Classifier Machine Learning - Task 5

Gabriel Gattaux

June 2021

1 Introduction

To complete this task I used Matlab,My files and the Toolbox *Deep Learning Toolbox* and *Text Analytics Toolbox*. To understand how the project is running, and how to run it please read the README.txt file. My algorithm is based on LSTM Network. Why LSTM Network ? Because LSTM Network is particularly useful with some long passage of text.

2 Creation of the Classifier

2.1 Data Processing

Import the Dataset, I used my function *ReadDataSet2* and the function *extractFileText* given by the *Text Analytics Toolbox*. First my data is in a a matrix of (number_of_data,2). e.g first row : Dataset(1,:) = [(Ad sales boost Time ....),'business']. We can see in the Figure 1 that the *business* and *sports* category has bigger data than the others.

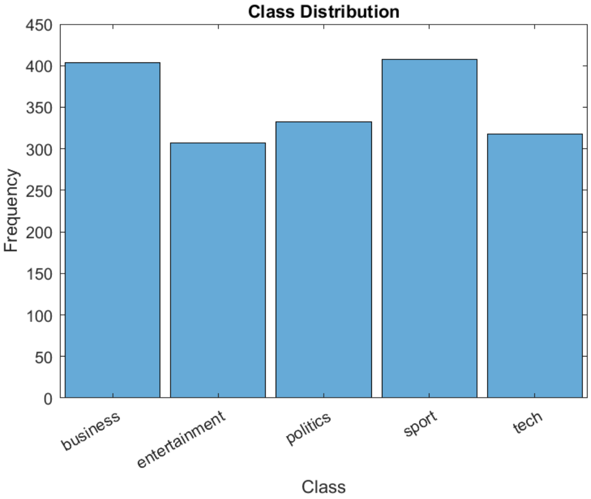
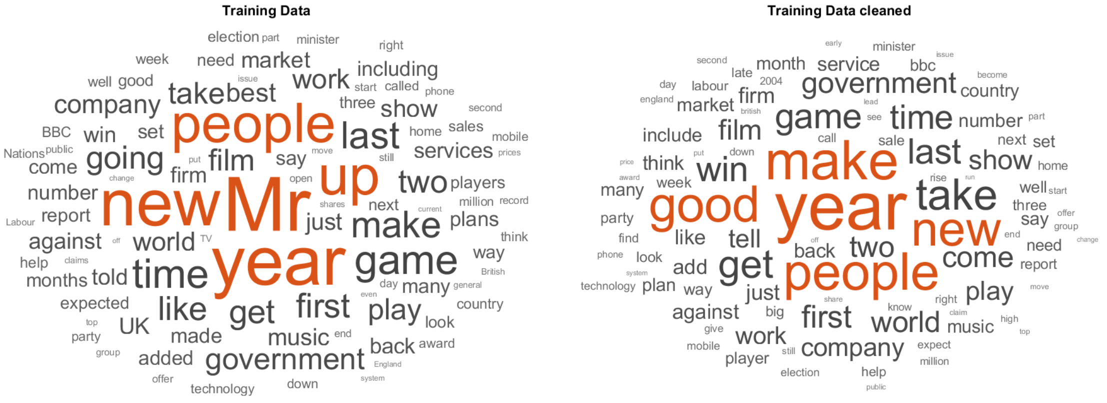


Figure 1: Histogram of the Category

Preprocessing was made by few different steps, e.g : Tokenizing, add speech details, remove stop word,Lemmatizing, erasing punctuation, removing long and short words and converting to lowercase. We can see the WordCloud before the Preprocess at Figure 2a and after at the Figure 2b. The partitioning of the Data at the begining is 10% for the Validation Set and 90% for the Training Set.

2.2 Feature Vector Construction

To input the documents into a LSTM Network, I used a **word encoding** to convert the documents into sequences of numeric indices. There was 21764 words encoded. In order to **Pad and truncate** the document, I had to choose a target length thanks to the histogram Figure 3. I choose a value of Sequence Length = 500.


Training Data
election part minister right week need market including well good issue work start called phone company take best three show second BBC win set people last home sales mobile Nations public come going put film say move services prices still change firm open up two players number new Mr shares next current million record Labour report just make plans against off world TV think way help claims year game British months told time day many general even expected like get first play country top UK look party made music end back award group added government England offer system technology down
Training Data cleaned
early minister second late month service bbc issue day labour 2004 government become england market firm country lead british include film game see time number part call sale next set price think put win down make last show home award week rise run many take well start three party good year offer say group find like tell off new end need change phone look back two come add get system report technology plan way people move just share right play against big know claim first high give world music top mobile work still company expect player million election help public

(a) WordCloud not cleaned

(b) WordCloud Cleaned

Figure 2: WordCloud

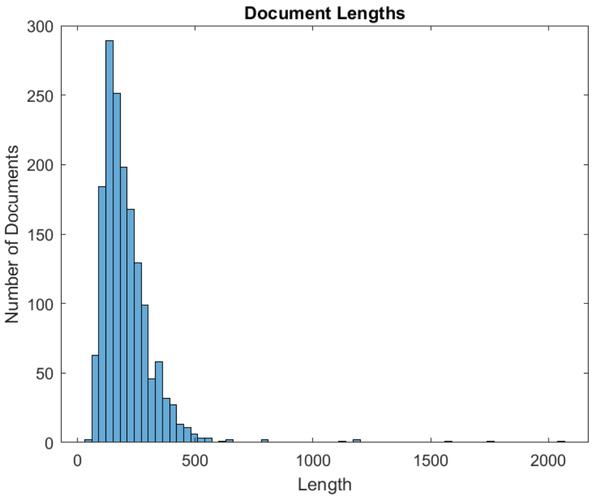
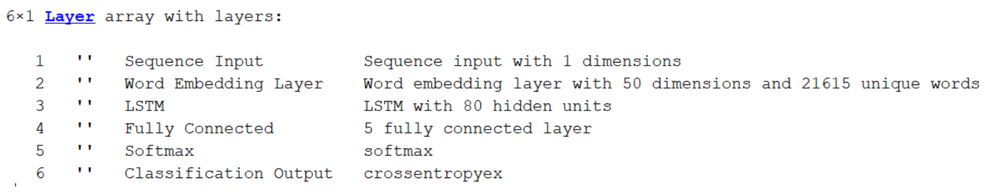


Figure 3: Histogram of Document Length

2.3 LSTM Network

In this Part I defined the LSTM Network Architecture, I defined the Layers as show in the Figure 4 and the training options that you can observe in the matlab file. This Layers and options could be changed to have a better results, we will speak about in the Improvement Section. In the Option I choose to have the plots of the Training Progress. In the options, we can choose if we uses a GPU or a CPU to compute, actually the GPU will be faster. By default it's the CPU.



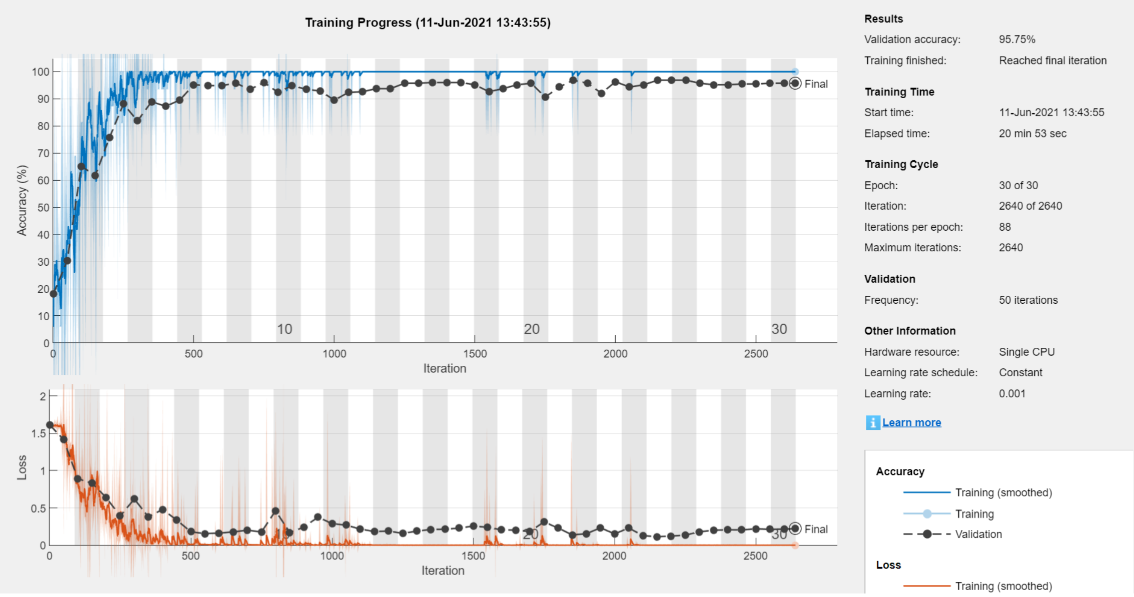
```
6×1 Layer array with layers:

   1   ''   Sequence Input           Sequence input with 1 dimensions
   2   ''   Word Embedding Layer     Word embedding layer with 50 dimensions and 21615 unique words
   3   ''   LSTM                     LSTM with 80 hidden units
   4   ''   Fully Connected          5 fully connected layer
   5   ''   Softmax                  softmax
   6   ''   Classification Output    crossentropyex
```

Figure 4: Layers and Options of the LSTM Network

3 Results

3.1 Training Analysis

If you run the program, you can observe the plot *'Training Progress'*. As you can also see in the Figure 5. We can see on this Figure Three type of curves, separate in two different plots. The first plot is for estimate the accuracy of the method, The curves "Training Accuracy" is the Classification accuracy on each individual mini-batch, the smoothed one is easier to analyze. The black one is the validation set. We can observe that after 500 iterations and 6 Epochs of the data set (An epoch is a full pass through the entire data set) This algorithm started to stabilize. The loss plots also. The Elapsed time is quite big, so future changes was made to decrease this time. The mean of the validation accuracy each time is almost 95%.



Results
Validation accuracy: 95.75%
Training finished: Reached final iteration

Training Time
Start time: 11-Jun-2021 13:43:55
Elapsed time: 20 min 53 sec

Training Cycle
Epoch: 30 of 30
Iteration: 2640 of 2640
Iterations per epoch: 88
Maximum iterations: 2640

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Figure 5: Training progress

3.2 Prediction with New Data

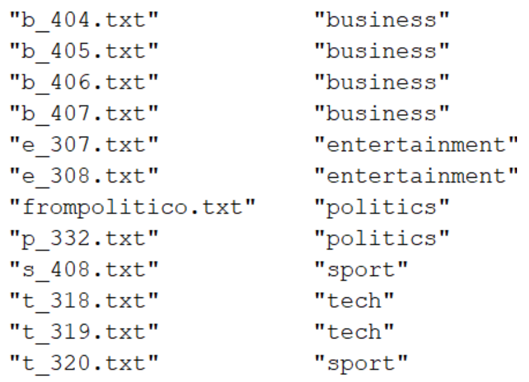
The variable Classification include the name of the file to classify and the classification made by the LSTM Network. So to test my Classifier I took some text from the data training and put inside the Data to classify, and I have also took an article from politico.com. As you can see on the Figure 6a we have only 1 files which is not true.

3.3 Confusion Matrix

We can observe on this matrix (fig. 6b) that almost all of the category are good predicted, and some are not like if the category is Sport, the predicted can be tech or entertainment. This confusion matrix was made by classifying the all Data for training.

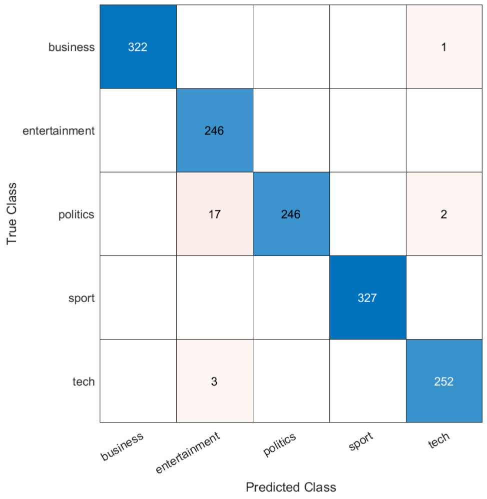
4 Improvement

In the Beginning I put the sequence length to 10, after thanks to the graph : 500, this fact passed my Accuracy of 84% to 95%. The number of SequenceLength influenced as well the Elapsed Time. For example with SequenceLength of 10 the time was 4 minutes. And for 500 is 20 minutes.We can notice on the Figure 5 that after 10 epochs, the algorithms doesn't change drastically, so we can just take into account this part. The changing value of *ValidationPatience* can save some time.



```
"b_404.txt"          "business"
"b_405.txt"          "business"
"b_406.txt"          "business"
"b_407.txt"          "business"
"e_307.txt"          "entertainment"
"e_308.txt"          "entertainment"
"frompolitico.txt"   "politics"
"p_332.txt"          "politics"
"s_408.txt"          "sport"
"t_318.txt"          "tech"
"t_319.txt"          "tech"
"t_320.txt"          "sport"
```

(a) Variable with data classified



(b) Confusion Matrix

Figure 6: Classify data and Confusion Matrix from other classified data

The mini-batch was also changed. I tried with the Classification Learner app, with a Naives Bayes algorithm, and decision tree, but they was an accuracy about 50%...

5 Conclusion

This Task was very interesting and very useful for my future and I learned a lot of things to compute an Neuronal Network in Matlab thanks to that. I learned also the LSTM Network which is an Recurrent Neural Network particularly used in the Text Classification. The best accuracy I had was 98% in 10 minutes of Training.

6 Literature

LSTM Network : *https : //towardsdatascience.com/lstm−networks−a−detailed−explanation−8fae6aefc7f9*

Text Analytics Toolbox : *https : //www.mathworks.com/help/textanalytics/index.html*

Deep Learning Toolbox : *https : //www.mathworks.com/help/deeplearning/index.html*

How to improve : *https : //machinelearningmastery.com/improve − deep − learning − performance/*

Create simple text model for classification : *https : //www.mathworks.com/help/textanalytics/ug/create−simple − text − model − for − classification.html*

Classify using deep learning : *https : //www.mathworks.com/help/textanalytics/ug/classify−text − data − using − deep − learning.html#ClassifyTextDataUsingDeepLearningExample − 4*

Training options : *https : //www.mathworks.com/help/deeplearning/ref/trainingoptions.html*

Setting up parametre : *https : //www.mathworks.com/help/deeplearning/ug/setting − up − parameters − and − training − of − a − convnet.html*