Hebbian Learning Rule

This rule, one of the oldest and simplest, was introduced by Donald Hebb in his book The Organization of Behavior in 1949.

It is a kind of feed-forward, unsupervised learning.

→ Hebb rule can be used for pattern classification, pattern association etc.

→ Logic functions (AND, OR, XOR) can be implemented using hebb network.

→ Hebb rule is best suited on bipolar data over binary data.

Implement logic 'AND' operation using 'Hebb Net'.

X1	X2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Bipolar inputs

Initialize weights and bias i.e. w1=w2=b=0

Update weights

1. For input pair X1,X2,b = (1,1,1)

X1	X2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

wi(new) = wi(old) + (xi * y)

w1(new) = w1(old) + (x1*y)

= 0 + 1*1 =1

w2(new) = w2(old) + (x2*y)

= 0 + 1*1 =1

b(new) = b(old) + y

Take the w1(old),w2(old),b(old) as previously updated for input pair 1st w1=1, w2=1, b=1

2. For input pair (1,-1,-1)

w1(new) = w1(old) + (x1*y)
= 1 + 1 * (-1) = 0
w2(new) = w2(old) + (x2*y)
= 1 + (-1)*(-1) = 2
b(new) = b(old) + y
= 1 + (-1) =0

X1	X2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

3. For input pair (-1,1,1)

w1(new) = w1(old) + (x1*y)
= 0 + (-1) * (-1) = 1
w2(new) = w2(old) + (x2*y)
= 2 + (1)*(-1) = 1
b(new) = b(old) + y
= 0 + (-1) = -1

4. For input pair (-1,-1,1)

w1(new) = w1(old) + (x1*y)
= 1 + (-1) * (-1) = 2
w2(new) = w2(old) + (x2*y)
= 1 + (-1)*(-1) = 2
b(new) = b(old) + y
= (-1) + (-1) =-2

X1	X2	b	Y	w1	w2	b
1	1	1	1	1	1	1
1	-1	1	-1	0	2	0
-1	1	1	-1	1	1	-1
-1	-1	1	-1	2	2	-2

Net Value

Net value = w1x1 + w2x2 + b

1. For input (1,1)
Net = 2*1 + 2*1 - 2 = 2
2. For input (1,-1)
Net = 2*1 + 2*(-1) - 2 = -2
3. For input (-1,1)
Net = 2*(-1) + 2*1 - 2 = -2
4. For input (-1,-1)
Net = 2*(-1) + 2*(-1) - 2 = -6

Final values	w1 = 2	w2 = 2	b=-2
X1	X2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Net Value

Net value = w1x1 + w2x2 + b

1. For input (1,1)
Net = 2*1 + 2*1 - 2 = 2
2. For input (1,-1)
Net = 2*1 + 2*(-1) - 2 = -2
3. For input (-1,1)
Net = 2*(-1) + 2*1 - 2 = -2
4. For input (-1,-1)
Net = 2*(-1) + 2*(-1) - 2 = -6

Final values	w1 = 2	w2 = 2	b=-2
X1	X2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

$$f(net) = \begin{cases} 1, & Net>=2 \\ -1, & Net<2 \end{cases}$$

Hopfield Network

Hopfield network is a special kind of neural network whose response is different from other neural networks. It is calculated by converging iterative process. It has just one layer of neurons relating to the size of the input and output, which must be the same.

In 1982, **John Hopfield** introduced an artificial neural network to collect and retrieve memory like the human brain. Here, a neuron is either on or off the situation. The state of a neuron(on +1 or off 0) will be restored, relying on the input it receives from the other neuron. A Hopfield network is at first prepared to store various patterns or memories. Afterward, it is ready to recognize any of the learned patterns by uncovering partial or even some corrupted data about that pattern, i.e., it eventually settles down and restores the closest pattern. Thus, similar to the human brain, the Hopfield model has stability in pattern recognition.

A Hopfield network is a single-layered and recurrent network in which the neurons are entirely connected, i.e., each neuron is associated with other neurons. If there are two neurons i and j, then there is a connectivity weight $\mathbf{w_{ij}}$ lies between them which is symmetric $\mathbf{w_{ij} = w_{ji}}$.

With zero self-connectivity, $\mathbf{W_{ii}}$ **=0** is given below. Here, the given three neurons having values **i = 1, 2, 3** with values $\mathbf{X_i}$**=±1** have connectivity weight $\mathbf{W_{ij}}$.

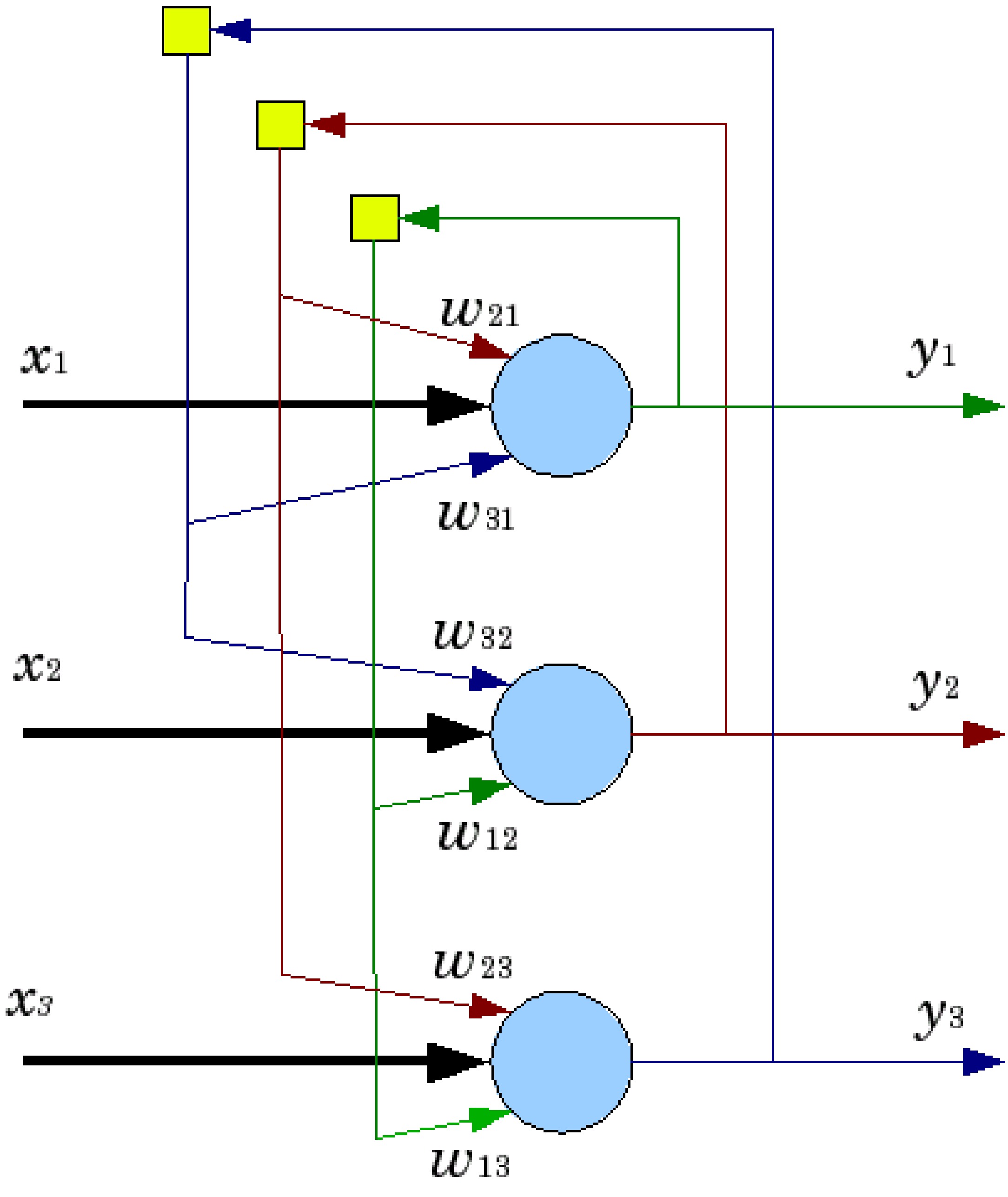
We have two different approaches to update the nodes:

Synchronously:

In this approach, the update of all the nodes taking place simultaneously at each time.

Asynchronously:

In this approach, at each point of time, update one node chosen randomly or according to some rule. Asynchronous updating is more biologically realistic.



x_1
x_2
x_3
w_{21}
w_{31}
w_{32}
w_{12}
w_{23}
w_{13}
y_1
y_2
y_3

Hopfield network finds a broad application area in image restoration and segmentation.

Hopfield neural networks are applied **to solve many optimization problems**. In medical image processing, they are applied in the continuous mode to image restoration, and in the binary mode to image segmentation and boundary detection.

1. **Memory capacity**. The number m of training patterns should be about the number of neurons n, or less

means that the memorizing capacity of a Hopfield network is severely limited. Catastrophic “forgetting" may occur if we try to memorize more patterns than the network is supposed to handle.

2. Discrepancy limitation. The new pattern to be recognized as one of the training patterns should not differ from any training pattern by more than about 25%.

3. Orthogonality between patterns. The more orthogonal (dissimilar) the training patterns, the better the recognition.

5. Weight symmetry. The weight matrix has to be symmetrical in order for the network to reach an equilibrium.

6· Local minima problem.A major disadvantage of the Hopfield network is that it can rest in a local minimum state instead of a global