Homework 3 **CSCE 633**
Due: 12.59pm on October 21, 2020

Instructions for homework submission
a) Please write a brief report and **include your code right after each answer**.
b) For each answer, please explain your thought process, results, and observations. Please do not just include your code without justification.
c) Create a **single pdf** and submit it on **eCampus**. Please do not submit .zip files or colab notebooks.
d) **This homework is a long one, therefore please start early :)**
e) The maximum grade for this homework, excluding bonus questions, is **10 points** (out of 100 total for the class). There are **2 bonus points**.

Question 1: Maximum likelihood estimate

(a) (1 point) Normal distribution: Suppose that data $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ provided in file *Q1_data.csv* in the **Google Drive** (under Homework3 folder) is drawn from the normal distribution $N(\mu, \sigma^2)$, where μ and σ^2 are unknown.

(a.i) (0.8 points) Show that the maximum likelihood estimate of parameters μ and σ is $\hat{\mu} = \frac{\sum_{n=1}^{N} x_n}{N}$ and $\hat{\sigma}^2 = \frac{\sum_{n=1}^{N} (x_n - \hat{\mu})^2}{N}$.
Hint: Compute the log-likelihood of the data and find its first order derivative with respect to μ and σ. You can assume that $\hat{\mu}$ is known when computing $\hat{\sigma}$.

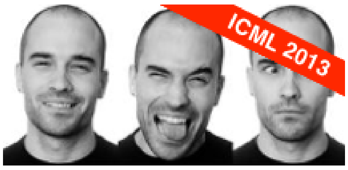
(a.ii) (0.2 points) Using the data provided in *Q1_data.csv*, provide an estimate of the mean μ and variance σ^2 based on which the data were generated using the above formula.

(b) (1 point) Multinomial distribution: Suppose the a gene manifests through three genotypes $\{G_1, G_2, G_3\}$ with probabilities $\{(1-\phi)^2, \phi^2, 2\phi(1-\phi)\}$. After testing a random sample of people, we find that N_1 individuals have genotype G_1, N_2 individuals have G_2, and N_3 individuals have G_3. Compute the maximum likelihood estimate of ϕ, assuming that N_1, N_2, and N_3 are known.
Hint: You are given three independent outcomes $\{G_1, G_2, G_3\}$, whose probabilities sum to one, therefore you can assume that they follow a multinomial distribution with corresponding probabilities $\{(1-\phi)^2, \phi^2, 2\phi(1-\phi)\}$.

Question 2: Machine learning for facial recognition

In this problem, we will process face images coming from the Facial Expression Recognition Challenge (presented in the International Conference of Machine Learning in 2013). The data is uploaded under *Homework3* folder in the shared **Google Drive**. You are given three sets of data: training set (i.e., *Q2_Train_Data.csv*), testing set (i.e., *Q2_Test_Data.csv*), and validation set (i.e., *Q2_Validation_Data.csv*).

The data consists of 48×48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in seven categories. More information on the data can also be found in this link.



All three files contain two columns:

- The column labeled as "emotion" contains the emotion class with numeric code ranging from 0 to 6 (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

- The column labeled as "pixels" contains the 2304 (i.e., 48×48) space-separated pixel values of the image in row-wise order, i.e., the first 48 numbers correspond to the first row of the image, the next 48 numbers to the second row of the image, etc.

(a) (0.5 points) Visualization: Randomly select and visualize 1-2 images per emotion.

(b) (0.5 points) Data exploration: Count the number of samples per emotion in the training data.

(c) (3 points) Image classification with FNNs: In this part, you will use a feedforward neural network (FNN) (also called "multilayer perceptron") to perform the emotion classification task. The input of the FNN comprises of all the pixels of the image.

(c.i) (2 points) Experiment on the validation set with different FNN hyper-parameters, e.g. # layers, #nodes per layer, activation function, dropout, weight regularization, etc. For each hyper-parameter combination that you have used, please report the following: (1) emotion classification accuracy on the training and validation sets; (2) running time for training the FNN; (3) # parameters for each FNN. For 2-3 hyper-parameter combinations, please also plot the cross-entropy loss over the number of iterations during training.
Note: If running the FNN takes a long time, you can subsample the input images to a smaller size (e.g., 24×24).

(c.ii) (1 point) Run the best model that was found based on the validation set from question (**c.i**) on the testing set. Report the emotion classification accuracy on the testing set.

(d) (3 points) Image classification with CNNs: In this part, you will use a convolutional neural network (CNN) to perform the emotion classification task.

(d.i) (2 points) Experiment on the validation set with different CNN hyper-parameters, e.g. # layers, filter size, stride size, activation function, dropout, weight regularization, etc. For each hyper-parameter combination that you have used, please report the following: (1) emotion classification accuracy on the training and validation sets; (2) running time for training the FNN; (3) # parameters for each CNN. How do these metrics compare to the FNN?

(d.ii) (1 point) Run the best model that was found based on the validation set from question (**d.i**) on the testing set. Report the emotion classification accuracy on the testing set. How does this metric compare to the FNN?

(e) (1 point) Bayesian optimization for hyper-parameter tuning: Instead of performing grid or random search to tune the hyper-parameters of the CNN, we can also try a model-based method for finding the optimal hyper-parameters through Bayesian optimization. This method performs a more intelligent search on the hyper-parameter space in order to estimate the best

set of hyper-parameters for the data. Use publicly available libraries (e.g., hyperopt in Python) to perform a Bayesian optimization on the hyper-parameter space using the validation set. Report the emotion classification accuracy on the testing set.
Hint: Check this and this source.

(f) (Bonus - 1 point) Fine-tuning: Use a pre-trained CNN (e.g., the pre-trained example of the MNIST dataset that we saw in class) and fine-tune it on the FER data. Please experiment with different fine-tuning hyper-parameters (e.g., #layers to fine-tune, regularization during fine-tuning) on the validation set. Report the classification accuracy for all hyper-parameter combinations on the validation set. Also report the classification accuracy with the best hyper-parameter combination on the testing set.

(g) (Bonus - 1 point) Feature design: In this part, you can try to extract image features rather than learning them from the FNN or CNN models. For example, you could try Histogram of Oriented Gradient (HOG) features or Gabor filterbanks. These features can be used as the input of a FNN which will take the emotion-specific decision.
Hint: Check this and this source.