Artificial Neural Networks: Exercise session 2

Stijn Staring (r0620003)

March 11, 2021

1 Exercises of Section 1: Hopfield Network

Create a Hopfield network with attractors T = [1 1;-1 -1; 1 -1]T and the corresponding number of neurons. You can use script rep2 as a basis and modify it to start from some particular points (e.g. of high symmetry) or to generate other numbers of points. Start with various initial vectors and note down the obtained attractors after a sufficient number of iterations. Are the real attractors the same as those used to create the network? If not, why do we get these unwanted attractors? How many iterations does it typically take to reach the attractor? What can you say about the stability of the attractors?

In Figure 1 all the found attractors can be seen which are more than the three desired ones that were specified to the Hopfield network. It can thus concluded that the despite the “newhop” matlab function tries to design a Hopfield network with the minimum of unspecified target attractors, they often occur. Table ?? shows the found attractor points. when designing the energy function for the system that has a corresponding local minima on at the attractor points, it can be that also unwanted attractor points become local minima. When the assumption of no bais term of the model is made, these correspond with the negative of the desired attractors. Also, a mixture of the desired states are possible. The amount of iterations it takes to reach an attractor depends on the start position. All the symmetric starting points of Figure 1 reached an atractor in less then 5 iterations. However, the amount of iterations can dramatically increase when the starting point is very close to an unstable attractor. For example when the starting point chosen is: $[-1; -10^{-20}]$ it takes 308 iteration to reach the stable attractor $[-1; -1]$. When the system is in a stable attractor and a little noise is added the system will move back to the stable attraction point. An unstable attractor on the other hand will move away when this happens. The stability of the diffent attractor points can be seen in Table 1.

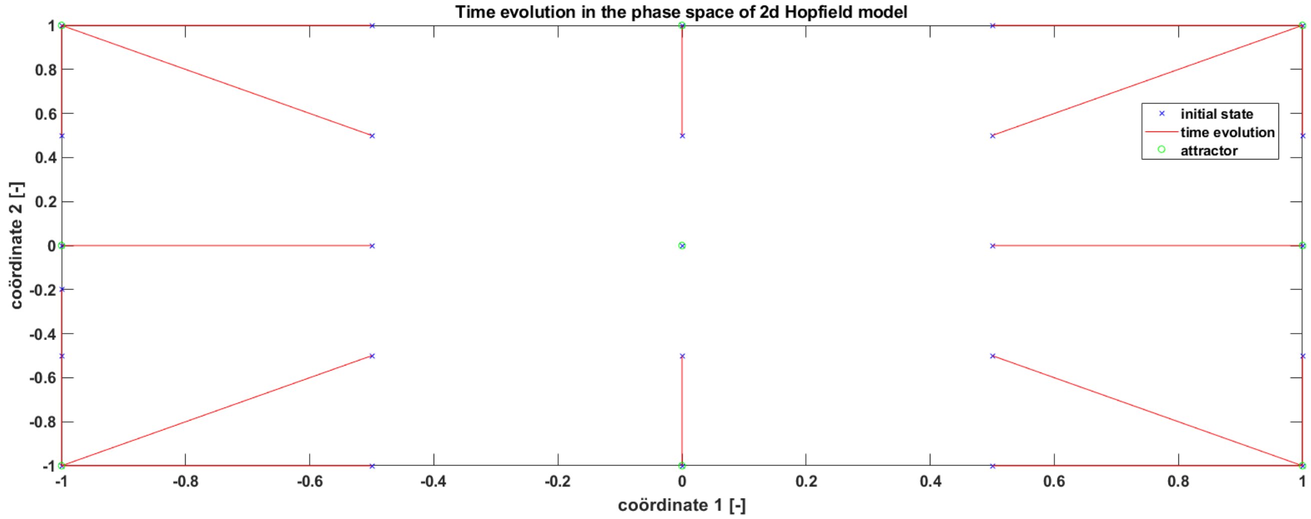

Figure 1: All the found attractors.

Table 2 shows an overview of the found attractors in a 3D space. Only the three specified target attractors were found during the experimentation with different starting points. All three are stabel points.

Attractor	Point	Stability
Attractor 1	$[1; 1]$	Stable
Attractor 2	$[-1; -1]$	Stable
Attractor 3	$[1; -1]$	Stable
Attractor 4	$[-1; 1]$	Stable
Attractor 5	$[0; 0]$	Unstable
Attractor 6	$[0; 1]$	Unstable
Attractor 7	$[0; -1]$	Unstable
Attractor 8	$[1; 0]$	Unstable
Attractor 9	$[-1; 0]$	Unstable

Table 1: Overview of the different attractor points found in a 2D plane.

Attractor	Point	Stability
Attractor 1	$[1; 1; 1]$	Stable
Attractor 2	$[-1; -1; 1]$	Stable
Attractor 3	$[1; -1; -1]$	Stable

Table 2: Overview of the different attractor points found in a 3D plane.

The function hopdigit creates a Hopfield network which has as attractors the handwritten digits 0,...,9. Then to test the ability of the network to correctly retrieve these patterns some noisy digits are given to the network. Is the Hopfield model always able to reconstruct the noisy digits? If not why? What is the influence of the noise on the number of iterations?

The noise that is added to the digits is gaussian noise. The noise variable determines the standard deviation of the gaussian that is sampled from to retrieve the noise that should be added. It can be concluded that when the noise is too high, the Hopfield model is not able to reconstruct the correct digits. The reason is because when the noise is too heavy the starting point can become closer to other digits than the original one. Increasing the amount of iterations will not solve this issue. When there is more noise, the system needs more iterations to reach the attractor points.

2 Exercises of Section 2: Long Short-Term Memory Networks

Train a MLP with one hidden layer after standardizing the data set. Investigate the model performance with different lags and number of neurons. Explain clearly how do you tune the parameters and what is the influence on the final prediction. Which combination of parameters gives the best performance on the test set?