,

CS 6673 fall 2018

Individual Project, Due December 11, 2018

Write a program to implement the discrete Hopfield neural network having n neurons. The algorithm is basically the same as that given in our lecture, except that the transfer function is the sign function with zero bias, and the external input term is absent.

The weights are determined by Hebb's rule with the diagonal elements set to zero, and biases are taken to be zero also. The training set consists of the following 4 vectors:

$$\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}^T, \quad \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix}^T, \quad \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}^T, \quad \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}^T$$

. Neurons should be updated asynchronously and randomly with each neuron updated with equal probability.

1. First check to see if all four stored patterns are equilibrium states of the system.
2. Then find all equilibrium states of the system. How many of them are spurious? How do those patterns look like. How many of them can be considered as the reversed states of something else?
3. For each of the equilibrium states find the basin of attraction. That is find the collection of states that converge to that equilibrium state.
4. What is the chance that an input pattern does not associate with any of the stored pattern?

You need to present as much results as deemed reasonable to convince me that you have done a through job. Make comments on your results and observations to show that you really understand what goes on with the Hopfield net in this problem.