Part 1: Research question

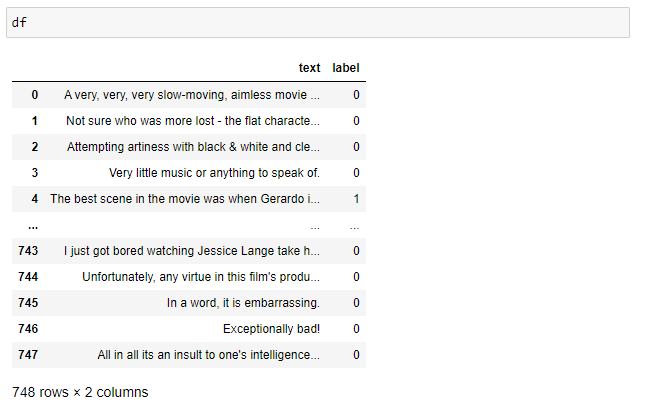
A.1 One research question I can answer using neural network techniques is “Is that review positive or negative?”

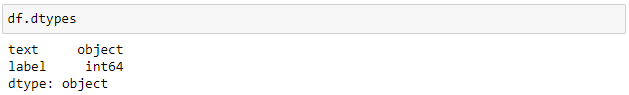
A.2. The objective of this analysis is to predict the connotation (positive or negative) of IMDB reviews given their text.

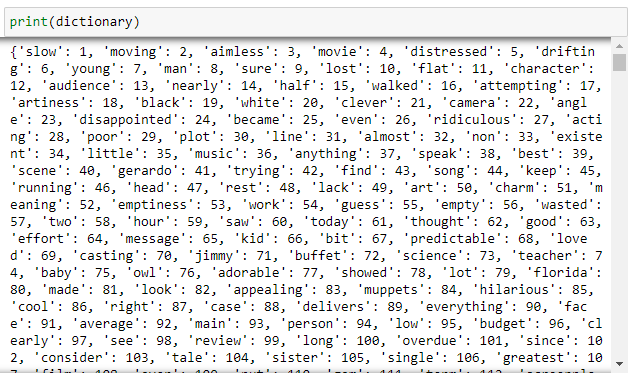
A.3. The types of neural networks that are capable of performing a text classification task that can be trained to produce useful predictions on the selected dataset are Convolutional Neural Networks and Recurrent Neural Networks (Text Classification).

Part II: Data preparation

B.1. My exploratory data analysis included viewing the dataframe, the column types, and the dictionary of used words.

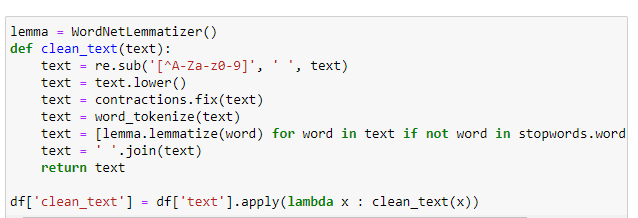






* After printing the vocabulary I confirmed there were no unusual characters outside of punctuation in the dataset.
* The vocabulary size for my dataset is 2746 words.
* My proposed word embedding length is 50.
* The justification for my chosen maximum sequence length is after printing all words in the dictionary, I made note that most appear to be 25 characters or less. 50 was a good length to make sure all words were included but don’t take up too much space.

B.2. The goal of the tokenization process is to break sequences (in this case reviews) into smaller parts. It is also a good place to standardize the text and remove any unwanted characters. I chose to tokenize the data set into words.The screenshots of my code are below.

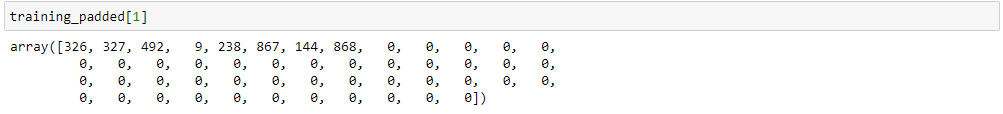


The packages I used above to complete tokenization are WordNetLemmatizer and contractions. Later in the program I call the tensorflow Tokenizer object to assist in padding sequences.

B.3. The first step of the padding process used to standardize the length of sequences is to tokenize the reviews into sequences instead of individual words. After that, pad\_sequences from tensorflow was used and then those padded sequences were converted into an array.

In this case, the padding occurs after the text sequence.

A screenshot of a single padded sequence is included below.



B.4. The number of categories of sentiment used is two (positive, negative).

The activation function for the final dense layer of the network is ‘softmax’.

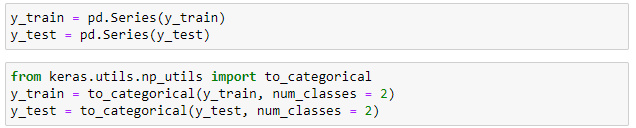
B.5. After the text was cleaned and padded, I converted the dataframe with the clean text into an array and defined my variables.



Next, I split the data using train\_test\_split with a training size of 80% of the data and a testing size of 20% of the data.



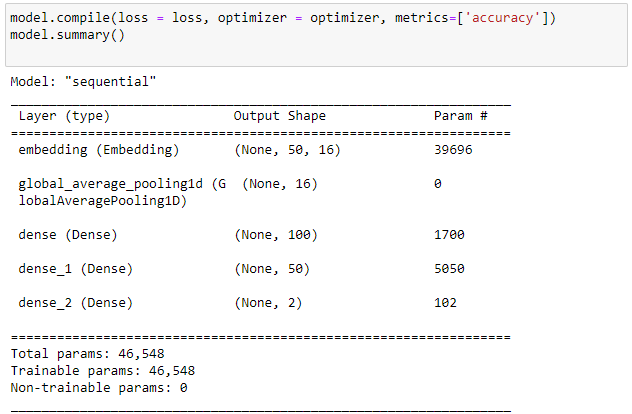
After that I converted the y variables into series with categorical variables and the data preparation was complete.



B.6. A copy of the cleaned dataset is included in the cleaned\_data.csv file of this submission. The prepared split dataset is included as well.

Part III: Network architecture

C.1. The output of the model summary of the function from Tensor Flow is below.



C.2. The number of layers in my network is 5.

1st layer: Core type = Embedding

* Layer type = Input
* Number of parameters = 39,696

2nd layer: Core type = Global average pooling 1d

* Layer type = Flatten
* Number of parameters = 0

3rd layer: Core type = Dense

* Layer type = Hidden
* Number of parameters = 1700

4th layer: Core type = Dense

* Layer type = Hidden
* Number of parameters = 5050

5th layer: Core type = Dense

* Layer type = Output
* Number of parameters = 102

C.3.

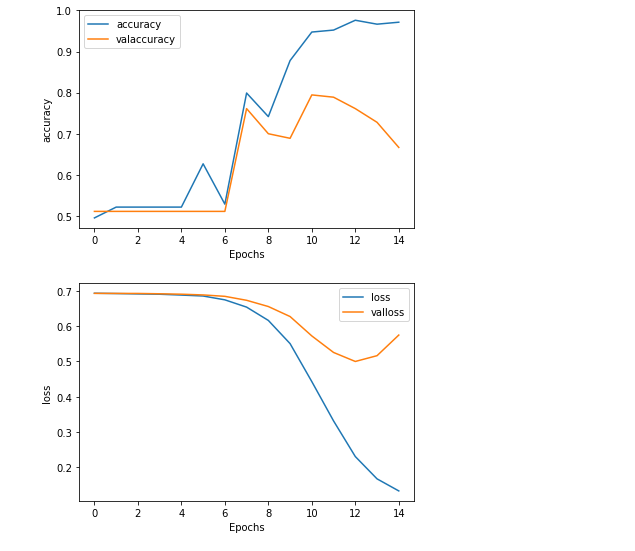
* The activation functions are ‘softmax’ and ‘relu’. Softmax is useful because it calculates relative probabilities. Relu is the most commonly used activation function for hidden layers.
* The number of nodes per layer is 100 for dense, 50 for dense\_1, and 2 for dense\_2. The only way to optimize these parameters is by retesting, so this is the first attempt at finding efficient ones.
* The loss function is ‘categorical\_crossentropy’ because it is used for classification models where there are two or more output labels.
* The optimizer is ‘adam’. It applies an algorithm called Adam that is based on adaptive estimation.
* The stopping criteria is a patience of 2. The patience parameter defines the number of epochs with no improvement in the model before it stops running. I chose this parameter because when only running < 20 epochs, 2 with no improvement is a large enough of a gap to consider stopping the program.
* The evaluation metric is test accuracy because it largely indicates how useful the model is in real life. This model received a score of 0.680.

Part IV: Model Evaluation

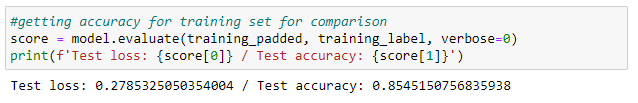
D.1. The impact of using stopping criteria instead of defining the number of epochs is the model is less likely to be overfitted. The final training epoch is depicted below.



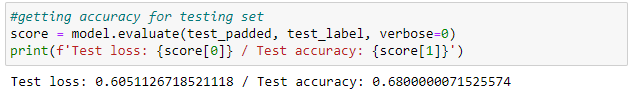
D.2. The model's training process is depicted in visualizations below.



D.3. To avoid overfitting, I defined stopping criteria instead of just the number of epochs. The model was fitted using the training data. For comparison to the testing data accuracy, I calculated the accuracy for the training data the model was fitted on. Here we got a value of 0.854 which is lower than expected but high enough to consider the model well fitted.



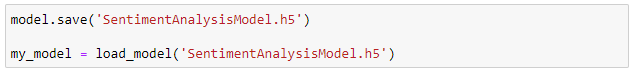
D.4. The predictive accuracy of the trained network is 0.680



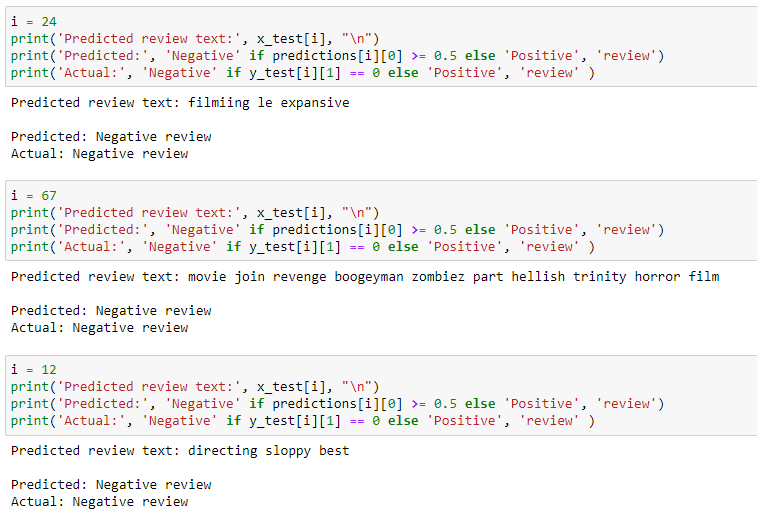
There is room for improvement that I hypothesize could be achieved by using a larger dataset.

Part V: Summary and Recommendations

E. The code used to save the trained neural network is below.



F. The functionality of my neural network is acceptable but not ideal. I would like an accuracy level of 0.9 or greater. Of the three reviews I viewed, however, all three were accurately labeled as pictured below.



The impact of the network architecture is that this program is accurate the majority of the time, while only taking less than a minute to run. If the number of epochs was greater, this could be much slower, and possibly risk overfitting the data.

G. Based on the results I recommend collecting more data or acquiring a larger dataset. There are tens of thousands of movies on IMDB with many reviews each. This dataset has approximately 750 entries. More data could drastically improve accuracy, which currently sits barely over the rate of chance.

Part VI: Reporting

H. A pdf of my notebook is included in this submission.

I. Third party code

Team, K. (n.d.). *Keras Documentation: Earlystopping*. Keras. Retrieved March 15, 2022, from https://keras.io/api/callbacks/early\_stopping/

J. Works cited

*Text classification: What it is and why it matters*. MonkeyLearn. (n.d.). Retrieved March 8, 2022, from https://monkeylearn.com/text-classification/#:~:text=The%20two%20main%20deep%20learning,a%20progressive%20chain%20of%20events.