Exercise 2: Multilayer Perceptron

Fabian Haas   **Markus Reichl**   **Florian Weingartshofer**

The goal of this exercise was the implementation and analysis of two types of Multilayer Perceptrons (MLPs) for a binary classification problem. The objective is to apply these neural networks to the **Ecoli** dataset, specifically to distinguish between two protein localization sites: cytoplasmic (*cp*) and inner membrane without signal sequence (*im*).

Data Preperation

We downloaded the dataset and used pandas to parse it and remove all the rows not containing `cp` and `im` as well as droping all non-numerical columns (except for the label)

Implementation from Scratch

- **Architecture**: the architecture of our self implemented MLP consists of an input layer, one hiden layer of size 5 and an output layer of size one. The size of the hidden layer was choosen because it is not too big to manually adjust weights, one output layer is enough since this classification is binary.

- **Activation function**: For activation we choose the simple and commonly used ***Sigmoid*** function.

- **Evaluation**: Since the task explicitly stated that we should not train the scratch implementation, the result can vary greatly depending on the choosen weights. After randomizing a lot of weights, and running the MLP multiple times, we got weigths that achive at least decent results of about 69% accurracy. We also got weigths that achieved result of less than 30% accurracy.

Class	Precision	Recall	F1-Score	Support
cp	0.67	1.00	0.81	143
im	1.00	0.10	0.19	77
Accuracy			0.69	220
Macro Avg	0.84	0.55	0.50	220
Weighted Avg	0.79	0.69	0.59	220

Implementation using Sklearn

- **Architecture**: in order too get the good results, we used `GridSearchCV` to find hyper parameters from a selection of different hidden layer sizes.
- **Training & Evaluation**: for training, we choose an **80/20** split, aka 80% of the data for training and 20% for evaluating the results. After running our cross validation, we got the best result from this hidden layer (10, 8, 5, 2)

Class	Precision	Recall	F1-Score	Support
cp	0.96	1.00	0.98	26
im	1.00	0.94	0.97	18
Accuracy			0.98	44
Macro Avg	0.98	0.97	0.98	44
Weighted Avg	0.98	0.98	0.98	44

In Summary the results of the scratch implementation are pretty much random, and while the result of the Sklearn implementation seem great, it did not have too much data for validation and thus might not generalize enough