Munier Antoine

# Homework 1 Recurrent Neural Networks

## 1. Introduction :

The goal of this exercise is to code an LSTM that can detect AI-generated code. We are free in its design, the goal being to have the most accurate model possible. We will therefore first compare different LSTMs and then we will try to improve the best one by playing on these hyperparameters. The first LSTM that we will set up is a basic LSTM, it will serve as a benchmark to compare the following ones. The second is a Bidirectional LSTM. The addition of bidirectionality is intended to improve the efficiency of the model by reading the sentence in reverse. We will also test an LSTM combined with a CNN. This is supposed to be the best because the CNN captures local paths and the LSTM keeps the direction over the long term. Once the model is chosen, we will optimize its parameter in order to have the best model.

## 2. Résultats :

The results for the classic LSTM are: 0.9858. This is very good, even too good to be true. A classic LSTM has results between 75 and 85%. We can also say that we wouldn't do an exercise in finding the best model when the simplest of them is already perfect. We deduce that there is a problem in our code. After checking the data, I can't find it. The other two models also have results close to 100%, so the problem comes from the data. This stops the rest of the study because we don't have a reliable basis to train and compare our models.

## 3. Conclusion :

The report ends quickly because we cannot move forward without resolving this data problem. It is not possible to conclude on the best model, even if we assume that the most complicated (LSTM + CNN) is the best.