Course: Numerical Analysis for Machine Learning
Prof. E. Miglio - January 20th 2023
Duration of the exam: 2.5 hours.

Exercise 1

Consider the following dataset (Petricoin et al., Lancet, 2002).

Each column of A represents measurements taken from a patient. There are 216 columns representing 216 patients, out of which 121 have ovarian cancer. Each row represents the concentration of a different protein in the genomic signature of the patient.

You can download the dataset using the following commands in Colab:

```
!wget https://www.dropbox.com/s/bal044f83ezxiia/ovariancancer_obs.csv
!wget https://www.dropbox.com/s/r52sn5gmkd3y797/ovariancancer_grp.csv
```

Then you can create the matrix as follows:

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

ovariancancer_obs_path = './ovariancancer_obs.csv'
ovariancancer_grp_path = './ovariancancer_grp.csv'

A = np.genfromtxt(ovariancancer_obs_path, delimiter=',').transpose()
f = open(ovariancancer_grp_path)
grp = np.array(f.read().split("\n"))
grp = grp[grp != '']

n_features = A.shape[0]
n_patients = A.shape[1]
print('%d patients' % n_patients)
print('%d features' % n_features)
```

1. Implement the randomized SVD.
2. Perform PCA on the data, by using the exact SVD algorithm. Then, plot the trend w.r.t. i of the singular values σ_i. Now, repeat the computation by considering the randomized SVD algorithm of rank $k = 1, 5, 10, 50$ and 100. Plot the approximate singular values together with the exact singular values. Comment what you see.
3. Use PCA (with exact SVD) to perform dimensionality reduction on the dataset for rank $k = 1$, 5, 10, 50 and 100. Compute the reconstruction error of the dataset as a function of k. Repeat the same exercise by using the randomized SVD algorithm. Repeat the same exercise by using the randomized SVD algorithm with a +50% oversampling, that is by increasing by 50% the number of columns in the matrix random P (round the quantity $1.5\,k$ to the closest integer). Finally, plot the trend of the reconstruction error of the dataset as a function of k in the three cases. Comment on the results.
4. Make a scatterplot of the first two principal components of the patients (obtained with exact SVD), by plotting in red patients in the group ”cancer” and in blue the patients in the remaining group. Repeat the same exercize with a rank $k = 2$ randomized SVD. Compute the execution time needed to run the SVD algorithm in the two cases. Comment on the results.

Exercise 2

Consider the subset of the Quantum Physics Dataset provided by Cornell University. The goal is to learn a classification rule that differentiates between two types of particles generated in high energy collider experiments.

Load the data as follows:

```
X = np.genfromtxt('X.dat')     # Features
y = np.genfromtxt('y.dat')      # Labels (+1, -1)
```

Classify the data by solving the following minimization problem

$$\min_{\mathbf{w}} J(\mathbf{w}) = \frac{1}{N}\sum_i J_i(\mathbf{w}), \tag{1}$$

where

$$J_i(\mathbf{w}) = L(\mathbf{w}^T\mathbf{x}_i, y_i), \tag{2}$$

with $L(s, y) = \log(1 + \exp(-sy))$.

1. Implement the Stochastic Gradient Descent algorithm (SGD) (batch size = 1) with constant learning rate equal to 0.5. Perform 5000 iterations and initialize the weight vector to zero.
2. Repeat the minization using the SGD with the following update for the learning rate

$$\eta_k = \frac{\eta_0}{1 + k/100}, \tag{3}$$

where k is the iteration number and $\eta_0 = 0.05$..

Comment the results.

Exercise 3

Show that a multi-layer neural network with linear activation function $s(x) = x$ is equivalent to a single layer linear network. Assume that in each layer the inputs follow a Normal distribution with mean zero and small variance, *i.e.* $\sigma \ll 1$. For which of the activation functions $s(x) = 1/(1 + \exp(-x))$, $s(x) = \tanh(x)$, $s(x) = \text{relu}(x)$ and $s(x) = \text{selu}(x)$ is a deep network equivalent to a linear network for the given distribution ? The selu function is given by:

$$\text{selu}(x) = \begin{cases} \lambda x \text{ if } x > 0 \\ \alpha(\exp(x) - 1) \text{ otherwise} \end{cases}, \tag{4}$$

where $\lambda \approx 1.0507$ and $\alpha \approx 1.75814$. (*Hint:* consider the case $\sigma \to 0$ using a Taylor expansion around 0.)

• consider for simplicity 1 input layer, 1 hidden layer and 1 output layer x z y

$$\left.\begin{aligned} y &= s(W_1 z + b_1) = W_1 z + b_1 \\ z &= s(W_2 x + b_2) = W_2 x + b_2 \end{aligned}\right\} \text{since } s(u)=u \text{ : linear activation function}$$

$$\Rightarrow y = W_1(W_2 x + b_2) + b_1 = W_1 W_2 x + W_1 b_2 + b_1 = Wx + b \quad \begin{cases} W = W_1 W_2 \\ b = W_1 b_2 + b_1 \end{cases}$$

• assume now the input $\sim N(0, \sigma^2)$ with $\sigma \ll 1$ meaning that $x \in [-\sigma, \sigma]$ very close to zero $\longrightarrow$ Taylor expand in 0 the activation functions

1) $s(x) = \frac{1}{1+e^{-x}} \sim \frac{1}{1+1-x} = \frac{1}{2-x}$ No ✗

2) $s(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1+x-(1-x)}{1+x+(1-x)} = x$ ok ✓

3) $s(x) = \text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$ ok only for positive values of x...

4) $s(x) = \text{SeLU}(x) = \begin{cases} \lambda x & \text{if } x \geq 0 \\ \alpha(e^x - 1) & \text{otherwise} \longrightarrow \alpha x \end{cases}$ ok ✓