Artificial Intelligence
Week-8 Lab Assignment-7

To model the low level image processing tasks in the framework of Markov Random Field and Conditional Random Field. To understand the working of Hopfield network and use it for solving some interesting combinatorial problems

Problem Statement 1:

Many low level vision and image processing problems are posed as minimization of energy function defined over a rectangular grid of pixels. We have seen one such problem, image segmentation, in class. The objective of image denoising is to recover an original image from a given noisy image, sometimes with missing pixels also. MRF models denoising as a probabilistic inference task. Since we are conditioning the original pixel intensities with respect to the observed noisy pixel intensities, it usually is referred to as a conditional Markov random field. Refer to (3) above. It describes the energy function based on data and prior (smoothness). Use quadratic potentials for both singleton and pairwise potentials. Assume that there are no missing pixels. Cameraman is a standard test image for benchmarking denoising algorithms. Add varying amounts of Gaussian noise to the image for testing the MRF based denoising approach. Since the energy function is quadratic, it is possible to find the minima by simple gradient descent. If the image size is small (100x100) you may use any iterative method for solving the system of linear equations that you arrive at by equating the gradient to zero.

Energy Minimization

Many vision task are naturally posed as energy minimization problems on a rectangular grid of pixels

$$E(u) = E\_data(u) + E\_smoothness(u)$$

The goal of energy Minimization is to find a set of coordinates representing the minimum energy conformation for the given structure or we can say for a given grid of pixels

Denoising

Given a noisy image with some pixel value representing in nd array form, where some measurement may be missing, recover the original image I(x,y) which is typically assumed to be smooth.

Markov's Random Field

Given a G(v,e) a graph, where X = set of node's associated with random variable u_i and there would be N_i neighbor node's.

Then markov's random field satisfies the,

"Probability distribution of ith Random Variable given that for all those random variable's j which belongs to the set of node's for given i. is equal to the probability distribution of a ith random variable given that for all random variables which are the neighbor of node i."

Potential

A potential is a non-negative function of variable x. Where a joint potential is non negative func of the set of variables.
We don't have to normalize it because it is a non-negative function.

Markov's Random Field

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{c=1}^{C} \phi_c(\mathcal{X}_c)$$

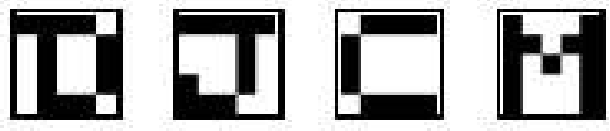
It is defined as the product of potential of the maximal clique(it is the set of random variables taken from set X) with 1/z as normalizer.

https://github.com/mayankmangalmourya/Image-Denoising---Markov-s-Random-Field-

---

Problem Statement 2 :

For the sample code hopfield.m supplied in the lab-work folder, find out the amount of error (in bits) tolerable for each of the stored patterns.



Hopfield networks **are a special kind of recurrent neural networks that can be used as associative memory. Associative memory is memory that is addressed through its contents. That is, if a pattern is presented to an associative memory, it returns whether this pattern coincides with a stored pattern.**

That is if we add a certain amount (tolerable) of error to any of the patterns that is known to the network , then it is able to retrieve the original pattern using the Hopfield Network.

For Example, 1) **2)**

Now Lets understand how it works

Testing Algorithm of Discrete Hopfield Net

Step 0: Initialize the weights to store patterns, i.e., weights obtained from training algorithm using Hebb rule.

Step 1: When the activations of the net are not converged, then perform Steps 2-8.
Step 2: Perform Steps 3-7 for each input vector X.

Step 3: Make the initial activations of the net equal to the external input vector X:

$$y_i = x_i \quad for\, i = 1\, to\, n$$

Step 4: Perform Steps 5-7 for each unit y_i. (Here, the units are updated in random order.)

Step 5: Calculate the net input of the network:

$$y_{ini} = x_i + \sum_j y_j w_{ji}$$

It is a **symmetrically weighted network** i.e., wii = 0 and wij = wji

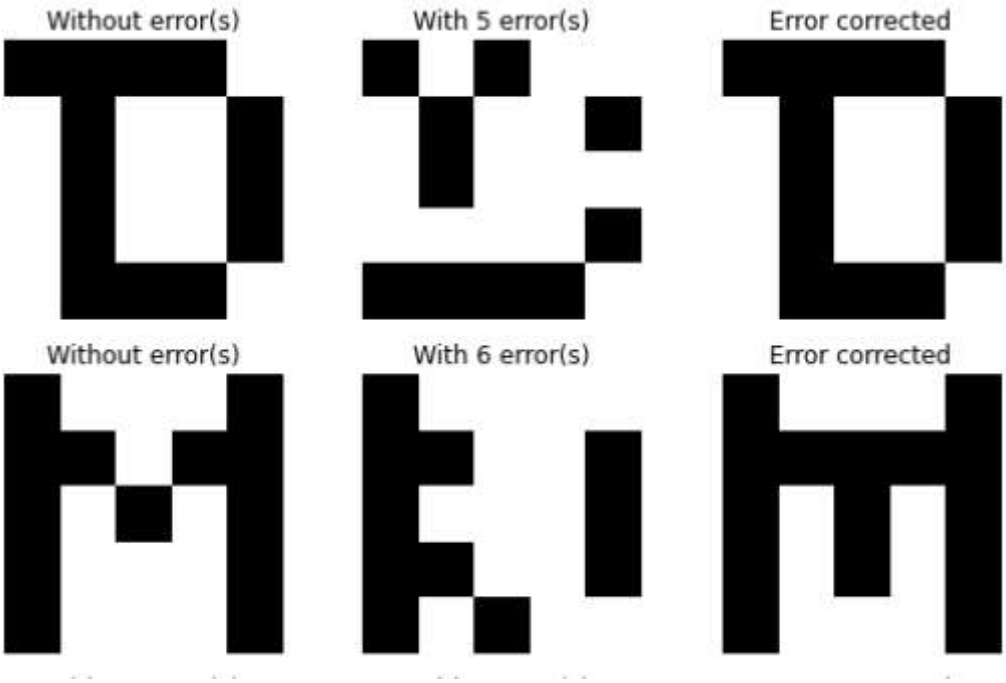
Step 6: Apply the activations over the net input to calculate the output:

$$y_i = \begin{cases} 1 & if\ y_{ini} > \theta_i \\ y_i & if\ y_{ini} = \theta_i \\ 0 & if\ y_{ini} < \theta_i \end{cases}$$

where θ_i is the threshold and is normally taken as zero.

Step 7: Now feed back the obtained output y_i to all other units. Thus, the activation vectors are updated.

Step 8: Finally, test the network for convergence.



Problem Statement 3 :

Solve a TSP (traveling salesman problem) of 10 cities with a Hopfield network. How many weights do you need for the network?

Ans.

Neuron

The architecture of a single neuron has a number I of inputs xi and one output(y). Associated with each input is a weight wi.
There may be an additional parameter w0 of the neuron called a bias which may view as being the weight associated with an input.

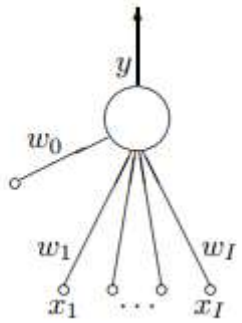


Image of a neuron

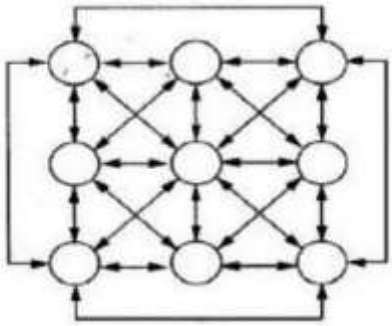
Hopfield Network : It is a network consisting of a set of neurons with output of each neuron as feedback to all other neurons. It is an auto associative fully interconnected single layer feedback layer.

Symmetrically weighted network.

In a Hopfield network, the energy function is used to determine the stability of the network and whether it has converged to a stable state.
The energy function is always minimized when the network reaches a stable state, which is a state where the energy cannot be further reduced.
Input Potential - It is the value that each neuron in the network receives from the input signal.

Solving Traveling Salesman Problem using Hopfield Network

Consider a set of K cities is given, there would be k*k neurons and a matrix of the K(K-1)/2 distances between them as neurons in the hopfield network are symmetrically weighted.



Fully connected Hopfield for 3 cities

The energy function in a HNN network should satisfy the following conditions:

The function should lead to a stable state and it should provide the shortest traveling path.

The weights play two roles, first, they must define an energy function which is minimized only when the state of the network represents a valid tour. Second, the weights must encode the objective function that we want to minimize – the total distance.

Hopfield network approach to solving the TSP involves minimizing an energy function, which includes penalty terms for using short edges and visiting cities more than once. These penalty terms are added to the energy function to encourage the network to explore solutions that do not include short edges and repeated visits to cities, which are generally not optimal solutions to the TSP.

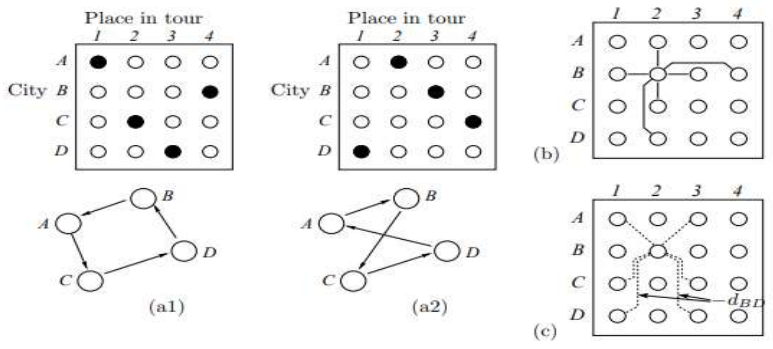


Figure 42.10. Hopfield network for solving a travelling salesman problem with $K = 4$ cities. (a1,2) Two solution states of the 16-neuron network, with activites represented by black = 1, white = 0; and the tours corresponding to these network states. (b) The negative weights between node $B2$ and other nodes; these weights enforce validity of a tour. (c) The negative weights that embody the distance objective function.

GitHub Link :

https://github.com/mayankmangalmourya/Image-Denoising---Markov-s-Random-Field-