Miniproject: Hopfield model of associative memory

Project in Neural Networks and Biological Modeling course, EPFL

Adrian Shajkofci, Elodie Manet

09/06/2015

Lausanne, academic year 2014 – 2015

INTRODUCTION

Human beings are able to recall important events of their life, like their first day at college or their wedding, which means that not only are humans able to store memories, but that they are equally capable of forgetting insignificant ones. For example, memories considered less significant can be partially stored with less information in order to save memory.

Human memory works with strong associations, which means that if you see a picture, you may spontaneously recall how the picture was taken and stories about what happens to this day. Moreover, memory retrieval implies that pattern completion can spring from a partial cue. Some abstract models of neural networks, like Hopfield's model of associative memory, already describe how the recalling of previously stored items from memory works.

The Hopfield model consists of a network of N neurons, characterized by an index i: $1 \leq i \leq N$ and these neurons have binary activities: ON and OFF. The state variable of a neuron "ON" is S_i (t) =1 and S_i (t) = -1 for an "OFF" neuron. Neurons are fully interconnected with synaptic weights w_{ij}, represented by a N x N matrix and acting as a memory array. The size of the matrix is fixed by the number of neurons in the networks and does not change no matter how many patterns are stored. In each time step, the network state is updated as following: $S_i(t + 1) = \text{sign}\left(\sum_{j=1}^{N} w_{ij} S_j(t)\right)$.

In the present simulation, the Hopfield model is slightly modified by updating the synaptic weights w_{ij} continuously in time: $w_{ij}(t + 1) = \lambda w_{ij}(t) + \frac{1}{N} S_i(t) * S_j(t)$.

λ is the weight decay factor ranged from 0 to 1. λ being close to 0 indicates that most of the previous memories are forgotten. A λ close to 1 shows that most of the memories have been kept.

The task of the network is to recall previously patterns and to store new ones. The brain is constantly stimulated by external signals, and continuously learning and reorganizing itself. We will therefore set the hypothesis that the information storage probability p_s (set to 0.8 in the present simulation) is higher than the recall probability (1-p_s). Both phases will then alternate randomly for a duration of c time steps (set to 5 in the simulation) in order to mimic external input and recall procedure.

However, one can ask how many patterns from the pattern dictionary P_p can be stored in a network of N neurons and be recalled without exceeding an error set to 0.05 in our project. It could also be interesting to investigate the impact of the number of neurons N (and therefore the network weights size N^2) on the maximum dictionary size P_{max} of patterns that can be stored and recalled with a reasonable error.

The error measure that is used is the Hamming distance between the recalled and the original pattern, computed as $\frac{1}{2}\left(1 - \frac{a \cdot b}{N}\right)$ where a and b are the vectors of both binary images. An error of 0.5 indicates a purely random attribution of pixels and the total absence of correlation between the recalled and original patterns. On the other hand, an error of zero comes out of a perfect image reconstruction, and an error of 1 indicates that the recalled pattern has every pixel flipped.

The recall phase starts by trying to compare the weights stored in memory with a noisy version of an input picture, then updating it incrementally in order to retrieve the original image. Noise is added at the beginning of the recall phase in order to mimic the external input fed by the eye. Indeed, the vision pathways extract specific features from the incoming picture. When a memory is recalled, some differences in the features (orientation, size, color for example) will differ in comparison to the stored memory.

A last question could be asked concerning the effect of forgetting previous memories on the network performance.

This project was able to investigate the aforementioned interrogations and attempted to shed light on them.

EXERCISES

1. GETTING STARTED

In the first exercise, the maximum value of pattern dictionary size $P_{\max}$ the network can recall without exceeding an error of 0.05, was examined. The network size is N=100.

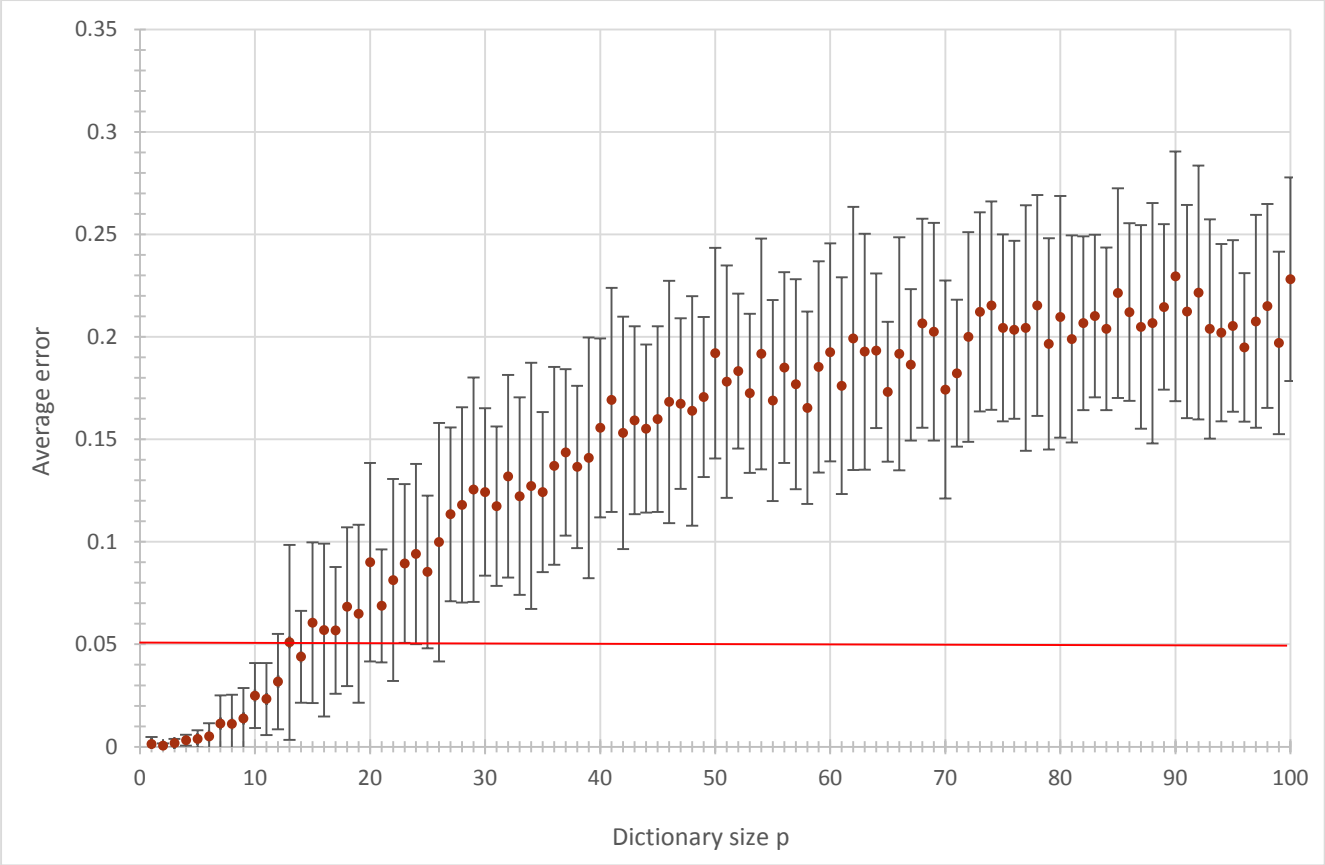

Figure 1: Average error for different p, the dictionary size, with parameters: N = 100, p_f = 0.1, p_s = 0.8, c = 5, λ = 1, for K=42 trials.

Figure 1 depicts the average error depending on different increasing values of p, the dictionary size. The error of 1 represents that every pixel is completely flipped (every 0 becomes 1 and vice-versa) therefore the maximal error possible is 0.5, which is a random assignation of pixels. An error of 0.2 means that 20% of the pixels are wrong. From this graph, it is clear that the average error generally increases while the dictionary size is increasing and seems to stabilise after p = 100. With a very low dictionary size (p less than 5), the average error is very low (about 0). As p increases, the average error will first increase in an exponential way until p =20, then, it follows a logarithmic curve to tends to an average value of ≈ 0.25 at p = 100. This increase can be explained by the fact that as the dictionary size increase, the model will make more mistakes when recalling the patterns. Before p=5, the model cannot make mistakes as there are few pattern to recall but as the size increase, it will be more complicated to handle the patterns. However, this error will not significantly vary after p = 50: no matter how many more patterns are stored in the synaptic weights, the performance will be roughly the same.

As a matter of comparison, a completely random draw would result in an error of 0.5; the Hopfield model results therefore have a better performance with 75% of the pixels correctly recalled.

From these results we can also see that the average error exceeds 0.05 from p = 13 therefore we chose to establish the maximum dictionary size $p_{\max}$ as $p_{\max} = 12$. However it has to be considered that the variance for the point p = 13 is rather high, which means that for a few simulations, $p_{\max}$ was slightly higher.

2. CAPACITY OF THE NETWORK

In the second exercise, the impact of the number of neurons (N) on the maximum dictionary size p_{max} is investigated. As in the first question, we keep the parameters fixed at p_f = 0.1, p_s = 0.8, c = 5, λ = 1.

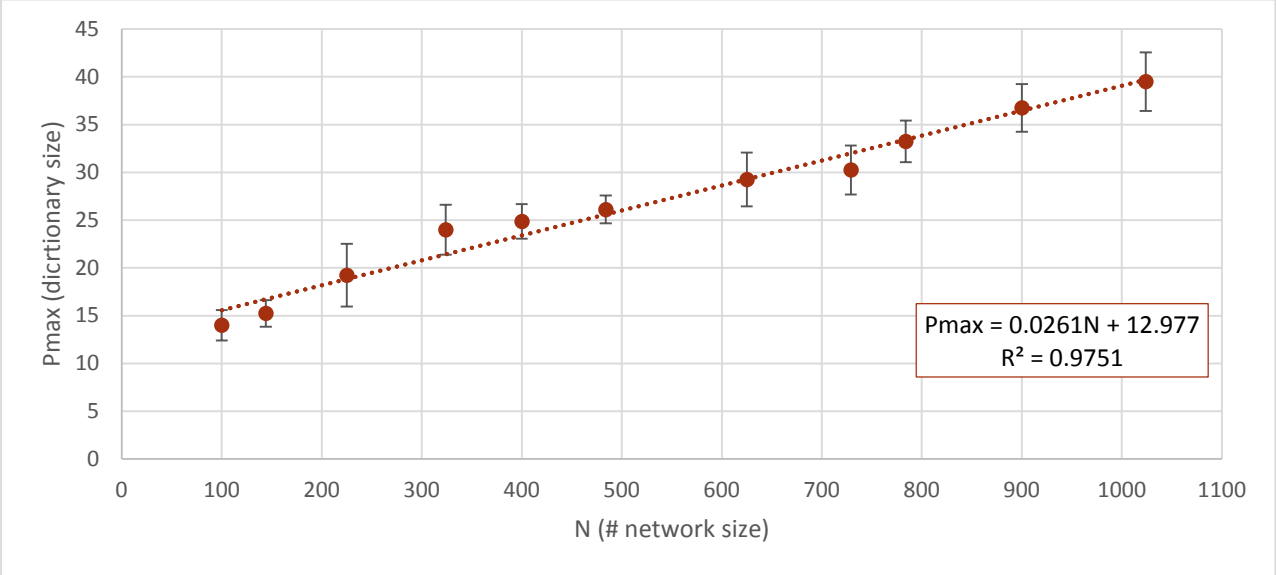

Figure 2: $P_{\max}$ for different network sizes N, with parameters: p_f = 0.1, p_s = 0.8, c = 5, λ = 1, for K=10 trials. A linear fit is then drawn.

Looking at the figure 2, we can observe that the higher the network size, N, the higher the $p_{\max}$, which means that maximum number of patterns, $p_{\max}$ which can be stored depends on the number of neurons in the network.

In the Hopfield model, $p_{\max} = \alpha N$, α being the capacity of the network[1]. This improvement of $p_{\max}$ comes from the fact that the information is stored in the connexions and not in the neurons. Then,

$$\alpha\ (capacity) = \frac{Number\ of\ pixels\ to\ store}{Number\ of\ connections} = \frac{p_{\max} * N}{N^2} = \frac{p_{\max}}{N} \Rightarrow p_{\max} = \alpha N$$

As in our simulation, we use a modified Hopfield model, we can see that $p_{\max}$ is slightly different but still has a linear relation to N.

3. IS FORGETTING BAD OR GOOD?

Until now, the weight decay factor, λ, has been at 1, which indicates that most of the memories have been kept. In this task, the effect of the variation of this parameter on the average error is examined. This procedure will address the fundamental question of which optimal value for λ produces the lower possible error. The sliding window is of size m = 5, therefore at every phase the pattern is drawn from a m-sized dictionary.

[1] Gerstner W., Kistler W., *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition* (french version), chapter 6.

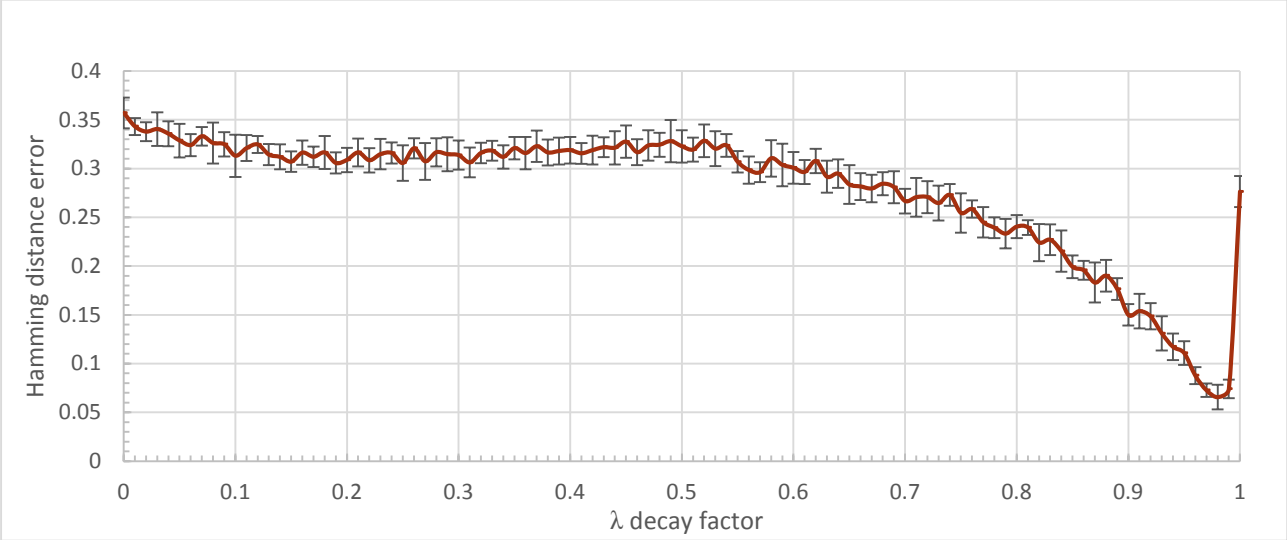

Figure 3: Error rate for different values of λ, the decay factor, with parameters: N=100, p_f= 0.1, p_s= 0.8, c = 5, m=5, Z=100 for K=10 trials.

This plot show that as the λ increases for 0 to 0.5, the error rate value does not significantly vary, in other words, it stays in a "plateau" at the average value of 0.33. With λ = 0, every past memory is erased once another picture is put into memory. Therefore there is at every recalling phase $\frac{1}{m}$ chance of retrieving the pattern that was previously stored. The average error with λ = 0 would therefore be $\frac{1}{2}(1-\frac{1}{m})$ = 0.4. The obtained results confirm this approximation.

After λ=0.5, the error rate decreases until reaching a minimum value of 0.065 at λ=0.97. As λ increases, the decay of learnt patterns becomes slower, meaning that only the oldest memories begin to fade away.

That means that the diminished patterns progressively are not part of the sliding pattern dictionary anymore, from which recalled patterns are drawn, therefore improving the performance. Eventually, the error rate increases abruptly to 0.25 for λ=1, which is coherent with our values for N = 100 (the final dictionary size with λ=1 is 55) in Exercise 1.

To conclude, the optimal value for λ to produce the lower error is around 0.97 and is the sweetest point of equilibrium between forgetting the patterns stored in the current window and the patterns that will never be drawn again.

4. INTERPRETATION

In this final section, we will examine the network performance of the joint effects on sub-dictionary size m for the sliding window operation also done in exercise 3 (m is ranging from 2 to 15) and λ (varying from 0 to 1).

With λ = 0, as before, the average error can be simply calculated with the probability formula $\frac{1}{2}\left(1-\frac{1}{m}\right)$. Taking m=2 and m=14 for example, we can find that the average error are $\frac{1}{2}\left(1-\frac{1}{2}\right)$ = 0.25 and $\frac{1}{2}\left(1-\frac{1}{14}\right)$ = 0.53 respectively which are very close to the obtained results. An average error of 0.5 indicates a purely random attribution of pixels and the total absence of correlation between the recalled and original patterns. This means that if the person remember nothing (λ = 0), with large m, the chance of recalling correctly is the same as piking random patterns in a sub-dictionary of size m since the system does not learn from anything other than the last stored pattern.

As λ increases, independently of m, we can notice than the average error rate decreases. Indeed, as λ increases, the decay of learnt patterns becomes slower and for that reason, only the oldest memories begin to vanish. Thus the weights of memories contained in the sub-dictionary are superior to the weights of the older memories that are not drawn during the recall phases, which improve the performance. Furthermore, the smaller value is m, the smaller the probability of forgetting a memory which comes from the sliding pattern

dictionary, m. In other words, the smaller value is m, the higher the performance in general, as we can see from our results.

In the final section, we will discuss the obtained results when λ = 1. Indeed, we can see that as m increases, the average error rate, even if it remains high, seems to decrease and converge to around 0.2-0.25 when m is very large (which is consistent with Figure 1 with P=100) . This could be due to the fact that with a small window size, the probability of recalling a pattern already stored in memory in the previous phases is way smaller than with a large sub-dictionary size, where the patterns contained in the sub-dictionary represents a larger part of the synaptic weights.

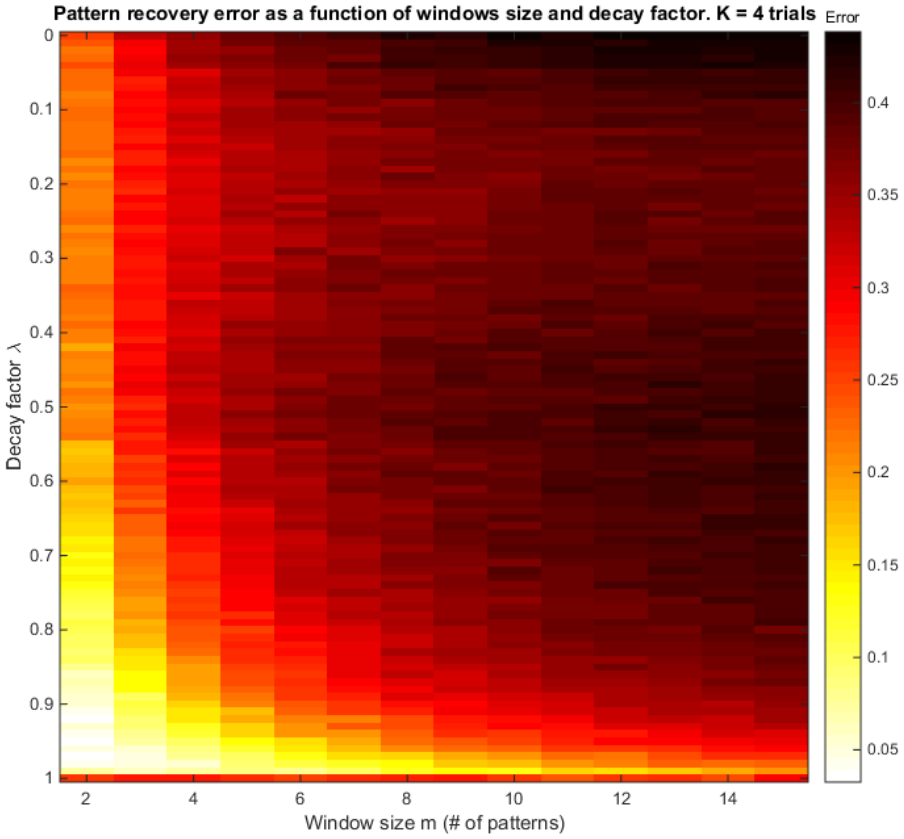

Figure 4: Error rate for different values of (m, λ), with parameters: p_f = 0.1, p_s = 0.8, c = 5, Z=100 for K=4 trials.

CONCLUSION

The Hopfield model of associative memory is a very simple mean of integrating the concept of memory storage in a network of neurons. However, by modifying a few parameters and assessing the model performance is it possible to uncover some of the principles of associative memory. First of all, we saw that the number of randomly-generated patterns than can be stored in a neuronal network is linearly dependent on the number of neurons in the network. Furthermore, when the model has to learn a large number of patterns, a usable way to overcome the size limitation is to have a working memory, modelled as a sub-dictionary, and a progressive fade of older memories. In that manner the performance while recalling recent events is still very good.

A further step in the modelling of memory would be to convert binary neurons, currently presented as weights of -1 or 1, to real neuron models with spike-generating capabilities. For that, more advanced frameworks such as NEURON or Nengo could be used.

In conclusion, even if there are some limitations regarding the performance with correlated pattern such as similar objects, letters or names[2], the Hopfield model of memory works well for randomly generated patterns.

[2] Gerstner W., Kistler W., *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, chapter 17.