CSCM38: Adv Topic - Artificial Intelligence and Cyber Security - Coursework 2: Comparing an RNNs LSTM and GRU Cells

Andy Gray
445348

18/12/2020

1 Introduction

We will be exploring the proposed solution and the libraries, dataset, preprocessing, algorithms, metrics that will get used for comparison and the NN parameters that we used. We will then be analysing the results, and discussing what insights they provide, ending with a conclusion.

2 Proposed Method

We will be creating an experiment that will compare the two different types of RNN cells, the LSTM and the GRU. We used several parameters to train and test the different RNNs as well as several other metrics to be able to compare the performance of the cells. The RNNs aim to be able to accurately predict if a Tweet posted on Twitter is about a real disaster or not. This experiment used many different Python 3 libraries.

2.1 Libraries & Frameworks

We used a collection of different Python 3 libraries to conduct this experiment. We used Tensorflow 2 [1] for creating the RNN model, LSTM and GRU cells. It got also used for preprocessing the text and sequence with 'Tokenizer' and 'Pad Sequence'. We used Sci-Kit Learn [3] for splitting the dataset and for creating the confusion matrix. Additionally, Pandas [] got used for handling the dataset and Numpy to allow the other libraries to be able to do their scientific calculations. We also used NLTK [] for NLP's stop words as well as some of Python's extra libraries os, time, re and string.

2.2 Dataset and Preprocessing

We used a dataset from Kaggle called "Natural Language Processing with Disaster Tweets" [2]. While a training and CSV file is available, we decided to use the training set due to the test dataset not having any labels. Therefore if we used this dataset, it would be hard to compare how well each cell performed on with an unseen dataset.

The dataset had a shape of 7613, 5. These include the features 'id', 'keyword', 'location', 'text' and 'target'. Due to the 'id' feature not having any relevance and the 'keyword' and 'location' containing null values, these features got dropped from the dataset. The dataset's targets were either a 0 for a non-disaster tweet or 1 for a disaster. There were 4342 non-disaster and 3271 disaster observations.

We then removed the characters [**List of Characters**] for the dataset's text. We have done this to make sure that the RNN focuses on the contents of the text and not have to focus on the punctuation as this could impact on the model's performance if we had left them in. We also removed all the stop words from the text by using the NLTK library. This action got done to make sure that these stop words also don't impact on the model's understanding of the text. The stop words included: [list stop words]. Along with removing the stop words, we removed any URLs that were in the tweets as these have no relevance on if the tweet is about a disaster or not. When the stop words, punctuation and URLs got removed from the text, we ended up with 17,971 different words contained within all the tweets. The most common words appearing were 'like' (345), 'I'm' (299), 'amp' (298), 'fire' (250) and finally 'get' (229).

Once the dataset was all preprocessed, we split the data set into 2/3 training 1/3 testing. We also then split the training set into an 80/20 split of training and validation data. We did this to see if the dataset was getting overfitted within in training and also to add an additional method of comparison between the two RNN cells. We then tokenised the unique words to create a word index to give the text a number representation for feeding through the RNNs. We then padded the sentence to 20 sequences. We did this to ensure that the length of text within the text would all match up, as the tweets have varying sizes anyway and with the previous clean up actions being down, potentially additional text has also been removed.

An example disaster text is "malaysia airlines flight 370 disappeared 17months ago debris found south indian ocean" and a non-disaster is "walk plank sinking ship".

2.3 Algorithms

For this experiment, we will use the LSTM and the GRU cells. These are both modifications of the RNN. The vanilla RNN, an unaltered RNN, is a robust network. However, it suffers from some issues. These issues are that it only has a short term memory, a suffers from a vanishing gradient point and an exploding gradient too.

2.3.1 LSTM

2.3.2 GRU

2.4 Metrics for Comparison

Time, loss, accuracy, validation loss and accuracy, confusion matrix

2.5 Neural Network Parameters

3 Results and Discussion

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Kaggle. Real or not? nlp with disaster tweets, 2020.

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.