LAB XII
Memory model with Hopfield network

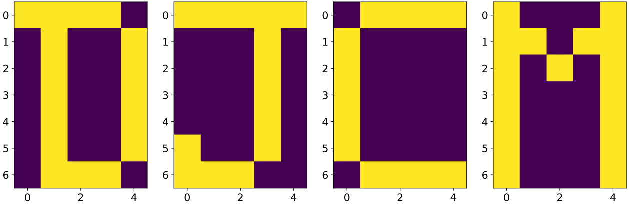
Jakub Tworzydło

Institute of Theoretical Physics UW
Jakub.Tworzydlo@fuw.edu.pl

28/05/2024 Pasteura 5, Warszawa

Task

Model a simple 7x5 Hopfield network using the letters as the stored patterns – use the four patterns below to train the network



Calculate the matrix of connections W (learning step). Perform $s = 5$ steps of the update procedure on the memorized patterns, check that they stay unchanged.

Randomly flip $n = 5$ pixels in each of the patterns, plot the noisy pictures, then run the “recognition” and plot the results. Use sub-plots (obligatory) for ease of comparision. How many errors do we need to get a halucination?

Hints

We can code patterns as `numpy arrays` eg.
`D = np.array( [ [1, 1, 1, 1, 0], ... ])` and then flatten
`x = D.flatten()*2-1` to get the desired vector of inputs.

The outer product $x^T x$ is easily computed by `np.outer(x, x)`. For the update step one can use the whole vector operation `np.sign( )`.

For fun: plot the matrix W with `plt.imshow` (Hinton diagram).

Extra

Propose a simulation, which will test capacity of the Hopfield network. To achive this, one should generate randomly N patterns for J neurons and memorize them with the matrix W. Perturb each pattern x_{per} by one bit flip and find the corresponding stable state x_{stab}. Calculate the overlap $\langle x_{\mathrm{per}} x_{\mathrm{stab}} \rangle / J$ averaged over all patterns and plot the average as a funcion of N/I. The overlap should clearly drop at the capacity threshold.