CS 509 - Pattern recognition

Assignment 4

Problem 1: Non-Parametric Methods

You are given a dataset $D = \{0,1,1,1,2,2,2,2,3,4,4,4,5\}$. Using techniques from parametric and non-parametric density estimation, answer the following questions:

1. Draw a histogram of D with a bin-width of 1and bins centered at $\{0,1,2,3,4,5\}$.
2. Write the formula for the kernel density estimate given an arbitrary kernel K.
3. In terms of their respective algorithms and their asymptotic performance, compare the Parzen window method and the $k - NN$ method of non-parametric density estimation.
4. Select a triangle kernel as your window function:

   $$K(u) = (1 - |u|)\delta(|u| \leq 1)$$

   Where u is a function of the distance of sample x_i to the value in question x divided by the bandwidth: $u = \frac{x - x_i}{h}$. Compute the kernel density estimates for the following values of $x = \{0,1,2,3,4,5\}$ bandwidths of 2.
5. Now, what if you assume that, rather, the density is a parametric density: it is a Gaussian. Compute the maximum likelihood estimate of the Gaussian's parameters.
6. Compare the histogram, the triangle-kernel density estimate, and the maximum-likelihood estimated Gaussian. Which best captures the data? What does each miss? Why would you choose one of another if you were forced to?

Problem 2: Neural networks

Consider a regression problem involving multiple target variables in which it is assumed that the distribution of the targets, conditioned on the input vector x, is a Gaussian of the form

$$p(t|x, w) = \mathcal{N}(t|y(x, w), \Sigma)$$

where $y(x, w)$ is the output of a neural network with input vector x and weight vector w, and Σ is the covariance of the assumed Gaussian noise on the targets. Given a set of independent observations of x and t, write down the error function that must be minimized in order to find the maximum likelihood solution for w, if we assume that Σ is fixed and known. Now assume that Σ is also to be determined from the data and write down an expression for the maximum likelihood solution for Σ.

Problem 3: Programming exercise: Dimensionality Reduction

The aim of this problem is to test three dimensionality reduction methods by using a real dataset. Iris dataset also known as Fisher's Iris is a multivariate dataset presented in 1936 by Ronald Fisher in his paper entitled by "The use of multiple measurements in taxonomic problems as an application example of linear discriminant analysis". The dataset includes 50 samples from each of the three iris species (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and width of the sepals and petals, in centimeters. Your task is to write a Python program to illustrate in 2D and in 3D the features' reduction of Iris dataset with the following three dimensionality reduction methods:

1. Principal Component Analysis,
2. Isometric Mapping,
3. Locally linear embedding,

Which of the three features' reduction methods is relevant for the Iris dataset? Justify your answer.

Hint: I provided you the implementation of the three reduction methods in Python. You only need to test them with the Iris dataset.

Problem 4: Programming exercise: Neural Network

In this problem, you have to write a Python program that can be reused for optimizing a neural network and performing hyperparameter tuning to obtain a high-performing model. In fact, the choice of the architecture of neural network and the parameters to use is a challenging task, because it depends on the dataset. The principle of hyperparameter tuning is to find the best parameters to use in a neural network to increase accuracy and to decrease misclassification. In fact, these parameters are not chosen manually, they are chosen with the concept of grid search to try out several values for our hyperparameters and compare the results. You may need the usage of GPU (Graphic Process Unit) to accelerate the computation of the hyperparameters. For that purpose, I recommend you run this program on Google Collaboratory instead of your personal laptop. To do so, sign in your Gmail account and click on the right as in Figure 1 and click on Driver. A new tab

window will get opened, click on **New → More → Google Colaboratory**. In the new opened tab, click on File → Upload Notebook and upload the notebook.

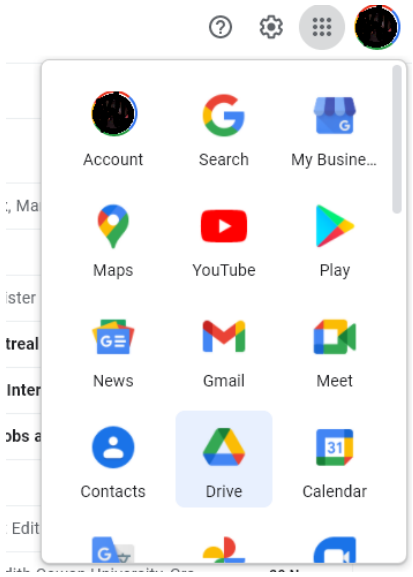

Figure 1

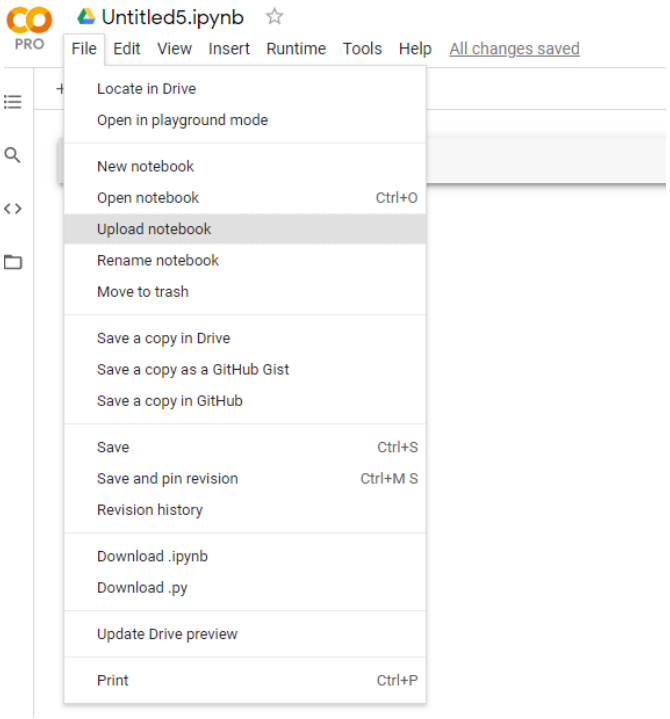

Figure 2

To use the GPU in Google Colaboratory, click on **Runtime → Change Runtime** and select GPU in Hardware Accelerator. I provided you the whole code, you need simply to run it and to illustrate in a table the best parameters that maximize the classification scores.

Submission

Please submit only one pdf file.