The weights for a Hopfield network are determined directly from the training data without need for training

- To store patterns s(p), p=1,2,…P

bipolar: $s(p)=(s_1(p),\ldots..s_i(p),\ldots.s_n(p))$ n dimensional P patterns

$$\boldsymbol{w_{ij}} = \sum_{\boldsymbol{p}} \boldsymbol{s_i}(\boldsymbol{p})\boldsymbol{s_j}(\boldsymbol{p}) \quad \boldsymbol{i} \neq \boldsymbol{j}$$

$$\boldsymbol{w_{ii}} = 0$$

$$w_{ij} = \sum_{p=1}^{P} x_p^{(i)} x_p^{(j)}$$

$$W_{ij} = W_{ji}$$

$$W_{ii} = 0$$

same as Hebbian rule (with zero diagonal)

binary: $$\boldsymbol{w_{ij}} = \sum_{\boldsymbol{p}} (2\boldsymbol{s_i}(\boldsymbol{p})-1)(2\boldsymbol{s_j}(\boldsymbol{p})-1) \quad \boldsymbol{i} \neq \boldsymbol{j}$$

$$\boldsymbol{w_{ii}} = 0$$

converting s(p) to bipolar [0 → -1] when constructing W.

So let's consider the case where we want our 5 node Hopfield net to store both the pattern $V^1 = (0\ 1\ 1\ 0\ 1)$ and another pattern $V^2 = (1\ 0\ 1\ 0\ 1)$.

WEIGHT MATRIX TO STORE V^1 = 01101 ,

$V_1 = 0$, $V_2 = 1$, $V_3 = 1$, $V_4 = 0$, and $V_5 = 1$.

0	W_{12}	W_{13}	W_{14}	W_{15}
W_{21}	0	W_{23}	W_{24}	W_{25}
W_{31}	W_{32}	0	W_{34}	W_{35}
W_{41}	W_{42}	W_{43}	0	W_{45}
W_{51}	W_{52}	W_{53}	W_{54}	0

So let's consider the case where we want our 5 node Hopfield net to store both the pattern $V^1 = (0\ 1\ 1\ 0\ 1)$ and another pattern $V^2 = (1\ 0\ 1\ 0\ 1)$.

WEIGHT MATRIX TO STORE FIRST PATTERN V^1 = 01101 ,

$V_1 = 0$, $V_2 = 1$, $V_3 = 1$, $V_4 = 0$, and $V_5 = 1$.

$$W_{ij} = (2V_i - 1)(2V_j - 1) = (2V_j - 1)(2V_i - 1) = W_{ji}$$

$W_{12} = (2V_1 - 1)(2V_2 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$
$W_{13} = (2V_1 - 1)(2V_3 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$
$W_{14} = (2V_1 - 1)(2V_4 - 1) = (0 - 1)(0 - 1) = (-1)(-1) = 1$
$W_{15} = (2V_1 - 1)(2V_5 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$
$W_{23} = (2V_2 - 1)(2V_3 - 1) = (2 - 1)(2 - 1) = (1)(1) = 1$
$W_{24} = (2V_2 - 1)(2V_4 - 1) = (2 - 1)(0 - 1) = (1)(-1) = -1$
$W_{25} = (2V_2 - 1)(2V_5 - 1) = (2 - 1)(2 - 1) = (1)(1) = 1$
$W_{34} = (2V_3 - 1)(2V_4 - 1) = (2 - 1)(0 - 1) = (1)(-1) = -1$
$W_{35} = (2V_3 - 1)(2V_5 - 1) = (2 - 1)(2 - 1) = (1)(1) = 1$
$W_{45} = (2V_4 - 1)(2V_5 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$

Replace 0 by -1 V = (0 1 1 01 1) becomes (-1 1 1 -1 1 1) and find weights

So now our weight matrix looks like this:

0	-1	-1	1	-1
W_{21}	0	1	-1	1
W_{31}	W_{32}	0	-1	1
W_{41}	W_{42}	W_{43}	0	-1
W_{51}	W_{52}	W_{53}	W_{54}	0

WEIGHT MATRIX FOR storing first vector : 0 1 1 0 1

By reflecting about the diagonal,

0	-1	-1	1	-1
-1	0	1	-1	1
-1	1	0	-1	1
1	-1	-1	0	-1
-1	1	1	-1	0

WEIGHT MATRIX TO STORE V^2 = 10101 ,

$V_1 = 1$, $V_2 = 0$, $V_3 = 1$, $V_4 = 0$, and $V_5 = 1$.

$$W_{ij} = (2V_i - 1)(2V_j - 1) = (2V_j - 1)(2V_i - 1) = W_{ji}$$

$$W_{12} = (2V_1 - 1)(2V_2 - 1) = (2 - 1)(0 - 1) = (1)(-1) = -1$$
$$W_{13} = (2V_1 - 1)(2V_3 - 1) = (2 - 1)(2 - 1) = (1)(1) = 1$$
$$W_{14} = (2V_1 - 1)(2V_4 - 1) = (2 - 1)(0 - 1) = (1)(-1) = -1$$
$$W_{15} = (2V_1 - 1)(2V_5 - 1) = (2 - 1)(2 - 1) = (1)(1) = 1$$
$$W_{23} = (2V_2 - 1)(2V_3 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$$
$$W_{24} = (2V_2 - 1)(2V_4 - 1) = (0 - 1)(0 - 1) = (-1)(-1) = 1$$
$$W_{25} = (2V_2 - 1)(2V_5 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$$
$$W_{34} = (2V_3 - 1)(2V_4 - 1) = (2 - 1)(0 - 1) = (1)(-1) = -1$$
$$W_{35} = (2V_3 - 1)(2V_5 - 1) = (2 - 1)(2 - 1) = (1)(1) = 1$$
$$W_{45} = (2V_4 - 1)(2V_5 - 1) = (0 - 1)(2 - 1) = (-1)(1) = -1$$

Replace 0 by -1 V = (1 0 1 0 1) becomes (1 -1 1 -1 1) and find weights

WEIGHT MATRIX FOR 0 1 1 0 1

By reflecting about the diagonal,

0	-1	-1	1	-1
-1	0	1	-1	1
-1	1	0	-1	1
1	-1	-1	0	-1
-1	1	1	-1	0

the weight matrix for the pattern (1 0 1 0 1):

0	-1	1	-1	1
-1	0	-1	1	-1
1	-1	0	-1	1
-1	1	-1	0	-1
1	-1	1	-1	0

ADDITION GIVES WEIGHT MATRIX TO STORE TWO PATTERNS AS :

0	-2	0	0	0
-2	0	0	0	0
0	0	0	-2	2
0	0	-2	0	-2
0	0	2	-2	0

train this network for following patterns

Pattern 1:-

Pattern 2:-

Pattern 3:-

Node 1 Node 2 Node 3

Number of nodes = dimension of pattern =3 in this case, Weight matrix dimension =3x3

$$W = \begin{bmatrix} 0 & -1 & -3 \\ -1 & 0 & 1 \\ -3 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} p1: -1 & -1 & 1 \\ p2: \ \ 1 & -1 & -1 \\ p3: -1 & 1 & 1 \end{bmatrix}$$

w_{12} : product of first and second columns for all patterns

W_{32} : product of third and second column for all patterns

$$w_{11} = 0$$

$$w_{12} = (-1)(-1) + (1)(-1) + (-1)(1) = -1$$

$$w_{13} = (-1)(1) + (1)(-1) + (-1)(1) = -3$$

$$w_{21} = (-1)(-1) + (-1)(1) + (1)(-1) = -1$$

$$w_{22} = 0$$

$$w_{23} = (-1)(1) + (-1)(-1) + (1)(1) = 1$$

$$w_{33} = 0$$

The behavior of a Hopfield network can depend on the update order.

- Computations can oscillate if neurons are updated in parallel.
- Computations always converge if neurons are updated sequentially.

When a "cue" – noisy pattern is given, there are now two ways to update the nodes:

Asynchronously: At each point in time, update one node chosen randomly or according to some rule, like even nodes/odd nodes.

Asynchronous updating is more biologically realistic

Synchronously: Every time, update all nodes together.

Asynchronous: Only one unit is updated at a time. This unit can be picked at random, or a pre-defined order can be imposed from the very beginning.

Synchronous: All units are updated at the same time.

Draw Asynchronous Recurrent Binary Hopfield Network for six dimensional input/output vector

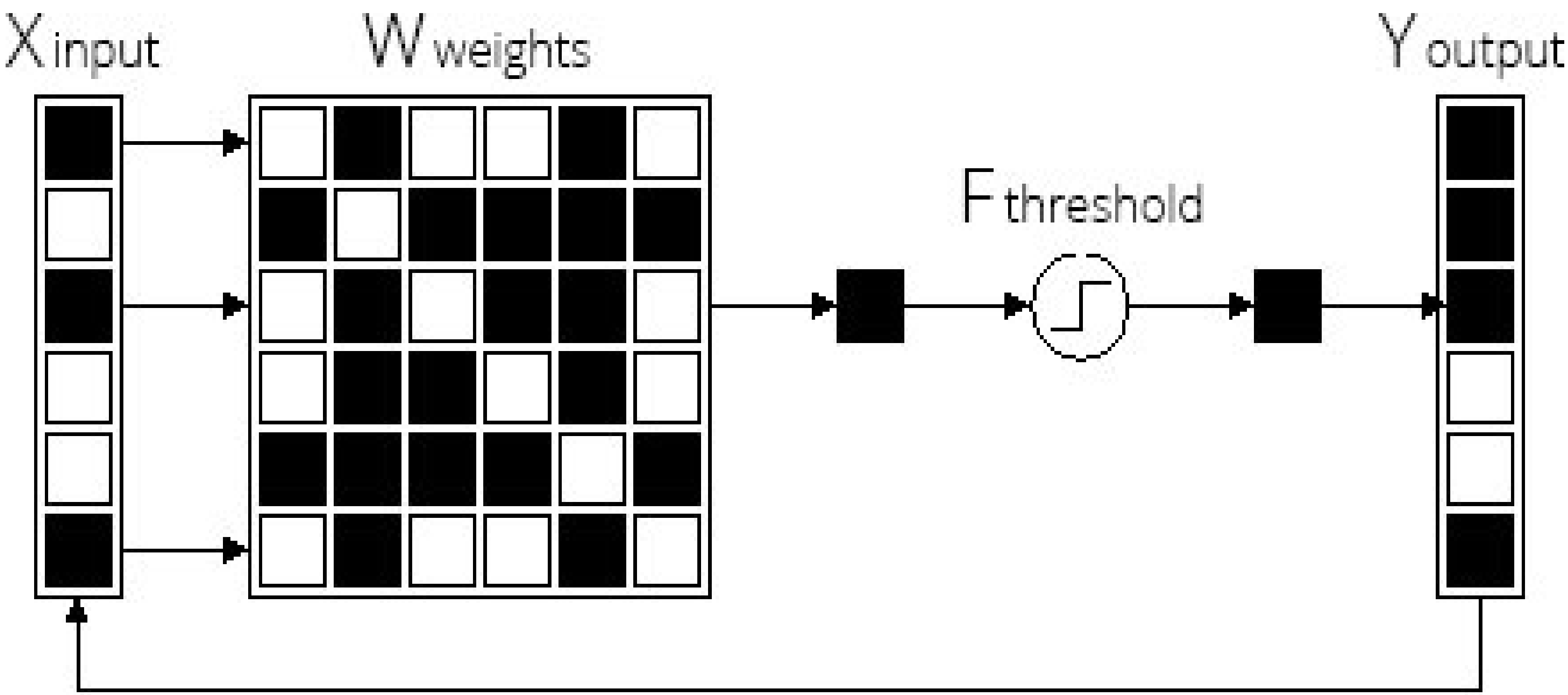


Fig 1. Asynchronous recurrent binary Hopfield network

So let's consider the case where we want our 5 node Hopfield net to store both the pattern $V^1 = (0\ 1\ 1\ 0\ 1)$ and another pattern $V^2 = (1\ 0\ 1\ 0\ 1)$.

weight matrix

node 1	node 2	node 3	node 4	node 5
0	−2	0	0	0
−2	0	0	0	0
0	0	0	−2	2
0	0	−2	0	−2
0	0	2	−2	0

start at the state (1 1 1 1 1) and see where it goes.

Testing

update the nodes in the fixed order 3, 1, 5, 2, 4, $\boxed{3, 1, 5, 2, 4,}$ etc.

start at the state (1 1 1 1 1) and see where it goes.

Testing

update the nodes in the fixed order 3, 1, 5, 2, 4, $\boxed{3, 1, 5, 2, 4,}$ etc.

$V^1 = (0\ 1\ 1\ 0\ 1)$ $V^2 = (1\ 0\ 1\ 0\ 1)$.

STEADY STATES

Testing again give 3,1,5,2,4,if no change- converged

- update node 3 -

  $V_3in = (0\ 0\ 0\ {-2}\ 2) * (1\ 1\ 1\ 1\ 1) = -2 + 2 = 0$

  since $0 >= 0$,

  $V_3 = 1$ Unchanged

- update node 1 -

  $V_1in = (0\ {-2}\ 0\ 0\ 0)\ .\ (1\ 1\ 1\ 1\ 1) = -2$

  since $-2 < 0$, $V_1 = 0$ *(it changed)*

- update node 5 -

  $V_5in = (0\ 0\ 2\ {-2}\ 0)\ .\ (\boxed{0}\ 1\ 1\ 1\ 1) = 0$

  since $0 >= 0$, $V_5 = 1$ *(it didn't change)*

- update node 2 -

  $V_2in = (-2\ 0\ 0\ 0\ 0)\ .\ (0\ 1\ 1\ 1\ 1) = 0$

  since $0 >= 0$, $V_2 = 1$ *(it didn't change)*

- update node 4 -

  $V_4in = (0\ 0\ {-2}\ 0\ {-2})\ .\ (0\ 1\ 1\ 1\ 1) = -4$

  since $-4 < 0$, $V_4 = 0$ *(it changed)* $(0\ 1\ 1\ \boxed{0}\ 1)$

weight matrix

N1	N2	N3	N4	N5
0	-2	0	0	0
-2	0	0	0	0
0	0	0	-2	2
0	0	-2	0	-2
0	0	2	-2	0

One of the stable states !

$V^1 = (0\ 1\ 1\ 0\ 1)$ $V^2 = (1\ 0\ 1\ 0\ 1)$.

Testing again give 3,1,5,2,4

Arrived at Stable state 1

- **update node 3 -**
  $V_3in = (0\ 0\ 0\ {-2}\ 2) \cdot (0\ 1\ 1\ 0\ 1) = 2$
  since $2 >= 0$, $V_3 = 1$ *(it didn't change)*

- **update node 1 -**
  $V_1in = (0\ {-2}\ 0\ 0\ 0) \cdot (0\ 1\ 1\ 0\ 1) = -2$
  since $-2 < 0$, $V_1 = 0$ *(it didn't change)*

- **update node 5 -**
  $V_5in = (0\ 0\ 2\ {-2}\ 0) \cdot (0\ 1\ 1\ 0\ 1) = 2$
  since $2 >= 0$, $V_5 = 1$ *(it didn't change)*

- **update node 2 -**
  $V_2in = ({-2}\ 0\ 0\ 0\ 0) \cdot (0\ 1\ 1\ 0\ 1) = 0$
  since $0 >= 0$, $V_2 = 1$ *(it didn't change)*
- **update node 4 -**
  $V_4in = (0\ 0\ {-2}\ 0\ {-2}) \cdot (0\ 1\ 1\ 0\ 1) = -4$
  since $-4 < 0$, $V_4 = 0$ *(it didn't change)*
- **Now we've updated each node in the net without them changing, so we can stop.**

start at the state (1 1 1 1 1) and see where it goes, when using a fixed node updating order of 2, 4, 3, 5, 1, 2, 4, 3, 5, 1(testing), etc.

update node 2 -

V_2in = (-2 0 0 0 0) . (1 1 1 1 1) = -2

since -2 < 0, V_2 = 0 *(it changed)* (1 0 1 1 1)

update node 4 -

V_4in = (0 0 -2 0 -2) . (1 0 1 1 1) = -4

since -4 < 0, V_4 = 0 *(it changed)* (1 0 1 0 1)

Update node 3 -

V_3in = (0 0 0 -2 2) . (1 0 1 0 1) = 2

since 2 >= 0, V_3 = 1 *(it didn't change)*

Update node 5 -

V_5in = (0 0 2 -2 0) . (1 0 1 0 1) = 2

since 2 >= 0, V_5 = 1 *(it didn't change)*

Update node 1 -

V_1in = (0 -2 0 0 0) . (1 0 1 0 1) = 0

since 0 >= 0, V_1 = 1 *(it didn't change)* *Went to other Steady state. If two patterns are very similar, the order in which you update the nodes can make a difference to which stable/attractor state it goes.*

weight matrix

N1	N2	N3	N4	N5
0	-2	0	0	0
-2	0	0	0	0
0	0	0	-2	2
0	0	-2	0	-2
0	0	2	-2	0

$V^1 = (0\ 1\ 1\ 0\ 1)$ $V^2 = (1\ 0\ 1\ 0\ 1)$.

STEADY STATES

Test node 2 -

$V_2in = (-2\ 0\ 0\ 0\ 0)\ .\ (1\ 0\ 1\ 0\ 1) = -2$

since $-2 < 0$, $V_2 = 0$ *(it didn't change)*

Test node 4 -

$V_4in = (0\ 0\ -2\ 0\ -2)\ .\ (1\ 0\ 1\ 0\ 1) = -4$

since $-4 < 0$, $V_4 = 0$ *(it didn't change)*

Test node 3 -

$V_4in = (0\ 0\ 0\ -2\ 2)\ .\ (1\ 0\ 1\ 0\ 1) = 2$

since$2 > 0$, $V_3 = 1$ *(it didn't change)*

Test node 5 -

$V_4in = (0\ 0\ 2\ -2\ 0)\ .\ (1\ 0\ 1\ 0\ 1) = 2$

since$2 > 0$, $V_5 = 1$ *(it didn't change)*

Test node 1 -

$V_4in = (0\ -2\ 0\ 0\ 0)\ .\ (1\ 0\ 1\ 0\ 1) = 0$

since$0 >= 0$, $V_4 = 1$ *(it didn't change)*

Now we've updated each node in the net without them changing,so we can stop.

weight matrix

N1	N2	N3	N4	N5
0	-2	0	0	0
-2	0	0	0	0
0	0	0	-2	2
0	0	-2	0	-2
0	0	2	-2	0

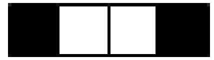
$V^1 = (0\ 1\ 1\ 0\ 1)$ $V^2 = (1\ 0\ 1\ 0\ 1)$.

STEADY STATES

Energy is associated with the state of the system.

During the recall phase of the Hopfield network the activity pattern strives to attain as low energy as possible, causing it to find local minima in the energy landscape, corresponding to stable patterns of activity.

Example:ENERGY CALCULATION

$$\mathbf{x}^1 = (1, -1, -1, 1)^T$$ 

$$\mathbf{x}^2 = (-1, 1, -1, 1)^T$$ 

$$\mathbf{W} = \begin{bmatrix} 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & -2 & 0 \end{bmatrix}$$

ENERGY= E(x) $= -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j$

½ is just a scaling factor

$E = -(1/2)\ [-2x_1 x_2 - 2x_2 x_1 - 2x_3 x_4 - 2x_4 x_3\ ]$

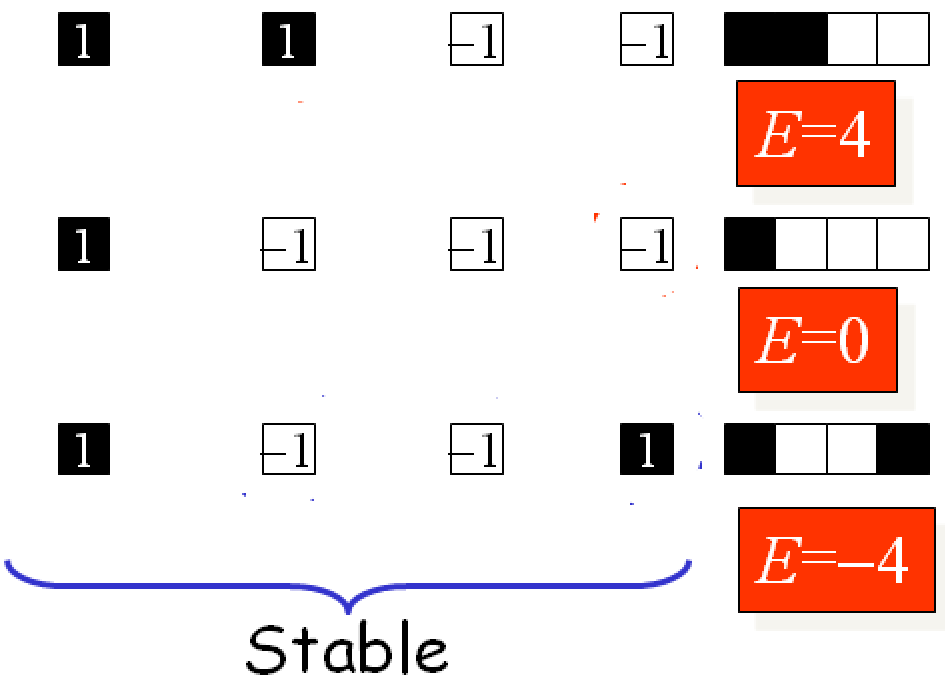
$= (-1/2)\ [-4[x_1 x_2 + x_4 x_3\ ] = 2[x_1 x_2 + x_4 x_3\ ]$

Energy for pattern 1 (1, -1, -1, 1) = 2[$x_1 x_2 + x_4 x_3$]= 2(-1 - 1) = - 4

Energy for pattern 2 (-1, 1, -1, 1) = 2[$x_1 x_2 + x_4 x_3$] = 2 (-1 - 1) = - 4

$$E(\mathbf{x}) = 2(x_1x_2 + x_3x_4)$$

$$\mathbf{W} = \begin{bmatrix} 0 & -2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$



END