Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 3
Radial basis functions, competitive learning and self-organisation

Samuel Leonardo Gracio and Martin Verstraete

September 26, 2018

1 Main objectives and scope of the assignment

List here a concise list of your major intended goals, what you planned to do and what you wanted to learn/what problems you were set to address or investigate, e.g.

Our major goals in the assignment were

- to familiarize with the Hopfield network and understand how it was possible to recall detailed images in a few iterations
- to understand the concept of energy

2 Methods

We worked with Python and with a Google Colab environment. We only used numpy functions and matplotlib.

3 Results and discussion

3.1 Convergence and attractors

- Stored patterns

```
Old pattern was  [-1. -1.  1. -1.  1. -1. -1.  1.] updated pattern is  [-1. -1.  1. -1.  1. -1. -1.  1.]
They're the same
Old pattern was  [-1. -1. -1. -1. -1.  1. -1. -1.] updated pattern is  [-1. -1. -1. -1. -1.  1. -1. -1.]
They're the same
Old pattern was  [-1.  1.  1. -1. -1.  1. -1.  1.] updated pattern is  [-1.  1.  1. -1. -1.  1. -1.  1.]
They're the same
```

The network is able to store all three patterns

- Distorted patterns

```
The new patterns is  [-1. -1.  1. -1.  1. -1. -1.  1.]  and the correct pattern was  [-1. -1.  1. -1.  1. -1. -1.  1.
They're the same
The new patterns is  [-1.  1. -1. -1. -1.  1. -1. -1.]  and the correct pattern was  [-1. -1. -1. -1. -1.  1. -1. -1.
The new patterns is  [-1.  1.  1. -1. -1.  1. -1.  1.]  and the correct pattern was  [-1.  1.  1. -1. -1.  1. -1.  1.
They're the same
```

x1d and x3d converge towards stored patterns. After 3 iterations, x2d does not converge toward x2

- Number of attractors in the network

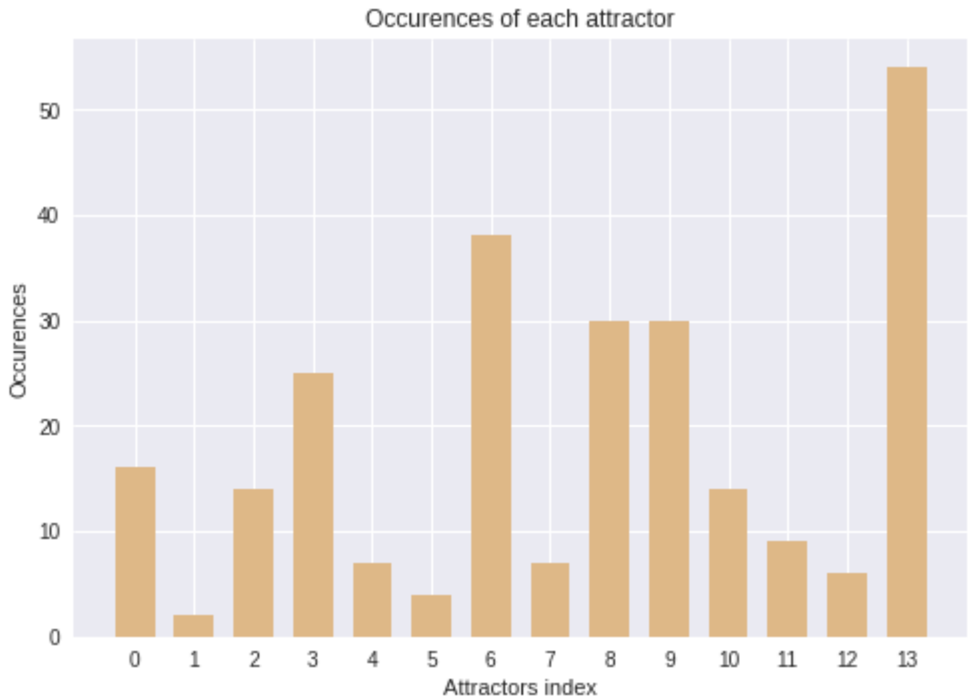


Figure 1: Occurrences for each attractor

We found 14 attractors in this network. Moreover, if a pattern is an attractor, the opposite pattern in an attractor too.

- More dissimilar patterns

```
The new patterns is  [ 1.  1.  1.  1.  1. -1.  1.  1.]  and the correct pattern was  [-1. -1.  1. -1.  1. -1. -1.  1.]  The convergence takes  3  iterations
They are different
The new patterns is  [ 1.  1.  1.  1.  1. -1.  1.  1.]  and the correct pattern was  [-1. -1. -1. -1. -1.  1. -1. -1.]  The convergence takes  3  iterations
They are different
The new patterns is  [ 1. -1. -1.  1.  1. -1.  1. -1.]  and the correct pattern was  [-1.  1.  1. -1. -1.  1. -1.  1.]  The convergence takes  2  iterations
They are different
```

It seems that if the input patterns are very dissimilar, the memory cannot recall the stored patterns

3.2 Sequential Update

- **Check that the three patterns are stable**

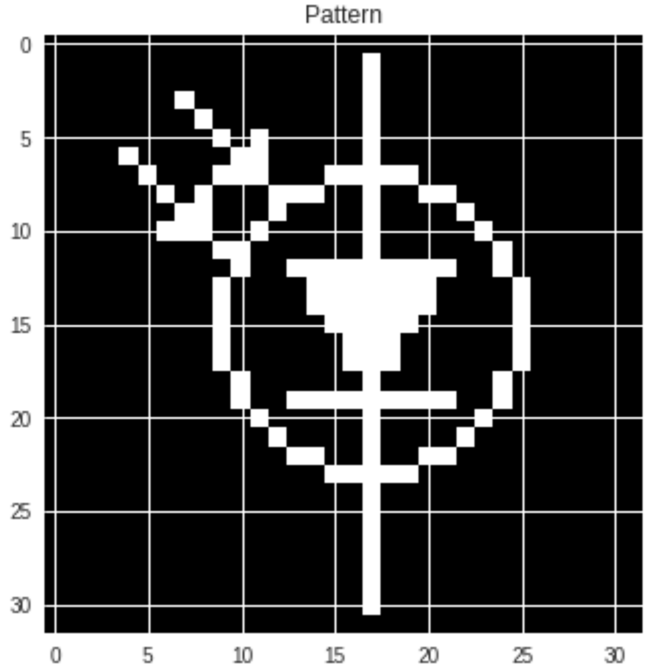


Figure 2: Pattern p3

```
Old pattern was  [-1 -1 -1 ... -1 -1 -1]  updated pattern is  [-1. -1. -1. ... -1. -1. -1.]  and the convergence takes  1  iterations
They're the same
Old pattern was  [-1 -1 -1 ... -1 -1 -1]  updated pattern is  [-1. -1. -1. ... -1. -1. -1.]  and the convergence takes  1  iterations
They're the same
Old pattern was  [1 1 1 ... 1 1 1]  updated pattern is  [1. 1. 1. ... 1. 1. 1.]  and the convergence takes  1  iterations
They're the same
```

The three patterns are stable

- **Can the network complete a degraded pattern?**

Figure 3: Degraded pattern p10

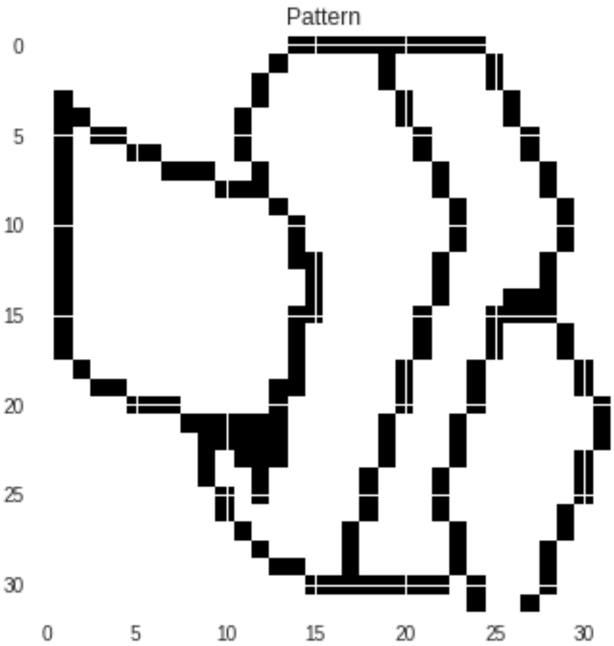


Figure 4: Restored pattern p1

The network can complete a degraded pattern

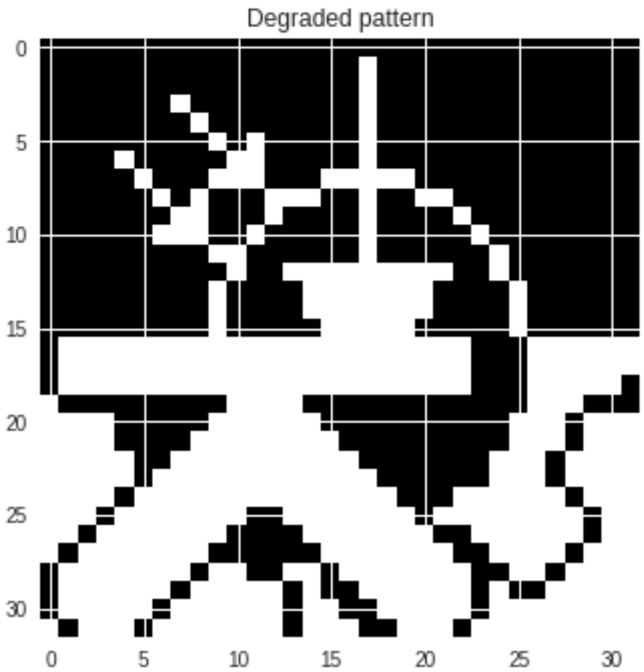


Figure 5: Degraded pattern p11

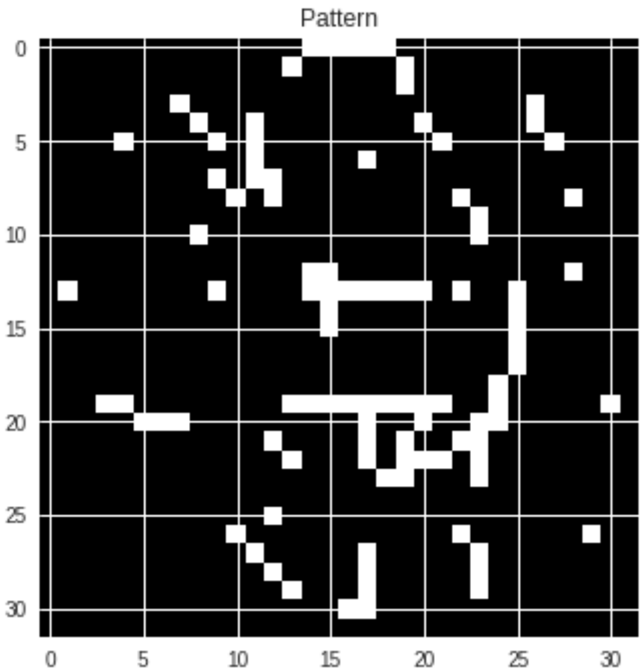


Figure 6: Restored pattern

So, the network cannot complete a pattern that is a mixture of two learnt patterns in batch mode

- **Sequential update with random units**

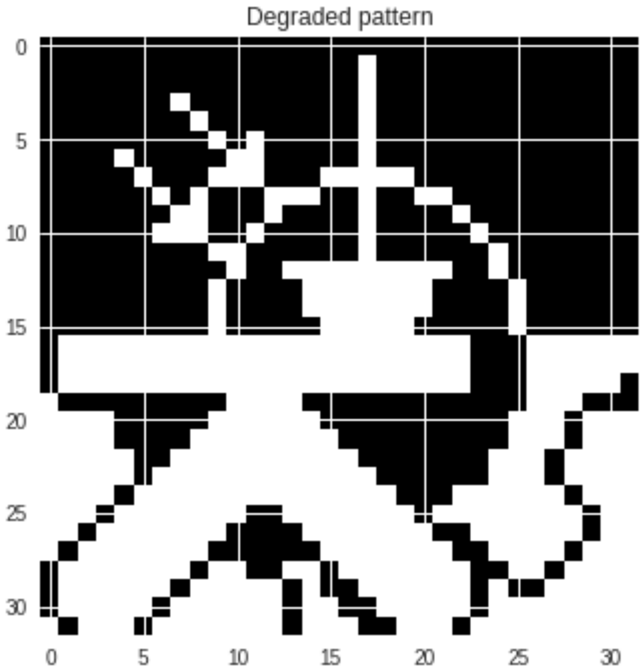


Figure 7: Degraded pattern p11

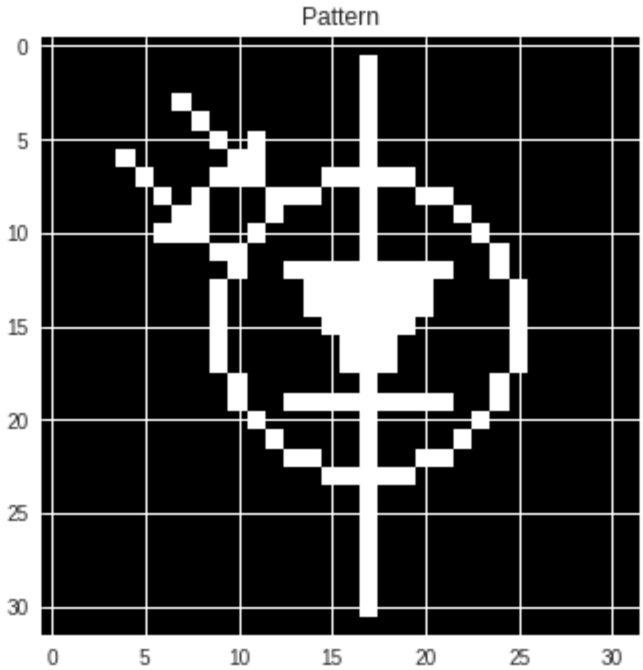


Figure 8: Restored pattern p3

With a sequential update, the pattern converges most of the time to the pattern p3 in a few iterations. The number of iterations needed for convergence is not constant

3.3 Energy

- **Energy at the different attractors**

```
The energy at attractor p1 is -1439.390625
The energy at attractor p2 is -1365.640625
The energy at attractor p3 is -1462.25
```

- **Energy at the points of the distorted patterns**

```
The energy at distorded pattern p4 is -720.48046875
The energy at distorded pattern p5 is -525.890625
The energy at distorded pattern p6 is -683.296875
The energy at distorded pattern p7 is -685.73046875
The energy at distorded pattern p8 is -171.546875
The energy at distorded pattern p9 is -267.51171875
The energy at distorded pattern p10 is -415.98046875
The energy at distorded pattern p11 is -173.5
```

- **Evolution of the energy**

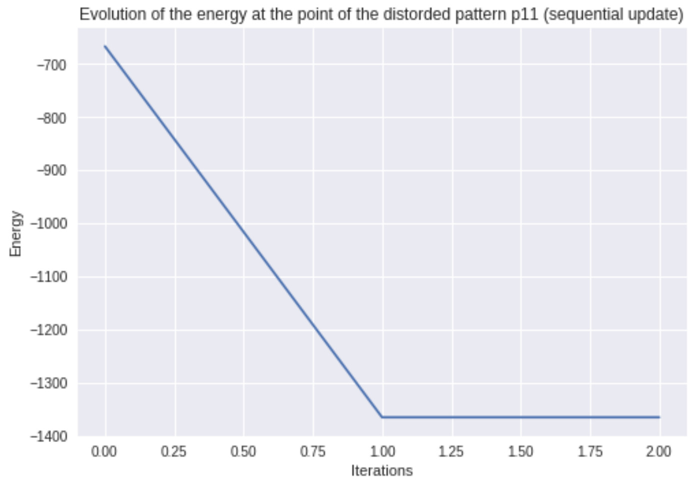


Figure 9: Sequential update

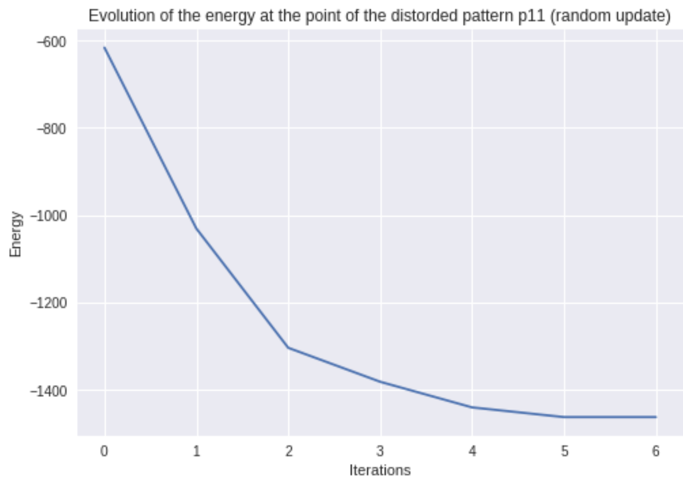


Figure 10: Random update

The sequential update converges faster than the random update, but the energy is lower in the case of the random update.

- **Behavior with a random matrix (sequential update)**

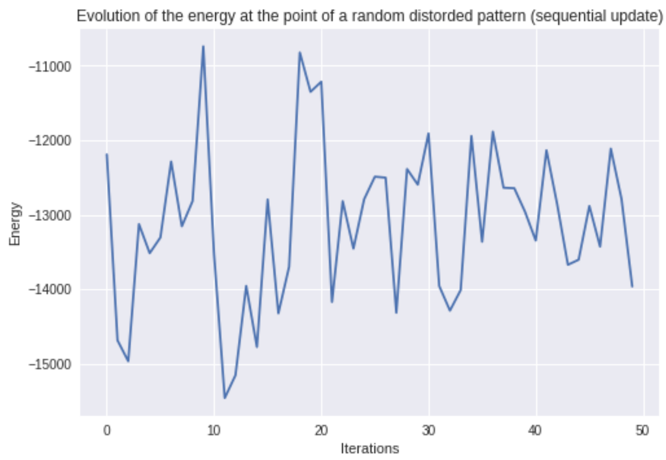


Figure 11: Random matrix

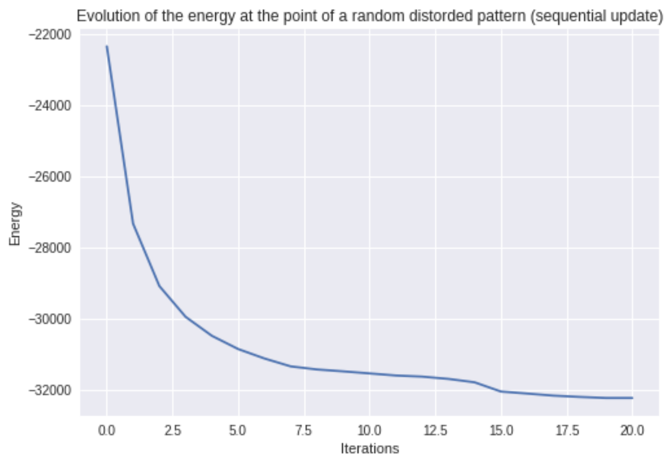
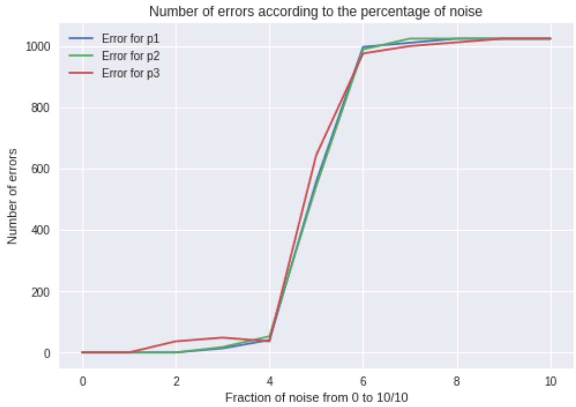


Figure 12: Random and symmetric matrix

With a random matrix, the convergence is not guarantee and the energy oscillates. With a symmetric random matrix, the convergence is possible and the energy decreases until it reaches a minimum.

3.4 Distortion Resistance



- We can see that we can remove almost 40% of the data by flipping it from 0 to 1, i.e by adding noise, without changing the success rate. After that, it increases quickly and you have 100% of error at the end, you have the exact contrary of your pattern.

- The results don't change whatever is the pattern.

- It doesn't converge to the right attractor, at the end it converges to the exact contrary of the wanted attractor. An extra iteration doesn't change that. We found one other noisy attractor.

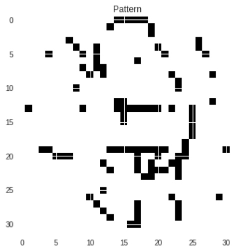


Figure 13: Noisy attractor sometimes appearing

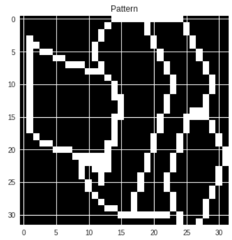


Figure 14 : Exact contrary of the initial attractor

3.5 Capacity

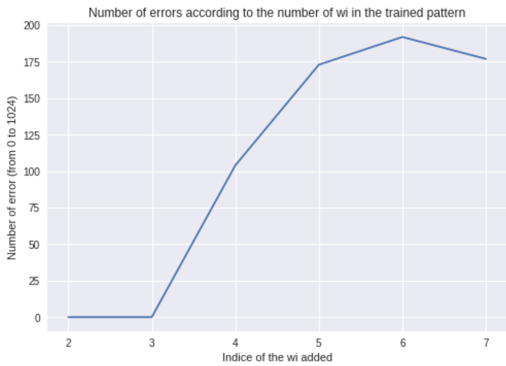


Figure 15 : Number of errors according to the number of wi

We choosed a moderatly distorted pattern, p2 with 20% of noise. We added some patterns to the network and we can easily see that adding just another pattern causes many errors. If you add others, it continues increasing until a certain value. It is a bit abrupt.

Figure 16 : Number of errors according to the number of w_i

Now we do it with random patterns. We can see that we can add more patterns without having a high number of errors. We arrive at 175 errors when we had almost 10 patterns, which was impossible without random patterns.

Random patterns seems to be less attractive for the network.

The 0.138N means that approximately 138 vectors can be recalled from storage for every 1000 nodes. Adding some random patterns can help you to store more patterns because they are less attractive than organized pattern like the pictures in our dataset. The fact that you have random 1 and 0 values makes less sense than having structured figures with coherence, it helps the network to learn, they are more attractive.

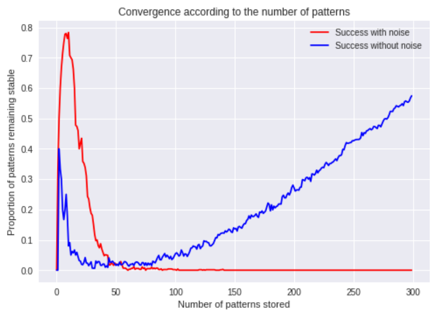


Figure 17 : Success rate for noisy and non-noisy random patterns

Now we've created 300 random patterns with a 100-unit network. We can see that adding some patterns have a bad influence without noise at the beginning and then we have better results. This is due to the fact that when you add several patterns, you will have high values on the diagonal of your weight matrix, that means a more attractive pattern, even if its random.

For the noisy ones, we have good results at the begining and then it gives pretty bad results due to the noise.

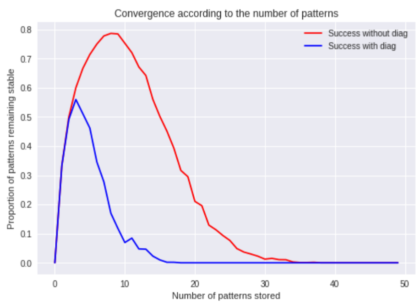


Figure 18 : Success rate with and without the diagonal values

Since the wii in the diagonal are really strong, we've decided to erase them and to compute again.

We obtain pretty better results without the diagonal than with it.

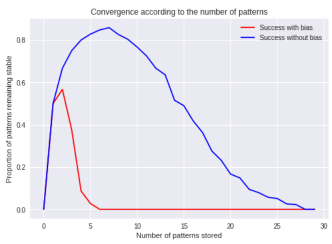


Figure 19 : Success rate with bias and without (p = 0,75)

We try to generate random patterns and patterns with bias, a probability of p = 0,75 to have 1 for instance. We can see that since the patterns are less random, adding a new pattern decreases quickly the success rate because each pattern have more attractivity.

3.6 Sparse pattern

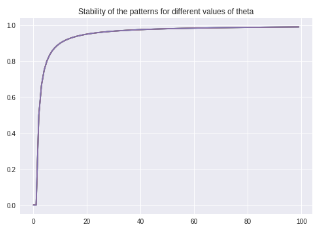


Figure 20 : Success rate for the different values of theta

We changed the way of updating the weights with theta as bias term and we found a better result rate when we had a high theta value but we had problem to understand the results…

4 Final remarks *(max 0.5 page)*

Please share your final reflections on the lab, its content and your own learning. Which parts of the lab assignment did you find confusing or not necessarily helping in understanding important concepts and which parts you have found interesting and relevant to your learning experience?

Here you can also formulate your opinion, interpretation or speculation about some of the simulation outcomes. Please add any follow-up questions that you might have regarding the lab tasks and the results you have produced.

This was a very interesting lab with application, the fact that we could see the patterns was very useful. Now Hopfield networks are much clearer for us.