Gurjus Singh

MSDS 458 – Deep Learning and AI

February 21st, 2021

Week 7: A.3 Third Research/Programming Assignment

**Abstract**

In this research, I explored the AG News Topics Dataset which consisted of 127,600 news headlines divided into 4 news categories World, Science/Tech, Business and Sports. The goal of this research was to use 1D CNN, RNN and LSTM architectures to do supervised learning multiclass classification on the news dataset. This involved heavy amount of preprocessing and encoding of the words in the articles. In the research, I did a total of 23 experiments.

**Introduction**

This research involved exploring a topic of Deep Learning which is Natural Language Processing, and how to find patterns in the words to classify. The dataset I was trying to classify on was the News dataset from Reuters which involved supervised learning multiclass classification on four categories which included World, Science/Tech, Business and Sports. I tried to play around with different architectures to observe different accuracy results. The architectures included 1D CNN, RNN and LSTM architectures. I also noticed preprocessing made a huge difference in the results.

**Literature Review**

I decided to read two papers on the topic of NLP and news text classification. In the first paper surrounding news classification, it went over techniques surrounding news classification [1]. Some things I took away from this paper was that classification involves steps such as collection of the news from a variety of different sources [1]. Then the preprocessing happens which involves tokenization [1]. In the process of tokenization, the words are broken up and treated as strings [1]. After tokenization, stop words removal happens which removes words that do not carry and information [1]. Stemming also happens in preprocessing which reduces a word to its root [1]. Feature selection also happens during preprocessing [1]. Once preprocessing happens, then it is time to classify the news [1]. The paper mentions several techniques for classification which are to use Naïve Bayes, Support Vector Machines, Artificial Neural Networks, Decision Trees, or K-Nearest Neighbors [1]. In the second article, it compared several modeling techniques to each other using the AG News Dataset [2]. It talked about using bag of words, and ngrams in the preprocessing stages. The results showed that Bag of Words with CNN performed with 88.76 percent accuracy [2].

**Methods**

The first step in the research involved importing the packages seen in 1-1. A picture containing shape

Description automatically generated

*Import Packages 1-1*

Several Packages were helpful in the research which was the Keras and Tensorflow packages. These were important to make the architecture for RNNs, LSTMs, and CNNs. I also checked the Tensorflow and Keras version to make sure they were both current. Next, I downloaded the AG News Dataset using the **tfds.load** function. What I observed after downloading is that there were 127,600 examples in the dataset. I then converted the dataset to a dataframe using **tfds.as\_dataframe** to see the first 10 examples as shown in 1-2. Graphical user interface, text, application

Description automatically generated

First 10 Examples 1-2

I also got information about the labels, and noticed there were 4 categories mainly World, Sports, Business, and Science/Tech. After observing the data and labels, I then split up the datasets into Train, Validation and Test.

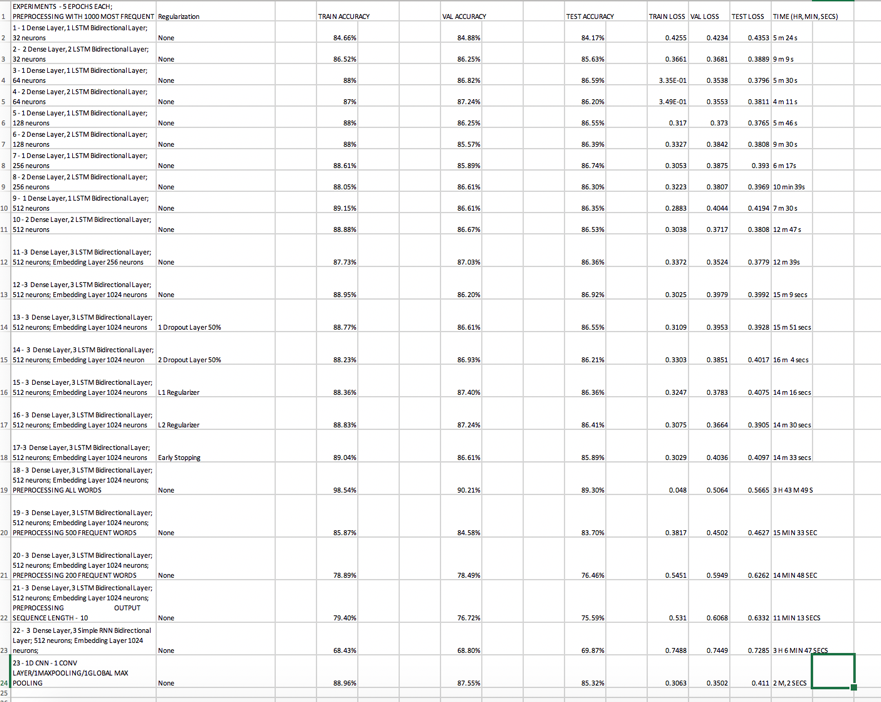
Next was the preprocessing part. The preprocessing T**extVectorization** was important for this as it gave me a way for the number of words we could use in our corpus. I first limited the dataset to the 1000 frequent words, which I would later change in the experiments. After limiting it to 1000 most words, the encoder was used to represent the words as numbers which then could be inputted into the neural network. After preprocessing, I then created my architectures. I tried out different number of layers such as dense, bidirectional , LSTM, and RNN layers. I also tried out using different neural networks such as switching between RNNs, CNNs, to LSTMs. I also played around with the **TextVectorization** function as explained above shrinking the vocabulary or increasing it from most frequent words.

In this research, there was a layer that was not used in previous research that I have done which was the embedding layer. The embedding layer is used to input 2D tensor of words mapped to integers [3]. The embedding layer functions in that it learns the relationships between the words and puts represents it in a geometric space [3].

**Results**

In this research, I did a total **of 23 experiments, 5 Epochs each**. I played around with the preprocessing, layering and regularization parameters. My goal was to get the Test Accuracy up 90 percent, but according to my results in 1-3, I could not obtain 90 percent and the closest I got was 89 percent. From the results in 1-3, it suggests that the best model was Experiment 12 which has a Training Accuracy of 88.92 percent and a Testing Accuracy of 86.92 percent. For Experiment 12, my architecture involved an embedding layer with 1024 neurons, three Dense/Bidirectional LSTM Layers with 512 neurons each. The reason I did not chose the model with Testing Accuracy of 89.30 percent because it was highly overfitting with a Training Accuracy of 98.54 percent.

In my experiments I also tried shrinking/increasing the number of frequent words used when I reduced most frequent words it started underfitting, but when I increased the number of most frequent words, the accuracy increased and started overfitting, which led to the experiment with 98 percent accuracy. I also tried one 1D-CNN model and my accuracy was 88.96 percent for Train Accuracy and 85.32 percent for Test Accuracy.



*Experiments and Results 1-3*

After examining each model and choosing Experiment 12 as the best model, I then wanted to examine the Confusion Matrix for Test Dataset as shown in 1-4. I also examined the precision and recall score which were both around 0.86 as seen in 1-5. This showed the model was doing very well in its results.

Graphical user interface, text, application, email

Description automatically generated

*Confusion Matrix Experiment 12 Test Dataset 1-4*

*Graphical user interface, text, application

Description automatically generated*

*Precision and Recall Score Test Dataset 1-5*

**Conclusion**

This research involved examining the AG News Dataset with 127,600 examples. The purpose of the research was to do a Supervised Learning Multiclass Classification on 4 category labels mainly World, Business, Sports and Science/Tech. After exploring my 23 experiments, I decided that the best model was Experiment 12 which involved an architecture of one embedding layer with 1024 neurons, three Dense/Bidirectional LSTM Layers with 512 neurons each. **Therefore, for my management recommendation, I recommend an architecture of one embedding layer with 1024 neurons, three Dense/Bidirectional LSTM Layers with 512 neurons each.**

For further research, I hope to examine more CNN layers as I felt on the first try the CNN did particularly well. I also hope to examine more data preprocessing techniques such as removing stop words and incorporating ntlk package. I did use all the words in one experiment, but my model ended up overfitting as explained above. I can try using regularizers on this overfitted model, or early stopping.

References

[1] Kaur, G., & Bajaj, K. (2016). *News classification and its techniques: a review*. Semantic Scholar. <http://www.iosrjournals.org/iosr-jce/papers/Vol18-issue1/Version-3/D018132226.pdf>

[2] SreeDevi, J., Rama Bai, M., & Chandrashekar Reddy, M. (2020, March 5). *Newspaper Article Classification using Machine Learning Techniques*. International Journal of Innovative Technology and Exploring Engineering. http://www.ijitee.org/wp-content/uploads/papers/v9i5/E2753039520.pdf

[3] Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Company.