ME8873 – Statistical Model Estimation
Homework Set No. 4
Due April 11th, 2022

***Note**:* Datasets for this assignment are posted under the data locker folder on Canvas.

Problem 1 : Error Backpropagation for Neural Networks

An important step in training a radial-basis-function (RBF) network is to determine the center location and the dilation parameter of each radial basis function so that a limited number of RBF functions may effectively approximate a nonlinear map. Shown below is an example of optimal allocation of RBF functions for voice data processing. Twenty RBF functions are placed optimally for covering approximately 300 data points in 2-dimensional input space. The dilation parameter, shown by the radius of each circle, is determined based on the variance of the data classified into the same RBF function.

A similar data set has been uploaded to the course data locker. You are requested to classify these data for the purpose of tuning a RBF network.

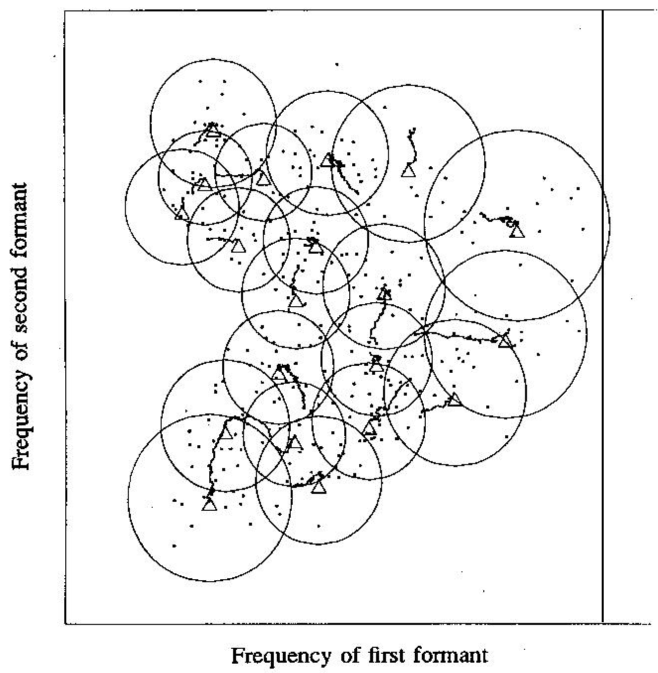

Figure 3 Two-dimensional classification example (vector quantization)
[J. Moody and C. Darken, 1989]

a). Implement the Generalized Lloyd Algorithm discussed in class for classifying *N* points of 2-dimensional input data into *m* clusters, i.e. *m* RBF functions. Download the data (data 3-3 from course locker) and test your program with the data. Set *m* = 9, create an equally-spaced 3-by-3 grid in the 2-dimensional space, and place the center points of the nine RBF functions initially at those grid points. After optimizing the center locations, compute the dilation parameter for each cluster. Plot the results in the same way as the above example.

b). Using the Least Square Estimate algorithm, obtain the scales of the RBF functions to approximate the downloaded training data.

c). To evaluate the validity of the tuned RBF network, another set of data has been uploaded to the data locker. These data are not used for training the RBF network but are used for evaluating the accuracy of the trained network. Using this data file, evaluate the mean squared error of the RBF network.

d). Repeat Parts a) through c) for m = 25. If time permits, try out a much larger number of RBF functions, say m = 100. Discuss pros and cons of using many RBF functions.

Problem 2 : Error Backpropagation for Neural Networks

A multi-layer neural network with two hidden layers is depicted below. All output functions in the hidden layers are logistic functions, and the one in the output layer is a linear function, i.e. $o_5 = g_5(z_5)$. During the training of the network using the <u>*Error Back Propagation Algorithm*</u>, the following data were obtained when one of the sample pairs, input x = 2 and target t = 2, were presented:

$$W_{21} = 2,\ W_{32} = 3,\ W_{42} = -2,$$

$$W_{52} = 1,\ W_{53} = 5,\ W_{54} = 2,$$

$$g_2(z_2) = 0.5,\ g_3(z_3) = 0.2,\ g_4(z_4) = 0.75,$$

Compute the weight change ΔW_{21} to be made for this sample presentation. Exclude momentum terms and use a learning rate of ρ = 0.2. Note that there is a direct connection between units 2 and 5 in the figure. Consider all the connections through which the weight change ΔW_{21} influence the squared error at the output. (***Hint**: Use the approach demonstrated in Lecture 14*)

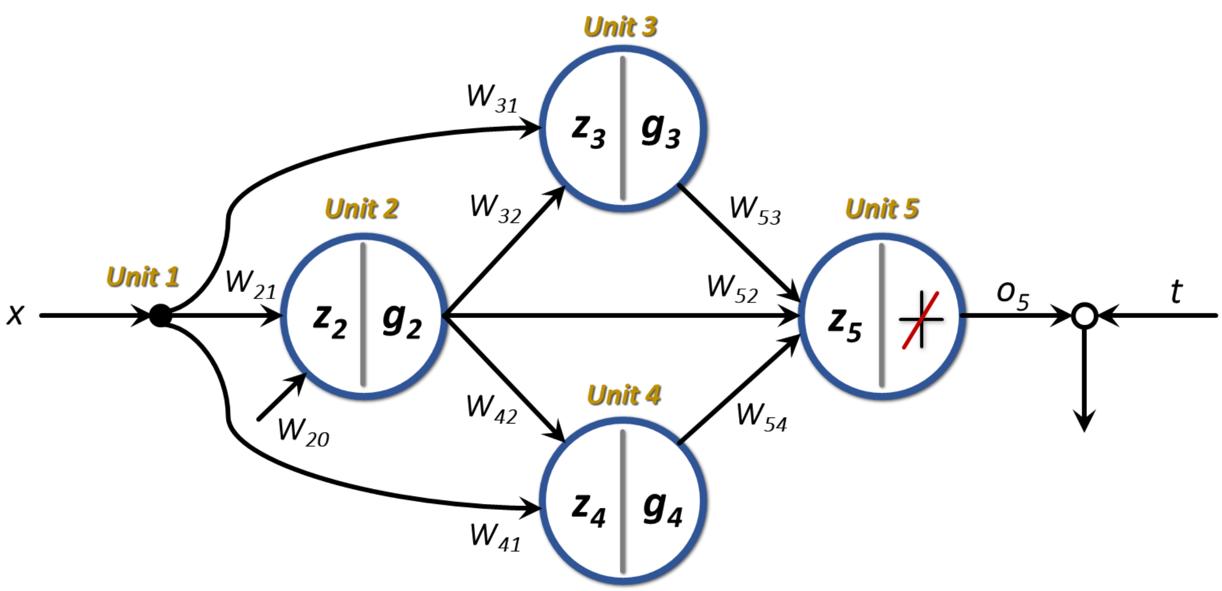

Figure 1: Multi-layer neural network with two hidden layers

Problem 3 : Neural Networks for Classification of Interspersed Data
A robotic manipulator uses sensors in its end effector determine the properties of fruit during a sorting task. Figure 2 shows how some of physical properties including diameter, density, mass, red light absorption, and firmness, present across apples, oranges, pears, and tomatoes. The file "***Data 4-2.csv***" in the Data Locker on Canvas contain four files containing 100 samples of each fruit.

Use the software package of your choice (MATLAB, R, or other) to create a simple feedforward artificial neural network (ANN) capable of accurately classifying the fruit. Choose your own learning rate, and percentage of samples (greater than 50%) for network training. You are free to choose whatever activation function you want, any number of hidden layers, and any number of neurons in the layers.

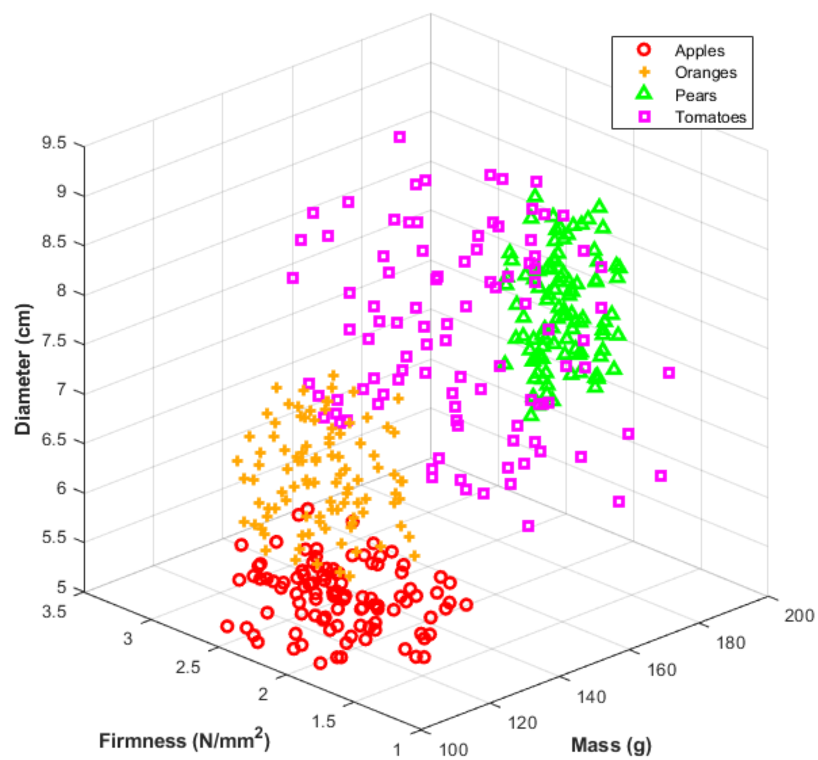

Figure 2: A plot of physical properties for four fruits being sorted by a robotic picker

a) Starting with an epoch size of five and increasing the number of epochs each run (e.g. n_{epoch} = 5, 10, 50, 100, etc.; at least five epoch sizes), train your neural network several times and report the accuracy of fruit classification with increasing epoch number.

1. Define your chosen neural parameters (# of layer, # of neurons, learning rate, and activation function)
2. Plot, in a single figure, the relationship between number of epochs and classification accuracy for **each** fruit. Label your plot axes and include a legend. At what number of epochs does your network's classification accuracy plateau?

b) Create four of your own "mutant fruits" (e.g. an extra dense apple) that lie outside the general property ranges of the four sample sets. How does your network classify these fruits? Note that you are not to add the mutant fruits to the training set. Plot a figure showing how classification accuracies for the mutant fruits change with epoch size. Explain, in a few sentences, why these changes happen (or do not).