Homework 3 **CSCE 633**
Due: 11.59pm on November 4, 2019

Instructions for homework submission
Please submit on eCampus a single zip file containing your pdf solutions and your code - Named FirstName_LastName_HW3.zip (and similar naming for the files within the zip). a) Submit PDF and .ipynb file (or similar code) of your solution Jupyter Notebook with latex explanations on eCampus, zipped into one file. b) Please write a brief report for the experimental problems. c) Please start early.

Question 1: Midterm Correction
Choose the question part (e.g. 1a or 2c) on the midterm you performed most poorly on. Please type up a solution guide that explains the solution and steps needed to arrive at this solution. Please show your work (or if it is a conceptual question, details on how you analyze the concept and evaluate the importance). Then include a section in which you detail your mistakes and explain your new understanding of the problem. Finally, attach an image of the question you are correcting to show the points taken off and the adjustments made. This question will count for both homework credit and give you the possibility of gaining up to 10 points back on the midterm question.
If you do not have a question that you lost 10 points on, you will receive full credit for the question and the remaining points will be considered extra credit on top of your overall exam score.

Question 2: Gradient Descent Boosting: Hastie et al. algorithm 10.4
(a) Please implement algorithm 10.4 for multi-class gradient boosting. You must manually implement this. If you use other sources for coding help you must cite them. (Please note - if you look up implementations for this and do not appropriately cite them - this is considered cheating).
(b) Comparison of your implementation with a popular package (XGBoost, GBM, or Light GBM - your choice): Using the Heart dataset - develop a multi-class classification for Heart Disease (AHD) and Thal crossed together (this requires you to create labels based upon the combination of Yes/NO and Fixed, Normal, Reversable for the final two columns). Investigate the number of iterations needed to optimize the testing error, then make the testing error start to rise for your implementation and the package you choose to compare against. For multiclass classification - please determine what metric you will be optimizing for comparison (note that AUROC only works in binary classification).
(c) Please provide your feature importance, and compare to the package choice's feature importance. Describe similarity vs. differences and explain (in particular with relation to hyperparameters).

Question 3: Activation Function in Neural Networks
Consider a three-layer fully-connected network with D input features, one hidden layer with M nodes, and one output layer. The activation function of each node is a sigmoid function of the form

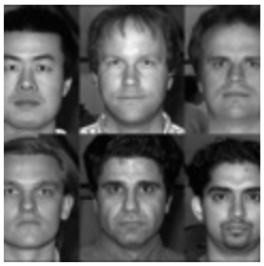
$$\text{sigmoid}(\alpha) = \frac{1}{1 + \exp(-\alpha)}$$

(a) Provide a schematic representation of the network. Define the variables for the input, the output, and the weights.
(b) Express the output as a function of the input.
(c) Calculate the number of parameters that need to be inferred.
(d) Show that there exists an equivalent network with hidden unit activation functions given by the hyperbolic tangent, which computes exactly the same function, where the hyperbolic tangent is given by

$$\tanh(\alpha) = \frac{\exp(\alpha) - \exp(-\alpha)}{\exp(\alpha) + \exp(-\alpha)}$$

Hint: First find the relation between $\text{sigmoid}(\alpha)$ and $\tanh(\alpha)$, then show that the parameters of the two networks differ by linear transformations.

Question 4: Image processing for human faces
In this problem, we will process face images coming from the Yale Face Dataset: `http://vision.ucsd.edu/content/yale-face-database`. This dataset contains images of the faces of 15 individuals. For each individual there are 11 images taken under a variety of conditions e.g., the person makes a happy expression, wears glasses etc.



(a) Download the dataset from the above URL. *Implement* Principal Component Analysis (PCA) on the input images. Assume that the input vector of PCA contains all rows of an image stacked one on top of the other. You can use available libraries that calculate eigenvalues and eigenvectors of a matrix. *Hint:* Don't forget to normalize the data.

(b) Plot a curve displaying the first k eigenvalues $\lambda_1, \ldots, \lambda_K$, i.e. the energy of the first K principal components. How many components do we need to capture 50% of the energy?

(c) Plot the top 10 *eigenfaces*, i.e. the eigenvectors $\mathbf{u_k}$, $k = 1, \ldots, 10$ obtained by PCA.

(d) Select a couple of images from the data. Use the first k eigenfaces as a basis to reconstruct the images. Visualize the reconstructed images using 1, 10, 20, 30, 40, 50 components. How many components do we need to achieve a visually good result?
Hint: Reconstruction of an input vector $\mathbf{x}$ based on the eigenvectors $\mathbf{u_1}, \ldots, \mathbf{u_K}$ is given by the following expression $\mathbf{x} \approx \mathbf{x_0} + \sum_{k=1}^{K} c_k \mathbf{u_k}$, where $c_k = \mathbf{u_k}^T \mathbf{x}$ is the projection of the input image to the k^{th} eigenvector.

(e) *Perform face recognition:* Split the input data into training and testing *making sure that every person is included in each set.* Use as input features the transformed feature space that resulted from PCA. Experiment with different number of PCA components through a 5-fold cross-validation. User an outer 5-fold cross-validation to build predictors using a boosting method (e.g. adaboost, xgboost), support vector machines (report kernel you used), and CNNs

(experiment with filter size, stride size, activation function, drop out). Report the recognition accuracy on the test set.

(f) *Perform type recognition with Gaussian Mixture Models:* Setting the number of components to be equal to the number of poses each subject can have, and using the data obtained by PCA, see if a Gaussian Mixture Model, trained with Expectation Maximization, naturally clusters the pictures by pose type.