Course: Numerical Analysis for Machine Learning

Prof. E. Miglio - July 8th 2021

Duration of the exam: 2.5 hours.

Exercise 1

Let us consider a database containing the relative consumption of certain food items in European countries. The numbers represent the percentage of the population consuming that food type.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('https://openmv.net/file/food-consumption.csv').fillna(0)
data
data.describe()
A = data.to_numpy()[:,1:].T.astype(np.float64)
countries = data.to_numpy()[:,0]
foods = np.array(data.columns[1:]
```

Each column of the matrix A refers to a country, and each row represents a specific food. The arrays countries and foods contain the list of countries and foods, respectively.

- How many samples and features are there in the dataset?
- Perform PCA on the dataset by means of the SVD decomposition. Then, plot the trend of
  - the singular values σ_k;
  - the cumulate fraction of singular values $\frac{\sum_{i=1}^{k}\sigma_i}{\sum_{i=1}^{q}\sigma_i}$;
  - the fraction of the "explained variance" $\frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{q}\sigma_i^2}$;
- Print the first two principal directions. Which are the more important foods for these two directions? Knowing that mediterranean cuisine envisages a large use of garlic and olive oil, compared to the European average, do you expect a positive or negative 1st principal component for Italy?
- Print the first two principal components. Based on these results:
  - In which European region do you expect a large consume of crisp bread? Why? Is this confirmed by the original data?
  - Which countries have food habits that are similar to Italy?
  - Based on the two first principal components, what do you expect about the food habits in England? Is this confirmed by the original data?

Exercise 2 Consider the data provided in file `ex2.zip`. Each row of the matrix X corresponds to one training example ($\mathbf{x}^{(i)} \in \mathbb{R}^2$) and the corresponding row of the vector $\mathbf{y}$ is the label. The dataset contains $m = 2000$ samples.

```
import numpy as np
X = np.genfromtxt('x.txt')
y = np.genfromtxt('y.txt')
```

- Consider the logistic regression model

$$h_{\mathbf{w},b}(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x}-b)}.$$

Implement the stochastic gradient descent method for the logistic regression. Leave the data in the original order. Start with $\mathbf{w} = \mathbf{0}, b = 0$, consider a learning rate equal to 0.01 and perform 5 epochs (*i.e* loops through your entire training set 5 times and hence, since you have 2,000 training examples, this corresponds to 10,000 iterations of SGD). Run the stochastic gradient descent and plot the parameter b as a function of the number of iterations taken. Do you see this plot having a "wavy" appearance? What do you think causes this?

- Repeat the previous problem, but now shuffle the training set first.

```
p = np.random.permutation(2000)
Xp = X[p,:]
yp = y[p,:]
```

Re-run stochastic gradient descent using the shuffled data (again start with $\mathbf{w} = \mathbf{0}, b = 0$, consider a learning rate equal to 0.01 and perform 5 epochs) and replot b as a function of the number of iterations. How is this plot different from the previous one?

- Report the obtained values for $\mathbf{w}$ and b.

Exercise 3 Consider the following mapping:

$$\begin{array}{llll} (0,0,0) \to 1, & (1,0,0) \to 0, & (0,1,0) \to 0, & (0,0,1) \to 0, \\ (0,1,1) \to 1, & (1,1,0) \to 0, & (1,0,1) \to 0, & (1,1,1) \to 1. \end{array} \tag{1}$$

- Is it possible to learn the previous map using only a single perceptron ?
- Propose a multi-perceptron neural network that is able to learn the previous mapping. Compute the weights and the biases of all the perceptrons in the network.