CS623: Introduction to Computing with Neural Nets *(lecture-10)*

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Tiling Algorithm (repeat)

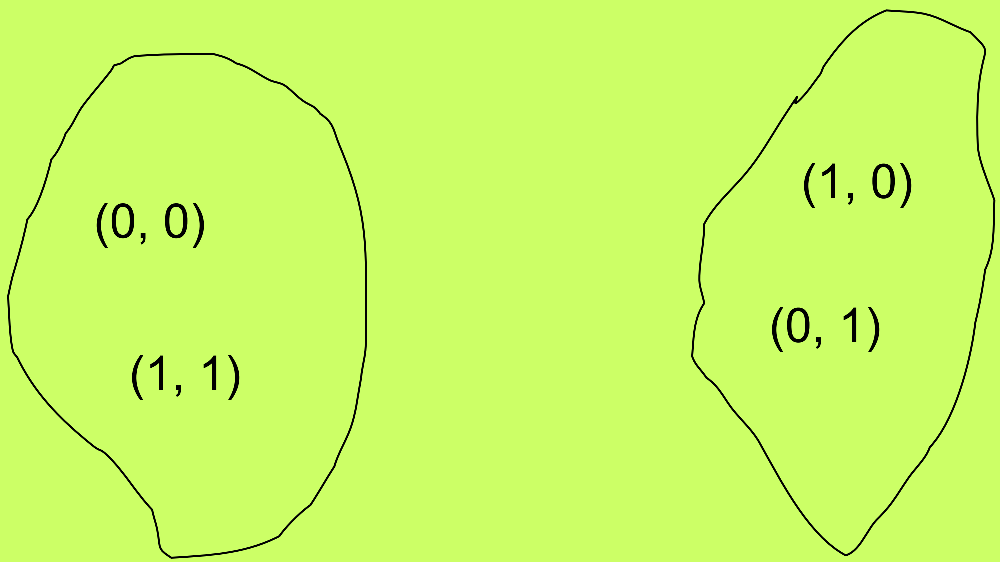
- A kind of divide and conquer strategy
- Given the *classes in the data*, run the perceptron training algorithm
- If linearly separable, convergence without any hidden layer
- If not, do as well as you can (*pocket* algorithm*)*
- This will produce classes with misclassified points

Tiling Algorithm *(contd)*

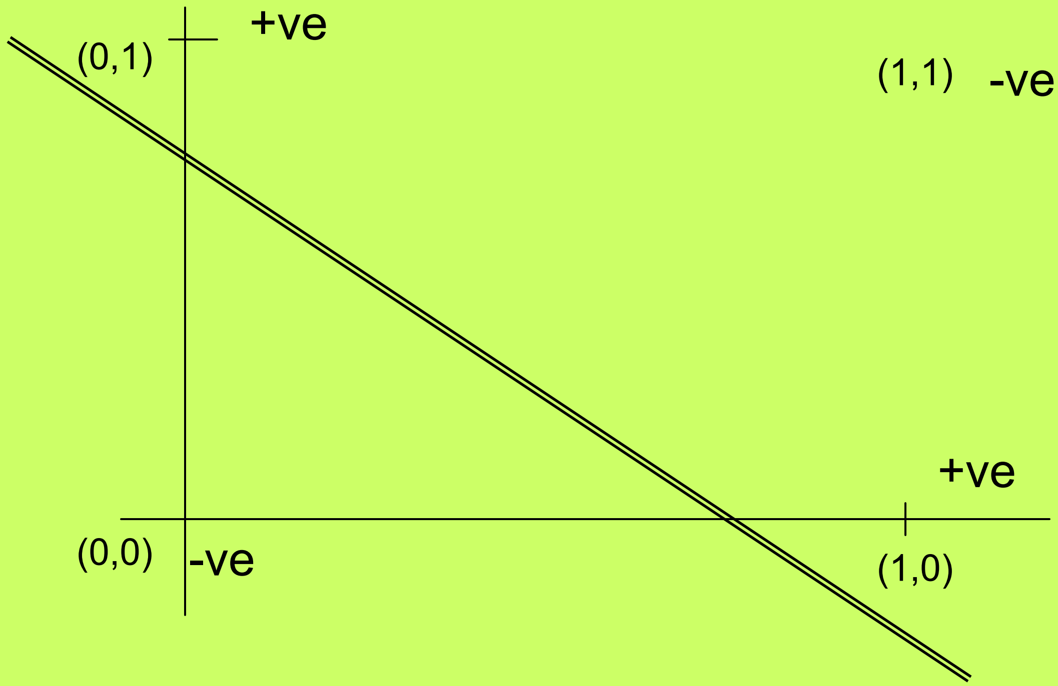
- Take the class with misclassified points and break into subclasses which contain no *outliers*
- Run PTA again *after recruiting* the required number of *perceptrons*
- Do this until *homogenous* classes are obtained
- Apply the same procedure for the first hidden layer to obtain the second hidden layer and so on

Illustration

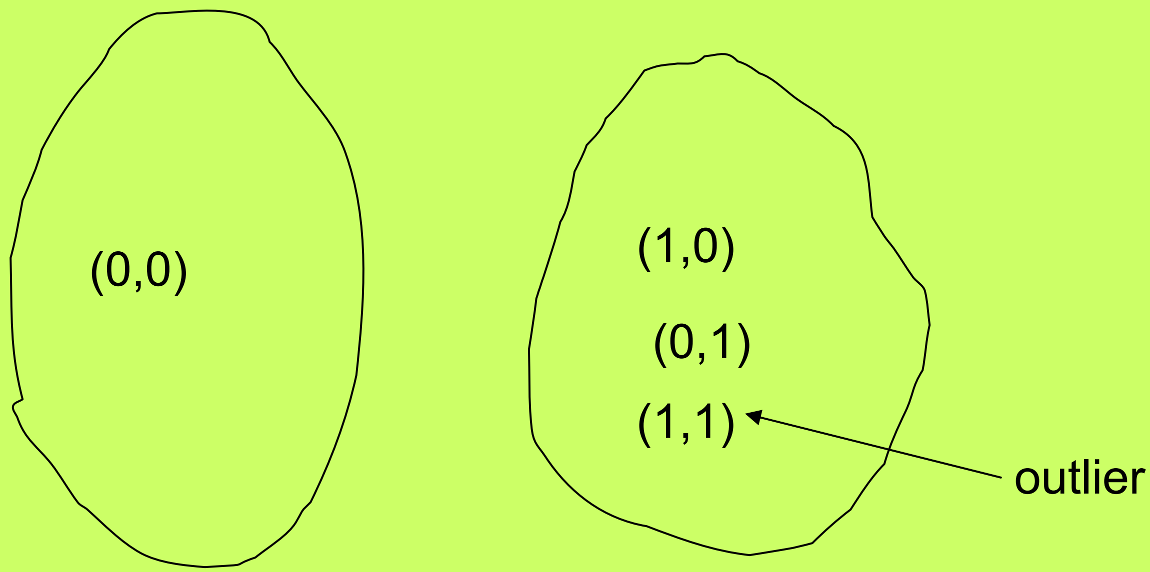
- XOR problem
- Classes are



As best a classification as possible

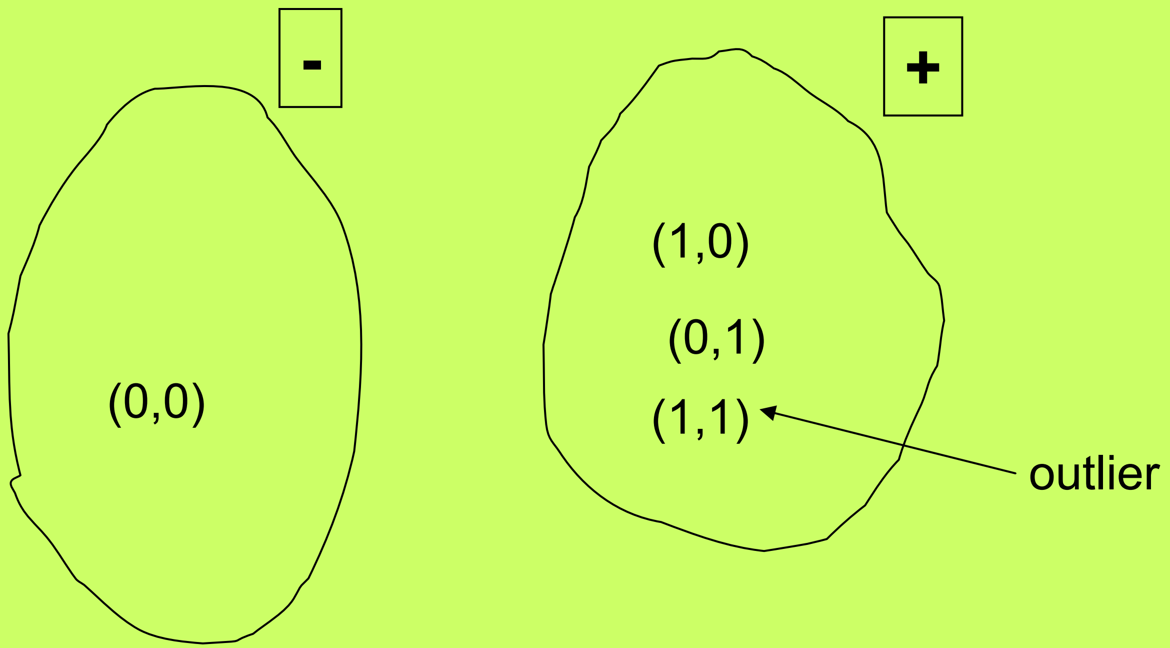


Classes with error



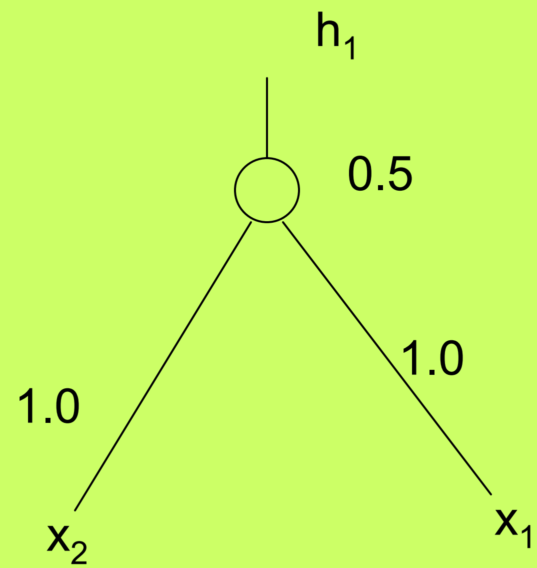
How to achieve this classification

- Give the labels as shown: eqv to an OR problem

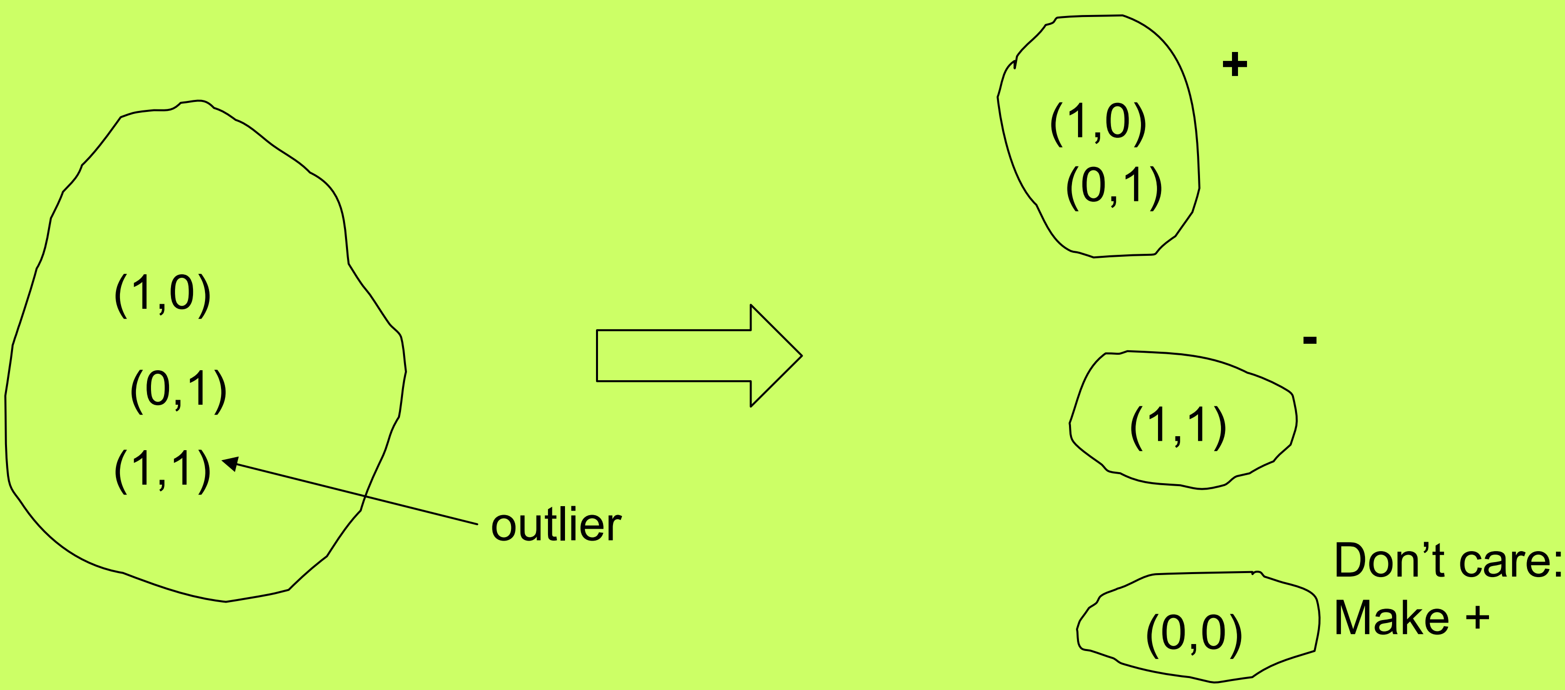


The partially developed n/w

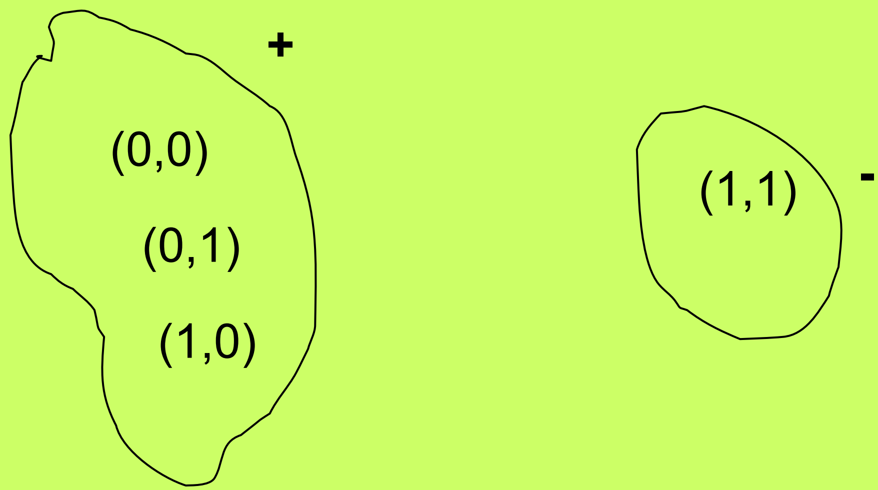
- Get the first neuron in the hidden layer, which computes OR



Break the incorrect class

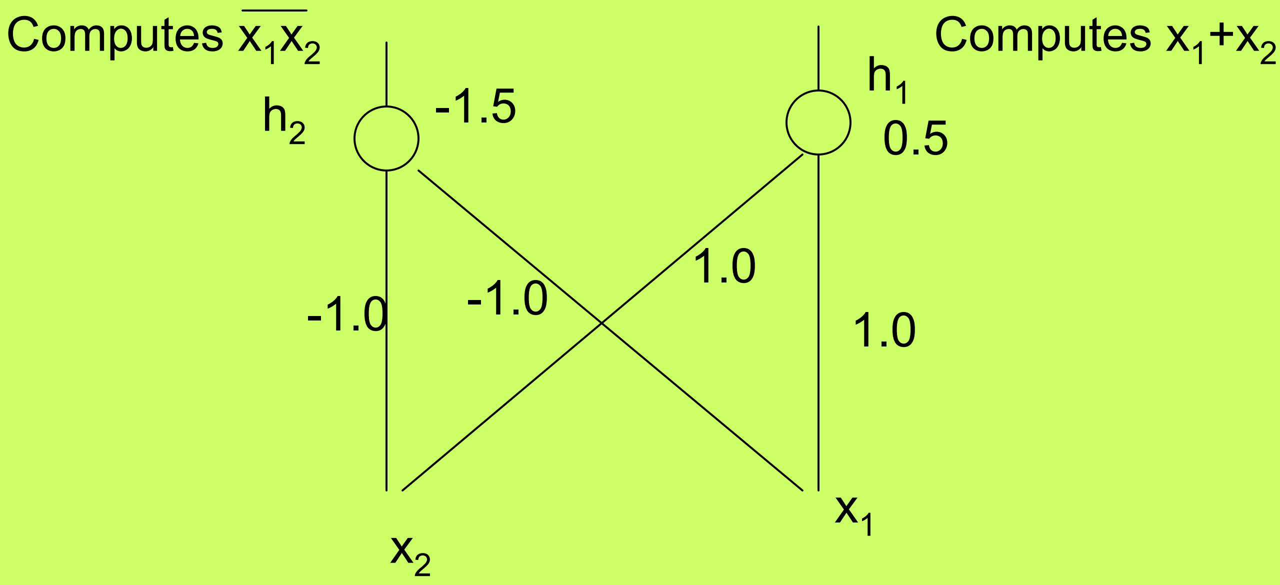


Solve classification for h_2



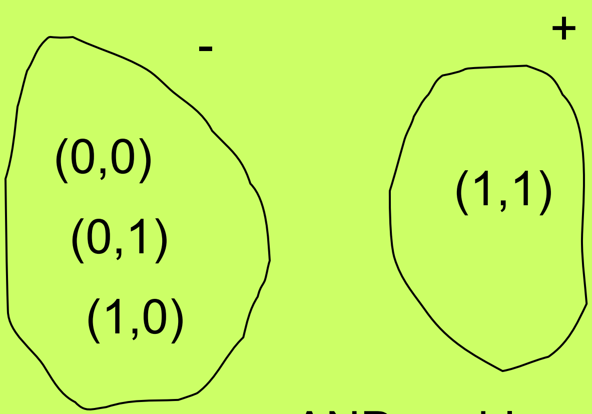
This is $\overline{x_1 x_2}$

Next stage of the n/w



Getting the output layer

- Solve a tiling algo problem for the hidden layer

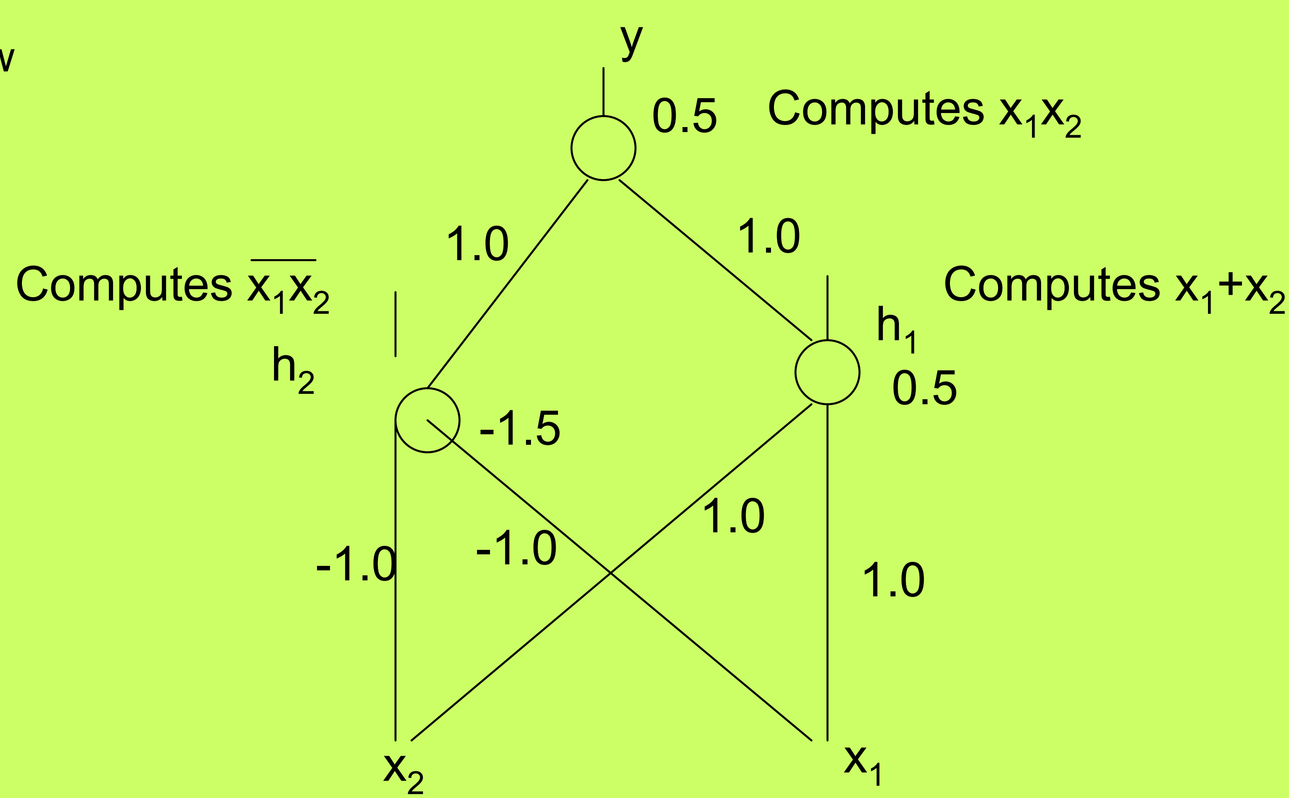


AND problem

x_2	x_1	h_1 (x_1+x_2)	h_1 $\overline{x_1 x_2}$	y
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

Final *n/w*

- AND n/w



Lab exercise

Implement the tiling algorithm and run it for

1. XOR
2. Majority
3. IRIS data

Hopfield net

- Inspired by associative memory which means memory retrieval is not by address, but by part of the data.
- Consists of

  N neurons fully connected with

  symmetric weight strength $w_{ij} = w_{ji}$
- No self connection. So the weight matrix is 0-diagonal and symmetric.
- Each computing element or neuron is a linear threshold element with threshold = 0.

Computation

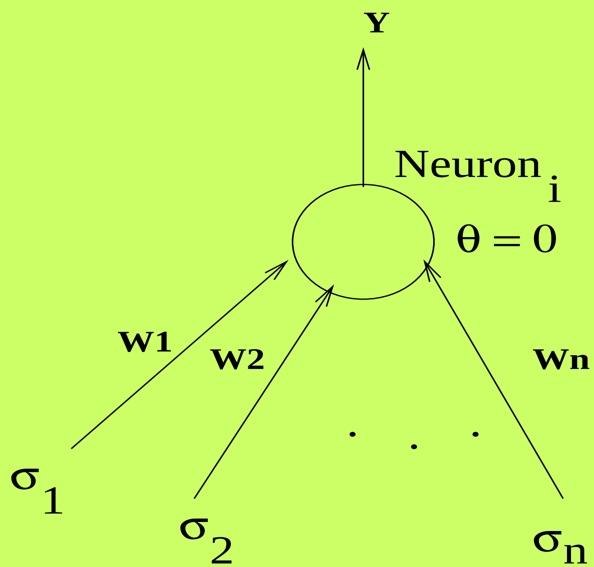


Figure: A neuron in the Hopfield Net.

Example

$w_{12} = w_{21} = 5$

$w_{13} = w_{31} = 3$

$w_{23} = w_{32} = 2$

<u>At time *t=0*</u>

$s_1(t) = 1$

$s_2(t) = -1$

$s_3(t) = 1$

Unstable state: Neuron 1 will flip.

A stable pattern is called an attractor for the net.

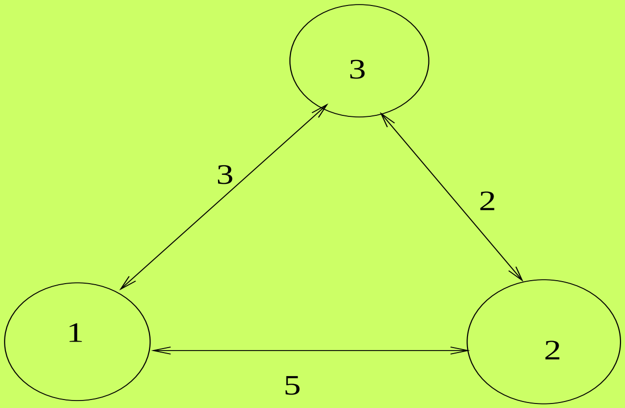


Figure: An example Hopfield Net

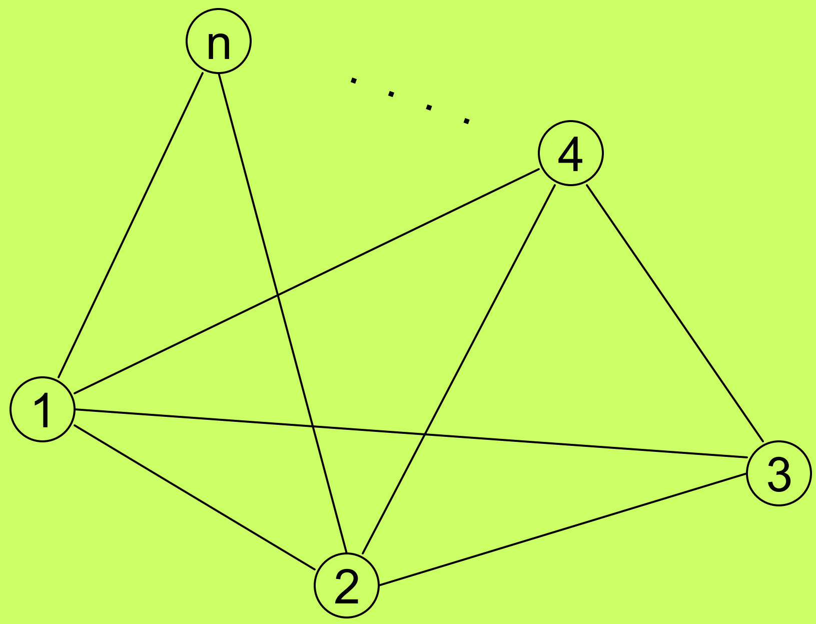
Stability

- Asynchronous mode of operation: at any instant a randomly selected neuron compares the net input with the threshold.
- In the *synchronous* mode of operation all neurons update themselves simultaneously at any instant of time.
- Since there are feedback connections in the Hopfield Net the question of *stability* arises. At every time instant the network evolves and finally settles into a stable state.
- How does the Hopfield Net function as *associative* memory ?
- One needs to store or stabilize a vector which is the memory element.

Energy consideration

- Stable patterns correspond to minimum energy states.
- Energy at state $<x_1, x_2, x_3, \ldots, x_n>$
- $$E = -1/2\sum_j\sum_{j<>i}w_{ji}x_ix_j$$
- Change in energy always comes out to be negative in the asynchronous mode of operation. Energy *always* decreases.
- Stability ensured.

Hopfield Net is a fully connected network



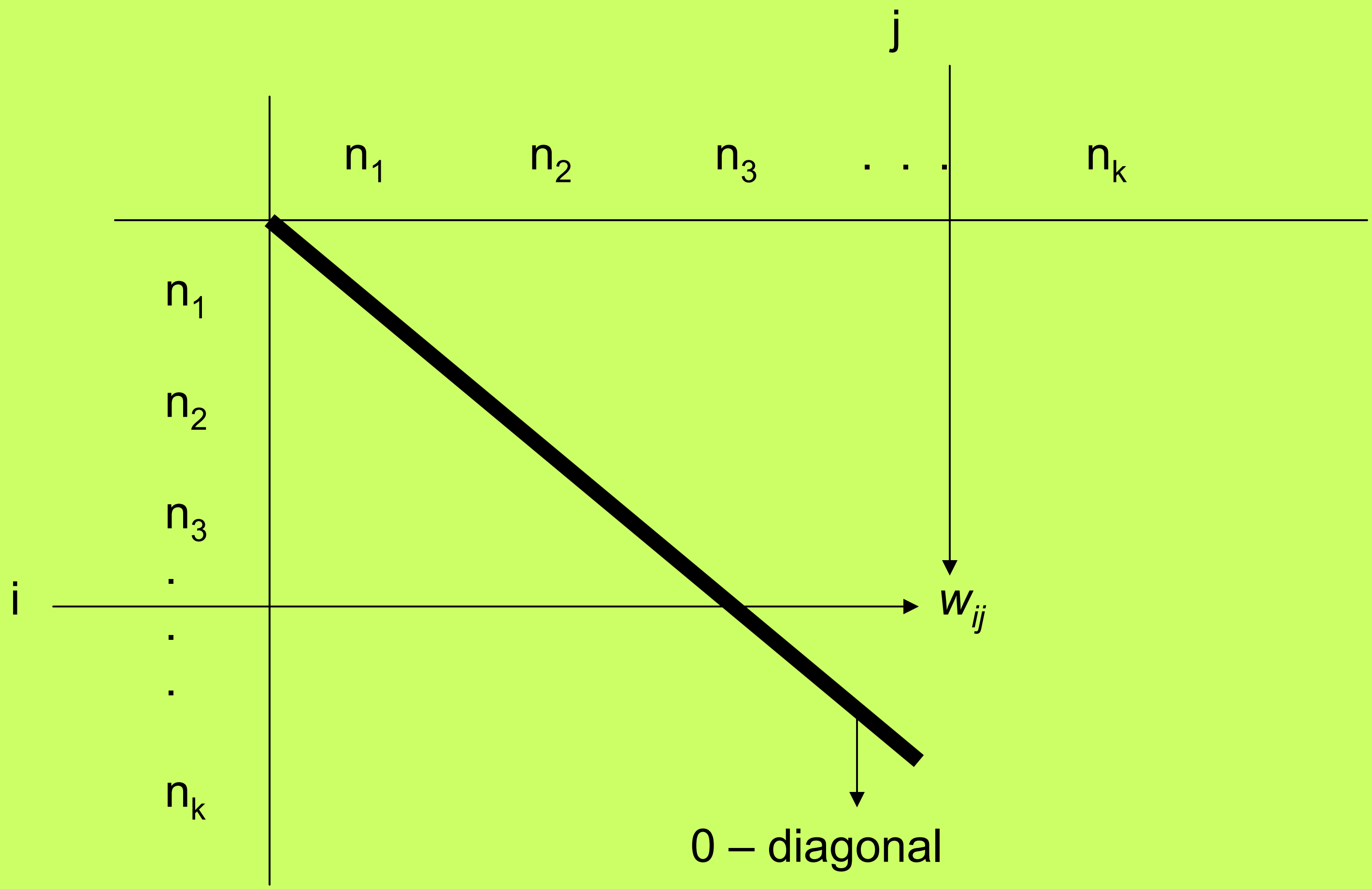
- i^{th} neuron is connected to *(n-1)* neurons

Concept of Energy

- Energy at state *s* is given by the equation:

$$
\begin{aligned}
E(s) = -\big[& w_{12}x_1x_2 + w_{13}x_1x_3 + \ldots + w_{1n}x_1x_n \\
& + w_{23}x_2x_3 + \ldots + w_{2n}x_2x_n + \\
& \vdots \\
& + w_{(n-1)n}x_{(n-1)}x_n \big]
\end{aligned}
$$

Connection matrix of the network, 0-diagonal and symmetric

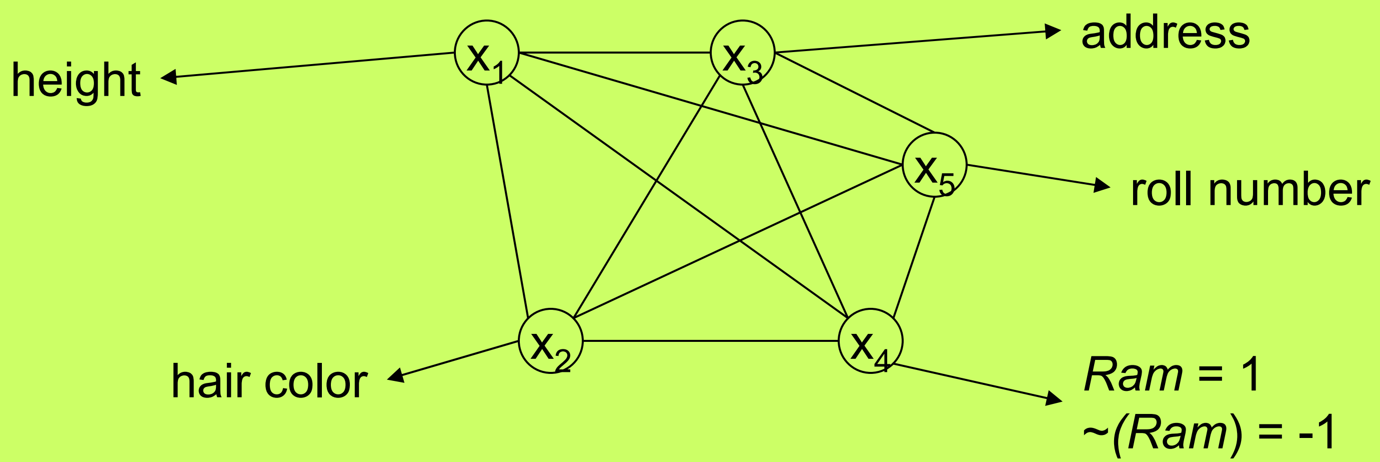


State Vector

- Binary valued vector: value is either 1 or -1

  $X = <x_n \; x_{n-1} \quad . \; . \; . \; . \quad x_3 \quad x_2 \quad x_1>$

- *e.g.* Various attributes of a student can be represented by a state vector



Relation between weight vector W and state vector X

$$W \cdot X^T$$

W → Weight vector

X^T → Transpose of state vector

For example, in figure 1,
At time *t = 0*, state of the neural network is:
s(0) = <1, -1, 1> and corresponding vectors are as shown.

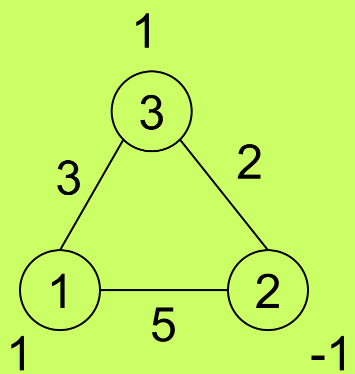


Fig. 1

$$W = \begin{bmatrix} 0 & 5 & 3 \\ 5 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \qquad X^T = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$W \cdot X^T = \begin{bmatrix} 0 & 5 & 3 \\ 5 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$W.X^T$ gives the inputs to the neurons at the next time instant

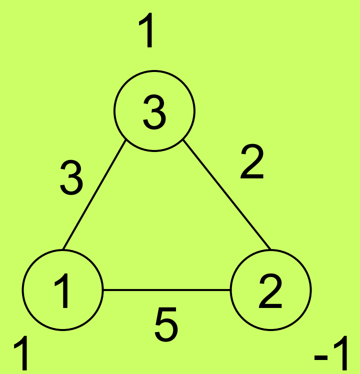
$$W \cdot X^T = \begin{bmatrix} \mathbf{0} & \mathbf{5} & \mathbf{3} \\ \mathbf{5} & \mathbf{0} & \mathbf{2} \\ \mathbf{3} & \mathbf{2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

$$= \begin{bmatrix} -2 \\ 7 \\ 1 \end{bmatrix}$$

$$\mathrm{sgn}(W.X^T) = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

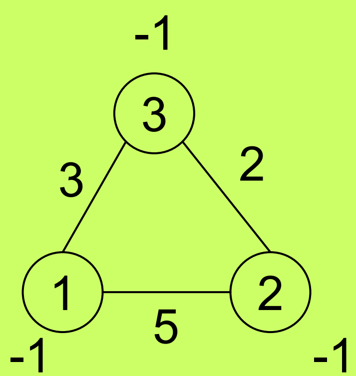
This shows that the n/w will change state

Energy Consideration



At time *t = 0*, state of the neural network is:
s(0) = <1, -1, 1>

- E(0) = -[(5*1*-1)+(3*1* 1)+(2*-1*1)] = 4



The state of the neural network under stability is *<-1, -1, -1>*

E(stable state) = - -[(5*-1*-1)+(3*-1* -1)+(2*-1*-1)] = -10

State Change

- $s(0) = <1, -1, 1\ldots>$
- $s(1)$ = compute by comparing and summing
- $x_1(t=1) = sgn[\Sigma_{j=2}^{n} w_{1j} x_j]$

  $= 1$ if $\Sigma_{j=2}^{n} w_{1j} x_j > 0$

  $= -1$ otherwise

Theorem

- In the asynchronous mode of operation, the energy of the Hopfield net <u>always</u> decreases.
- Proof:

$$
\begin{aligned}
E(t_1) = -\big[& w_{12}x_1(t_1)x_2(t_1) + w_{13}x_1(t_1)x_3(t_1) + \ldots + w_{1n}x_1(t_1)x_n(t_1) \\
& + w_{23}x_2(t_1)x_3(t_1) + \ldots + w_{2n}x_2(t_1)x_n(t_1) + \\
& \vdots \\
& + w_{(n-1)n}x_{(n-1)}(t_1)x_n(t_1) \big]
\end{aligned}
$$

Proof

- Let neuron 1 change state by summing and comparing
- We get following equation for energy

$$
\begin{aligned}
E(t_2) = -\big[& w_{12}x_1(t_2)x_2(t_2) + w_{13}x_1(t_2)x_3(t_2) + \ldots + w_{1n}x_1(t_2)x_n(t_2) \\
& + w_{23}x_2(t_2)x_3(t_2) + \ldots + w_{2n}x_2(t_2)x_n(t_2) + \\
& \vdots \\
& + w_{(n-1)n}x_{(n-1)}(t_2)x_n(t_2) \big]
\end{aligned}
$$

Proof: *note that only neuron 1 changes state*

$$\Delta E = E(t_2) - E(t_1)$$

$$= -\{[w_{12}x_1(t_2)x_2(t_2) + w_{13}x_1(t_2)x_3(t_2) + \ldots + w_{1n}x_1(t_2)x_n(t_2)]$$

$$-[w_{12}x_1(t_1)x_2(t_1) + w_{13}x_1(t_1)x_3(t_1) + \ldots + w_{1n}x_1(t_1)x_n(t_1)]\}$$

$$= -\sum_{j=2}^{n} w_{1j}\left[x_1(t_2) \cdot x_j(t_2) - x_1(t_1) \cdot x_j(t_1)\right]$$

Since only neuron 1 changes state, $x_j(t_1)=x_j(t_2)$, j=2, 3, 4, …n, and hence

$$= \sum_{j=2}^{n} \left[w_{1j} \cdot x_j(t_1)\right]\left[x_1(t_1) - x_1(t_2)\right]$$

Proof *(continued)*

$$= \sum_{j=2}^{n} \underbrace{\left[w_{1j} \cdot x_j(t_1) \right]}_{(S)} \underbrace{\left[x_1(t_1) - x_1(t_2) \right]}_{(D)}$$

- Observations:
  - When the state changes from -1 to 1, *(S) has to be +ve and (D) is –ve;* so ΔE becomes negative.
  - When the state changes from 1 to -1, *(S) has to be -ve and (D) is +ve;* so ΔE becomes negative.
- Therefore, Energy for any state change always decreases.

The Hopfield net has to “converge” in the asynchronous mode of operation

- As the energy *E* goes on decreasing, it has to hit the bottom, since the weight and the state vector have finite values.
- That is, the Hopfield Net has to converge to an energy minimum.
- Hence the Hopfield Net reaches stability.