

Exercise 2- Fall term 2020. Due Jan 20, 2021 (in Moodle Box)

This is 25% of your final grade. It will only count if you pass the final exam. Please submit on time. You should work in groups of 2 or 3.

The deadline is January 17, 2021 (23:59). Please submit on time.

Late penalties: There is a 5 point penalty (out of 100 for the exercise) for submissions until January 29. From January 30 – Feb. 4 the penalty is 15 points total. No acceptance after Feb 4. (So a grade of 0 on the project.)

Include on your submission report the date of submission. Make a clear report. (Be sure to include .ר.ן of all the submitters.

1. **Implement** a 100 neuron Hopfield network from **your own code**. (It should use asynchronous updating.) The neurons should be simple McCulloch Pitts neurons .
2. Set the weights so that you are storing the following 9 input vectors (expressed here as a 10 x 10 matrix, but they are really vectors of length 100. Each vector is a 10 x 10 bipolar picture of the Hebrew letters .ט ,נ ,ר ,י ,ה ,ד ,ג ,כ ,ח (i.e. a 10 x 10 picture of +1/-1 that gives a visual sketch of the letter.)
3. Are all these pictures stable points? If not, which ones are not? Explain. If you store less than all the 9 vectors, are the results better?
4. Initialize the network with the picture of the Hebrew letter "נ" with addition of 10% random noise. Do this 10 times (i.e. ten different runs with initialization of the letter "נ" but with randomly chosen 10 neurons flipped as input. Do this 10 separate times.)
 - i) Does it always converge? Does it always converge to "נ"?
 - ii) Now repeat for 20%, 30%, 40%, 50% ... random noise. Does it always converge in these cases? Does it always converge to "נ"? Explain your results (briefly). Present a table of results.

You should report in each section with snapshots of the examples

Include one sequence of changes of an epoch. (An epoch is a sequence where you have updated all 100 neurons.)

Be sure to report how many epochs it took for each pattern to converge (if it ever does).