Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 3

Hopfield networks

Nouhaila AGOUZAL, Abdessamad BADAOUI and Nasr Allah AGHELIAS

October 2, 2023

1 Main objectives and scope of the assignment

Our major goals in the assignment were :

- to Explain the principles governing autoassociative networks' operation and functionality.
- to Train Hopfield networks for pattern storage and retrieval.
- to Examine storage capacity and strategies for enhancement in associative memories.

2 Methods

The programming was done in python. We used the numpy library for mathematical calculations and matplotlib.pyplot to generate graphs.

3 Results and discussion

3.1 Convergence and attractors :

In this section, the convergence criterion states that a pattern is considered converged when it remains unchanged after a single synchronous update. The patterns x1d, x2d & x3d converge towards the stored patterns. Out of the 256 possible candidates, there are 10 attractors. Figure 1 illustrates the convergence ratios for all possible distorted patterns of x1, x2 & x3. From this figure, we can deduce that when a larger number of components in the original pattern is modified, the distorted pattern may not converge back to the original one through a synchronous Hopfield network update.

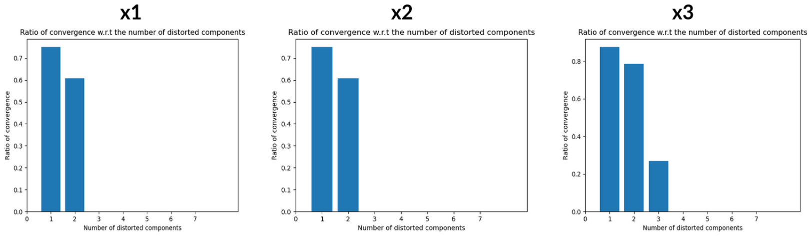


Figure 1: Convergence ratios for all possible distorted patterns of x1, x2 & x3

3.2 Sequential update :

The three patterns p1, p2 & p3 are stable. The network can successfully complete the pattern p10, which is a degraded version of p1. However, as depicted in Figure 2, the network encounters difficulties when trying to complete the pattern p11, which is a mixture of p2 and p3. This issue is resolved using the original sequential Hopfield dynamics, as demonstrated in Figure 3.

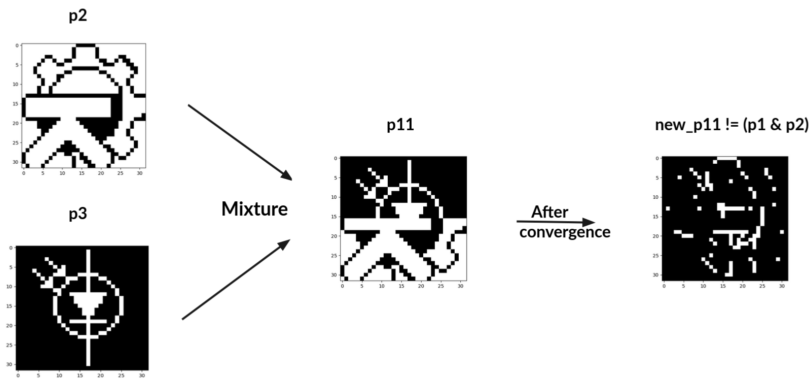


Figure 2: Convergence of p11 with synchronous update

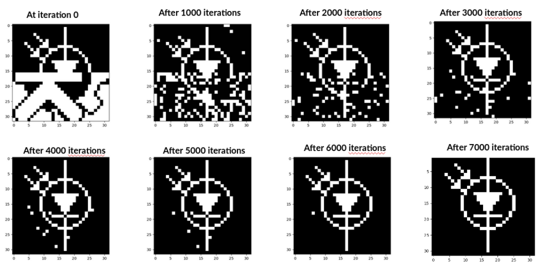


Figure 3: Convergence of p11 with sequential update

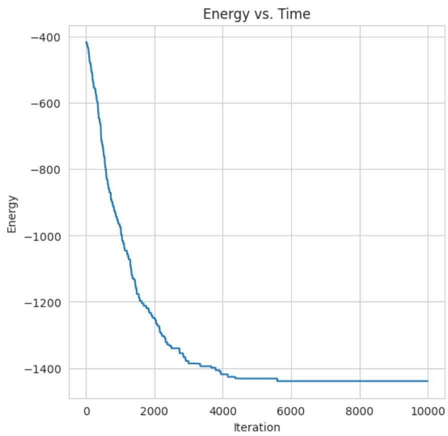
3.3 Energy

The energy depends on weights, which determine the landscape of this energy, and also on the state where we are calculating it. After training the Hopfield network on the patterns p1, p2, and p3, which are stable states as demonstrated previously and represent some of the local minima of this energy, the energies at those states are as follows:

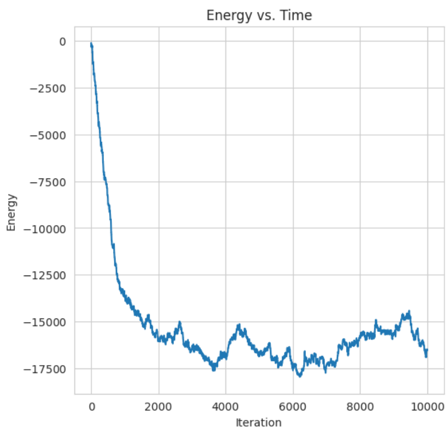
The energy at state p1 is: -1439.39
The energy at state p2 is: -1365.64
The energy at state p3 is: -1462.25

The energies at the distorted patterns p10 and p11 are:

The energy at state p10 is: -415.98
The energy at state p11 is: -173.5



(a) The mean squared error calculated across all the scenarios.



(b) Energy over time for a non symmetric weight matrix.

In Figure 4a, we illustrate the evolution of energy when we apply p10 as the input state. This input state lies in the basin of the attractor p1, so over time, the Hopfield network will converge to p1. This convergence is guaranteed because we use asynchronous updates (the energy decreases at each state update) with no self-connections and a symmetric weight matrix.

When generating a weight matrix with normally distributed random numbers, we observe the energy evolution shown in Figure 4b. As we can see, we no longer have a decreasing energy, but instead, we have many oscillations. This occurs because we no longer have a guarantee of convergence since the weight matrix is not symmetric. Consequently, the network does not necessarily lead to stable states. However, in the case of a symmetric matrix, we observe a similar behavior in Figure 4a, indicating a decrease in energy over time (convergence).

3.4 Distortion Resistance

Figure 5 displays the percentage of pixels that were successfully restored for different levels of noise. We can observe that for noise levels between 0% and 20%, the network can restore the actual clean pattern. However, as the noise becomes more significant, it becomes increasingly challenging for the network to converge to the correct attractor. Sometimes it converges to another attractor from the training patterns, and sometimes it converges to entirely new patterns (spurious memories). Another noteworthy observation is that with noise levels between 80% and 100%, the network converges to the anti-memory, which are inevitable stable states in this kind of associative memory.

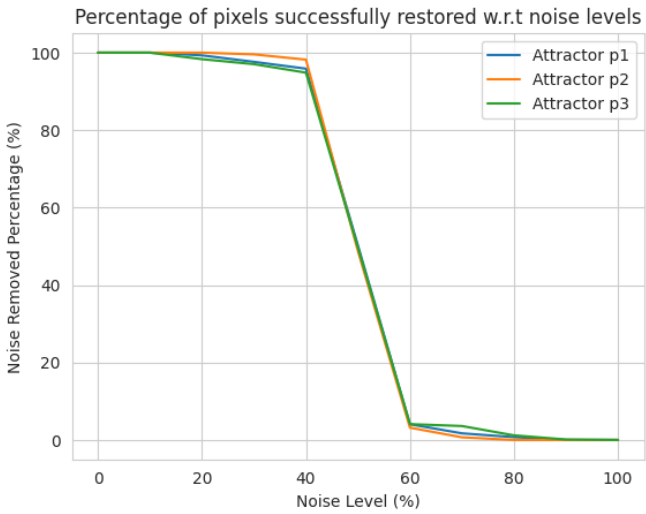


Figure 5: Percentage of pixels successfully restored w.r.t noise levels

3.5 Capacity

When we try to learn another picture e.g. p4 on top of the three previous pictures (p1, p2 & p3), we notice a sharp drop in performance. By learning one additional picture, the network fails to retrieve the previous ones starting from a moderately distorted version.

In contrast, when our network learns random patterns, it becomes capable of retrieving a significantly larger number of distorted (20% noise) learned patterns compared to the previous experiment. The figure 6 shows the number of retrievable patterns with respect to learned patterns.

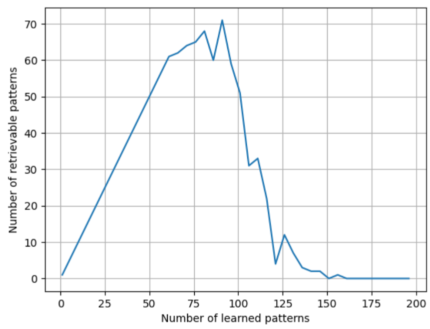
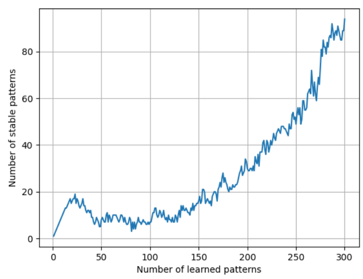


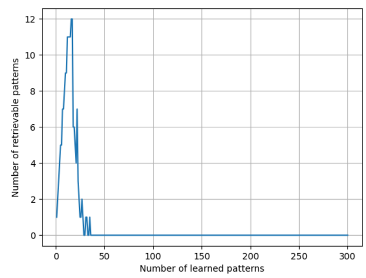
Figure 6: Number of retrievable patterns wto. the number of learned patterns (1024-unit sized)

This is due to the high chance of cross-talk between pictures compared to randomly generated patterns which are more likely to be independent from each other and subsequently not impacting the network's performance.

When we train our network on 300 100-unit random patterns, we notice the number of stable patterns increases proportionally with the number of learned patterns. However, when the convergence from a noisy pattern is used, the performance drops in a sharp manner.

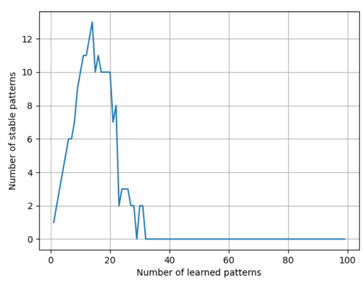


(a) Number of stable patterns wto. number of learned patterns (100-units)

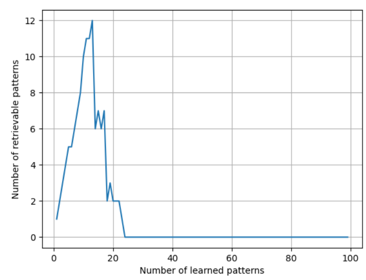


(b) Number of retrievable patterns from moderately distorted versions (20% noise) wto. learned patterns.

In the above figure, we should keep in mind that W had self-connections. When we eliminate self-connections, we obtain the following results. Self-connections promote spurious memory patterns and are not robust towards noise.



(a) Number of stable patterns wto. number of learned patterns (100-units and no self-connections)

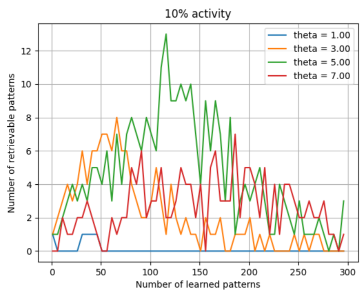


(b) Number of retrievable patterns from moderately distorted versions (20% noise) wto. learned patterns (no self-connections)

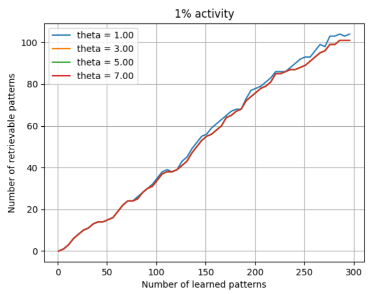
When we bias the patterns (e.g. our patterns contain more +1 than -1), our network's capacity drops. This is due to the high chance of cross-talk between biased patterns. This is coherent with the capacity results of the picture patterns since they are biased (background colour is more present).

3.6 Sparse Patterns

Sparser patterns are less likely to cross-talk with each other due to the large number of zeros they contain. In contrast, we retrieve less patterns when they are less sparse (e.g. 10% activity). In this case, θ has a significant impact since a larger θ implies a high chance of updating with the value 0, and a smaller θ implies a higher chance of updating with the value 1.



(a) Number of retrievable patterns wto. number of learned patterns (100-units and 10% activity)



(b) Number of retrievable patterns wto. number learned patterns (100-units and 1% activity)

4 Final remarks

This lab helped us program a Hopfield Network, train it on certain patterns (can be pictures like in our case) and retrieve them from distorted version.

Moreover, we witnessed how the energy profile of the distorted version evolve through time when using a symmetric weight matrix. We saw how the level of noise impacts the performance of the network

Finally, we tested the capacity of our network using multiple pattern with different scenarios. When patterns are slightly biased the capacity of the network takes a hit compared to the unbiased patterns. Furthermore, sparser patterns (i.e. less activity) are less likely to cross-talk with each other thus the networks performs well on them.