Artificial Intelligence — Lab Sessions 3-4

ESIR – Université Rennes 1

2021–2022

For any questions outside of the lab sessions, feel free to e-mail me at : **Tom.Bachard@inria.fr**.

The goal for this second lab session is to learn how to use recurrent neural networks, in particular LSTMs and word embedding. These techniques should enable you to construct text classifiers using deep learning.

Evaluation : At the end of the AI practical session, you are required to submit a report. The report should consist of all the materials that you learned, experiments along with the insights. While we expect you to submit the final report after the last practical session, you should start editing the document from the first session on. Your final evaluation will be based on your gitlab (see Exercise 0 from session 1-2) repository and on your report.

Exercise 1 : Recurrent Neural Network and IMDB classification

Objective : In this exercise you will familiarize yourself with recurrent neural networks and in particular with LSTM.

Dataset : Download and load the IMDB dataset.

Analyze this LSTM code up to testing. It is the code given in example in the Keras library. It demonstrates how to construct a text classifier using LSTM for a text classification task. We would like to predict the sentiment of short textual comments about movies, recorded on the site IMDB. Based only on the text content, we would like to predict whether a comment is positive or negative.

Run the example code. Analyze the performance of the classifier on the test set and training set :
- If your training loss is smaller than test loss, then your network is likely to *overfit*. In that case, add (or increase) dropout and/or decrease the network size.
- If your training loss and test loss is almost the same, then your model is likely to *underfit*. In that case, increase the size of the network and/or the number of nodes/layer.

Do not forget to include all your remarks in your report !

Further tutorial and other usefull links that we (**strongly !**) advise you to read :
- Tensorflow text classification with RNN ;
- Sequence classification using LSTM. Unfortunately, the blog is not updated and the tutorial is implemented in tensorflow 1.0. You may tune it for running on tensorflow2.0 environment ;
- Understanding LSTM ;
- The notebook file *Explore Overfitting and Underfitting with IMDB dataset* in Moodle.

Exercise 2 : Text classification on the Ohsumed dataset

Objective : The goal of this exercise is to realize a text classifier using deep neural networks. Your task is to construct a classifier, using the available training set, and evaluate it using the test set. The classifier should predict the category for the articles.

Dataset : We will work with the Ohsumed dataset that contains abstracts of scientific articles from a large medical publication database. The articles are related to one of 23 categories of cardiovascular diseases. The dataset is uploaded on moodle.

Dataset description : The dataset has two versions :
— One that contains the first 20000 articles (split into training and test set), available on Moodle ;
— A more complete version that has all articles (50000 articles), available here.
You should work with the version that has 20000 articles, and only use the complete version **only once** you constructed and analyzed an efficient classifier for the smaller set. As usual, the dataset is split into two parts : a training set and a test set. Each article of the collection is labeled with one of the 23 categories.

Overall strategy : To realize a classifier you should use an **iterative** approach : try to realize a simple classifier first, then analyze its performances (e.g. using the accuracy metric), then chose techniques to improve the performance (in terms of accuracy) of your classifier. Try to describe and document the development process in your report : which experiments did you run ? What were the results ? What did you do to improve the performance and why ?

To implement your first text classifier from scratch, we advise you to :
— Look into the data, get familiar with its structure and the type of text you will process ;
— Explore the data and compute some basic statistics. For example, what is the size of the vocabulary ? What is the number of examples in the training/test set per category ? What is the frequency of the words ? What is the most common, the least common ? Feel free to add anything you find relevant on the dataset ;
— Preprocessing : adjust your data to be able to pass them into a neural network ;
— Create a neural network with a simple architecture and train it ;
— Evaluate the performance of your classifier on the training set and the test set. These numbers can give hints about the quality of your classifier and can also can guide you to decide what to do in order to improve the performance. Beware of under-fitting and over-fitting ;
— Interpret your results (in plain text) and set your next objective (for example, if you had an over-fitting and you decide to add regularization, explain it) ;
— Iterate (many many **many** times) and try to improve your model. Tunning a network is sometimes difficult due to all the possibilities for the hyper-parameters.

In order to improve your network during your iterations, you can try to :
— Invest time to do more preprocessing. For example, remove frequent words, remove stopwords, perform stemming. A quick introduction here ;
— Change the parameters of your algorithms and your optimizers, cost functions, *etc* ;
— Add dropout, early stopping, regularization ;
— Change the network topology/use different techniques (multilayer perceptron, LSTM, *etc*) ;
— Only when you managed to obtain good results, witch to the bigger dataset.