Optimal Learning Rate

- What is the optimal value η_{opt} of the learning rate?

Consider 1-dim. case. Use first-order Taylor expansion around current weight w_c

$$E(w) = E(w_c) + (w - w_c)\frac{\partial E(w_c)}{\partial w}.$$

Differentiating both sides with respect to w gives:

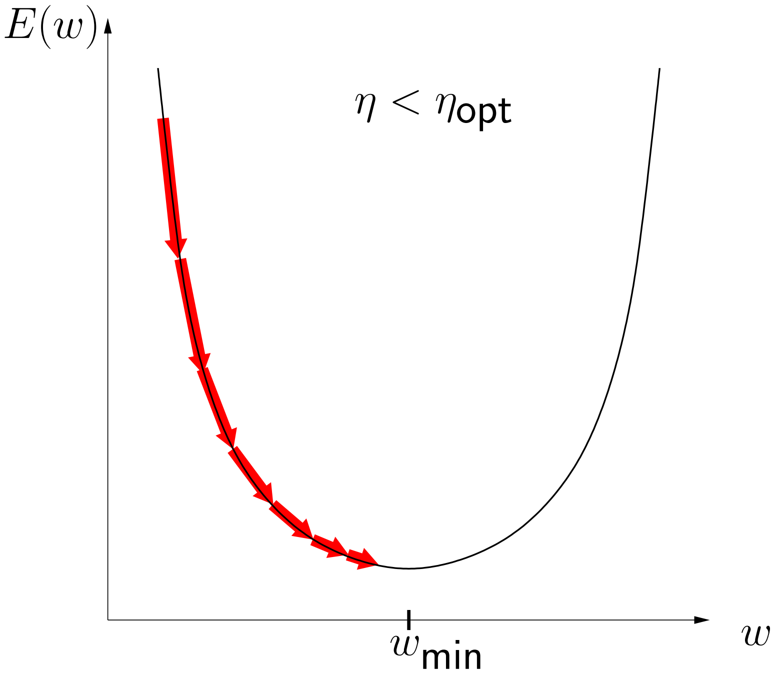
$$\frac{\partial E(w)}{\partial w} = \frac{\partial E(w_c)}{\partial w} + (w - w_c)\frac{\partial^2 E(w_c)}{\partial w^2}$$

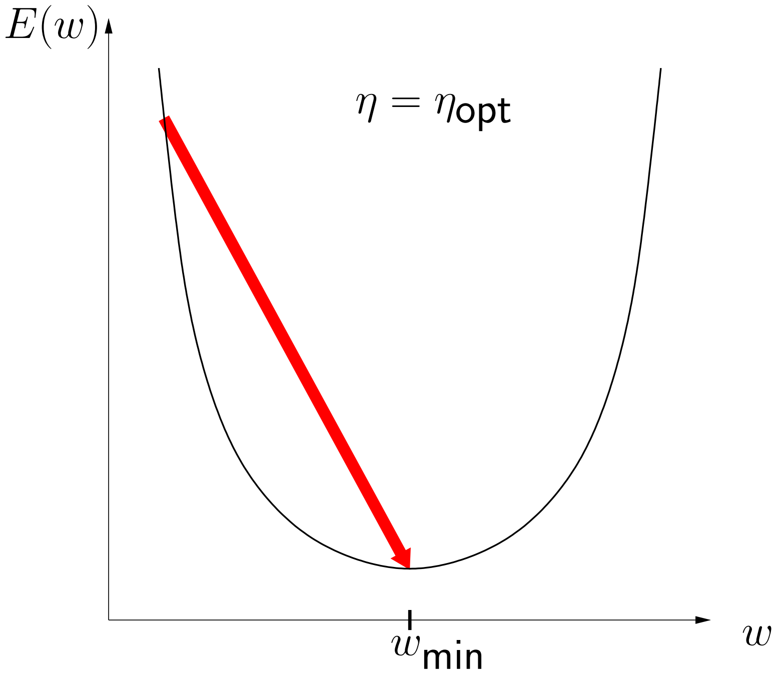
Setting $w = w_{\text{min}}$ and noting that $\frac{\partial E(w_{\text{min}})}{\partial w} = 0$, one obtains

$$0 = \frac{\partial E(w_c)}{\partial w} + (w_{\text{min}} - w_c) + \frac{\partial^2 E(w_c)}{\partial w^2}$$

Optimal Learning Rate (cont.)

$$w_{\text{min}} = w_c - \underbrace{\left(\frac{\partial^2 E(w_c)}{\partial w^2}\right)^{-1}}_{\eta_{\text{opt}}} \frac{\partial E(w_c)}{\partial w}$$

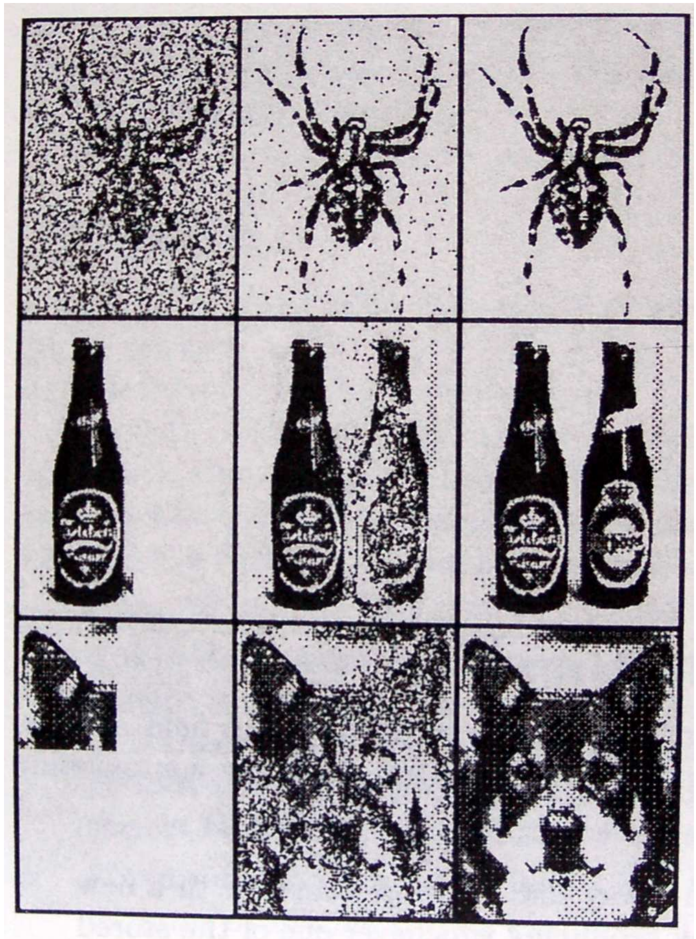




Hopfield Network Introductory Example

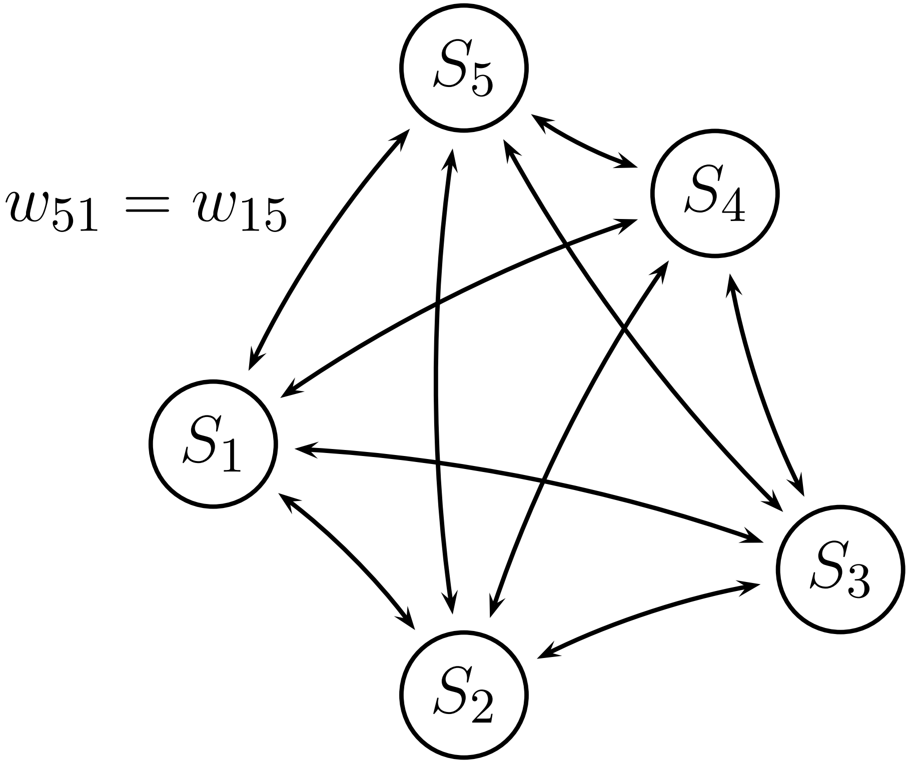
- Suppose we want to store N binary images in some memory.
- The memory should be *content-addressable* and insensitive to small errors.
- We present corrupted images to the memory (e.g. our brain) and recall the corresponding images.

recalled by the memory



presentation of corrupted images

Hopfield Network



- w_{ij} denotes weight connection from unit j to unit i
- no unit has connection with itself $w_{ii} = 0$, $\forall i$
- connections are symmetric $w_{ij} = w_{ji}$, $\forall i, j$

State of unit i can take values ± 1 and is denoted as S_i. State dynamics are governed by activity rule:

$$S_i = \text{sgn}\left(\sum_j w_{ij} S_j\right), \text{ where } \text{sgn}(a) = \begin{cases} +1 & \text{if } a \geq 0, \\ -1 & \text{if } a < 0 \end{cases}$$

Learning Rule in a Hopfield Network

Learning in Hopfield networks:

- Store a set of desired memories $\{\mathbf{x}^{(n)}\}$ in the network, where each memory is a binary pattern with $x_i \in \{-1, +1\}$.
- The weights are set using the sum of outer products

$$w_{ij} = \frac{1}{N} \sum_n x_i^{(n)} x_j^{(n)},$$

where N denotes the number of units (N can also be some positive constant, e.g. number of patterns). Given a $m \times 1$ column vector $\mathbf{a}$ and $1 \times n$ row vector $\mathbf{b}$. The outer product $\mathbf{a} \otimes \mathbf{b}$ (short $\mathbf{a}\,\mathbf{b}$) is defined as the $m \times n$ matrix

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \otimes [b_1\, b_2\, b_3] = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}, \quad m = n = 3$$

Learning in Hopfield Network (Example)

Suppose we want to store patterns $\mathbf{x}^{(1)} = [-1, +1, -1]$ and $\mathbf{x}^{(2)} = [+1, -1, +1]$.

$$\begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} \otimes [-1, +1, -1] \quad = \quad \begin{bmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{bmatrix}$$

$$+$$

$$\begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix} \otimes [+1, -1, +1] \quad = \quad \begin{bmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{bmatrix}$$

Learning in Hopfield Network (Example) (cont.)

$$\mathbf{W} = \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}$$

Recall: no unit has connection with itself.

The storage of patterns in the network can also be interpreted as constructing stable states. The condition for patterns to be stable is:

$$\text{sgn}\left(\sum_j w_{ij}x_i\right) = x_i \,, \forall i.$$

Suppose we present pattern $\mathbf{x}^{(1)}$ to the network and want to restore the corresponding pattern.

Learning in Hopfield Network (Example) (cont.)

Let us assume that the network states are set as follows: $S_i = x_i, \ \forall i$. We can restore pattern $\mathbf{x}^{(1)} = [-1, +1, -1]$ as follows:

$$S_1 = \text{sgn}\left(\sum_{j=1}^{3} w_{1j} S_j\right) \quad = \quad -1 \qquad S_2 = \text{sgn}\left(\sum_{j=1}^{3} w_{2j} S_j\right) = +1$$

$$S_3 = \text{sgn}\left(\sum_{j=1}^{3} w_{3j} S_j\right) \quad = \quad -1$$

Can we also restore the original patterns by presenting "similar" patterns which are corrupted by noise?

Updating States in a Hopfield Network

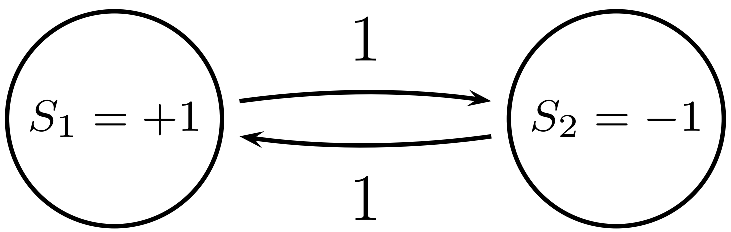
Synchronous updates:

- all units update their states $S_i = \text{sgn}\left(\sum_j w_{ij} S_j\right)$ simultaneously.

Asynchronous updates:

- one unit at a time updates its state. The sequence of selected units may be a fixed sequence or a random sequence.

Synchronously updating states can lead to oscillation (no convergence to a stable state).



Aim of a Hopfield Network

Our aim is that by presenting a corrupted pattern, and by applying iteratively the state update rule the Hopfield network will settle down in a stable state which corresponds to the desired pattern.

Hopfield network is a method for

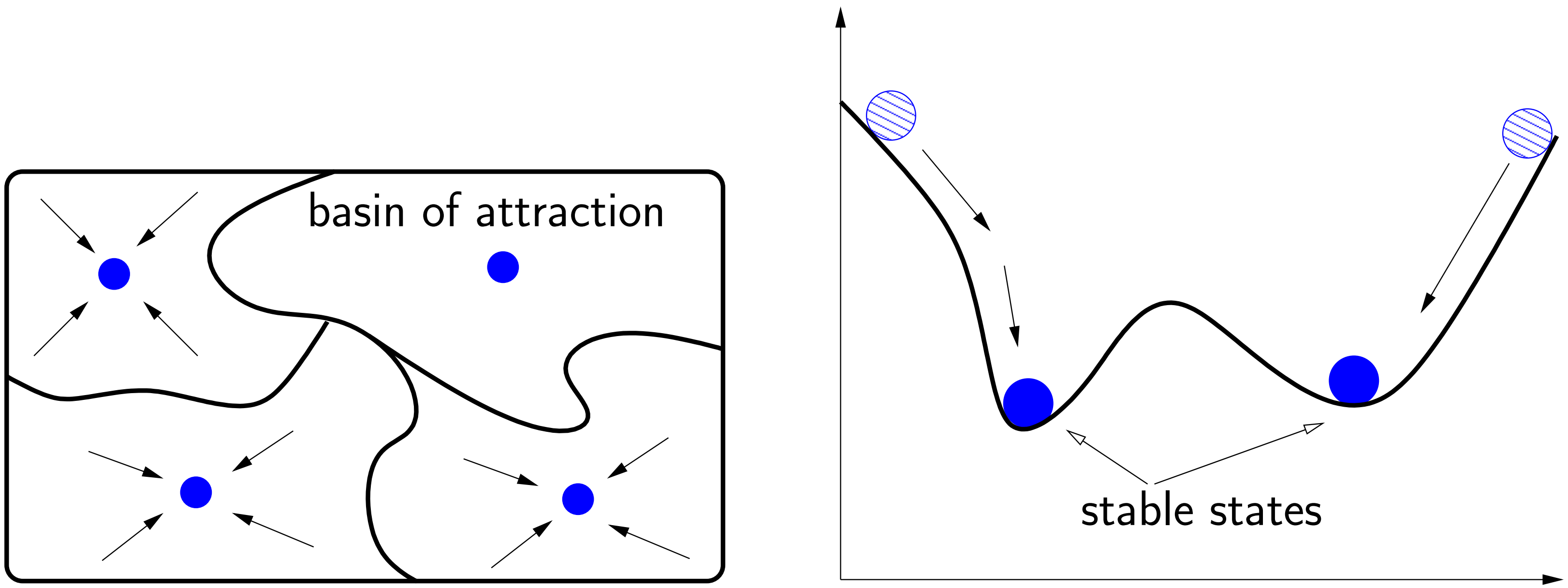
- pattern completion
- error correction.

The state of a Hopfield network can be expressed in terms of the energy function

$$E = -\frac{1}{2}\sum_{i,j} w_{ij} S_i S_j$$

Hopfield observed that if a state is a local minimum in the energy function, it is also a stable state for the network.

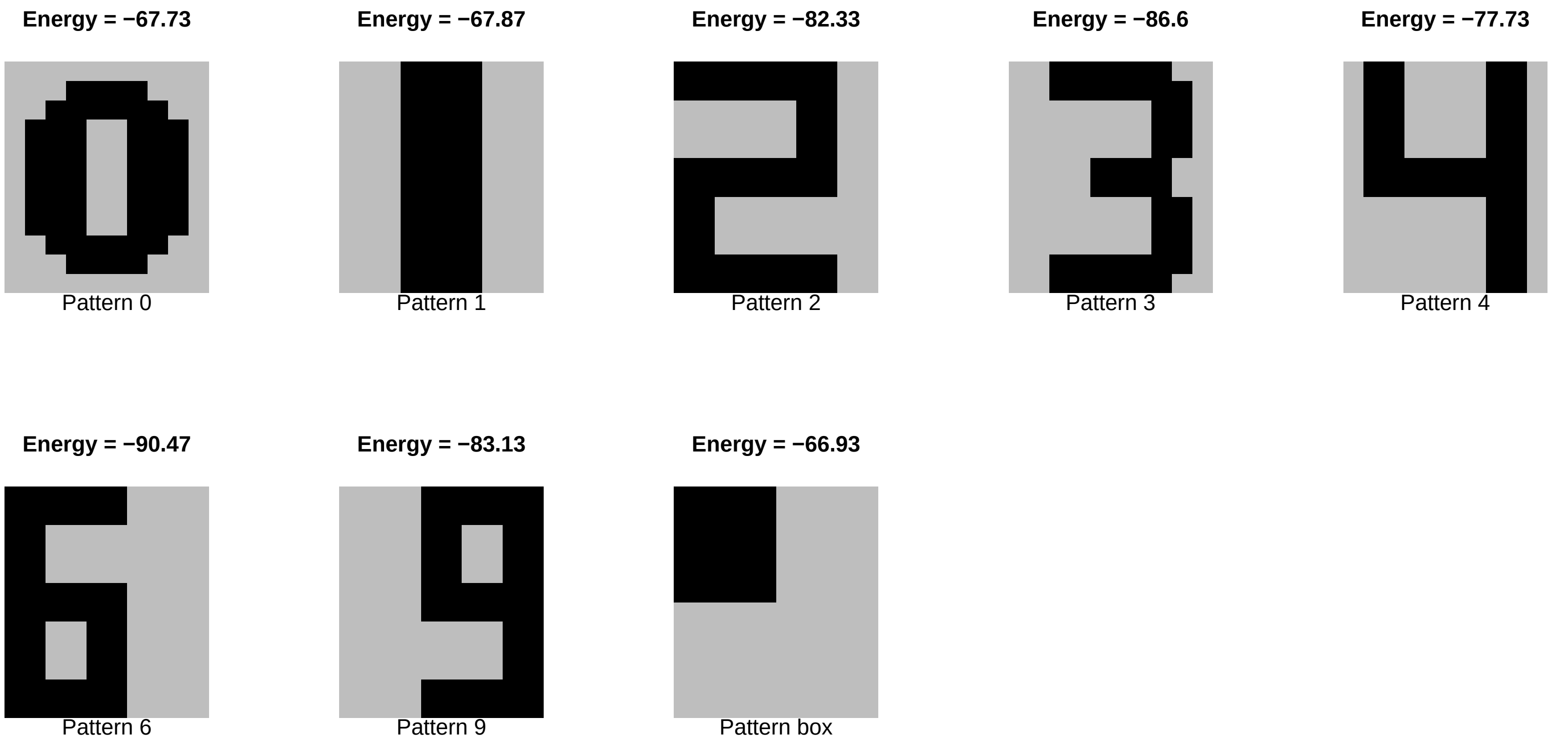
Basin of Attraction and Stable States



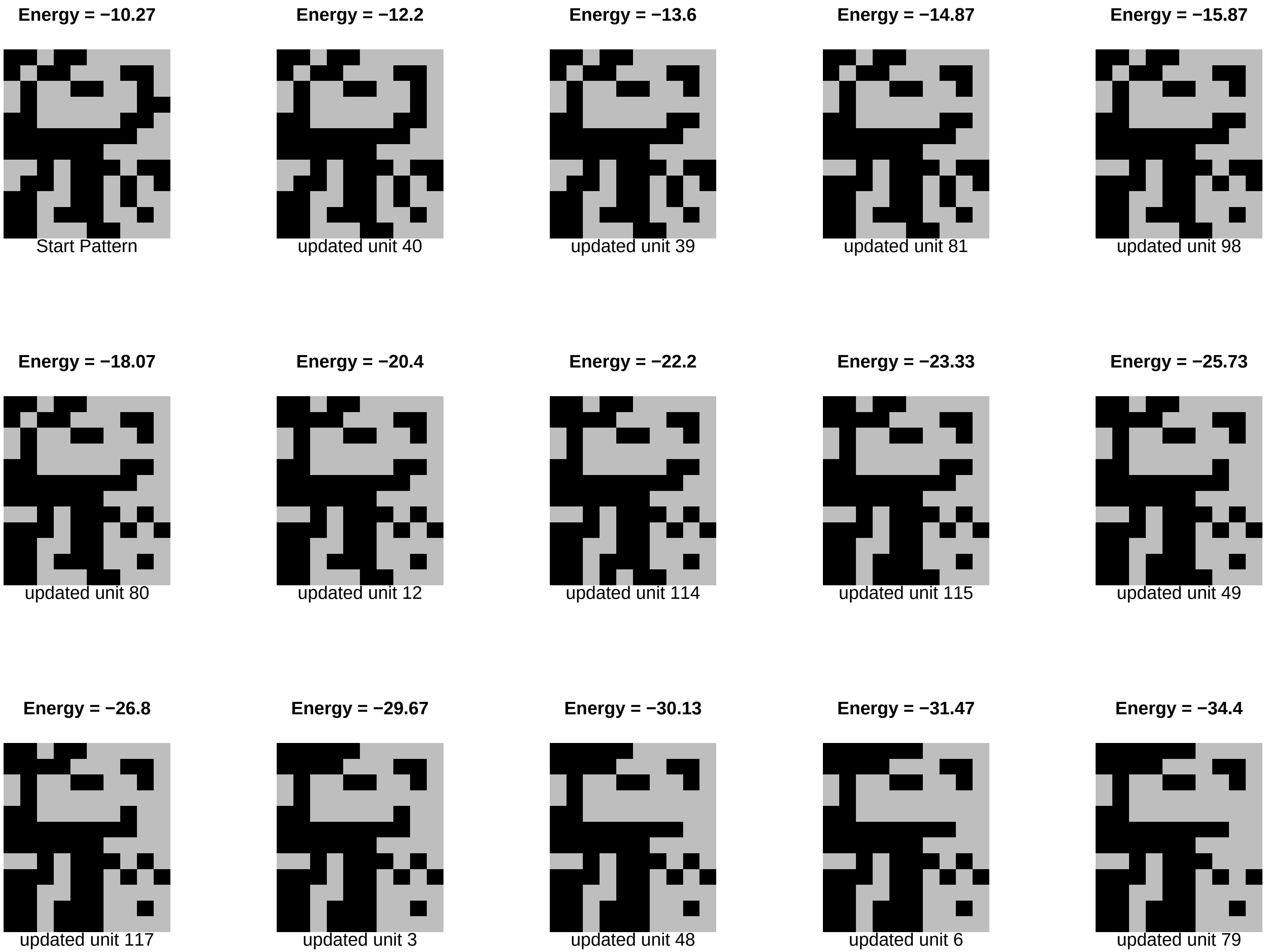
Within the space the stored patterns $\mathbf{x}^{(n)}$ are acting like attractors.

Haykin's Digit Example

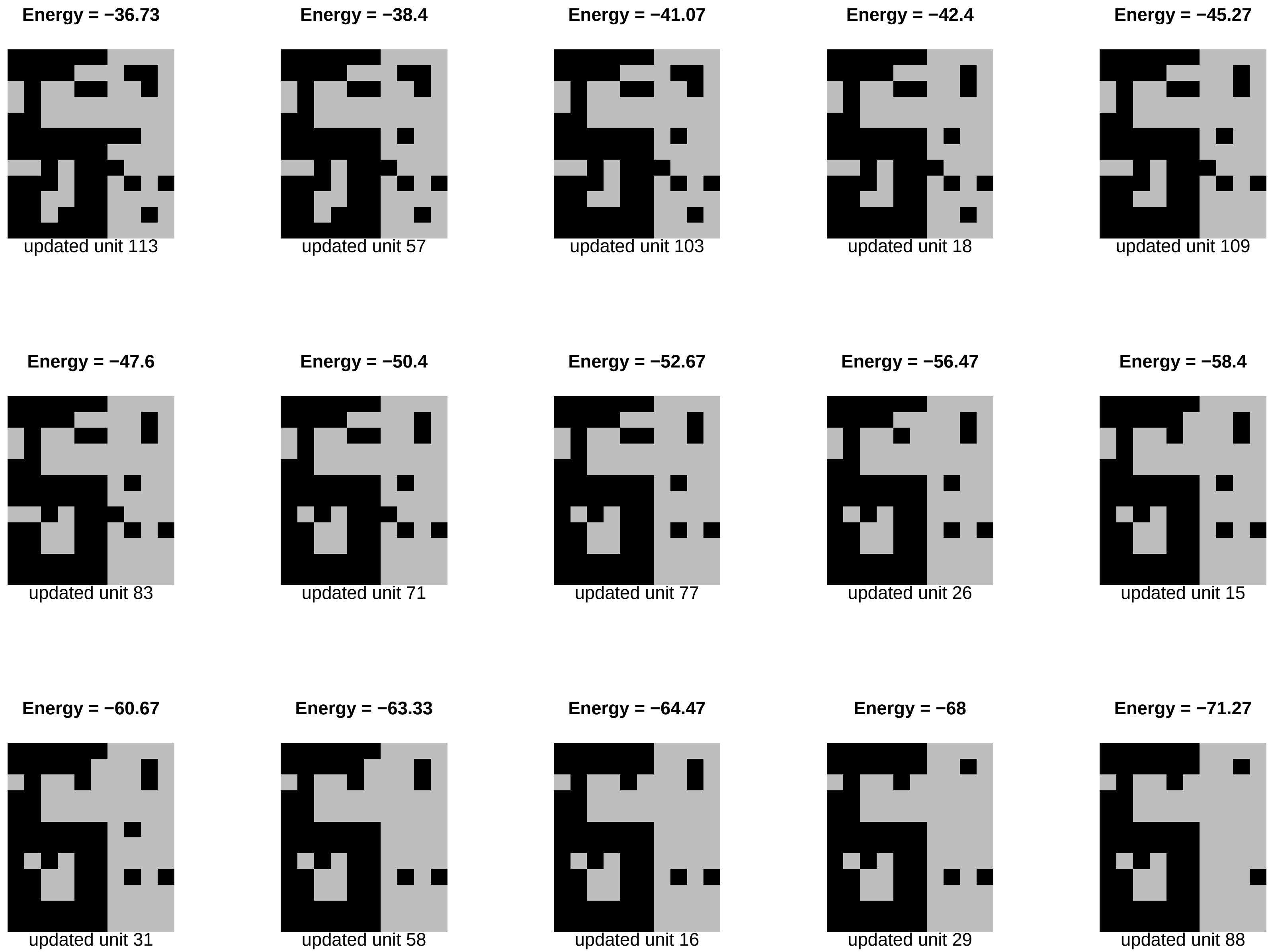
Suppose we stored the following digits in the Hopfield network:



Updated States of Corrupted Digit 6

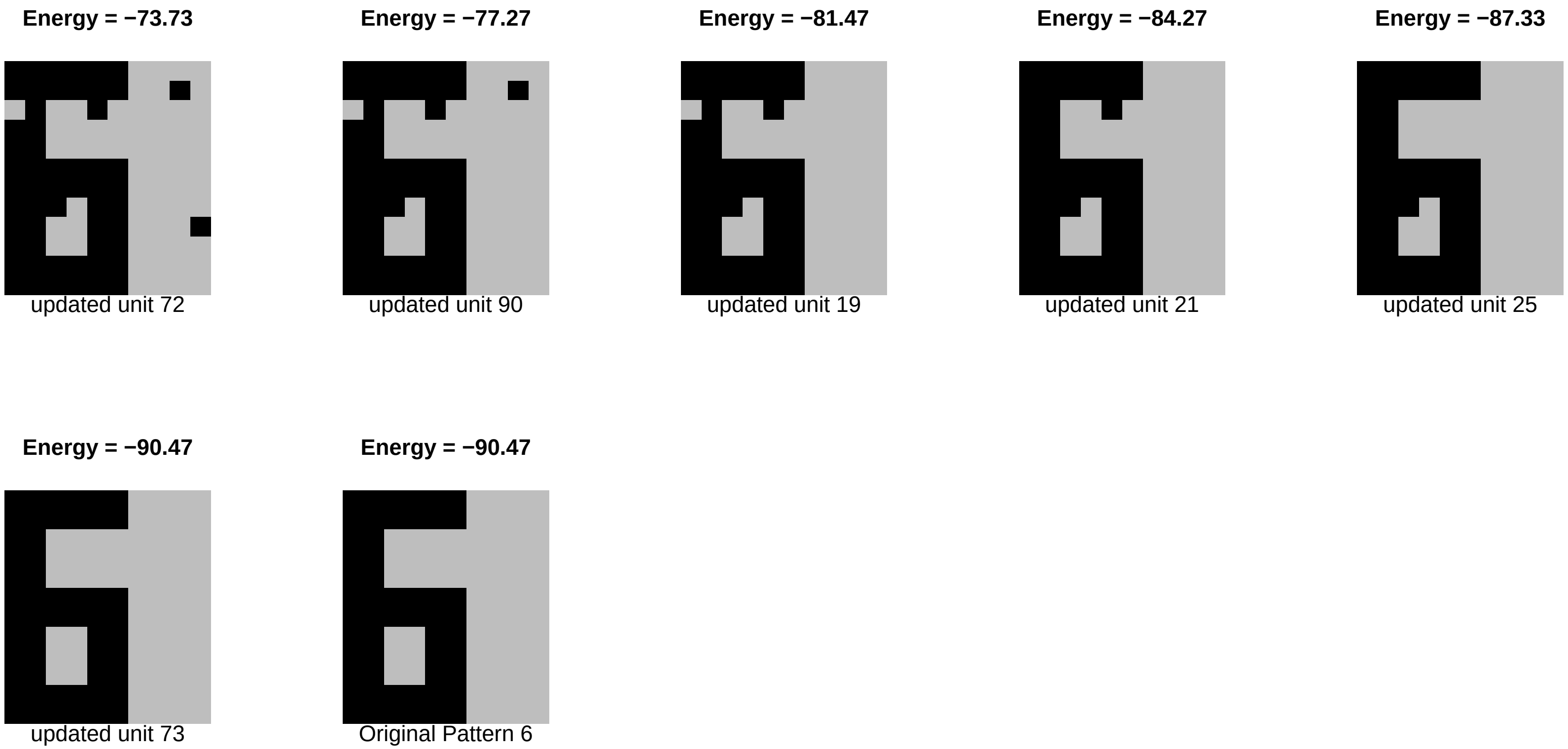


Updated States of Corrupted Digit 6 (cont.)

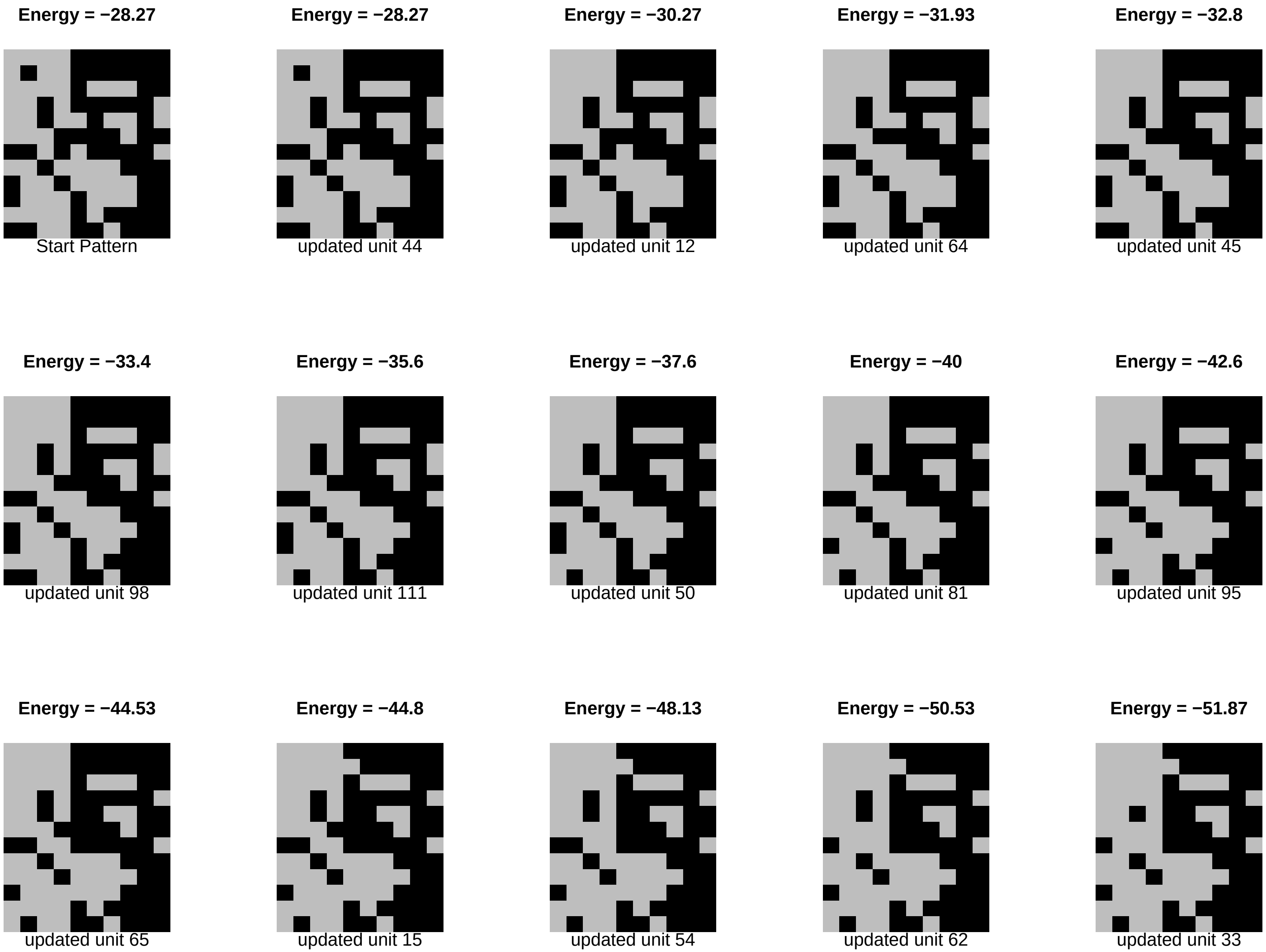


Updated States of Corrupted Digit 6 (cont.)

The resulting pattern (stable state with energy -90.47) matches the desired pattern.

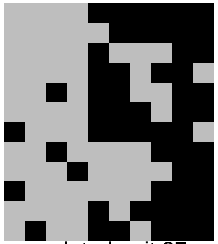


Recall a Spurious Pattern



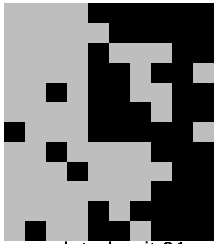
Recall a Spurious Pattern (cont.)

Energy = −53.73



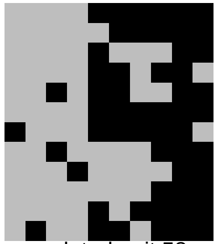
updated unit 37

Energy = −56.53



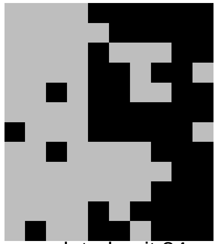
updated unit 91

Energy = −59.93



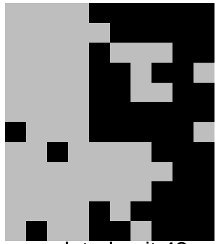
updated unit 58

Energy = −61.6



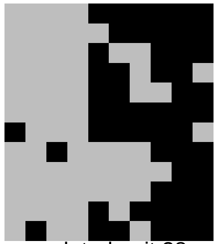
updated unit 84

Energy = −63.2



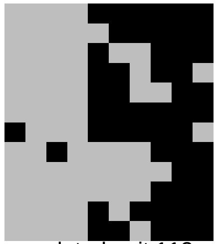
updated unit 43

Energy = −63.73



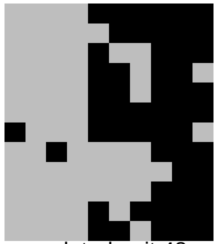
updated unit 28

Energy = −66.8



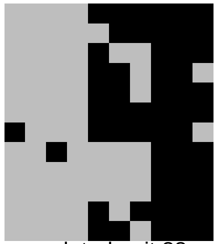
updated unit 112

Energy = −67.6



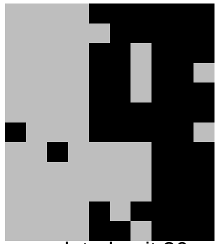
updated unit 48

Energy = −69



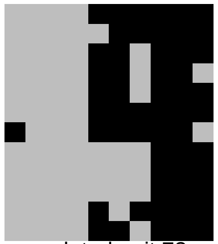
updated unit 88

Energy = −70.4



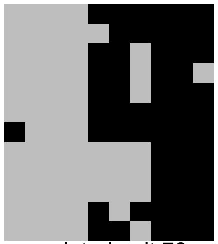
updated unit 26

Energy = −71.93



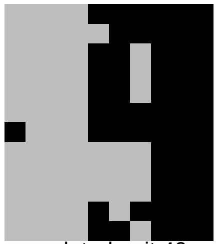
updated unit 73

Energy = −74.13



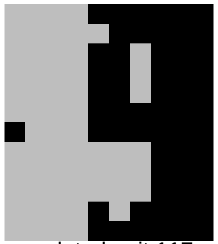
updated unit 70

Energy = −76.6



updated unit 40

Energy = −80.27



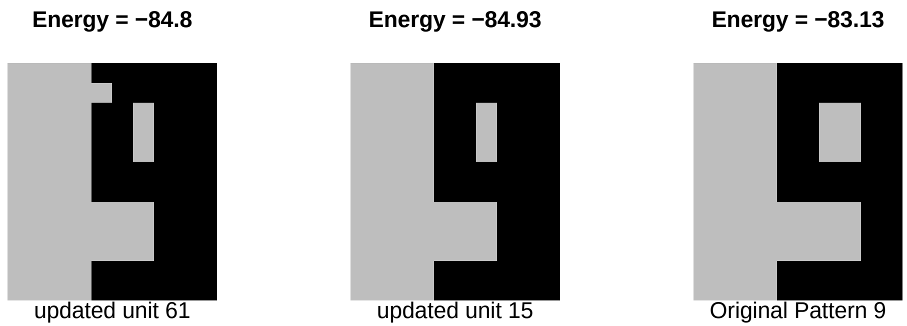
updated unit 117

Energy = −81.4

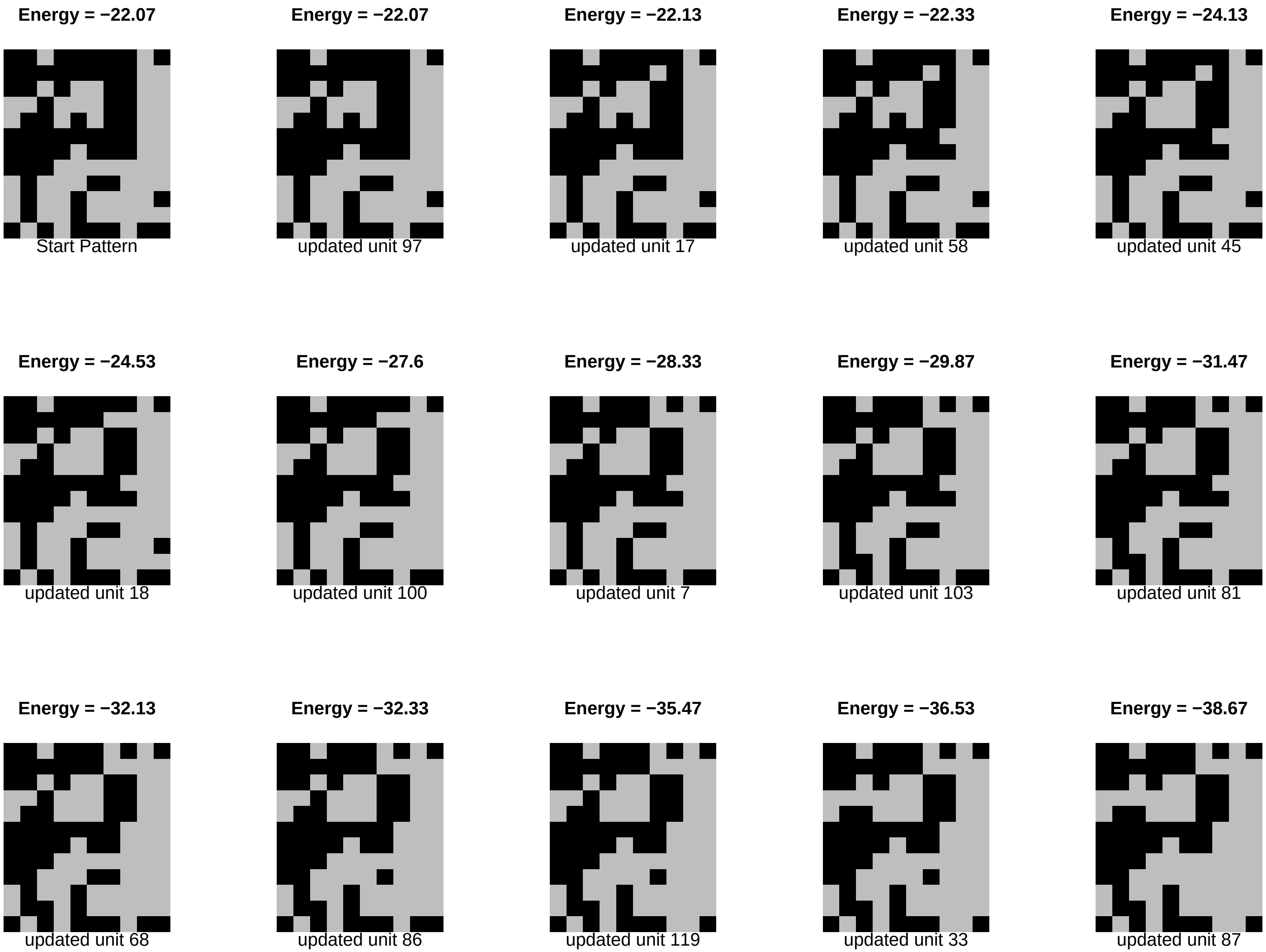
updated unit 106

Recall a Spurious Pattern (cont.)

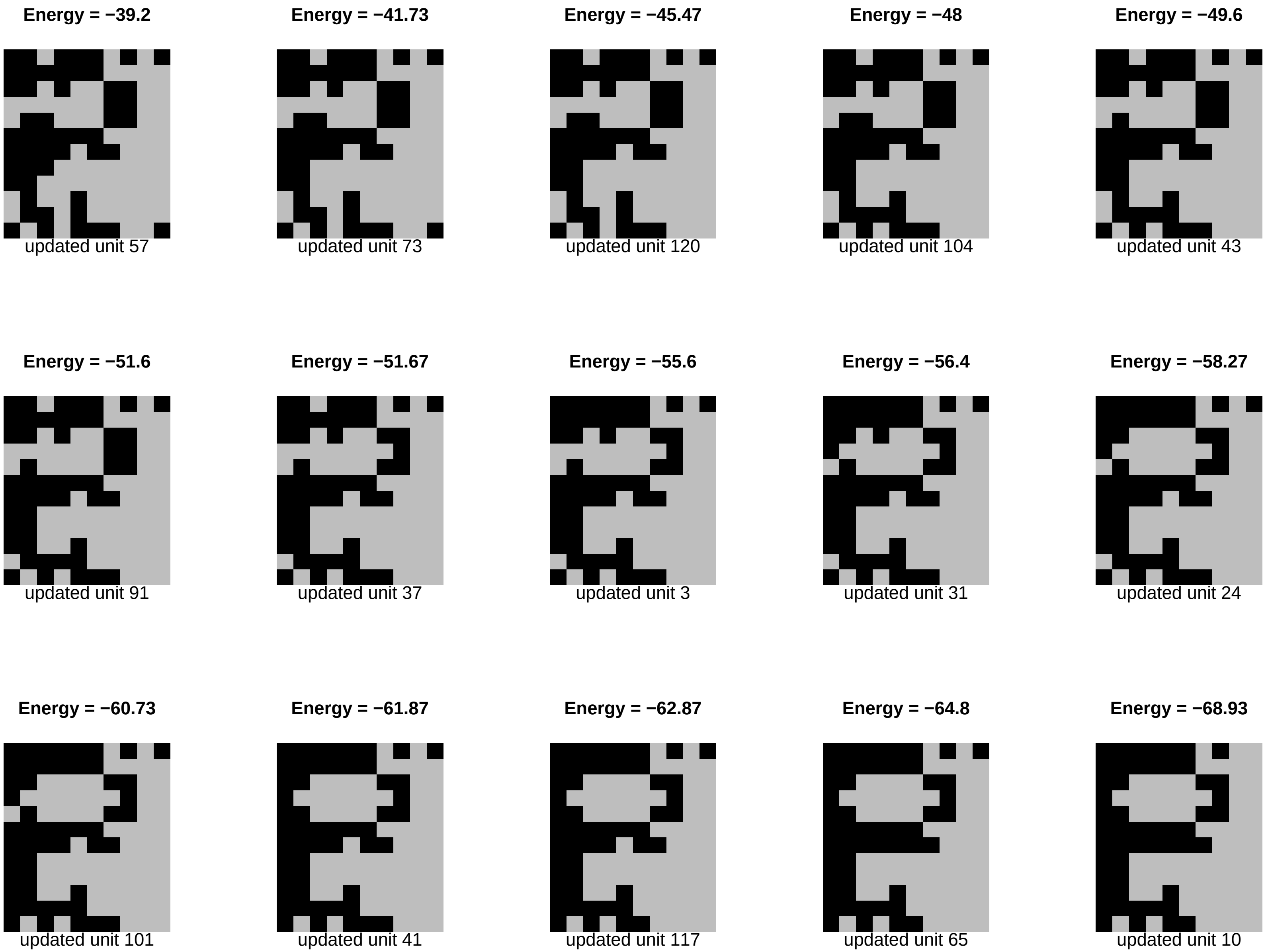
The Hopfield network settled down in local minima with energy -84.93. This pattern however is *not* the desired pattern. It is a pattern which was not stored in the network.



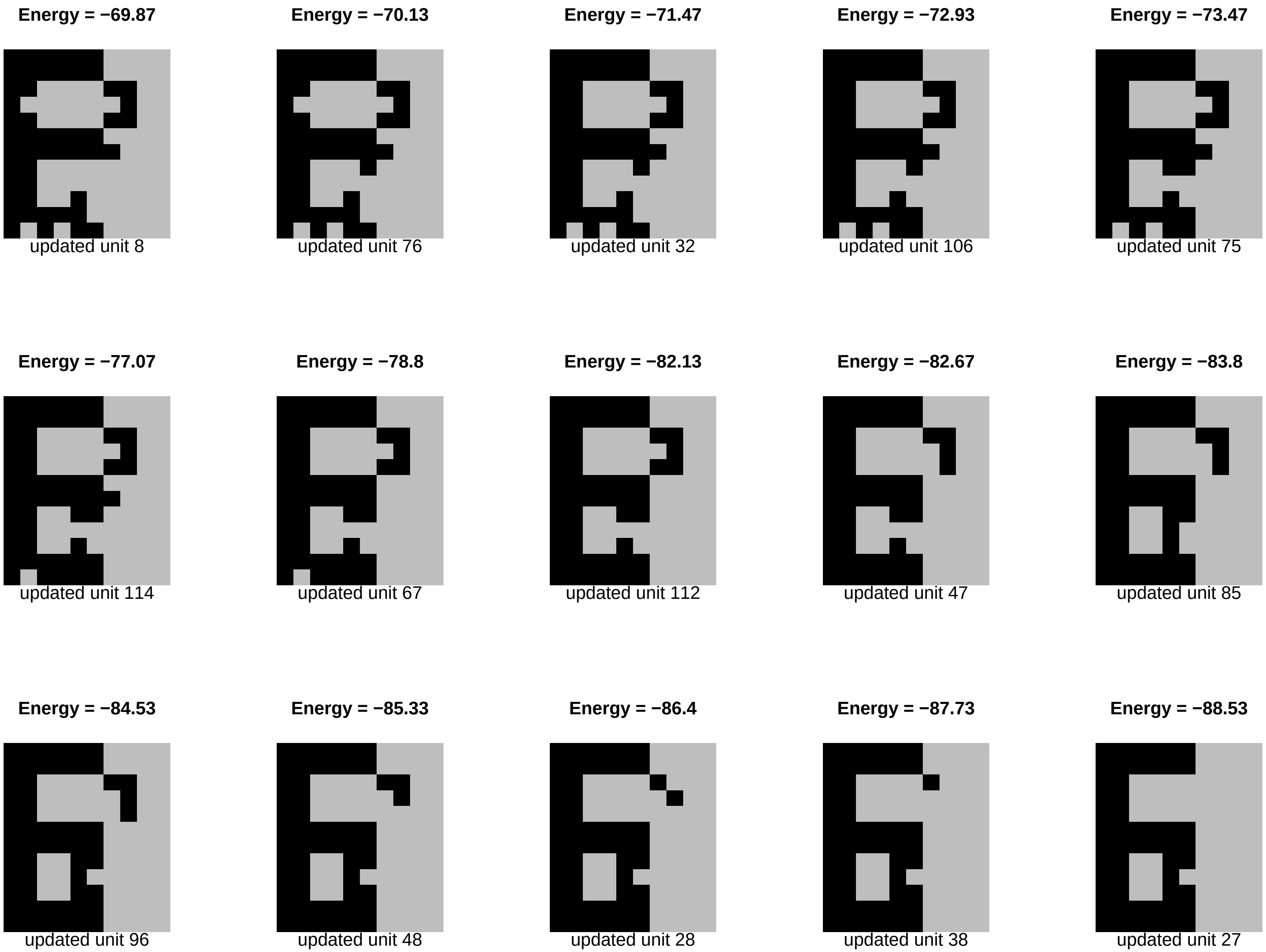
Incorrect Recall of Corrupted Pattern 2



Incorrect Recall of Corrupted Pattern 2 (cont.)

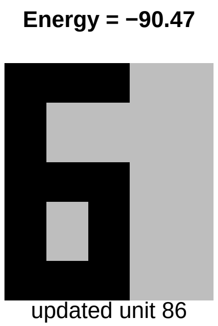


Incorrect Recall of Corrupted Pattern 2 (cont.)



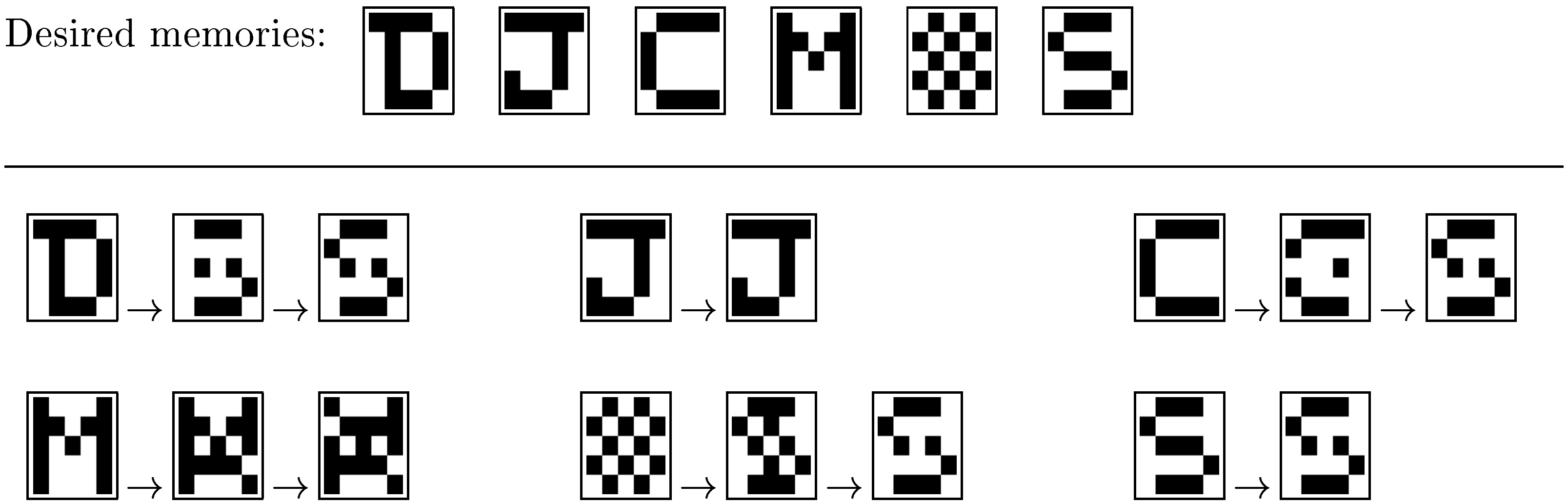
Incorrect Recall of Corrupted Pattern 2 (cont.)

Although we presented the corrupted pattern 2, the Hopfield network settled down in the stable state that corresponds to pattern 6.



MacKay's Example of an Overloaded Network

Six patterns are stored in the Hopfield network, however most of them are not stable states.



Spurious states represent stable states that are different from the stored desired patterns.

Spurious States and Capacity

- Reversed states $((-1) \cdot \mathbf{x}^{(n)})$ have same energy as the original patterns $\mathbf{x}^{(n)}$.
- Stable mixture states are not equal to any single pattern. They corresponds to a linear combination of an odd number of patterns.
- Spin glass states are local minima that are not correlated with any finite number of the original patterns.

Capacity:
What is the relation between the number d of units and the maximum number $N_{\max}$ of patterns one can store by allowing some small error. If $N_{\max} = \frac{d}{4 \log d}$ then most of stored patterns can be recalled perfectly.