Neural Networks (April 2020)
Assignment #2:
Hopfield Neural Network and Kohonen Self-Organizing Map
Due Date: Ordibehesht 25, 1399

PART A:

Datasets: ‘yaleface’

In this part, you will design a parallel Hopfield neural network to perform normal facial image recognition through facial expression images. You are given a dataset (Yale Face Database) which contains 165 grayscale images of 15 different people. There are 11 images per person (subject), one per each facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and winked. Here for this assignment, the first 7 subjects have been chosen.

In this assignment, you will design a simple face recognition system by using the Hopfield networks. This system must be trained by normal facial images. After that, it should have the ability of recognizing different people with different facial expressions. To achieve this purpose:

1. First resize the images from 243x320 to 32x32 pixels, then reshape each image into a single column vector. Obviously, this vector will contain 1024 elements. Next, change the representation of each element of the resulted vector from decimal into 8 bit binary representation. This will cause an eight columned matrix like the following figure (see Fig. 1).

   Do the same process for all 7 training images, then you will have 7 matrices with dimensionality of 1024x8.

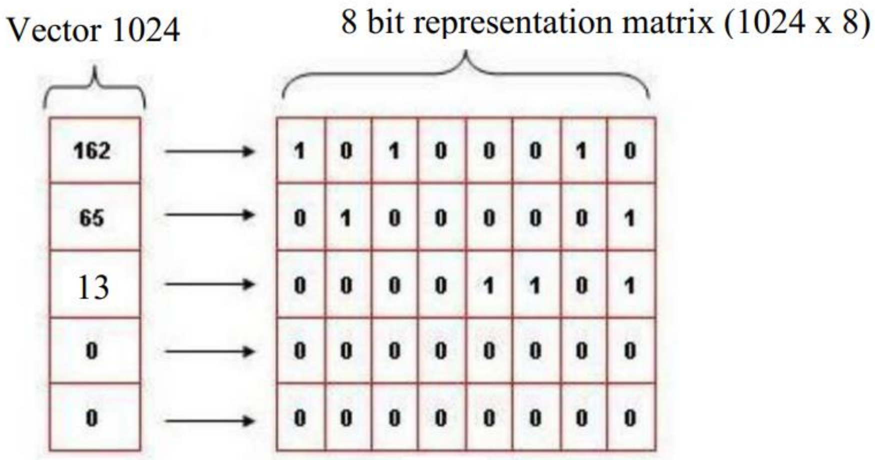


Fig. 1

2. In the second phase, you should train 8 parallel Hopfield network for each column of the resulted images. Each network will have 1024 inputs and 1024 outputs. For training the first network, use the first column of each matrix. So, you will have 7 training vectors for the first network. Then, for training the second network, use the second column of the matrices and so on.
3. In this phase, you should test the trained network. After training the network, perform the first phase of the assignment again and obtain the appropriate 1024x8 matrices for test facial expression images of different people. Then, feed the trained networks with these matrices (feed each network with the proper column of the test matrix) and get the output of the networks for each test sample. After that, convert the binary representation of the resulted image (network output) into one vector of decimal numbers and reshape this vector into a 32x32 image. Then, use the following formula in order to find the difference between retrieved images and various normal facial expressions:

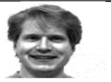
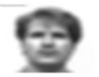
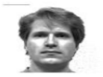
$$dif = \min_{I \in T} \left\| I - \hat{I} \right\|$$

Where $\hat{I}$ is the retrieved image, I is one of the normal facial images and T is the set of all normal facial images. Any I which minimizes the above formula, can express the class of the retrieved image.

Your tasks:

a) Implement the above face recognition system.

b) Apply all test data to the network and report each experiment in a table like this:

Row#	Test image	Retrieved image	Most matched normal image	class
1				1

c) Report the accuracy of networks for each person's data.

d) Report the total accuracy of networks for all test data.

Hint:

These MATLAB functions may be found useful: imread, imwrite, imresize, dec2bin, str2double, num2str, bin2dec. Also, hardlim or hardlims for net output.

PART B:

Dataset: ‘Data.xlsx’

In this part, you will apply a Self-Organizing Map (SOM net) to a dataset. You are given a dataset which contains 1000 samples from 4 classes that are not linearly separable. Split the data into training and test sets. Select 20% of samples for test data. Build a SOM with a 10x10 grid topology to learn the 2D distribution of the data. Use the Euclidean distance to determine the winner neuron in the output (map) layer. You can experiment with different size of the neighborhood and learning rates and determine the ones that work best. You should reduce the size of the neighborhood and the learning rate in consecutive epochs.

1. Start with random weights and generate a 2D scatter plot of the weights along with training data.
2. Then, train the network and generate the scatter plot once again. What is the relationship between the distribution of training examples and the SOM neurons?
3. For each test sample, determine the winner neuron and compute their distance. Sum up this value for all test samples to evaluate your result.

Notes:

- Feel free to use any predefined functions.
- Due Date: Ordibehesht 25, 1399.
- Pay extra attention to the due date.
- Be advised that all submissions after the deadline **would not be graded**.
- Prepare a complete report.
- Email your files as a folder in this format (HW#_student#_name_family.zip).
  Email: hashemi.mahshadd@gmail.com

Kind Regards