Artificial neural networks

Assigment 3: Recurrent neural networks

Lood, Cédric
Master of Bioinformatics

May 29, 2016

1 Context

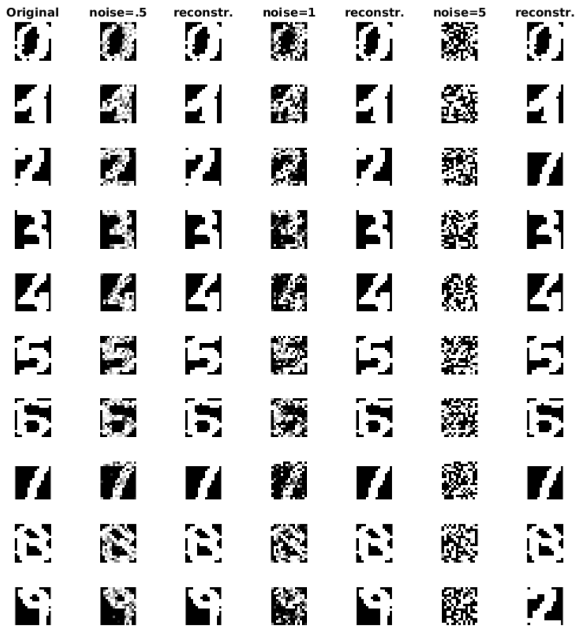
In this exercise, we explore the use of recurrent neural networks and their applications as associative memories. Items can be stored as equilibrium points of the network, similarly to those in dynamical systems.

2 Hopfield network

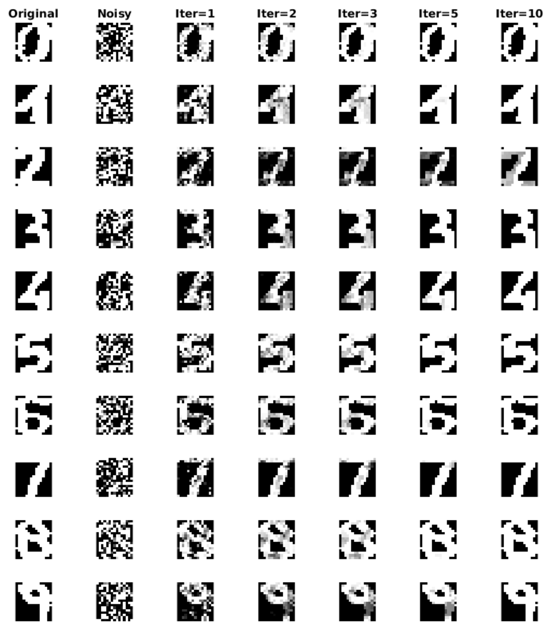
In this section, we use hopfield neural networks to store 10 digitized handwritten digits. The network has an architecture of 240 neurons, fully connected and arranged in a single layer, with a saturation function *satlins*. It is trained in such a way that each digit is an attractor point of the network, where they represent minimums of the energy function. To note is that with an architecture consisting of 240 neurons, the storage of 10 patterns is possible with a small chance of error.

The first column of the left-hand side of figure 1a corresponds to the patterns that are stored in the network. Each digit is encoded as a 15 by 16 vector of pixels. Each 3 pairs of columns following the digits in the first column corresponds to an attempt at retrieving a noisy digit (the first column in a pair) through a fixed amount of iterations. As can be seen, the process is fairly good, failing only when the amount of noise is large - examples of failures are located in the last column of the digits 2, incorrectly reconstructed as a 7, and the digit 9, incorrectly reconstructed as a 2. In both cases, the error do seem to make sense to the human eye. What happens there is that we jump from the neighborhood of the correct attractor due to the noise to the neighborhood of another attractor.

Similarly, the first column of the figure 1b displays the original pattern. A fixed amount of noise is then introduced (2.5 in our example), then an iterative reconstruction process is attempted.



(a) Fixed amount of iterations (50)



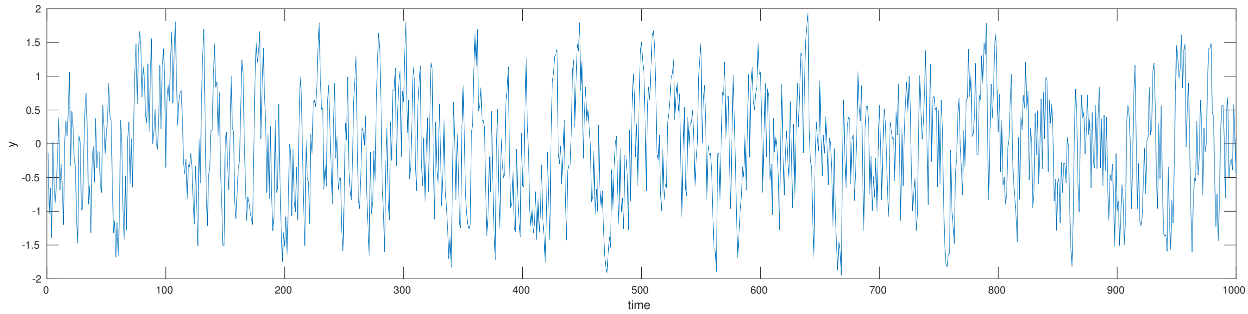
(b) Fixed amount of noise (2.5)

Figure 1: Hopfield network reconstruction of noisy digits

I did not detect spurious states - which are attractors not put in explicitely in the network. The probability that some exists though is not null.

3 Elman network

In this section, I investigated the use of Elman recurrent networks, known to be able to deal with time varying patterns, on a so-called *Hammerstein system* (see figure below). Elman network consists in two-layers, a hidden one, that uses sigmoid transfer function and has a feedback to the input, and an output layer that uses a linear transfer function.



To determine appropriate numbers of epochs to use and the number of hidden neurons for the network, I investigated systematically multiple settings and looked at the test MSE. The figure **??** summarizes the results. First, I determined the number of training examples that worked best on the standard setting (20 neurons, 1000 epochs). This number fluctuated significantly across multiple runs as the system generated is stochastic. I kept the value of 300 for the rest of the analysis.

I then proceeded to determine the combination of epochs and number of neurons that resulted in the lowest test MSE before finally building a model of the system.