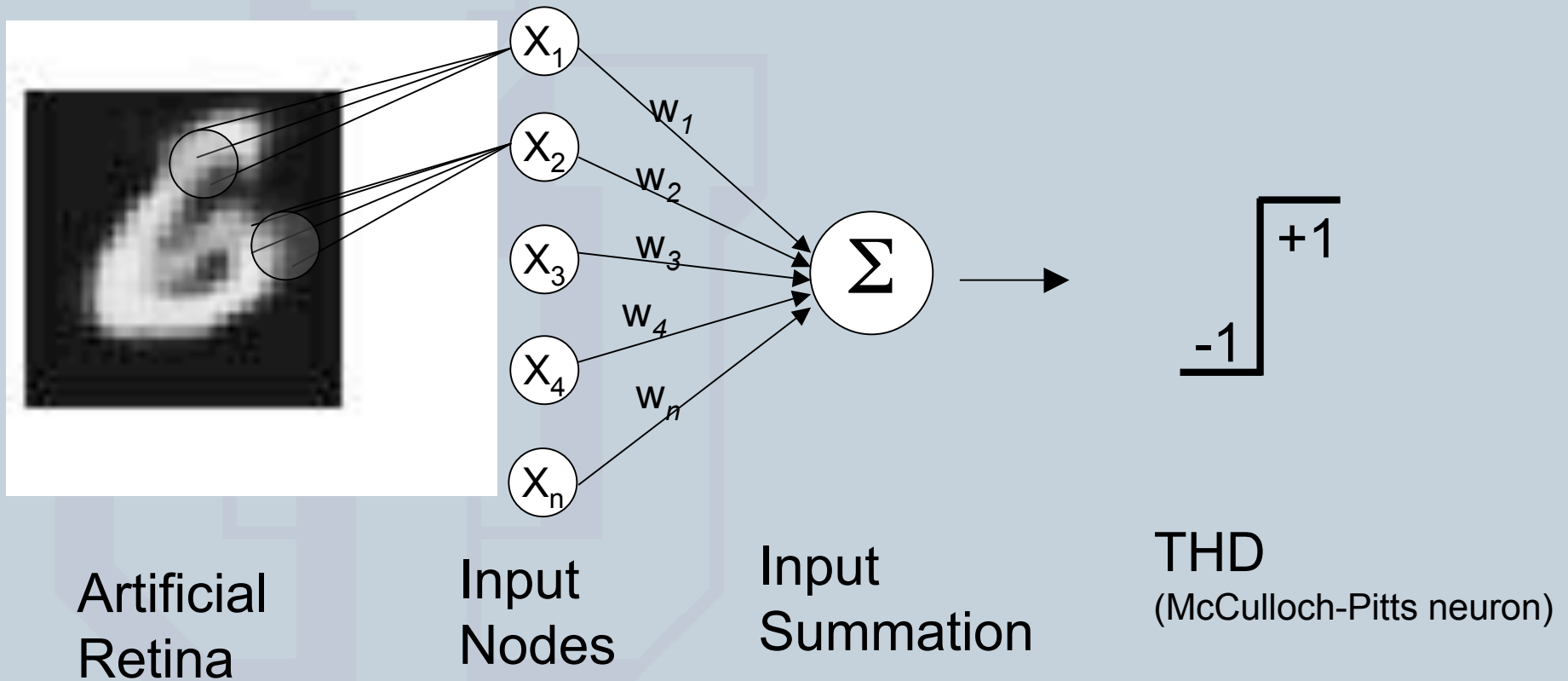


# Multilayer Networks

Q550: Models in Cognitive Science



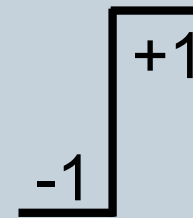
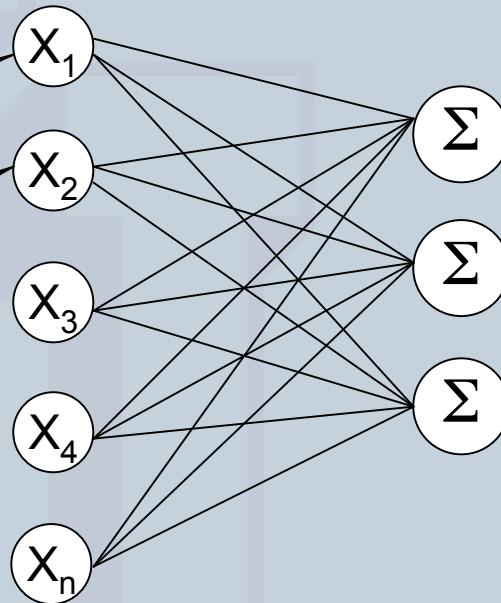
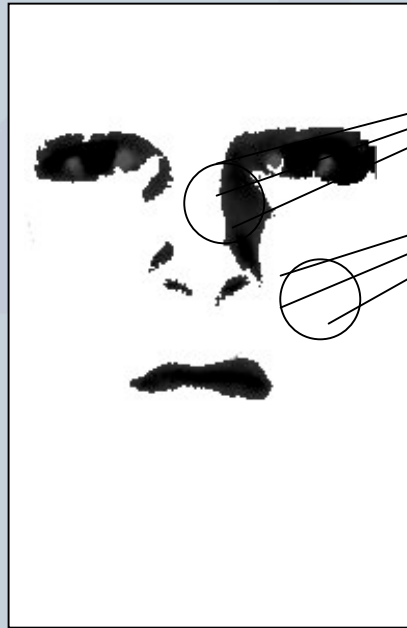
# Single-Layer Perceptron



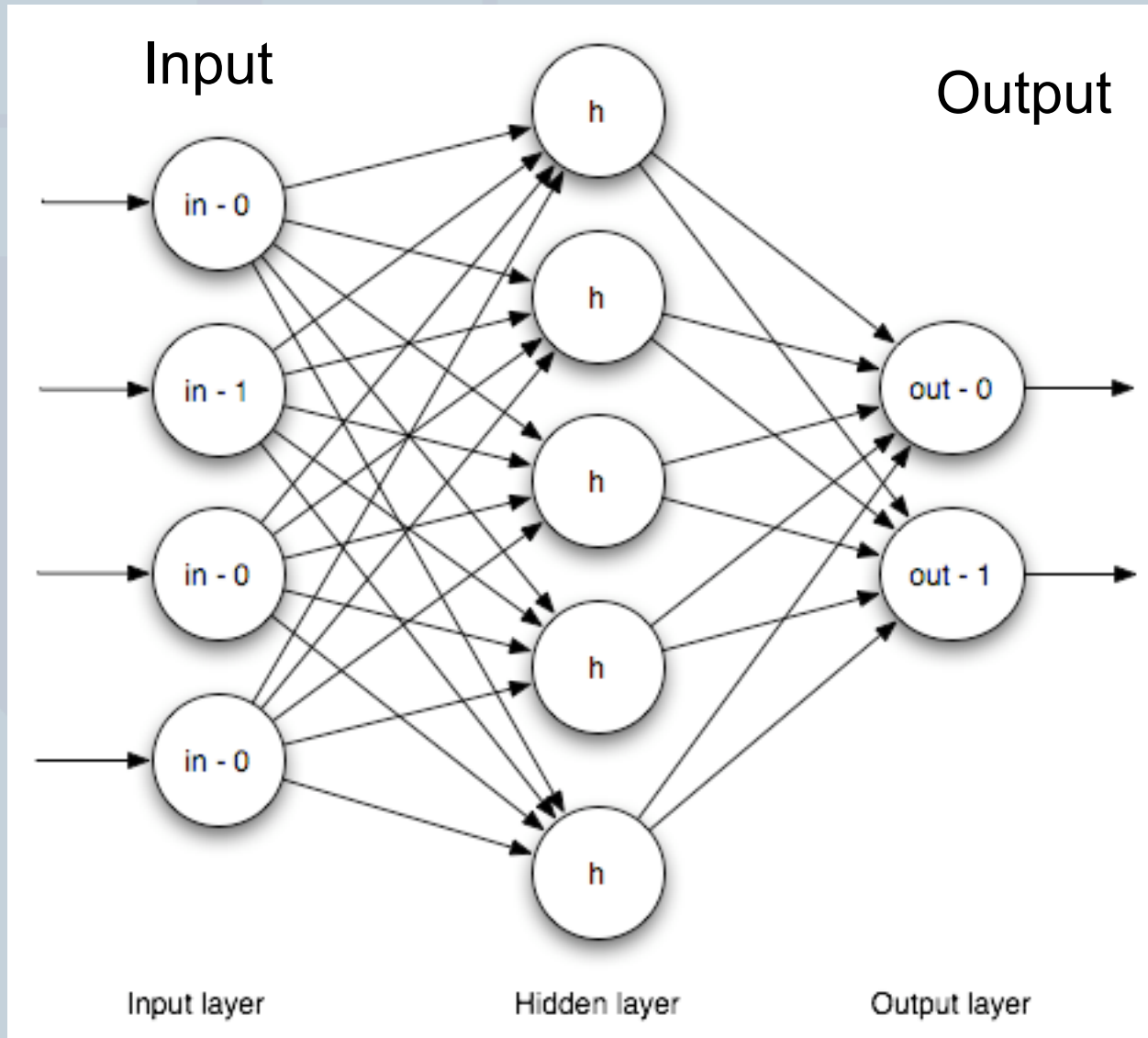
If  $y = t$ , do nothing  
If  $y \neq t$ , then delta update:

$$\Delta w_{ij} = \alpha(t_i - y_i)x_j$$

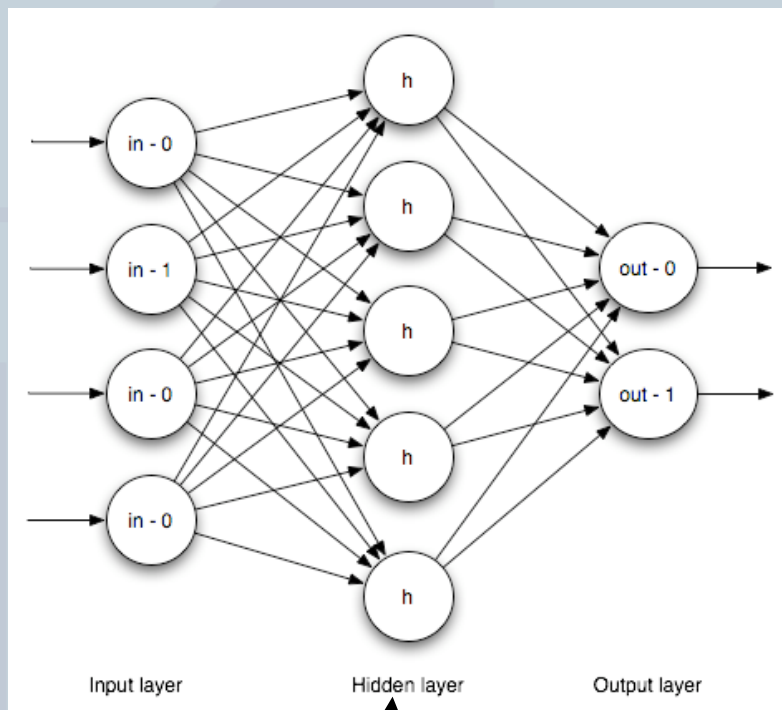
# Multiclass Single-Layer Perceptron



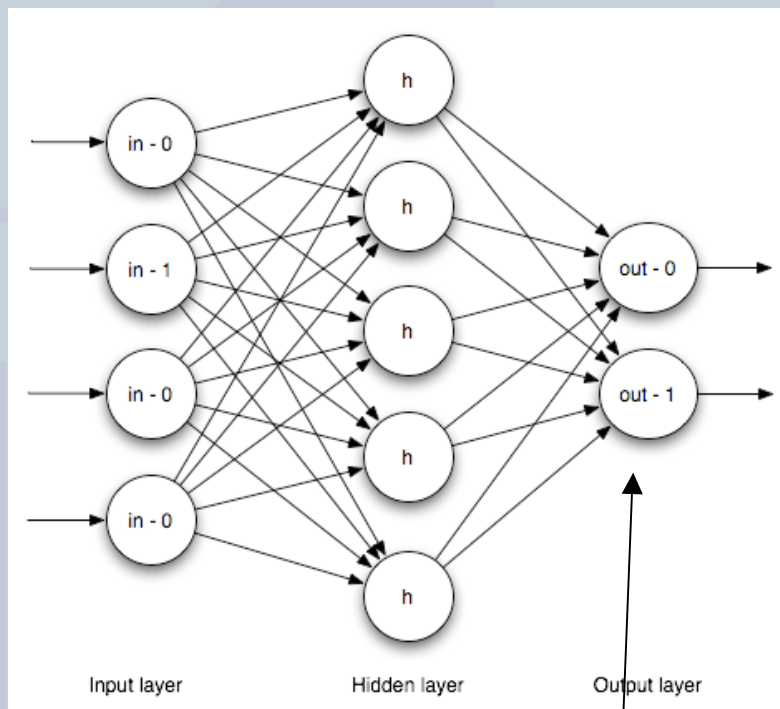
# Hidden



Heteroassociator  
Autoassociator



$$net_i = \sum_{j=1}^n i_j w_{ij}$$



Logistic sigmoid transform

$$net_i = \sum_{j=1}^n h_j w_{ij} \longrightarrow o_i = \frac{1}{1 + \exp^{-net_i}}$$

# Training the network

---

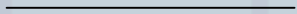
- Backwards propagation of errors (“backprop”)

## **DO i = 1 to N\_Training\_Examples**

- Present a training example and compute output
- Compare actual output to desired output; determine error for each node
- For each node, calculate what the output should have been, and a scaling factor to produce the desired output
- Adjust the weights of each node to minimize error
- Assign “blame” for error to nodes at the previous level, giving more blame for nodes more responsible for the error
- Repeat for the previous layer, using its blame as error

**ENDDO**

$i_j$



$h_j$



$o_k$

$i_j$



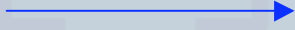
$h_j$



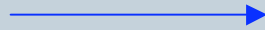
$o_k$

$$h_j = \sum_{i=1}^I i_i w_{ji}$$

$$i_j$$



$$h_j$$



$$o_k$$

$$h_j = \sum_{i=1}^I i_i w_{ji}$$

$$net_k = \sum_{j=1}^J h_j w_{kj}$$

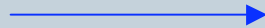
$$o_k = \frac{1}{1 + \exp^{-net_k}}$$

$t_k$  is the desired output

$i_j$



$h_j$



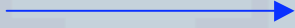
$o_k$

$$\delta_k = (t_k - o_k)o_k(1 - o_k)$$

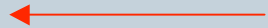
output error for  
each node

$t_k$  is the desired output

$i_j$



$h_j$

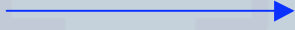


$o_k$

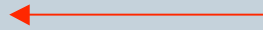
$$\Delta w_{jk} = \eta \delta_k h_j$$

Update H-->O

$i_j$



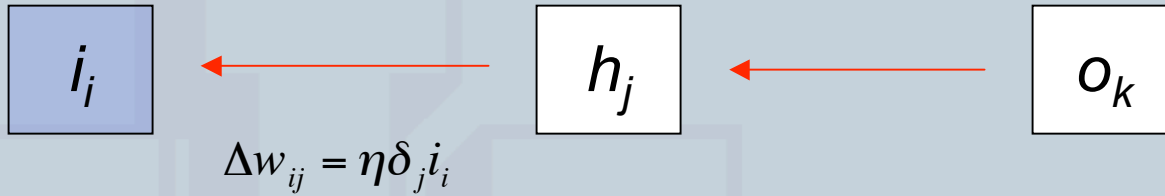
$h_j$



$o_k$

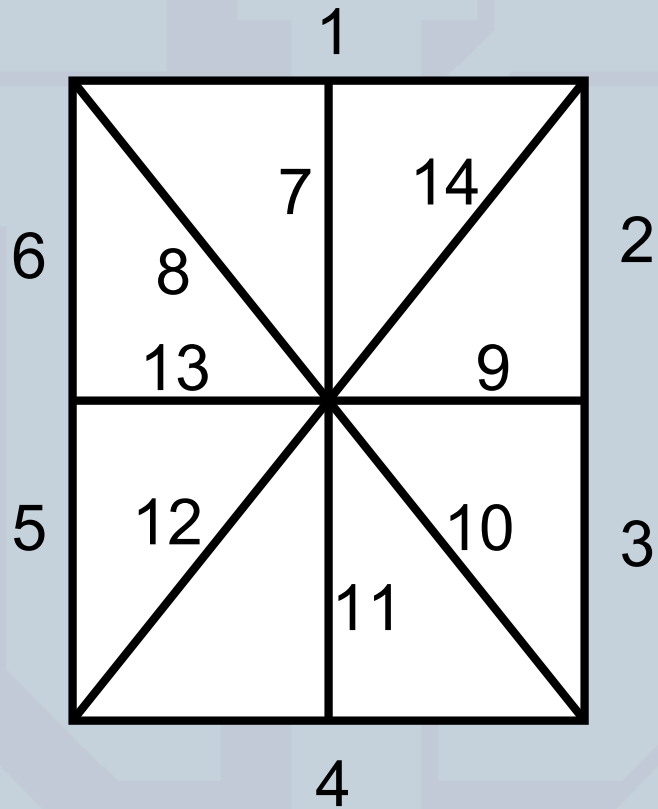
$$\delta_j = h_j(1 - h_j) \sum_{k=1}^K w_{kj} \delta_k$$

error for each  
hidden node

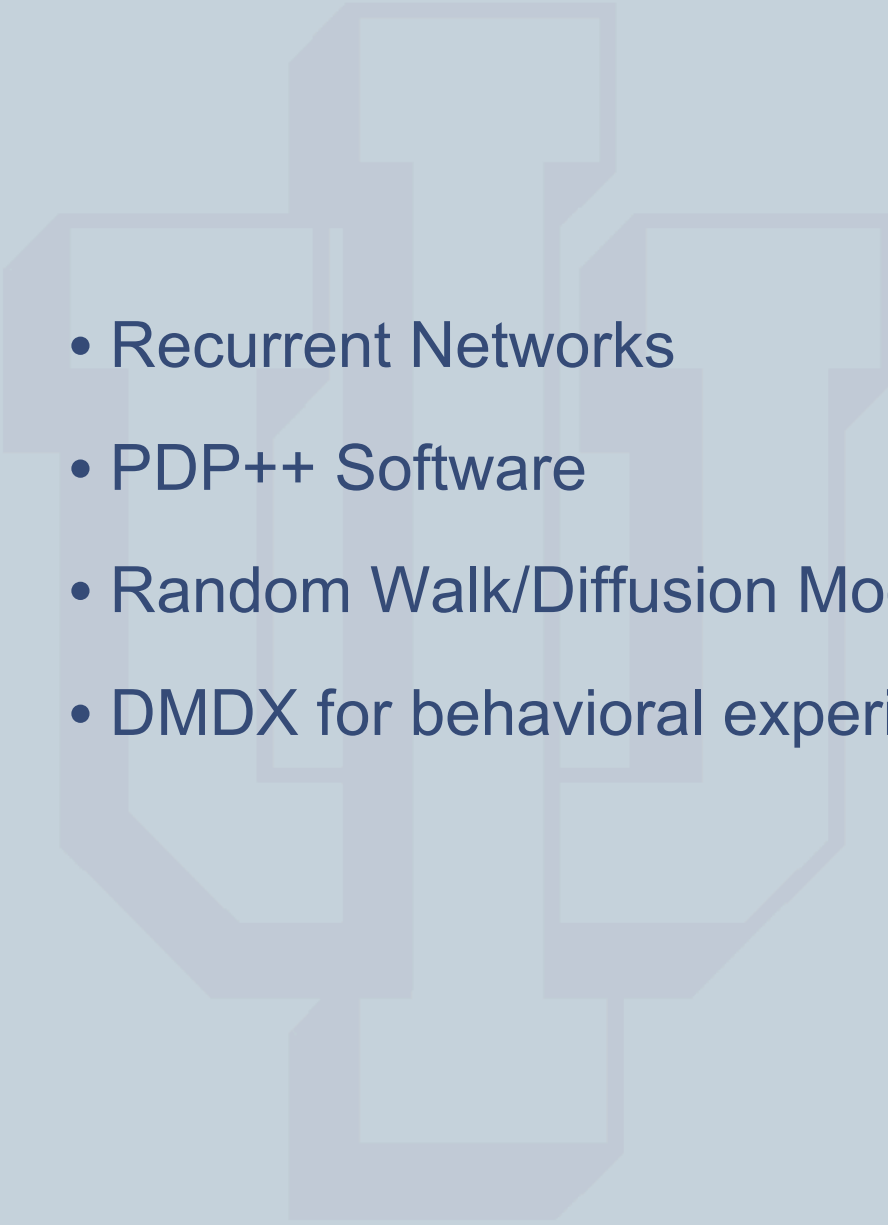


Update I-->H

Then try with a new training pattern and backpropagate the errors until the system is “trained up”



$$E = [1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0]$$

- 
- Recurrent Networks
  - PDP++ Software
  - Random Walk/Diffusion Model
  - DMDX for behavioral experiments