Course: Numerical Analysis for Machine Learning
Prof. E. Miglio - January 16th 2025
Duration of the exam: 2.5 hours.

Exercise 1

Load the dataset contained in the file `faces.mat` using the following commands:

```
import scipy.io as sio
data = sio.loadmat('faces.mat')
X = data['X']
```

The dataset contains a collection of 5000 32×32 grayscale face images. You can plot a single face as follows:

```
import numpy as np
import matplotlib.pyplot as plt
x0 = np.transpose(np.reshape(X[0,:],(32,32)))
plt.imshow(x0,cmap='gray')
```

1. Compute the normalized matrix $\tilde{X}$.
2. Perform the PCA on $\tilde{X}$ and plot the first 25 eigenfaces.
3. Reduce the dimension of the sample from 1024 (32 by 32) to 100 by projecting the matrix $\tilde{X}$ onto U.
4. Plot the original images (pick the first 100 images) and the ones reconstructed from only the first 100 principal components. Plot also the error.

Exercise 2

Consider the following simple linear network

$$a(\mathbf{v}, \mathbf{w}) = \mathbf{w} \cdot \mathbf{v}, \tag{1}$$

where $\mathbf{w} = [w, b]^T$ is the parameter vector (w is the weight and b the bias) and $\mathbf{v} = [x, 1]^T$ is the input vector.

Consider the samples $(x_1, y_1) = (2, 0.5)$ and $(x_2, y_2) = (-1, 0)$ and the cost function

$$J(\mathbf{w}) = (y_1 - a(\mathbf{v}_1, \mathbf{w}))^2 + (y_2 - a(\mathbf{v}_2, \mathbf{w}))^2, \tag{2}$$

where $\mathbf{v}_1 = [x_1, 1]^T$ and $\mathbf{v}_2 = [x_2, 1]^T$.

1. Rewrite equation (2) as $J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T A \mathbf{w} + \mathbf{d}^T \mathbf{w} + c$ (write explicitly A, $\mathbf{d}$ and c.).
2. Compute the value of the exact parameter vector $\mathbf{w}^*$ that minimize $J(\mathbf{w})$.
3. Plot the surface that represents J.
4. Implement the gradient descent method and use it to compute $\mathbf{w}^*$: set the initial guess $\mathbf{w}^{(0)}$ equal to $[1, 1]^T$ and the learning rate η equal to 0.05.
5. What is the maximum value of the learning rate that can be used ?

Exercise 3

Consider a logistic regression

$$\sigma(\boldsymbol{\beta}^T \mathbf{x}) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2) \tag{3}$$

where $\sigma(c) = \dfrac{1}{1 + \exp(-c)}$.

Let us consider the following sets of data:

	Set 1			Set 2			Set 3		
Point ID	x_1	x_2	Label	x_1	x_2	Label	x_1	x_2	Label
1	0	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	1	0	0
3	0	-1	0	-1	0	0	0	1	0
4	-1	0	1	1	0	1	-1	0	0
5	0	1	1	0	-1	1	0	-1	0

1. Plot in 3 pictures the data contained in the 3 datasets (use different colors or symbols for the two classes). What is the main difference between set 1, set 2 and set 3 ?

2. Compute the vectors $\boldsymbol{\beta}$ that allow to use (3) to classify the data contained in set 1 and set 2 assuming a threshold $\epsilon = 0.5$ for the positive class. Is the solution unique ? Motivate your answer.

3. Consider the following alternative system of coordinate (ξ_1, ξ_2) to define the data contained in set 3

	Set 3		
Point ID	ξ_1	ξ_2	Label
1	0	0	1
2	0	1	0
3	0	1	0
4	1	0	0
5	1	0	0

   Explain how we can use ξ_1 and ξ_2 to classify the data contained in set 3.

4. Propose a neural network to determine the parameters to be used to classify set 3.