P2.2: Recurrent Neural Networks (RNNs).

- Deadline: April 20, 16:00.

- Objective. To develop different deep recurrent neural networks to solve a text classification problem.

- Dataset.

  - We will use the Amazon Reviews for Sentiment Analysis (Kaggle dataset). This dataset consists of a few million Amazon customer reviews (input text) and star ratings (output labels).

  - The classes are __label__1 and __label__2, and there is only one class per row. __label__1 corresponds to 1- and 2-star reviews, and __label__2 corresponds to 4- and 5-star reviews. 3-star reviews, i.e. reviews with neutral sentiment, were not included in the original dataset.

  - Most of the reviews are in English, but there are a few in other languages, like Spanish.

  - The original dataset has 3,600,000 examples for training and 400,000 for testing. We will use a reduced version of the dataset, with 25,000 examples for training and 25,000 examples for testing.

- Code.

  - The file generateAmazonDataset.ipnb loads the data (readData) and transforms the text (transformData). The output of transformData is the train and test data that must be used.

  - transformData transforms text input to integer number input based on a vocabulary using the Keras function TextVectorization. It requires two hyperparameters: the size of the vocabulary (maxFeatures) and the maximum length of the text (seqLength). By default, seqLength has been set to the average length of the training samples plus two times their standard deviation.

- Evaluation. The evaluation metric that must be used is the classification accuracy.

- Task to be carried out.

  - Use the training set to train the network, and the test set to provide the final result.

  - You model can have any type of Keras layer (Dense, SimpleRNN, LSTM, etc.), including combinations of them. However, **at least one of the layers of the model must be any kind of RNN**.

  - You can try any hyperparameters for your model: number of layers, number of units, activation function, regularization, batch size, etc. You can also modify the following hyperparameters:

    - maxFeatures: size of the vocabulary.

        - seqLength: maximum length of the text for each sample. If the text of a sample is shorter than seqLength, TextVectorization will add 0 at the end to complete the input vector.

        - The embedding dimension. You will need an embedding layer after the input layer. The embedding layer turns positive integers (indexes) into dense vectors of fixed size (the embedding dimension). E.g. [[4], [20]] -> [[0.25, 0.1, -0.3], [0.6, -0.2, 0.81]] for an embedding dimension of 3.

- Submission.

    - The exercise will be developed using a Jupyter Notebook and the Keras framework.

- The notebook should include:

    - The practice can be carried out alone or in pairs, so the first cell of the notebook must be the full names of the authors.

    - The code for each of the models developed should be included and it should be a complete ML process: data loading and manipulation, network creation, training, and results.

    - The notebook will be saved with the results of its execution included.

    - The code shall be accompanied by cells with an explanatory report containing a description of the process followed, detailing the problems encountered and justifying the decisions taken. It should also include a section on results and discussion of them.

- Submission process.

    - The exercises will be submitted using the virtual campus of each university:

        - Universidade da Coruña: https://campusvirtual.udc.gal/

        - Universidade de Vigo: https://moovi.uvigo.gal/

        - Universidade de Santiago de Compostela: https://cv.usc.es/

    - Each member of the practice group must submit the notebook in their corresponding Moodle task.

    - There is a strict deadline for each assignment. Past due submissions will be rejected.

- Evaluation criteria.

    - Quality of the predictions made. Take as a reference:

        - A classification accuracy of at least 84% on the test set.

    - Quality of the design.

    - The network design follows the recommendations on how to create RNNs.

- Quality of explanations:
    - The process is adequately detailed, and the decisions made are justified.
    - The results are commented and interpreted correctly.