Session 4

In this session, we will be using one of the classification tasks found in OpenML as the basis for a mock-up exam. More precisely, the task *semeion* is selected. This is another handwritten digit dataset. Specifically, it is composed by 1593 handwritten digits from around 80 different people. Each digit is represented as a binary image of 16x16 pixels (256 values).

You may need to run this code if this is the first time you are running this notebook.

In [ ]:

```
!pip install scikit-learn
```

Before starting the mock-up exam, you must download the data (**semeion_X.npy** and **semeion_y.npy**) and the logistic regression library (**Logistic_Regression.py**) from poliFormat.

In [ ]:

```
# Execute this cell only when running in Google Colab

# You need to upload LogisticRegression.py
from google.colab import files
uploaded = files.upload()

# You need to upload semeion_X.npy
from google.colab import files
uploaded = files.upload()

# You need to upload semeion_y.npy
from google.colab import files
uploaded = files.upload()
```

Below you can find a baseline result achieved with the logistic regression classifier using default parameters with batch size 10, and devoting 80% of the samples to training and 20% to test (with seed random_state=23).

In [2]:

```
from LogisticRegression import LogisticRegressionClassification, LogisticRegressionTraining
import warnings; warnings.filterwarnings("ignore"); import numpy as np
from sklearn.model_selection import train_test_split

# Load data
X = np.load('semeion_X.npy'); y = np.load('semeion_y.npy')
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=True, random_state=23)
N = len(X_train); M = len(X_test)

# Train and classify
W = LogisticRegressionTraining(X_train, y_train, bs=10)
haty_test = LogisticRegressionClassification(X_test, W)
accuracy = np.sum(haty_test==y_test)/M
print(f"Test error: {1.0-accuracy:.1%}")
```

```
Test error: 12.5%
```

Exercise 1

Applying the logistic regression classifier with default parameter values and batch size 10, adjust the maximum number of epochs in logarithmic scale to determine an optimal value. Report the classification error rate on the training and test sets. Is overfitting observed? If so, from what epoch?

Exercise 2

Using maximum number of epochs 2000, learning rate (eta) 1e-2 and applying the logistic regression classifier, adjust the batch size in logarithmic scale to determine an optimal value. Report the classification error rate on the training and test sets.

Exercise 3

Applying the logistic regression classifier with default parameter values and batch size 10, adjust both the maximum number of epochs and the learning rate (eta) to determine the optimal values. Use in both cases a logarithmic scale. Report the classification error rate on the training and test sets. Discuss the results obtained.

Exercise 4

According to the results you have obtained, could you claim that this task is linearly separable? Why?