Practical work 02 – 25/02/2025
Model Selection

Objectives

The main objectives of this Practical Work for Week 2 are the following :

a) Implement and test CE cost function.

b) Implement k-fold cross validation for hyperparameter tuning.

Submission

- **Deadline** : Tuesday 4 March, 3pm
- **Format** :
  - Exercise 1 (Gradient Descent Implementation - CE-Cost and k-fold Cross-Validation) :
    - Jupyter Notebook `k-fold_cross-validation_stud.ipynb` completed with your solutions.

  Submission of all files in a single zip-file using the naming convention (for team of two students #1, #2) :

  `family name_given name #1- family name_given name #2.zip`

Exercise 1 Gradient Descent Implementation - CE-Cost and k-fold Cross-Validation

In a first step complete the implementation for gradient descent learning for the generalised perceptron with CE Cost and analyse the results. Then extend the notebook with k-fold cross validation for hyperparameter tuning.
To do so use the Jupyter Notebook `k-fold_cross-validation_stud.ipynb`. It is an extension of the Notebook `generalised_perceptron_stud.ipynb` (c.f. PW-01).
Use **numpy** functionality only. The sections of code that you need to implement are marked with

START YOUR CODE

END YOUR CODE

Proceed as follows :

(a) Study the class `NeuralNetwork` and implement the gradient calculation (`back_propagate`) and the cost function (`cost_funct`) for CE cost.

(b) In cell [12] complete - where required - the section to extend the data-set split into train-, *validation-* and test-set.

(c) Run the training (chosing an appropriate learning rate and number of epochs) for CE cost and determine the matrices for any combinations of binary classifications possible (c.f. Table 1 in `FTP-DeLearn_Lecture-Notes`).

(d) In cell [14] implemented k-fold cross validation and perform hyperparameter tuning by changing the number of training epochs between 50 and 1000 for a learning rate $\alpha = 0.5$. Plot the mean validation error - including the standard deviation - similar to Figure 79 in the lecture notes.

Hints :

— Keep an eye on the shapes of the arrays (as used in the dummy implementation).

— In case of problem you may want to try using PyCharm debugger to analyse problems.