CISC 867: Bonus Project 3 (10 points)

Building a recurrent language model

A language model can predict the probability of the next word in the sequence, based on the words already observed in the sequence. In this problem, you will need to prepare text for developing a word-based language model, design and fit a neural language model with a learned embedding and a recurrent hidden layer, and to use the learned language model to generate new text with similar statistical properties as the source text.

You will use *The Republic by Plato* as the source text. A cleaned version (no hyphens or punctuations, removed nonalphabetic words, and all words in lowercase) can be downloaded from https://www.gutenberg.org/cache/epub/1497/pg1497.txt. The file should be about 15,802 lines of text. Now we can develop a language model from this text. You need to:

1. Organize the text into sequences each of 50 input words and 1 output word. You will have exactly 118,633 training patterns to fit our model.
2. **T**rain a statistical language model using a recurrent architecture from the prepared data that
    a. uses a distributed representation for words so that different words with similar meanings will have a similar representation.
    b. learns the representation at the same time as learning the model.
    c. learns to predict the probability for the next word using the context of the last 100 words.
3. Try different types of recurrent nodes (GRU, LSTM) and different number of nodes/layers and report how this has affected the results

Submission: Please include the following files in the archive: (project_3.zip):

- Report_3.pdf/docx
- Project_3.py/ipynb