Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program
605-447/625-438
Dr. Mark Fleischer

Module 8.2: Hebbian Learning Continued

This Sub-Module Covers …

- Performance evaluation.
- Proper training does not merely involve minimizing the error function using the training data.
- Must have some appropriate way of gauging how well the network performs using data it hasn't seen during the training phase.

Network to Matrix

Each output x_i for a node *i* with *N* perceptrons can be written as:

$$x_i = \sum_{j=1}^{N} w_{ij} x_j$$

But this is just the definition of the i^{th} element of a vector after a matrix/vector multiplication!

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k$$

where index *k* corresponds to the iterate number.

Network to Matrix

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k$$

But how should we define **W**?

Sometimes in order to answer a question relating to a problem, we must change the problem!

$$\mathbf{p} = \mathbf{W}\mathbf{p}$$

Instead of deciphering a general dynamical system, let's examine a special case --- a fixed point!

A Test Exemplar

Given some input pattern (or exemplar)

$$p = \left\{ \begin{array}{ccc} \oplus & \cdot & \cdot \\ \cdot & \oplus & \cdot \\ \cdot & \cdot & \oplus \end{array} \right\},$$

$$\text{where } \oplus = \text{ On and } \cdot = \text{ Off}$$

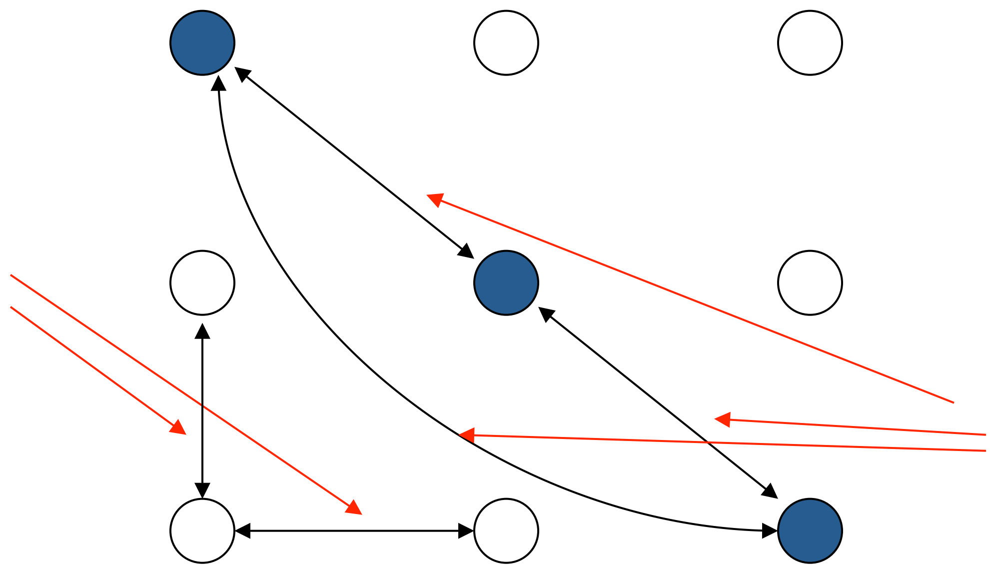
We want to design a network that will converge to a set of exemplars given a partial or noisy representation of the exemplars.

Mutual Reinforcement ala Hebbian Learning

The ● represent a 1 input, the ○ represent a 0 input.

Initially, all weights are 0.

These weights stay 0.



These weights are increased by 1.

Note: This doesn't show all the edges (weights).

How can we model this mathematically?

Change the input pattern into a vector of 1's and 0's.

$$\mathbf{p}^{\mathrm{T}} = \{\oplus \cdot \cdot \,|\, \cdot \oplus \cdot \,|\, \cdot \cdot \oplus\}$$
$$= \begin{pmatrix}100 & 010 & 001\end{pmatrix}$$

Now we compute a matrix **W** such that **Wp** = **p**.

How should such a matrix be defined?

$\mathbf{W} = ?$ $\cancel{\mathbf{W} = \mathbf{I}}$? Trivial and generic!

Hint: Must be somehow linked to the vector **p** to facilitate the calculation

$$\mathbf{W}\mathbf{p} = \mathbf{p}.$$

Can we create a matrix from the vector **p**?

$$\mathbf{pp}^{T}=\begin{pmatrix}1\\0\\0\\0\\1\\0\\0\\0\\1\end{pmatrix}(100010001)=\left(\begin{array}{ccc|ccc|ccc}1&0&0&0&1&0&0&0&1\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\\hline 0&0&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0&1\\0&0&0&0&0&0&0&0&0\\\hline 0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0&1\end{array}\right)$$

We will need to get rid of the 1's on the diagonal. Can't have a node reinforce itself. Why?

We will create/introduce a function Z which operates on matrices and sets diagonal elements to 0.

So let $\Delta\mathbf{W} = Z\left(\mathbf{p}\mathbf{p}^{\mathrm{T}}\right)$ and

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + Z\left(\mathbf{p}\mathbf{p}^{\mathrm{T}}\right)$$

We will also use a hard-limiting activation function: $F_h(x) = \begin{cases} 1, \text{if } x > 0 \\ 0, \text{if } x \le 0 \end{cases}$

So does $\mathbf{W}_{\text{new}}\mathbf{p} = \mathbf{p}$?

What is the Z function?

- For any 1, 0 vector **p** producing $\mathbf{pp}^T$ we want to subtract a modified identity matrix to get rid of entries on the diagonal with a 1.

E.g., p = (1 0 1) then pp^T =

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{matrix} \longrightarrow \begin{matrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{matrix}$$

Thus we must subtract the I^* = $\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$

Note the diagonal is the same as the vector p and that $I^*p = p$ for any p.

Another example

$$p^T = (1\ 0\ 1\ 0\ 1) \text{ then } pp^T = \begin{bmatrix} 1&0&1&0&1\\ 0&0&0&0&0\\ 1&0&1&0&1\\ 0&0&0&0&0\\ 1&0&1&0&1 \end{bmatrix} \qquad I^* = \begin{bmatrix} 1&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&1 \end{bmatrix}$$

$$\text{and } I^*p = \begin{bmatrix} 1&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&1 \end{bmatrix} \begin{bmatrix} 1\\0\\1\\0\\1 \end{bmatrix} = \begin{bmatrix} 1\\0\\1\\0\\1 \end{bmatrix}$$

$$\text{Define } \Delta\tilde{\mathbf{W}} = \mathbf{p}\mathbf{p}^{\mathrm{T}}; \quad \text{hence} \quad \Delta\mathbf{W} = Z\left(\mathbf{p}\mathbf{p}^{\mathrm{T}}\right)$$

$$\begin{aligned} \mathbf{W}_{\text{new}} &= \mathbf{W}_{\text{old}} + Z\left(\mathbf{p}\mathbf{p}^{\mathrm{T}}\right) \\ &= \Delta\mathbf{W} + \mathbf{W}(0) \\ &= \Delta\mathbf{W} \end{aligned}$$

$$\text{So does } F_h\left(\mathbf{W}_{\text{new}}\mathbf{p}\right) = \mathbf{p}?$$

$$\begin{aligned} F_h\left(\mathbf{W}_{\text{new}}\mathbf{p}^{\mathrm{T}}\right) &= F_h\left[\Delta\mathbf{W}\mathbf{p}\right] \\ &= F_h\left[\left(\Delta\tilde{\mathbf{W}} - \mathbf{I}_{\mathbf{n}}^{*}\right)\mathbf{p}\right] \\ &= F_h\left[\Delta\tilde{\mathbf{W}}\mathbf{p} - \mathbf{p}\right] \\ &= F_h\left[\mathbf{p}\mathbf{p}^{\mathrm{T}}\mathbf{p} - \mathbf{p}\right] \end{aligned}$$

$$
\begin{aligned}
F_h\left(\mathbf{W}_{\text{new}}\mathbf{p}\right) &= F_h\left[\mathbf{p}\mathbf{p}^{\mathrm{T}}\mathbf{p}-\mathbf{p}\right] \\
&= F_h\left[d\mathbf{p}-\mathbf{p}\right] \\
&= F_h\left[\left(d-1\right)\mathbf{p}\right]
\end{aligned}
$$

where $\mathbf{p}^{\mathrm{T}}\mathbf{p} = d$, a scalar equal to the number of 1's in the vector (do you see why?)

Note that $d \geq 1$ in non-trivial cases. Therefore, the above

$$= \mathbf{p}$$

What About Noisy Inputs?

$$I_3 = \sum_{j=1}^{9} w_{3j} x_j \text{ with } w_{33} = 0. \text{ In general, } I_i = \sum_{j=1}^{N} w_{ij} x_j$$

for N = number of neurons = number of elements in a pattern.

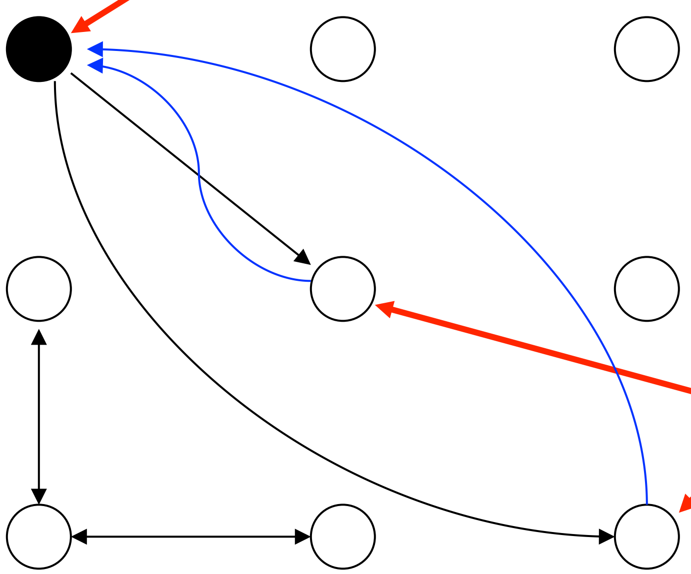
Now make a noisy version of the pattern p:

$$\tilde{p} = \left\{ \begin{matrix} \oplus & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix} \right\} = (100 \mid 000 \mid 000)^{\mathrm{T}}$$

$$\mathbf{W}_{\text{new}}\tilde{\mathbf{p}} = \left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

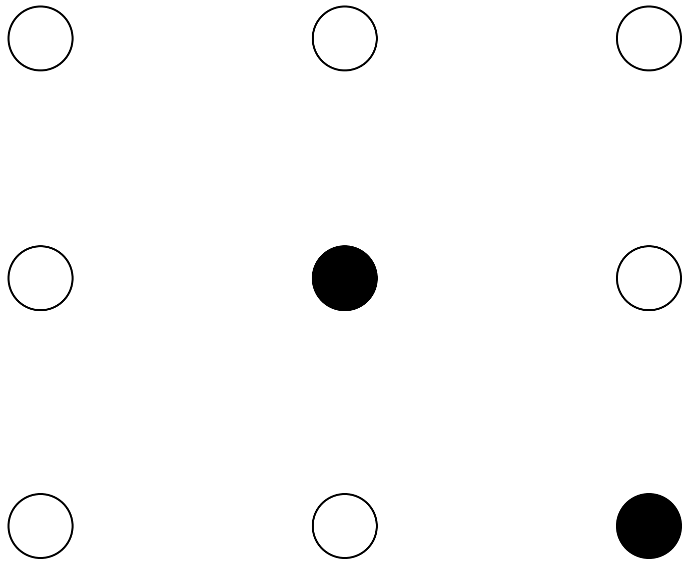
Putting this into F_h produces no change.

This "on" node turns on these other two nodes.



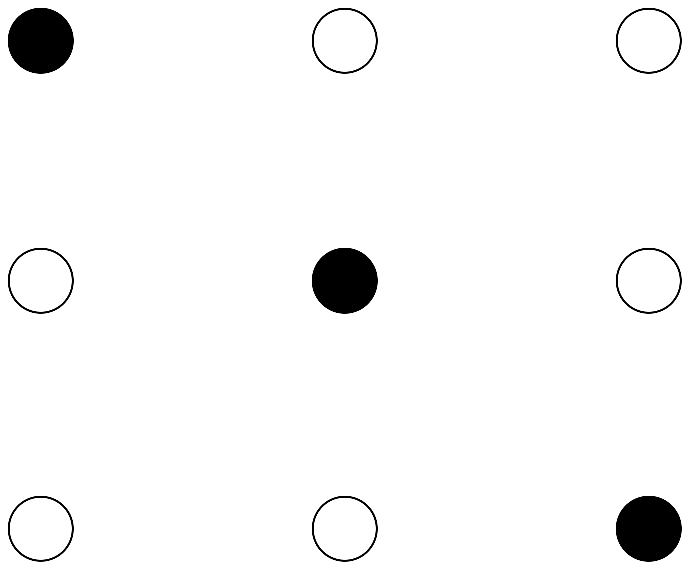
These "off" nodes turn off the first "on" node.

The resulting pattern:



Now using this as the input in a second iteration, and using the same logic, we get

Our original pattern.

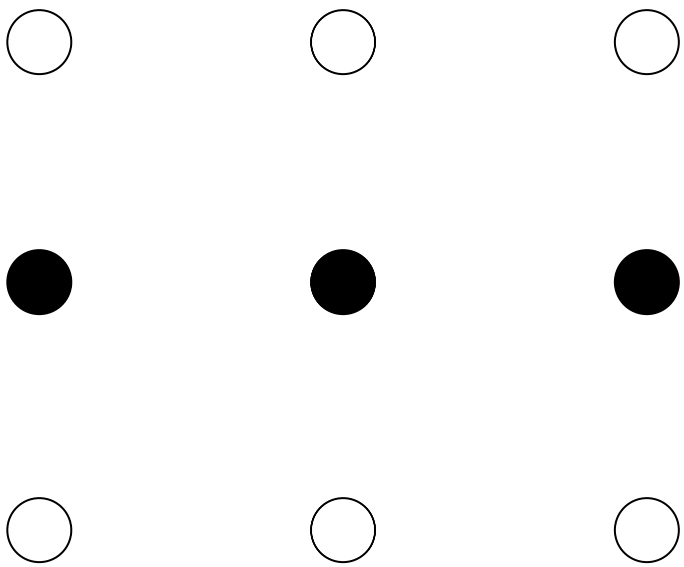


But hold on. Not all input patterns result in our "trained" pattern.

$$\mathbf{W}\begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} \text{ and in fact } \mathbf{W}\begin{pmatrix}0\\\times\\\times\\\times\\0\\\times\\\times\\\times\\0\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}$$

This is because the 1 in the first vector was not associated with any 1 in the exemplar, hence, the other 0s turned this 1 to off (0).

What About Another Exemplar?



$$= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Delta \mathbf{W} = Z\left(\mathbf{q}\mathbf{q}^{\mathrm{T}}\right)$$

$$\mathbf{q}\mathbf{q}^{\mathrm{T}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \left(000111000\right) = \left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$Z\left(\mathbf{q}\,\mathbf{q}^{\mathrm{T}}\right)=\left(\begin{array}{ccc|ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + Z\left(\mathbf{q}\mathbf{q}^{\mathrm{T}}\right) = \left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right)$$

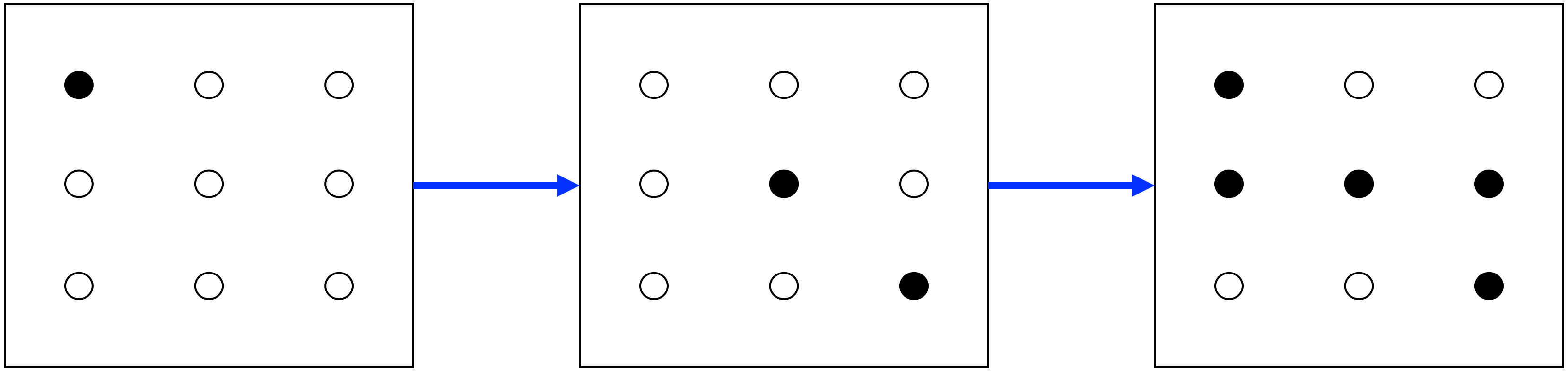
This system "knows" (is trained for) the two exemplars.

$$F_h\left(\mathbf{W}_{\text{new}}\tilde{\mathbf{p}}\right) = F_h\left(\left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right)\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$F_h\left(\mathbf{W}_{\text{new}}\tilde{\mathbf{p}}\right) = F_h\left(\left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right)\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} \overset{1}{\cancel{2}} \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$F_h\left(\mathbf{W}_{\text{new}}\tilde{\mathbf{p}}_1\right) = F_h\left(\begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \tilde{\mathbf{p}}_2$$

$$F_h\left(\mathbf{W}_{\text{new}}\tilde{\mathbf{p}}_2\right) = F_h\left(\left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right)\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$



All exemplar patterns “on”. This is an example of “heat death”.