Neural Networks

Carlos III University of Madrid

April 22, 2024

Practice III – RNNs for Sentiment Analysis.

Pablo Gradolph Oliva - 100458456
Raquel Parajuá Delgado - 100454359

TABLE OF CONTENTS

1. Introduction.

A RNN is a type of artificial neural network designed especially for the use in processing sequences of data. They have recurrent connections that allow them to maintain an internal state or memory and whose architecture allows the output to be dependent not only on the input in that moment but also of the internal state calculated in the previous time steps.

This characteristic that differentiates them form traditional neural networks, make them really useful to perform sentiment analysis. Since comments, opinions or sentences are sequences of words, a RNN can process these sequences word by word considering the order and structure of the text, and given that RNN maintain memory of previous states, they can give a context and a relationship between words in a sentence, which is important because opinions can be influenced by a previous context.

In this third project we've studied how this is done with a real data set of financial news, analyzing the accuracy of a LSTM, and comparing it with the one of a MLP classifier (optional).

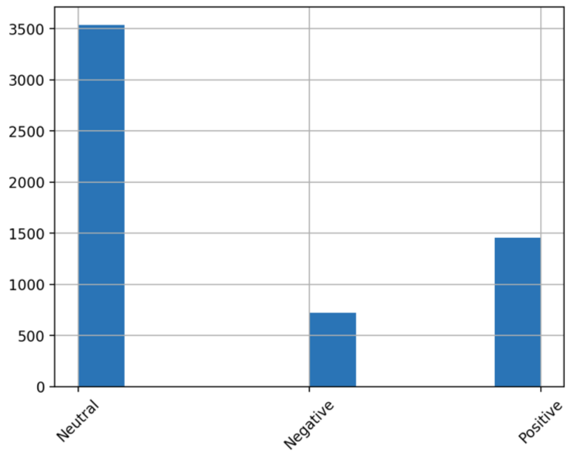
And then, we've studied what happens when applying a simple attention layer before the classifier, and whether performance improves or not.

If for some reason, you cannot see the images in the attached Jupyter Notebook, please access this GitHub repository to see them: https://github.com/PabloGradolph/Neural-Networks/blob/main/Project3/Project3.ipynb

2. Procedure and Results.

2.1 No Attention layer.

To start off, sentences are divided and labeled as 0 for neutral, 1 for negative and 2 for positive and a histogram is plotted that shows them.



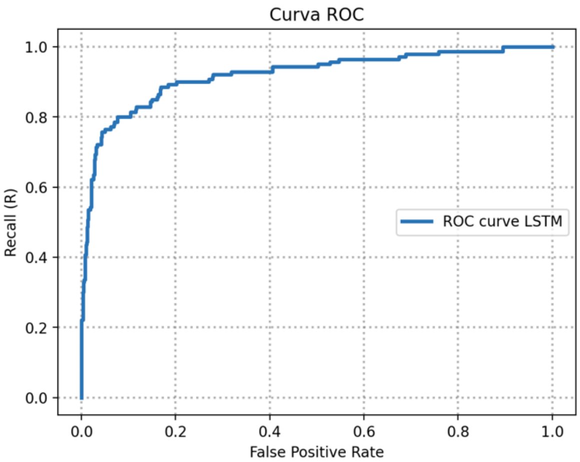
Then, each sentence is pre-processed by removing words that are either not very informative or punctuation marks, so we first have the original sentence and then the normalized one. This is going to allow us to get the word embedding vector which is the most important attribute for us, and whose dimension is 300.

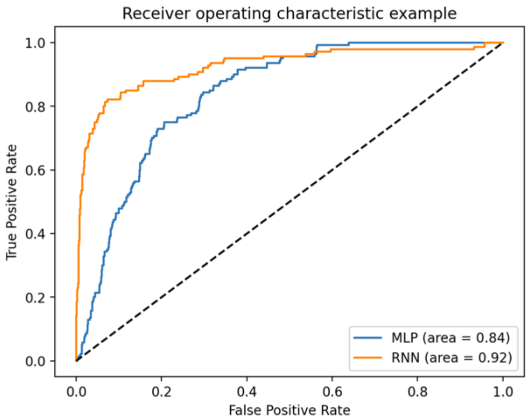
After this, we need to normalize the whole dataset, so all sequences have the same length. We do this by artificially making them the same length by adding junk tokens, even though we keep in a list the real length of each data, since the sentiment analysis prediction will be using the RNN state corresponding to the last real token.

For the RNN classification we want to implement an LSTM that takes as an input the sequence of word embeddings and predicts the binary label. Long Short-Term Memory is a type of recurrent unit used in RNNs that faces and eases the problem standard RNNs have of the vanishing gradient. The key feature of LSTM is the memory cell structure, which contain gates that regulate the flow of information, including an input gate, a forget gate and an output gate, that allow to retain and update information over time more effectively than standard recurrent units.

We tested the accuracy of LSTM and an AUC-ROC, which is a metric used to evaluate the performance of a binary classification model by plotting the true positive rate against the false positive rate for different threshold values. Generally, an AUC-ROC value above 0.5 tells us that the model performs better that random guessing and the closer to 1 it is, the nearest to perfection.

In this case, accuracy for LSTM was of 0.9326923076923077, and the AUC ROC for it is 0.9202618099032442. This value suggests that the LSTM model is quite effective at distinguishing between the two classes and indicates that the model has a strong ability to correctly classify instances of the positive class while minimizing false positives.





For the optional part, we've also plotted the ROC curve for MLP, and this is what we got:

AUC ROC for MLP: 0.8384319863403529, this shows us a slightly lower performance of the model with a Multilayer Perceptron.

Reasons for this might be that LSTM models are specifically designed to handle sequential data and are more equipped to sense dependencies, while MLPs might struggle to capture these relationships effectively. However, it still indicates a value well over 0.5, therefore we can consider it pretty effective as well.

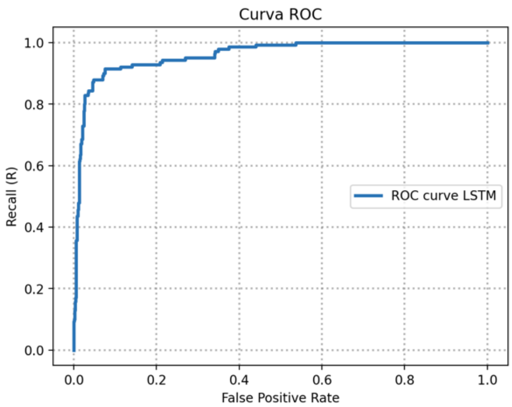
All together, these are the different values and the differences between both models reflected in accuracy and AUC ROC:

MLP Test Accuracy: 0.8776223776223776
RNN Test Accuracy: 0.9361888111888111
MLP ROC AUC: 0.8385813887307911
RNN ROC AUC: 0.924338360842345

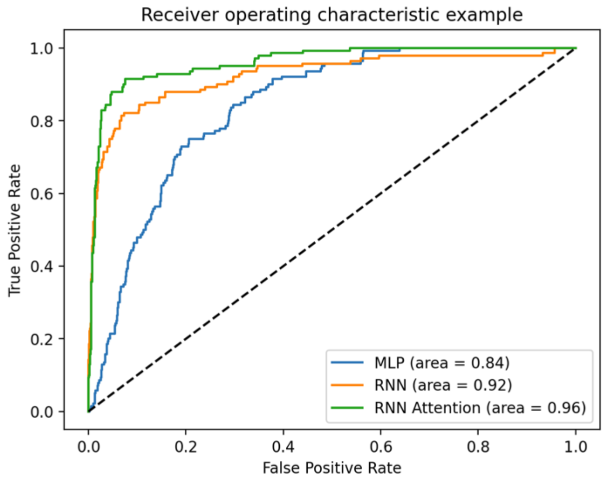
2.2 With Attention

An attention layer is a fundamental part of many neural network models, especially when sequences or sets of data have concrete parts where some are more relevant than others in different moments. (in this case for example there are more relevant words for the purpose than others). Its main function is to make the model focus on specific parts of the input and with those characteristics that are relevant on each specific time. An attention layer accurately designed can be significative in terms of the performance of the model, since it will focus on the important parts, the model's capacity of sensing complex data connections will improve and lead to an overall better performance. Let's see how our attention layer affects our previous model.

First, the same pre-processing procedure was done as in the previous section and following it, accuracy and AUC ROC was obtained for the model with the attention layer.
The following was obtained:



The test accuracy is 0.9554195804195804 and the AUC ROC for RNN with Attention is 0.960394137734775. We can already see a value closer to 1 than in the model without the attention layer, but let's look at all the models together.



With all three together we can see the different values of the AUC ROC. The lowest value and therefore the worst is for the MLP model, for the reason that was previously explained; then the RNN model without the attention layer follows with a decent value; and finally the highest and best values are for the RNN model with the attention layer.

This tells us not only that the attention layer has done what theoretically promises and has improved the overall performance of the model, but also that we accurately designed it, because otherwise, if it were badly or poorly designed or implemented, the models performance would have been negatively affected by the unwanted introduction of noise or biases in the model, which can diminish the capacity to generalize to new data and negatively affect the model's precision.

3. Conclusions.

Overall, we can conclude that when we are faced with a problem of processing sequences of data, like performing a sentiment analysis, in which a memory for context and relationships between sentences is crucial, the best model found that achieves the best test accuracy and highest AUC ROC is the RNN with the attention layer accurately designed, which was tested against the same RNN without the attention layer and the MLP model. This shows us that the attention layer indeed focuses on the important part of the data on each moment and allows a better capacity and performance of the model by improving complex connections, however, we have to keep in mind that if the attention layer is not well designed or implemented it might have the opposite to the desired effect.