# Laboratory Assignment 2 - 2 (LAB2-2)
# Hopfield Networks

Solve the following assignment, whose completion is required to access the oral examination. Upload the assignments all-together in the Moodle platform of the course (once you have completed all the labs, not only this single one) as a compressed folder including one subfolder for each laboratory.

The subfolder for this lab should be called "*LAB2_2*" and should include the Matlab scripts and the other files as requested in the assignment below. You can organize the code as you wish, implementing all the helper functions that you need, provided that these are included in the subfolder and are appropriately called in the scripts.
Properly organize the files requested for the different assignments into different sub-folders. E.g. "LAB2_2/Assignment1" for the first assignment, "LAB2_2/BonusTrack1" for the first bonus track, etc.

Bonus track assignments are meant to be for those who finish early, but they are not formally required for completing the Lab Assignment.

Supporting material for this assignment is listed below:

**Lectures' Slides**
Hopfield networks from CNS lectures "Part2_Lecture2".

**Matlab documentation**
*Matlab User's Guide* https://www.mathworks.com/help/index.html
Matlab documentation using the `help` command

## Assignment 1 – Small Image Dataset

The assignment consists in the following points:

1) Download and unzip the "lab2_2_files.zip". The archive contains
   - "lab2_2_data.mat", containing the input patterns (representing images) to use for the assignment, stored into 3 vectors p0, p1, p2
     Note: each input pattern represents a digit (i.e. p0 -> '0', p1 -> '1', p2 -> '2')
   - "distort_image.m", a (supporting) Matlab function that you need to use for distorting images as a part of the assignment
2) Implement all the code that is required for the different stages in the operation of a Hopfield network, i.e. the storage phase (learning) and the retrieval phase (initialization, iteration until convergence, outputting)
   Note: in the code also include the computation of the overlap function (with respect to the training patterns) and of the energy function.
3) Use the 3 input vectors p0, p1 and p2 to train the Hopfield network
4) Generate distorted versions of the 3 patterns using the distort_image() function, whose arguments are (ordered): the pattern to distort and the percentage of distortion. For each pattern p_i, generate 3 distorted versions, with percentage of distortion 0.05, 0.1 and 0.25. I.e., d_i_1 = distort_image(p_i,0.05); d_i_2 = distort_image(p_i,0.1); d_i_3 = distort_image(p_i,0.25); In this step, you will have generated a total number of 9 patterns.
5) Feed the trained Hopfield network with the 9 input patterns generated in the previous point. For each of the 9 input patterns:
   - plot the energy as a function of time;
   - plot the overlap computed for each one of the training patterns, as a function of time; use the same figure for plotting all the overlaps (use Matlab command hold on to plot more than one line in the same figure)
   - plot the reconstructed image and compute a measure of discrepancy between the reconstructed image and the optimal memory (add this result as title caption to the figure, e.g. with command title). Use the command imagesc for this plot, see more info below (and in Matlab documentation)

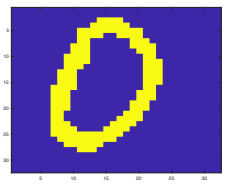
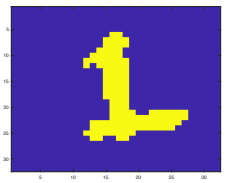
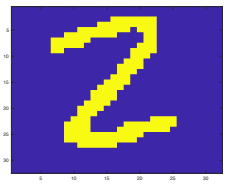
   Note: 3 plots in total for each of the 9 input patterns.

The output of this assignment should then consist in the following data:

- The script .m file(s)
- The plots generated in point 5), i.e. 3 x 9 = 27 plots (in .fig or .png format, please use a significant name for each file, e.g. "distorted_0_1_energy.fig", "distorted_0_1_overlap.fig", "distorted_0_1_reconstructed.fig")

Some hints/notes on the manipulation of the images in the input patterns (for this assignment):

- each input pattern represents a 32x32 bitmap (+1/-1)
- you can reshape a 1024 vector to a 32×32 matrix by img = reshape(vec,32,32);
- you can visualize an image with command imagesc(img);

# Bonus Track Assignment # 1 – Synthetic data

Use the code for the implementation of the Hopfield network (written for Assignment1) and train a new Hopfield network to store the following input patterns

$$p1=[-1\ -1\ +1\ -1\ +1\ -1\ -1\ +1]';$$

$$p2=[-1\ -1\ -1\ -1\ -1\ +1\ -1\ -1]';$$

$$p3=[-1\ +1\ +1\ -1\ -1\ +1\ -1\ +1]';$$

In this case, use an implementation without the input bias. Check if the network has effectively stored the 3 patterns as fixed points.

Now consider the following distorted inputs

$$p1d=[+1\ -1\ +1\ -1\ +1\ -1\ -1\ +1]';$$

$$p2d=[+1\ +1\ -1\ -1\ -1\ +1\ -1\ -1]';$$

$$p3d=[+1\ +1\ +1\ -1\ +1\ +1\ -1\ +1]';$$

Feed them to the network and apply the activation update until convergence.

P1) Did all the patterns converge to the appropriate stored memory? Store the activations of the network neurons (as a function of time) in a matrix (e.g. Nneurons x Ntimes). Once the network has converged, plot such activations (as a function of time) on the same figure (cf. hold on command in Matlab).

The output of this assignment should then consist in the following data:

- The script .m file(s)
- The matrices of neurons' activations generated at point P1) (i.e., 3 matrices); save these matrices into a .mat file.
- The plots generated at point P1) (i.e., 3 plots), in .fig or .png format (use significant names for the files)

# Bonus Track Assignment # 2 – How much can you remember?

In this bonus track exercise we study how many digits our Hopfield Network of Assignment 1 can store and recover.

To do so, you can train the Hopfield Network incrementally. The update rule is the following:

$$W^{new} = W^{old} + \frac{1}{N}\xi\xi^T - I,$$

where $\xi$ is the new pattern to be stored. Remember to add the *input bias* when computing the activations of a neuron, and to remove self-connections among neurons.

Load the 10 digits from the *digits.mat* file. Then, proceed as follows:

1) Learn the first 3 digits, and generate their noisy version with a percentage of distortion of 0.05.
2) Now repeat:
    a. Compute the average discrepancy of all the patterns used so far.
    b. Compute the overlap for each of them over "time" with respect to the original image.
    c. Add the next pattern (proceed in order) to the Hopfield Network, and generate its distorted version.

Plot the average discrepancy with as a function of the number of stored patterns. Is there a drop in reconstruction performance at some point? How bad is it?

Finally, plot the overlaps you computed every time you added a pattern. Is there any of them which is better recalled than the others?

The output of this assignment should then consist in the following data:

- The script .m file(s)
- One plot for the average discrepancy
- 8 plots with the overlaps (the first one with digits {1,2,3}, the last one with digits {1,2,…,10}