

Exercises L2: Hopfield networks

Exercise 1

The goal of the exercise is to write a working implementation of a Hopfield network. In the files `hopfield.py` and `test_hopfield.py` you will find a frame you can use to get you started.

- Open file `hopfield.py` and read the docstrings to see how the functions that you'll have to write look like
- How can we go about testing them?
- Start writing tests for the `update_activity` function, and then implement it until the tests pass
- Now write tests and body for the function `store_pattern`

Exercise 2 (optional)

The first version of the `store_pattern` function is likely to be quite inefficient for large networks. Optimize it by substituting the for-loops with the `numpy.outer` function.

Exercise 3

We would like to use our Hopfield network to store images of African animals:



The script `animal.py` contains three functions:

- `load_binary_image`: load a binary image and transform it in a +1/-1 activity pattern
- `show_pattern`: display an activity pattern in a matplotlib figure
- `random_flip`: add noise to the pictures

There is already a snippet of code that loads the first three animals and displays the first one. (Note: you can use the larger versions of the animals if you wrote an efficient network.)

- Create new weights for an Hopfield network and store the first three animals
- In the file `mystery.png` there is a very noisy picture taken by an amateur explorer. What animal did he photograph?

c) Explore the robustness of the Hopfield network:

- write a function that computes the difference between two different activation patterns (it should return the number of different elements)
- for each image, add random noise using the `random_flip` function, and try to reconstruct the image by updating the activity of the network for a few times
- make a plot of reconstruction error as a function of the percent noise in the image

d) Store the fourth animal in the Hopfield network and try recovering one of the animals. What happened? Why did the network break down so quickly?