Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 3

Hopfield networks

Fredrik Danielsson, Franco Ruggeri and Kevin Dalla Torre

February 17, 2020

1 Main objectives and scope of the assignment

Our major goals in the assignment were

- to learn to implement a Hopfield network.
- to learn to use Hopfield networks for noise reduction and pattern completion.
- to understand the dynamics of Hopfield networks related to the energy.
- to understand the question of storage capacity and the effects of different features on that.

2 Methods

We used *python* for all the tasks with the support of just the *numpy* and *matplotlib* libraries, implementing a Hopfield network from scratch.

3 Results and discussion

3.1 Convergence and attractors

In this section we are going to discuss the results obtained for a 8-neuron Hopfield network trained with 3 patterns (x1, x2, x3) using Hebbian learning which was tested using 3 distorted patterns (x1d, x2d, x3d), using synchronous updates. Two versions of network are analyzed: with and without self connections.

The network managed to store the 3 patterns as fixed points. Both with and without self connections, it could recall only a subset of the stored patterns starting from the distorted ones (x1 and x3 with self connections, just x1 without). This is due to the synchronous updates, which do not guarantee convergence. In particular, the network can end up alternating between two states with the same energy, and this is the case for x2d.

Feeding the network with all the possible patterns ($2^8 = 256$), we found also that there are other attractors. Without self connections, there were:

- 3 attractors corresponding to the patterns used to train the network.
- 3 extra attractors corresponding to the inverse of the patterns.

In fact, the negations of the stored patterns are always spurious patterns. If the starting pattern is distorted more than half with respect to an original pattern, it converged to the inverse of it.

Adding self-connections, the number of attractors increased to 14, so there were more spurious patterns, confirming the disadvantage of using them.

3.2 Sequential update

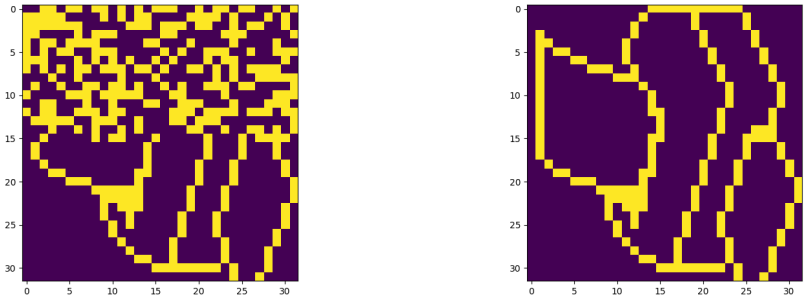
In this and in the following sections we are going to use a 1024-nodes Hopfield network trained with 3 pictures (p1, p2, p3). In this section we want to study the capability of this network to complete degraded patterns (pictures).

As shown in figure 1, the network completes perfectly p10, which is a degraded version of p1, both with synchronous and asynchronous updates. In figure 2, we can instead observe a convergence to a spurious pattern using synchronous updates. In this case, sequential updates perform better and complete almost perfectly p11, which is a mixture of p2 and p3 (figure 3). The term *almost* is due to the corrupted bits, another problem of Hopfield networks. The reconstructed picture is anyway clearly recognizable.

A note about sequential updates is that the reached attractor can be different depending on the random order of updates of the neurons. In some cases, the network found a spurious pattern for p10 (figure 1).

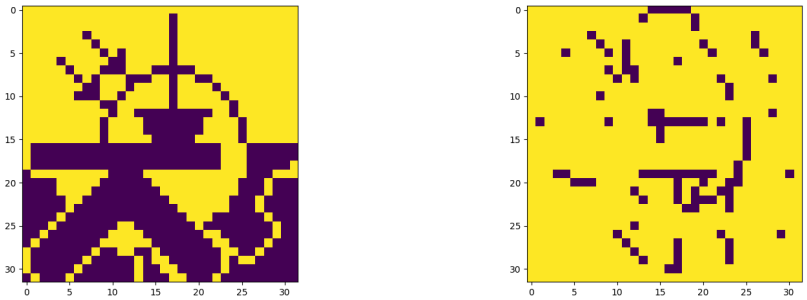
3.3 Energy

In table 1, the energies of attractors and starting states are reported. For the stored patterns, which are stable points, the attractor coincides with the starting state (this is not always true as we shall see in section 3.5). We can notice that



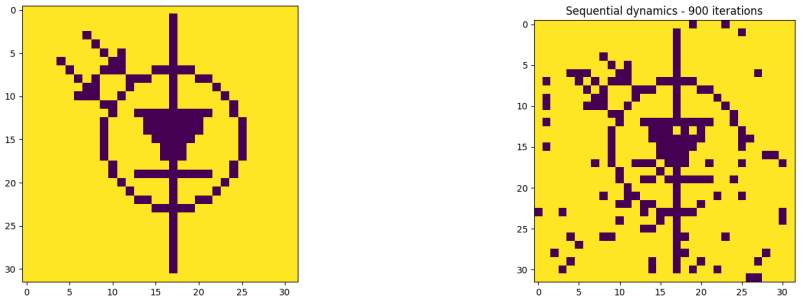
(a) (b)

Figure 1: Completion of p10 (a), degraded version of p1. The network reconstructs perfectly p1 (b). The result is the same with synchronous and asynchronous updates.



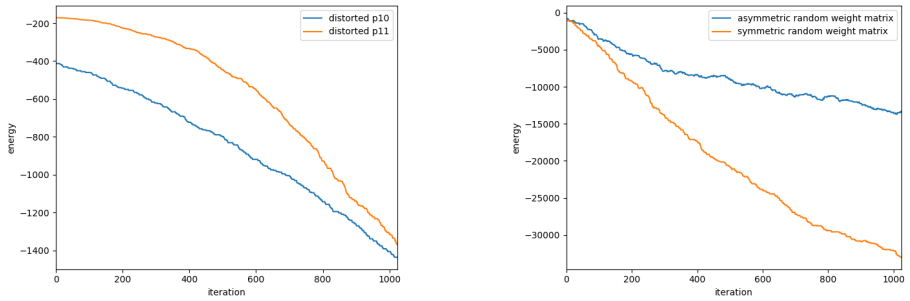
(a) (b)

Figure 2: Attempt of completion of p11 (a), mixture of p2 and p3, with synchronous updates. The result is a spurious pattern (b).



(a) (b)

Figure 3: Completion of p11, mixture of p2 and p3 (a), resulting in p3. The result has however some corrupted bits (b).



(a) (b)

Figure 4: Energy trend during updates for completion of p10 and p11 (a) and for symmetric and asymmetric random weights matrices (b).

the attractor always has lower (or equal, in case of coincidence) energy than the starting state. In fact, each state change during the update rule decreases the energy and an attractor is a local minimum in the energy landscape. This is also shown in figure 4a.

State	Energy
Attractor p1	-1436.39
Attractor p2	-1362.64
Attractor p3	-1459.25
Attractor p10	-1436.39
Attractor p11	-1368.23
Starting p10	-412.98
Starting p11	-170.50

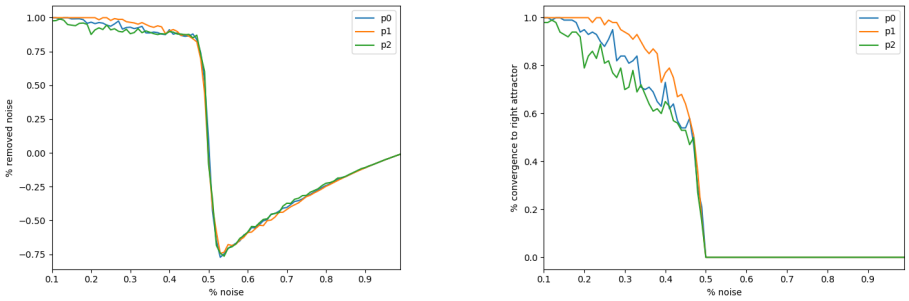
Table 1: Energies in attractors and starting points.

Figure 4b shows a comparison of symmetric and asymmetric weight matrix. The energy (for a random starting state and a a random weight matrix) is not monotonically decreasing for an asymmetric matrix. In fact, a constraint for this (implying of course sure convergence to a local minimum) is to have a symmetric zero-diagonal weight matrix.

3.4 Distortion resistance

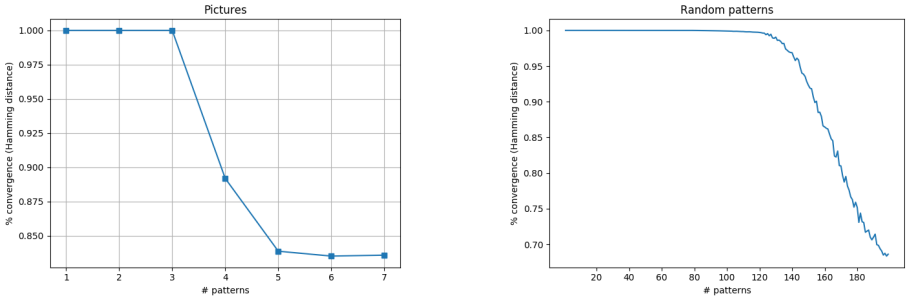
A very common application of Hopfield networks is noise reduction. As shown in figure 5, the noise is removed almost completely (above 80%) when less than 50%. The sudden drop at 50% is explained by looking at the right figure. If the noise is higher than 50%, the network converges to a wrong attractor, so the percentage of removed noise with noise greater than 50% is not significant.

We can also notice that the 3 different patterns behave slightly different, because each one has its own basin of attraction.



(a) (b)

Figure 5: Percentage of removed noise (a) and percentage of convergence to the pure pattern (b) for a varying amount of noise.



(a) (b)

Figure 6: Percentage of convergence measured as the ratio between Hamming distance and number of nodes.

3.5 Capacity

As shown in figure 6, the network can only store a maximum of 3 pictures while around 140 random patterns. The huge difference is due to the cross-talk between patterns, that is high in the former case (pictures are close) and low in the latter (random patterns are uniformly distributed and so they are closer to being orthogonal). In fact, Hebbian learning works well only if the patterns are near orthogonal.

We can also notice that the drop is abrupt (discussed better in the following) and that the capacity in the second case corresponds to the theoretical result $0.138N$.

Regarding the capacity, Hopfield networks show the so-called catastrophic forgetting, illustrated in figure 7. If the number of patterns is over the capacity, the number of stable points sharply decreases, i.e. the network forgets the stored patterns.

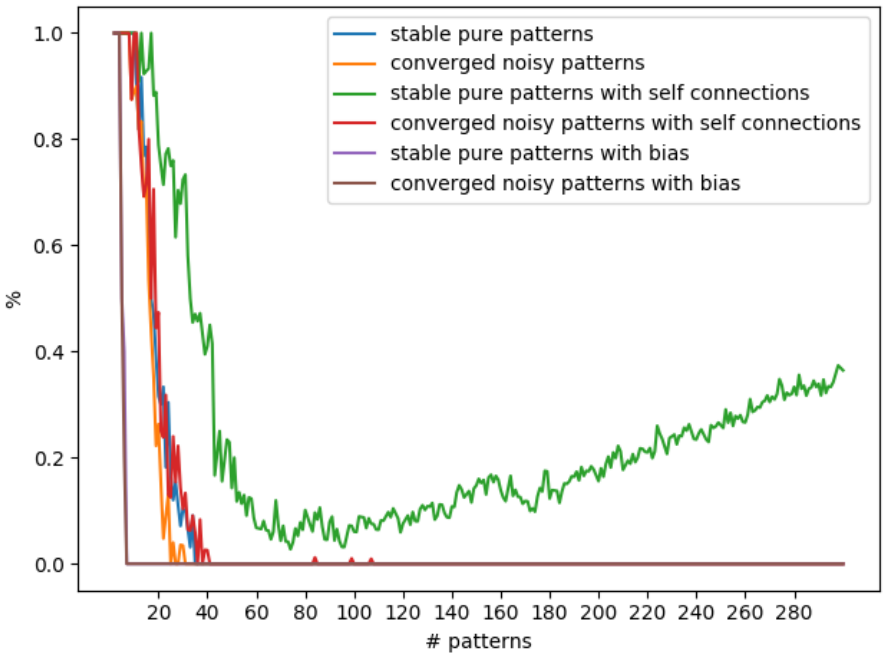
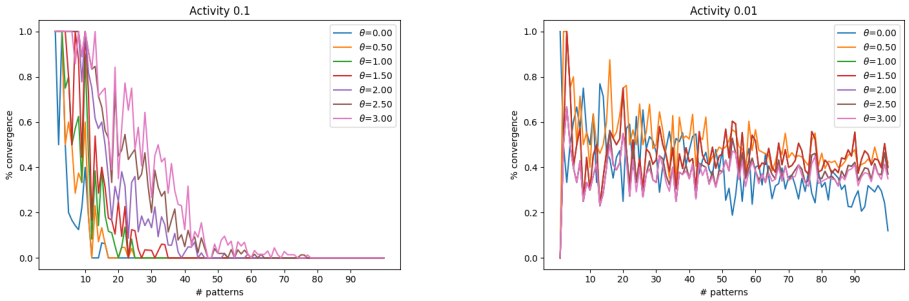


Figure 7: Percentage of stable pure patterns or noisy patterns converged to the pure pattern, for a varying number of patterns.

Self connections make it more difficult for a node to change state, so they mitigate this effect (but they promote the formation of spurious patterns, as discussed in section 3.1). Also, continuing to add patterns after the drop (catastrophic forgetting), the number of stable patterns increases again. This happens because self connections get stronger and stronger, so the network returns just the pattern fed into. However, the network behaves in such way also for noisy patterns (it just returns the noisy patterns themselves, no noise reduction), so the network does not work as wanted and the measure of stable patterns is misleading.

Without self-connections, the capacity is a bit worse and the number of stable points is zero for a high number of patterns, so the trend is the expected one (not misleading).

Finally, with biased patterns (having more 1s than -1s), the capacity is even worse because the points are closer each other and as a consequence there is more cross-talk.



(a) (b)

Figure 8: Sparse patterns

3.6 Sparse patterns

In figure 8 we can see that the sparser the data (i.e. smaller activity), the higher the capacity. Also, we can notice that the best bias term is proportional to the activity and this is quite intuitive, since for higher values of activity it has to compensate more bias.

The disadvantage of having sparse patterns is that, of course, we store less information (given by 1s).

4 Final remarks

The assignment was very useful to understand the key concepts of Hopfield networks (attractors, energy, capacity) and their applications (noise reduction, pattern completion).