PA2: Classifying words with a perceptron

The goal of this exercise is to consolidate the basic notions of machine learning by implementing a perceptron classifier for a semantic task.

We will classify a given set of words into two classes: those more associated with war and those more associated with peace. For this, we have provided the training data, which you can find in the file pa2_input.txt in Materials. As you will notice, B-words (= features) are proper names in this case and the T-words (objects of classification) are various nouns. All the words and the counts are extracted from the novel "War and Peace" by Leo Tolstoy.

Your task is write a Python script that takes as input a co-occurrence matrix (feature matrix) and finds the optimal weights for a sigmoid perceptron-like classifier described in this tutorial: https://nbviewer.jupyter.org/github/Christof93/perceptron/blob/master/perceptron_algorithm.ipynb

To solve this task, you are allowed to copy parts of the code from the tutorial, but you will need to find out yourself how to:

- Initialise the weights (e.g. how many you need)
- Calculate the output
- Calculate the error
- Update the weights as a function of the error (use the same formula as in the tutorial)
- Define the stopping criterion

In this way, you will refresh the following notions:

- input instance space
- weight vector
- weight updating
- error
- learning rate.

Note: Please make sure that we can run your script from the command line. The input file should be passed as an argument from the command line too.

Submission

Upload to OLAT by 17.04.2023 at 15h

- your Python script (named pa2.py)
- your weights as a text file, one number per line, named weights_pa2.txt

Plotting

This part is not obligatory, but you can try to plot the input data with the function from PA1, just to see where these T words end up in a two-dimensional space.