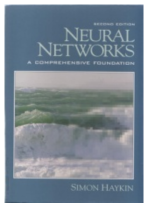
Definition

ANN Learning

- *Learning is a process by which the free parameters of a neural network are adapted in an desired way through a process of stimulation by the environment in which the network is embedded.*
  The type of learning is determined by the manner in which the parameter change takes place.

$$e.g. \quad w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}$$

This is chosen s.t. it minimizes the cost function



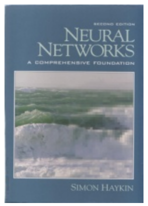
Definition

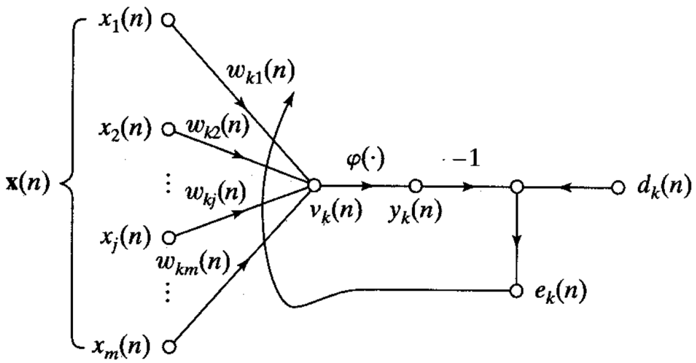
1st Learning Rule: Error Correction

- Error $\boldsymbol{e_k}$ actuates a controls mechanism

n	time step
k	neuron number
$d_k(n)$	desired signal
$y_k(n)$	observed signal

$$e_k(n) = d_k(n) - y_k(n)$$





Definition

1st Learning Rule: Error Correction

- Error e_k actuates a controls mechanism
- Cost function E on top (here: instantaneous at time n, local at output node k)

n	time step
k	neuron number
$d_k(n)$	desired signal
$y_k(n)$	observed signal

$$e_k(n) = d_k(n) - y_k(n)$$

$$E(n) = \frac{1}{2} e_k^{\,2}(n)$$

Definition

1st Learning Rule: Error Correction

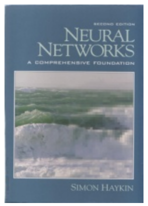
- Error e_k actuates a controls mechanism
- Cost function ***E*** on top (here: instantaneous at time n, local at output node k)
- Minimization of cost function: ***Widrow-Hoff rule (delta rule)***

n	time step
k	neuron number
$d_k(n)$	desired signal
$y_k(n)$	observed signal

$$e_k(n) = d_k(n) - y_k(n)$$

$$E(n) = \tfrac{1}{2} e_k^2(n)$$

$$\boxed{\Delta w_{kj}(n) = e_k(n) x_j(n)}$$



Definition

1st Learning Rule: Error Correction

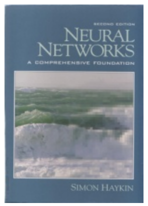
- Error $\boldsymbol{e_k}$ actuates a controls mechanism
- Cost function ***E*** on top (here: instantaneous at time n, local at output node k)
- Minimization of cost function: ***Widrow-Hoff rule (delta rule)***
- *The adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question.*

n	time step
k	neuron number
$d_k(n)$	desired signal
$y_k(n)$	observed signal

$$e_k(n) = d_k(n) - y_k(n)$$

$$E(n) = \tfrac{1}{2} e_k^{\,2}(n)$$

$$\Delta w_{kj}(n) = e_k(n) x_j(n)$$



Definition

1st Learning Rule: Error Correction

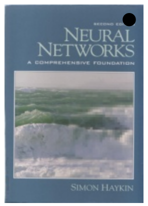
n	time step
k	neuron number
$d_k(n)$	desired signal
$y_k(n)$	observed signal

- Error $\boldsymbol{e_k}$ actuates a control mechanism
- Cost function ***E*** on top (here: instantaneous at time n, local at output node k)
- Minimization of cost function: ***Widrow-Hoff rule (delta rule)***
- *The adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question.*
- Learning rate η and final eq. $w_{kj}(n+1) = w_{kj}(n) + \eta\,\Delta w_{kj}$

$$e_k(n) = d_k(n) - y_k(n)$$

$$E(n) = \tfrac{1}{2} e_k^{\;2}(n)$$

$$\boxed{\Delta w_{kj}(n) = e_k(n)\,x_j(n)}$$



Definition

2nd: Memory based learning

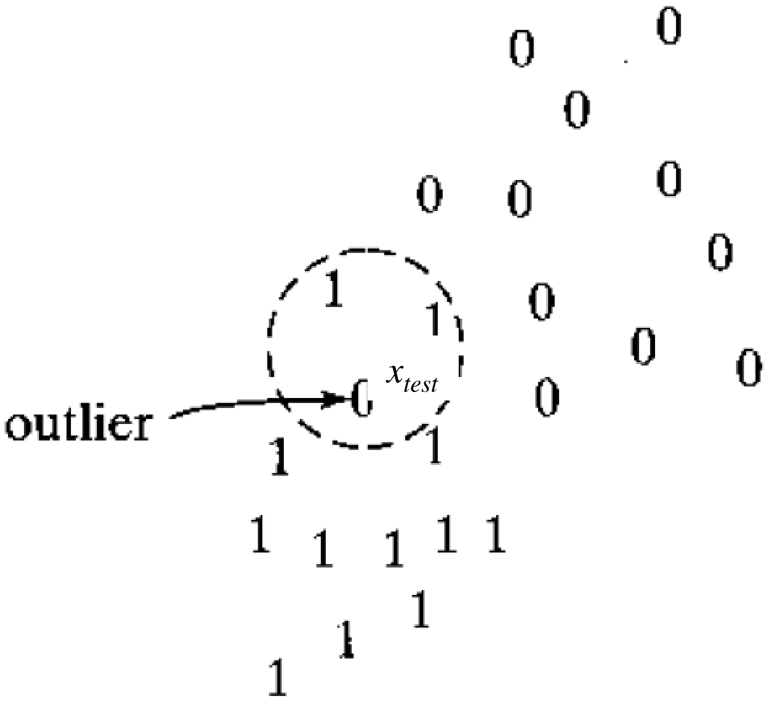
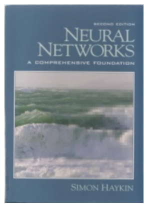
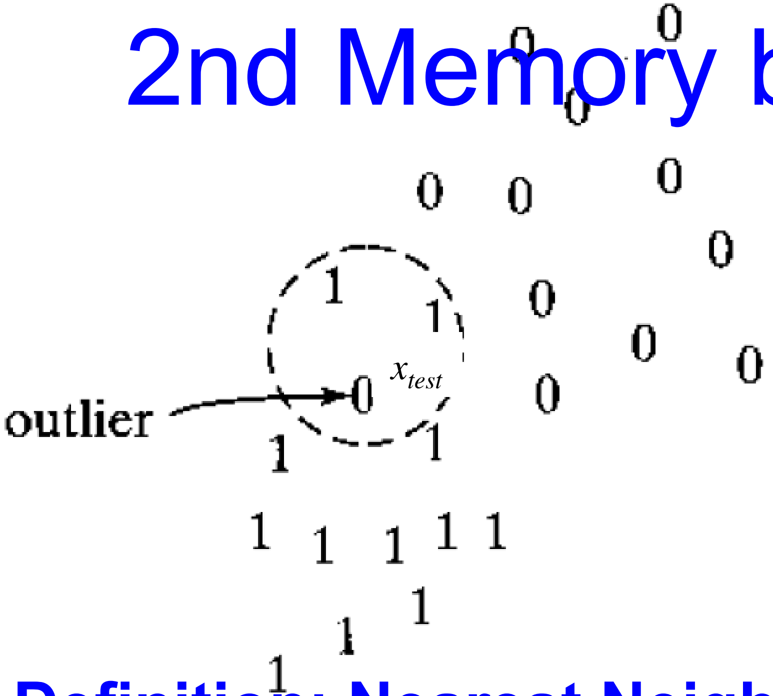


FIGURE 2.2 The area lying inside the dashed circle includes two points pertaining to class 1 and an outlier from class 0. The point *d* corresponds to the test vector $\mathbf{x}_{test}$. With $k = 3$, the *k-nearest* neighbor *classifier* assigns class 1 to point even though it lies closest to the *outlier*.

(majority vote)



2nd Memory based learning cont.



Definition: Nearest Neighbor

$$Given\ L = \{x_1, x_2, \ldots, x_N\}\ and\ x_{test} \notin L.$$

$$Then\ x' \in L\ is\ called$$

$$nearest\ neighbor\ to\ x_{test}\ in\ L :\Leftrightarrow$$

$$\min_i d(x_i, x_{test}) = d(x', x_{test})$$

Algorithm: k-nearest neighbor learning

$$Given\ L, x_{test} \notin L, k \in IN\ \ fixed,$$

$$class\ \ function : class\_of()\ on\ L$$

$$Set\ x' = \{\}, L_0 = L, Classf = empty\_list$$

$$for\ \ j = 1 \ldots k\ do\ \{$$

$$L_j \leftarrow L_{j-1} \setminus x';$$

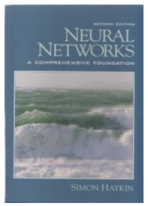
$$x' \leftarrow find\ NN\ to\ x_{test}\ in\ L_j;$$

$$c \leftarrow class\_of(x');$$

$$Classf \leftarrow push(c);$$

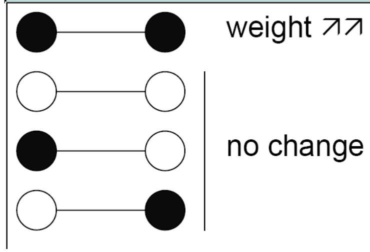
$$\}$$

$$set\ c(x_{test}) := most\ \ frequent\ value\ in\ Classf$$



3rd Hebbian learning

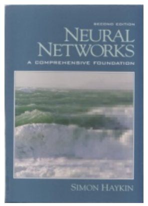
Definition



Donald Hebb, Canadian psychologist, wrote a revolutionary paper in 1949:

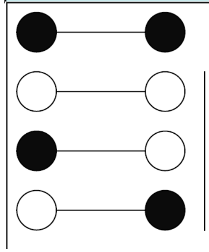
”Let us assume that the persistence or repetition of a reverberatory activity (or “trace”) tends to induce lasting cellular changes that add to its stability.… When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”

1. If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously), then the strength of that synapse is selectively increased.



3rd Hebbian learning

Definition

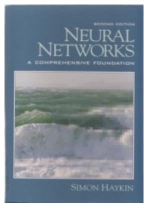


Donald Hebb, Canadian psychologist, wrote a revolutionary paper in 1949:

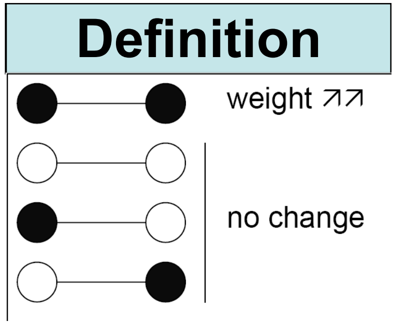
”Let us assume that the persistence or repetition of a reverberatory activity (or “trace”) tends to induce lasting cellular changes that add to its stability.… When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”

1. If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously), then the strength of that synapse is selectively increased.

2. If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.



3rd Hebbian learning

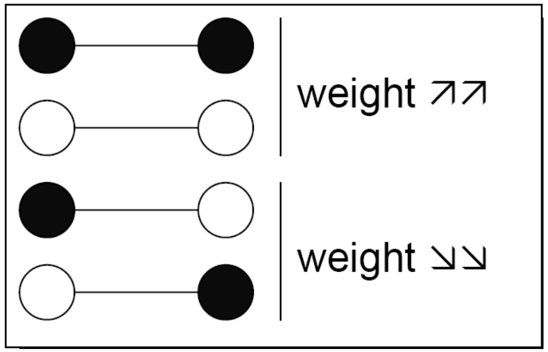


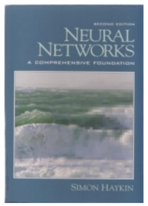
Donald Hebb, Canadian psychologist, wrote a revolutionary paper in 1949:

"Let us assume that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability.… 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."

1. If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously), then the strength of that synapse is selectively increased.

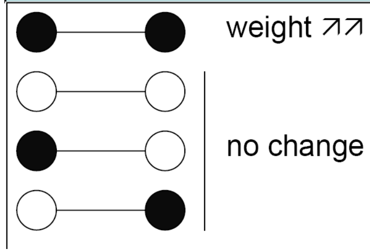
2. If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.





3rd Hebbian learning

Definition

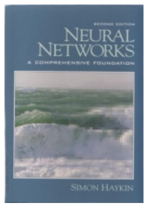


- Time-dependent mechanism
- Local mechanism
- Interactive mechanism
- Conjuctional or correlational mechanism

correlation of the nodes

- In general $$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$
- Most simple $$\Delta w_{kj}(n) = \eta y_k(n) x_j(n)$$
- Covariance hypothesis $$\Delta w_{kj}(n) = \eta(y_k - \bar{y})(x_j - \bar{x})$$

summarized: cells that fire together, wire together.



Example

Multi-valued Perceptron

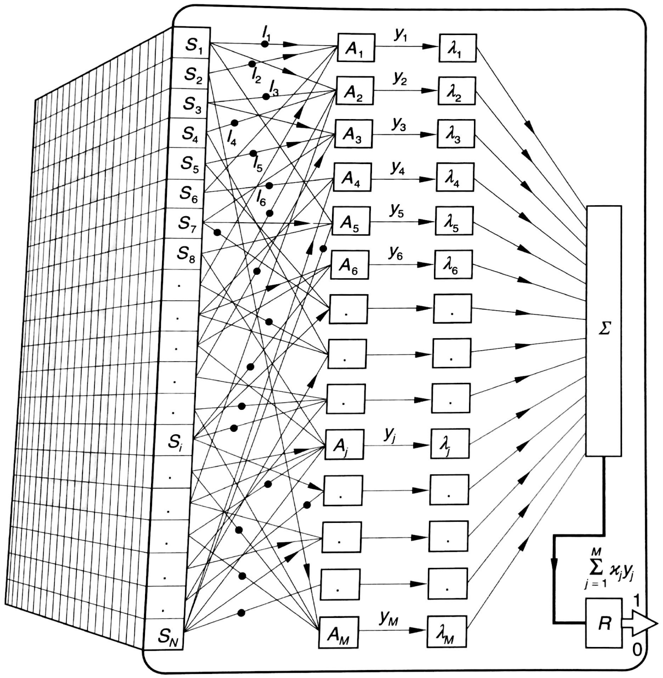


Abb. 3.14

- fixed receptive fields to connect a patch of sensor cells S_i to A_j
- adjustable weights λ_i
- multiple linear combiners, one for each letter

Definition

4th Competitive Learning

- In competitive learning the output neurons compete among themselves to become active (fired).

- Hebbian learning ⇔ several output neurons may be active simultaneously,

- competitive learning ⇔ only a **single output neuron** is active **at any one time**.

Definition

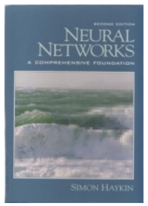
4th Competitive Learning

- In competitive learning the output neurons compete among themselves to become active (fired).

- Hebbian learning ⇔ several output neurons may be active simultaneously,

- competitive learning ⇔ only a **single output neuron** is active **at any one time**.

- A set of neurons (x) that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.
- A limit imposed on the "strength" of each neuron (sum == 1)



4th Competitive learning (cont)

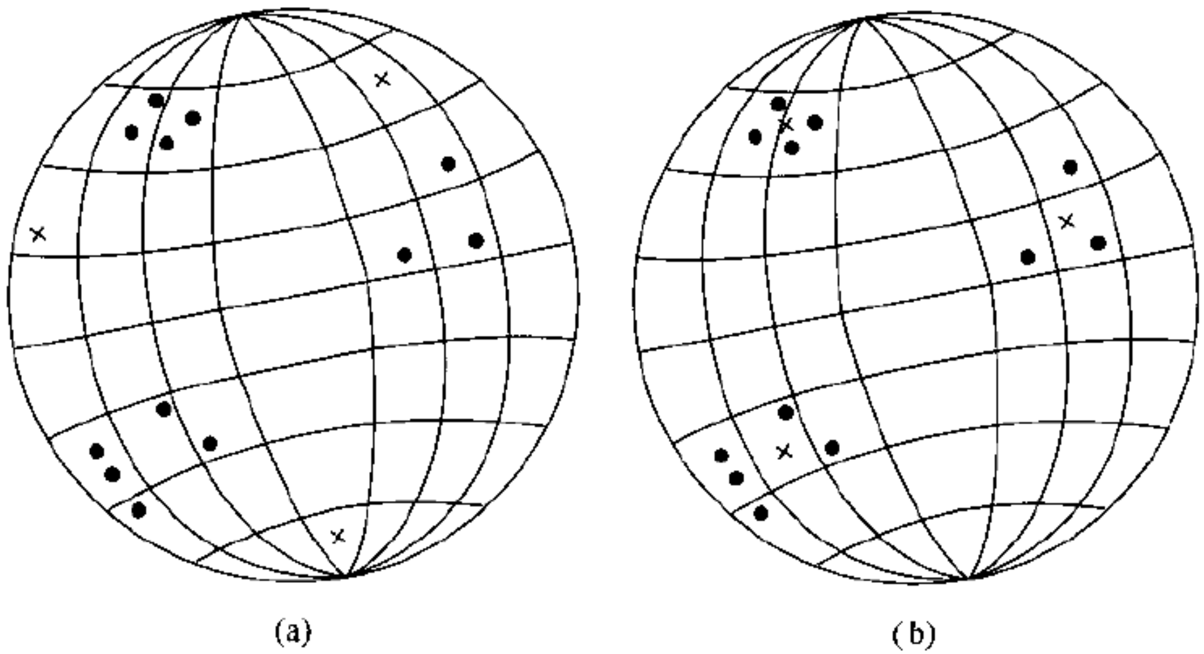
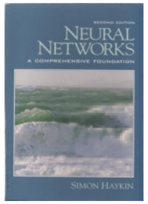


FIGURE 2.5 Geometric interpretation of the competitive learning process. The dots represent the input vectors, and the crosses represent the synaptic weight vectors of three output neurons. (a) Initial state of the network. (b) Final state of the network.



Definition

4th Competitive Learning

- In competitive learning the output neurons compete among themselves to become active (fired).

- Hebbian learning $\Leftrightarrow$ several output neurons may be active simultaneously,

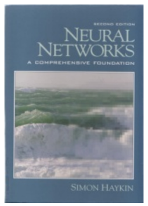
- competitive learning $\Leftrightarrow$ only a **single output neuron** is active **at any one time**.

- A set of neurons (x) that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.
- A limit imposed on the "strength" of each neuron (sum == 1)

- A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, or only one neuron per group, is active (i.e., "on") at a time.

The neuron that wins the competition is called a **winner-takes-all neuron.**

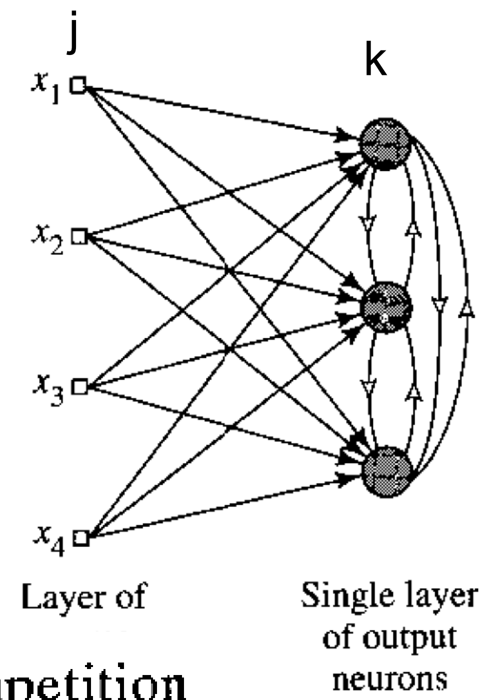
(Rumelhart and Zipser,1985)



Definition

4th Competitive Learning (cont)

Ex: SOM's

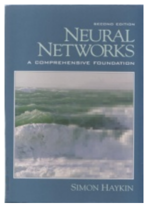


$$\sum_j w_{kj} = 1 \qquad \text{for all } k$$

$$y_k = \begin{cases} 1 & \text{if } v_k > v_j \text{ for all } j, j \neq k \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases}$$

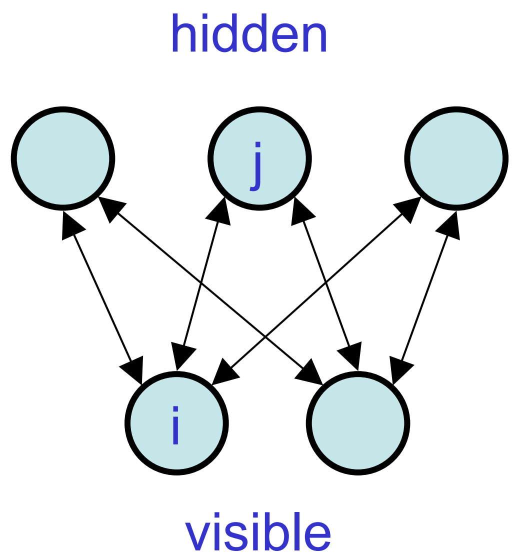
- Combination of 2 NNs: a first layer regular one, and a second competing one

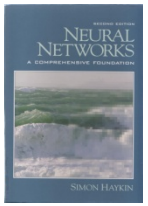


Definition

Sketch: 5th Boltzmann learning

- Recurrent ANN
- Two groups: visible and hidden neurons
- Biparted graph, reduced Boltzmann Machine == RBM
- Binary neurons (+/- 1 as state)
- operate by flipping

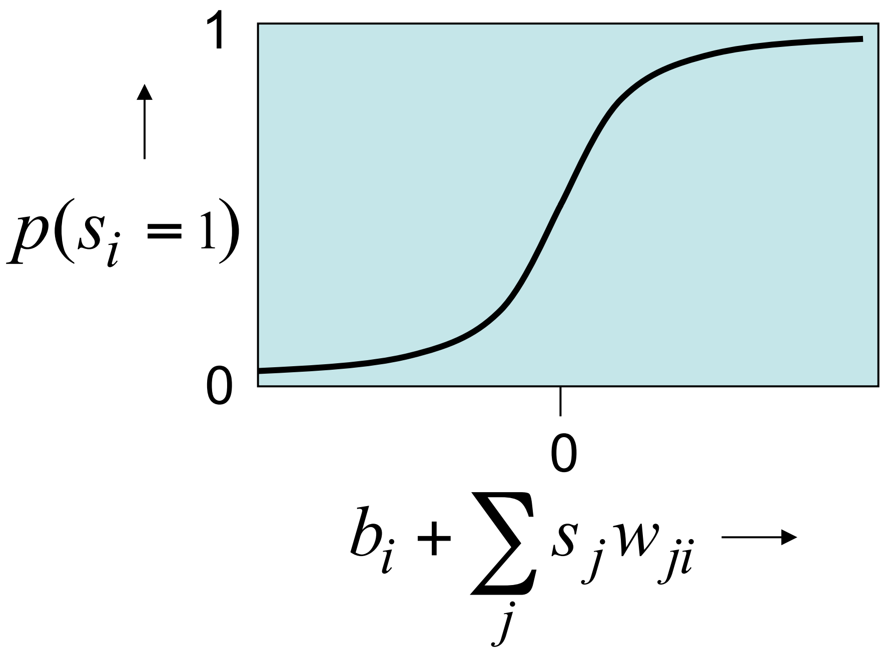




Stochastic binary units

(Bernoulli variables)

- These have a state of 1 or 0.
- The probability of turning on is determined by the weighted input from other units (plus a bias)



$$p(s_i = 1) = \frac{1}{1 + \exp(-b_i - \sum_j s_j w_{ji})}$$

Cited from:
2007 NIPS Tutorial on:
Deep Belief Nets
by Geoffrey Hinton

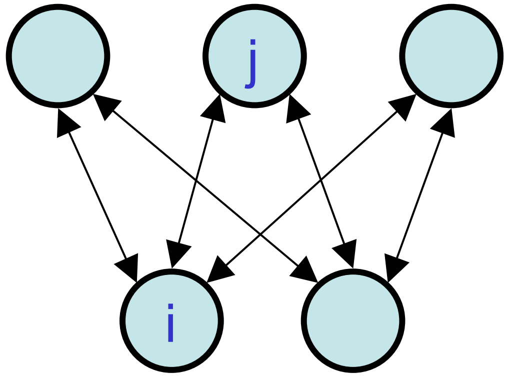
Sketch: 5th Boltzmann learning

- Recurrent ANN
- Two groups: visible and hidden neurons
- Binary neurons (+/- 1 as state)
- operate by flipping
- Modes of operation: clamped / free running conditions

error between clamped and correlated should be 0

restricted as there is no connection bewteen nuerons of the same layer

hidden



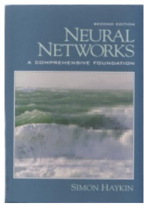
visible

ρ_{kj}^{+} $\quad$ *correlation in clamped state* input doesn't change

rebuilding the input

ρ_{kj}^{-} $\quad$ *correlation in free running state*

$$\Delta w_{kj} = \eta(\rho_{kj}^{+} - \rho_{kj}^{-}) \qquad j \neq k$$



Summary

- Learning minimizes a cost function
- The way how this is accoplished is called the leaning type (rule)
- 5 candidates: Widrow-Hoff (or delta) rule, memory based, Hebbian, competetive, Boltzmann
- Sometimes we find an implicit changing of weights.