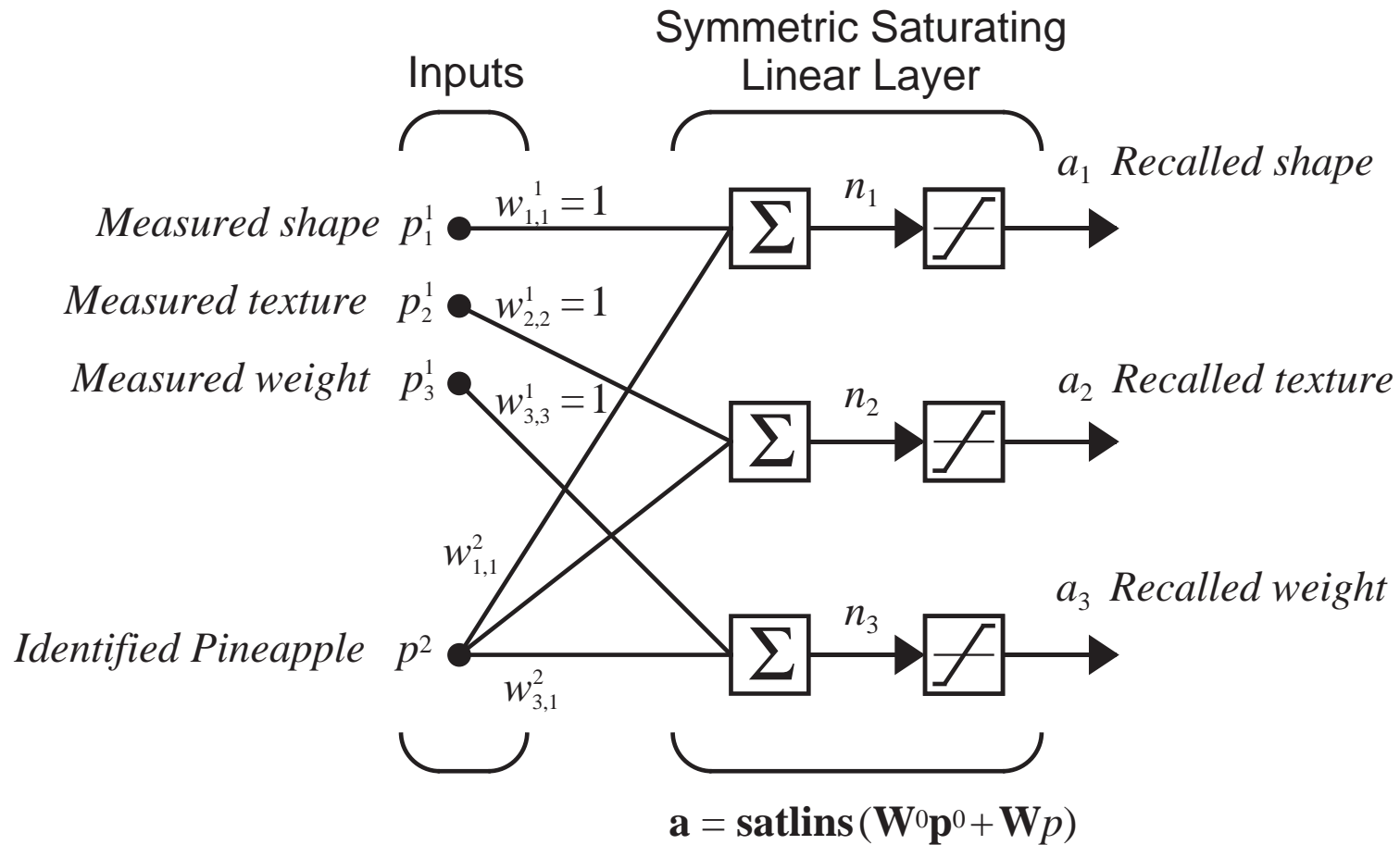
If we make the decay rate γ equal to the learning rate α ,

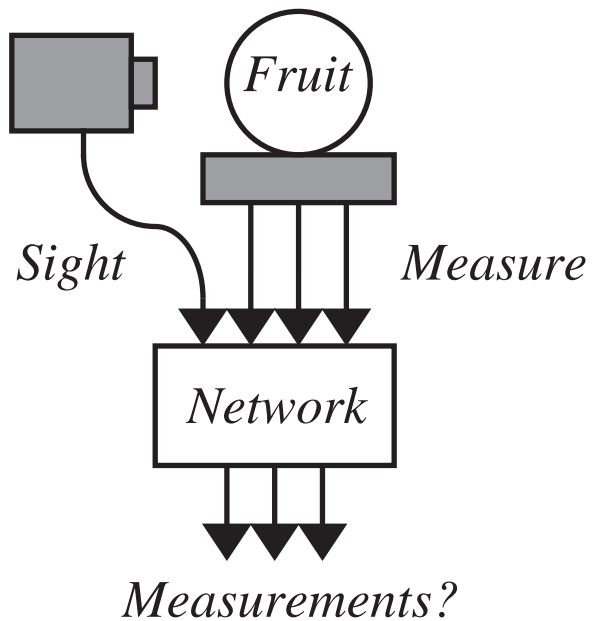
$$w_{ij}(q) = w_{ij}(q-1) + \alpha(a_i(q) - w_{ij}(q-1))p_j(q)$$

Vector Form:

$$\mathbf{w}_j(q) = \mathbf{w}_j(q-1) + \alpha(\mathbf{a}(q) - \mathbf{w}_j(q-1))p_j(q)$$

Example - Pineapple Recall





$$\mathbf{a} = \text{satlins}(\mathbf{W}^0 \mathbf{p}^0 + \mathbf{W}p)$$

$$\mathbf{W}^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{p}^0 = \begin{bmatrix} \textit{shape} \\ \textit{texture} \\ \textit{weight} \end{bmatrix}$$

$$\mathbf{p}^{\textit{pineapple}} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

$$p = \begin{cases} 1, & \text{if a pineapple can be seen} \\ 0, & \text{otherwise} \end{cases}$$



$$\left\{ \mathbf{p}^0(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, p(1) = 1 \right\}, \left\{ \mathbf{p}^0(2) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}, p(2) = 1 \right\}, \dots$$

$$\alpha = 1$$

$$\mathbf{a}(1) = \mathbf{satlins} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} 1 \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{no response})$$

$$\mathbf{w}_1(1) = \mathbf{w}_1(0) + (\mathbf{a}(1) - \mathbf{w}_1(0))p(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) 1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



$$\mathbf{a}(2) = \text{satlins} \left(\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} 1 \right) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \quad (\text{measurements given})$$

$$\mathbf{w}_1(2) = \mathbf{w}_1(1) + (\mathbf{a}(2) - \mathbf{w}_1(1))p(2) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \left(\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) 1 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

$$\mathbf{a}(3) = \text{satlins} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} 1 \right) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \quad (\text{measurements recalled})$$

$$\mathbf{w}_1(3) = \mathbf{w}_1(2) + (\mathbf{a}(2) - \mathbf{w}_1(2))p(2) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} + \left(\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \right) 1 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$