Course: Numerical Analysis for Machine Learning
Prof. E. Miglio - September 1st 2022
Duration of the exam: 2.5 hours.

Exercise 1 Load the image of a dog using the following commands:

```
from matplotlib.image import imread
import matplotlib.pyplot as plt
import numpy as np
import os
plt.rcParams['figure.figsize'] = [16, 8]

A = imread(os.path.join('.','dog.jpg'))
X = np.mean(A, -1); # Convert RGB to grayscale

img = plt.imshow(X)
img.set_cmap('gray')
plt.axis('off')
plt.show()
```

1. Compute the economy SVD;
2. Let $\mathbf{X}$ the matrix representing the true image and $\tilde{\mathbf{X}}$ the approsimation of rank r obtained using the SVD. Compute and plot the relative reconstruction error of the truncated SVD in the Frobenius norm as a function of the rank r. The expression of the relative reconstruction error is

$$\frac{\|\mathbf{X} - \tilde{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F}. \tag{1}$$

   Remember that the Frobenius norm is given by $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |X_{ij}|^2}$.
3. Square this error (and plot it) to compute the fraction of the missing variance as a function of r;
4. Find the rank r_v where the reconstruction captures 99% of the total variance;
5. Compare r_v with the rank r_F where the reconstruction captures 99% in the Frobenius norm and with the rank r_c that captures 99% of the cumulative sum of singular values.

Exercise 2
Consider the function

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2. \tag{2}$$

1. Compute the gradient ∇f and the Hessian $D^2 f$ of (2). Prove that the function has a unique minimizer x^* and find the minimizer.
2. Implement the Gradient Descent algorithm with constant stepsize to approximate the minimizer. The code must take as input the following parameters

```
max_iter = 50000                   # maximum number of iterations
tol      = 1e-8                    # tolerance for the stopping criterium
x_0      = np.array([-1.2,1.2])    # initial guess
stepsize = 0.001                   # constant stepsize
func     = see equation (2)        # the expression of the function to be minimized
dfunc    = gradient of equation (2) # the expression of the gradient of the function to be minimized
```

Use the following expression for the stopping criterium

$$\epsilon^{(k)} = |f(x^{(k)}, y^{(k)}) - f(x^{(k-1)}, y^{(k-1)})| \tag{3}$$

3. Consider the following matrix

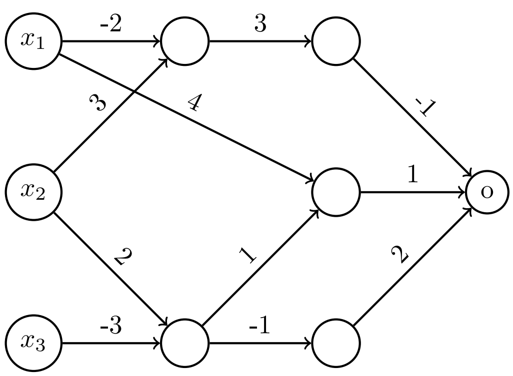
$$H = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix}. \tag{4}$$

Compute the eigenvalues of H and decide if it is positive-definite or not. Towards the end of the algorithm (when you are close to the minimum) replace the descent direction $d = -\nabla f(x)$ with $d = -H^{-1}\nabla f(x)$. Is this still a descent direction ? What behaviour do you observe ? Can you explain this behaviour ?

Hint: in order to check if you are close to the minimun you have to introduce a second tolerance to be used used either on the value of $\epsilon^{(k)}$ or on the value of $f(x^{(k)}, y^{(k)})$.

Exercise 3

Consider the following network where on each edge (i, j) the value of $\dfrac{\partial y(j)}{\partial y(i)}$ is given; $y(k)$ denotes the activation of node k.



The output o is equal to 0.1 and the loss function is $L = -log(o)$. Compute the value of $\dfrac{\partial L}{\partial x_i}$ for each input x_i using the backpropagation method.