EECE5644 Spring 2020 – Take Home Exam 4

Submit: Monday, 2020-April-13 before 11:45ET

Please submit your solutions on Blackboard in a single PDF file that includes all math and numerical results. Only the contents of this PDF will be graded.

For control purposes, please also provide any code you write to solve the questions in one of the following ways:

1. Include all of your code in the PDF as an appendix.
2. In the PDF provide a link to your online code repo where the code resides.

This is a graded assignment and the entirety of your submission must contain only your own work. Do not make your code repo public until after the submission deadline. If people find your code and use it to solve questions, we will consider it cheating for all parties involved. You may benefit from publicly available literature including software (that is not from classmates), as long as these sources are properly acknowledged in your submission.

Copying math or code from each other is clearly not allowed and will be considered as academic dishonesty. Discussing with the instructor and the teaching assistant to get clarification, or to eliminate doubts is acceptable.

By submitting a PDF file in response to this take home assignment you are declaring that the contents of your submission, and the associated code is your own work, except as noted in your acknowledgement citations to resources.

Start early and use the office periods well. The office periods are on gcal and contain video links for remote participation. You may call 1-617-3733021 to reach Deniz if you have difficulty using the video links for the office periods.

It is your responsibility to provide the necessary and sufficient evidence in the form of explanations, math, visualizations, and numerical values to earn scores. Do not provide a minimal set of what you interpret is being precisely requested; you may be providing too little evidence of your understanding. On the other hand, also please do not dump in unnecessarily large amounts of *things*; you may be creating the perception that you do not know what you are doing. Be intentional in what you present and how you present it. You are trying to convince the reader (grader) that your mathematical design, code implementations, and generated results are legitimate and correct/accurate.

Question 1 (30%)

The data generation script for this question is called exam4q1_generateData.m. Generate two-dimensional $\mathbf{x} = [x_1, x_2]^T$ samples with this Matlab script; specifically, 1000 samples for training and 10000 samples for testing.

Train and test a single hidden layer MLP function approximator to estimate the value of X_2 from the value of X_1 by minimizing the mean-squared-error (MSE) on the training set. (The first coordinate of each data vectors will go into the MLP, and the output will try to approximate the second coordinate of the data vectors.)

Use a softplus (SmoothReLu) activation function as the nonlinearity for the perceptrons in the hidden layer. Useing 10-fold cross validation, select the the best number of perceptrons that your training set can justify using. Leave the output layer of the MLP as a linear unit linear (no nonlinearity). Once the best model structure is identified using cross-validation, train the selected model with the entire training set. Apply the trained MLP to the test set. Estimate the test performance.

You may use existing software packages for all aspects of this solution. Make sure to clearly demonstrate that you are using the packages properly.

Hint: $logistic(z) = 1/(1+e^{-z})$ & $softplus(z) = ln(1+e^{z})$

Note: The theoretical minimum-MSE estimator is the conditional expectation of X_2 given X_1, which can be analytically derived using the joint pdf, if it had been known, but in practice we do not know the true pdf, so in this exercise we try to approximate the theoretically optimal solution with a neural network model.

Question 2 (35%)

For this question use the same multiring data generation script from the third assignment. Generate a two-class training set with 1000 and testing set with 10000 samples. The class priors should be equal. Train and evaluate a support vector machine classifier with a Gaussian kernel (radial-basis function (RBF) kernel) on these datasets.

Specifically, us a spherically symmetric Gaussian/RBF kernel. Using 10-fold cross-validation, select the best box constraint hyperparameter C and the Gaussian kernel width parameter σ (notation based on previously covered math in class).

Train a final SVM using the best combination of hyperparameters with the entire training set. Classify the testing dataset samples with this trained SVM to assess performance.

You may use existing software packages for all aspects of this solution. Make sure to clearly demonstrate that you are using the packages properly.

Question 3 (35%)

In this question, you will use GMM-based clustering to segment the color images *3096_color.jpg* and *42049_color.jpg* from the Berkeley Image Segmentation Dataset. We will refer to these images as the airplane and bird images, respectively.

As preprocessing, for each pixel, generate a 5-dimensional feature vector as follows: (1) append row index, column index, red value, green value, blue value for each pixel into a raw feature vector; (2) normalize each feature entry individually to the interval $[0,1]$, so that all of the feature vectors representing every pixel in an image fit into the 5-dimensional unit-hypercube. All segmentation algorithms should operate on these normalized feature vectors.

For each image do the following: (1) Using maximum likelihood parameter estimation, fit a GMM with 2-components, use this GMM to segment the image into two parts; (2) Using 10-fold cross-validation, and maximum everage validation-log-likelihood as the objective, identify the best number of clusters, then fit a new GMM with this best number of components and use this GMM to segment the image into as many parts as there are number of Gaussians.

For GMM-based clustering, use the GMM components as class/cluster-conditional pdfs and assign cluster labels using the MAP-classification rule.