Artificial Intelligence Lab Report 8

1st Dipean Dasgupta
202151188
BTech CSE
IIIT,Vadodara

2nd Shobhit Gupta
202151149
BTech CSE
IIIT,Vadodara

3rd Rahul Rathore
202151126
BTech CSE
IIIT,Vadodara

4th Rohan Deshpande
202151133
BTech CSE
IIIT,Vadodara

***Abstract*—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.**

I. Problem Statement

1) 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.
2) Setup the energy function for the Eight-rook problem and solve the same using Hopfield network. Give reasons for choosing specific weights for the network
3) Solve a TSP (traveling salesman problem) of 10 cities with a Hopfield network. How many weights do you need for the network?

II. Explanation

A. Problem 1

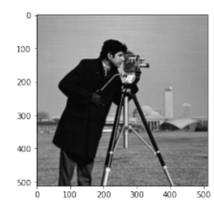
For this task, we converted the hopfield.m MATLAB codes to Python so that we could do it in the same Notebook as the other parts. The network was trained using Hebb's rule, as in the file and then they were noised by changing some random pixel values. At max, 15 pixel values were noised. We can see that suprisingly, the network can correct up to max 8 errors.

B. Problem 2

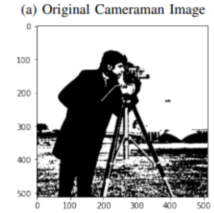
This is a 512x512 grayscale image. We normalize the pixel values to be between 0 and 1, by dividing all values by 255, and then 'binarizing' it for the Markov Random Field by converting all the normalized pixel values below 0.5 to 0 and the rest to 1 Markov random fields sue a quadratic potential function to measure the energy potential of the image when changing a particular pixel, with respect to the neighbouring pixels. The quadratic potential function is given by:

$$E(U) = \Sigma_{n=1}^{N}(u_n - v_n)^2 + \lambda\Sigma_{n=1}^{N-1}(u_{n+1} - u_n)^2 \qquad (1)$$

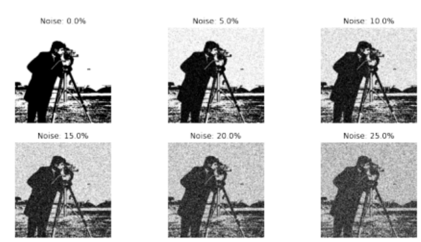
where v is the smooth ID signal, u is the IID and E is the energy function. We ran the algorithm for 5*512*512 = 1310720 iterations.



(a) Original Cameraman Image

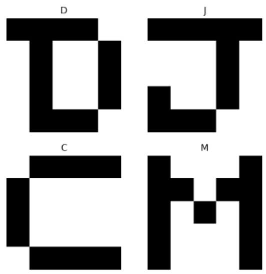


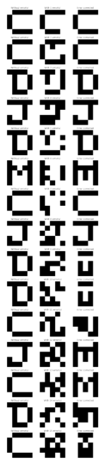
(b) 'Binarized' Cameraman Image



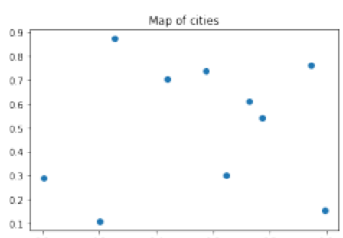
C. Problem 3

This is the usual famous NP-hard problem of Travelling Salesman, that is done using the Hopfield Networks. Since in





a Hopfield Network, each node is connected to the other node, we needed a total of 10x10 = 100 weights. We first generate 10 cities randomly. And then let the hopfield network predict an optimal least path cost.



III. References

1) http://www.cs.utoronto.ca/ strider/Denoise/Benchmark/http://www.cs.utoronto.ca/ strider/Denoise/Benchmark/
2) https://web.cs.hacettepe.edu.tr/ erkut/bil717.s12/w11a-mrf.pdfMRF Image Denoising
3) http://www.inference.phy.cam.ac.uk/mackay/itila/Single Neuron and Hopfield Network: Chapter 40, 41, 42 Information Theory, Inference and Learning Algorithms, David MacKay