Name: Anto Murugan
SID: 862154748

CS173 Final Project

Story Generation

Introduction

The goal of this project is to create a program that can automatically generate a story based on a seed text and input for number of words. Using a pertained model on the modified dataset of the Stories of American Movies, the program uses the seed text to generate a story.

Python Libraries used for this project:

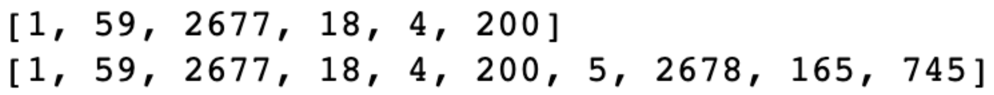
1. Tensorflow
2. Numpy

Approach

I chose the Stories of American Movies dataset from Kaggle (https://www.kaggle.com/revanth9/wikipediaplotdata). But the entire dataset had about 500 000 words which was too large for my computer to process. Therefore I randomly deleted stories in the dataset to bring the total words under 6000.

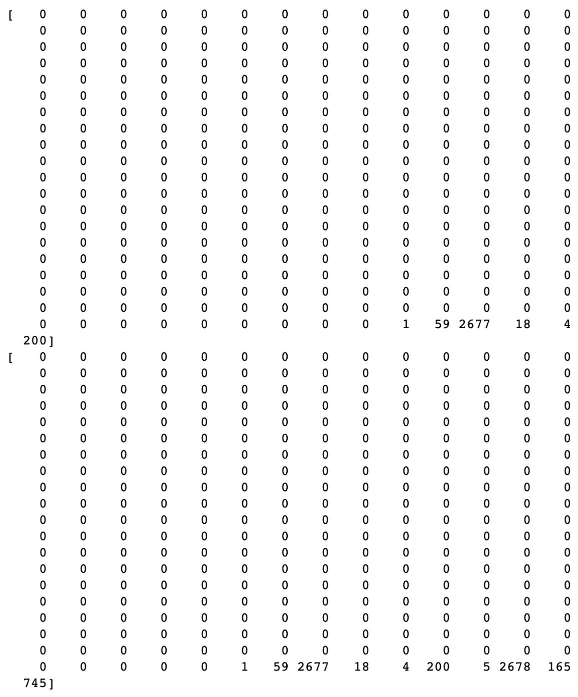
Then I created a corpus with all the words in the working dataset converted to lowercase and removed the newline characters. I fit the Tokenizer( ) function from the tensorflow library to tokenize the words in my corpus.

Following the tokenization, I created input sequences for the training model using the texts_to_sequences( ) function. This allowed me to create numerical representations of the sentences with the respective word index. I also had to pad each sequence using the pad_sequences( ) function to match the length of the longest sequence in the corpus. This allowed my training and testing data to be balanced.



```
[1, 59, 2677, 18, 4, 200]
[1, 59, 2677, 18, 4, 200, 5, 2678, 165, 745]
```

The image above shows the sequence before they are padded. Since each sentence have different length, the sequences have different length, resulting in the data being unbalanced.



```
[    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    1   59 2677   18    4
   200]
[    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    1   59 2677   18    4  200    5 2678  165
   745]
```

After padding the sequences are all the same length. As a result the data that is fed into the training model is balanced.

I used the Sequential model from the tensorflow library to create a Bidirectional Long-Short Term Memory (LSTM) neural network. The LSTM model is a variant of the Recurrent Neural Network (RNN). Unlike RNN, the units in LSTM include a 'memory cell' that can maintain information in memory for long periods of time. The Bidirectional LSTM adds more context than regular LSTM for the input sequences. Bidirectional LSTMs can be used to train both sides of the input sequence. First from left to right and the second in reversed order of the input sequence.

A RNN:

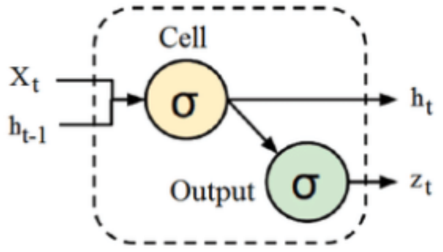


image source: https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359

A LSTM:

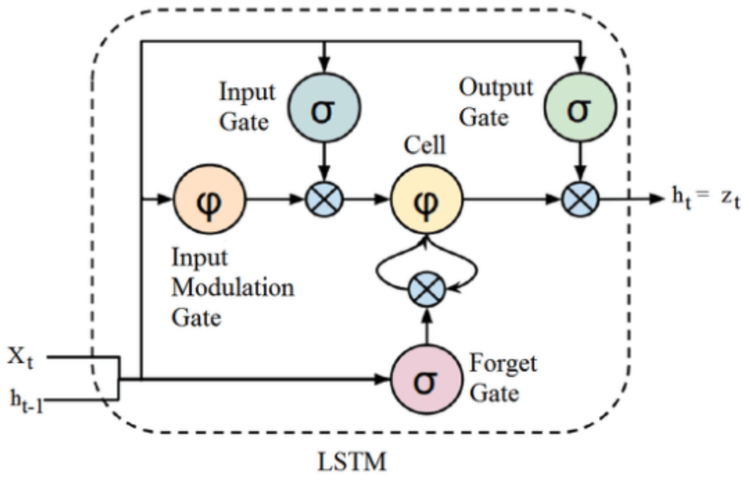


image source: https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359

A Bidirectional LSTM

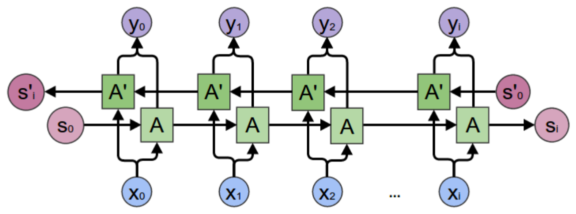
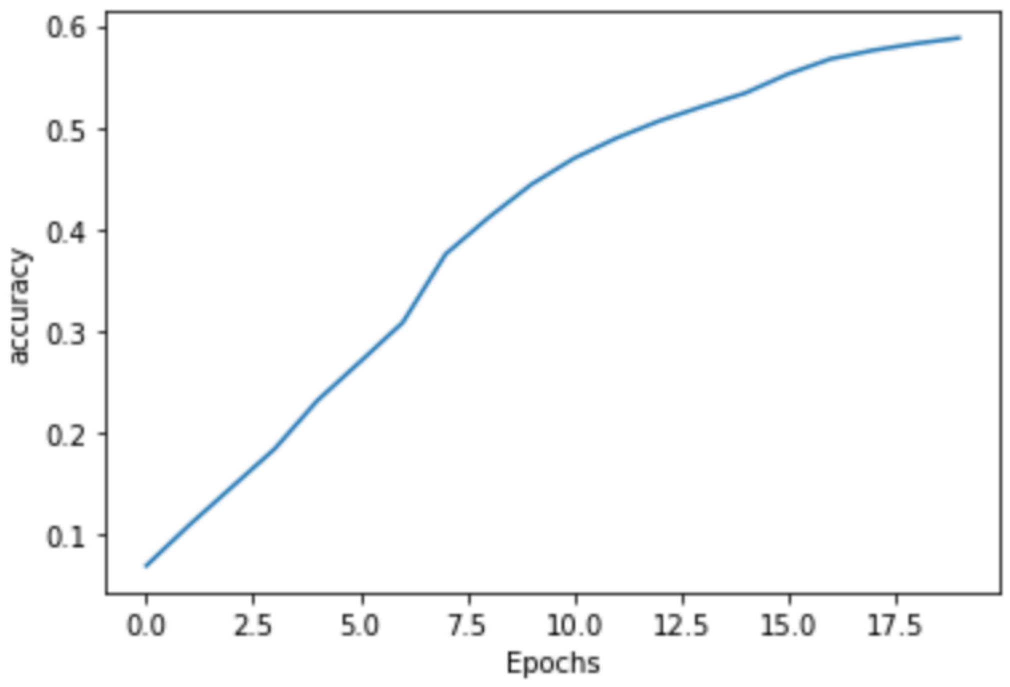


image source: http://colah.github.io/posts/2015-09-NN-Types-FP/

After training the data, I was able to save the trained model to test the story generation aspect of my project. The maximum accuracy attained by the training model is: 62.6%

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 280, 50)           291750
_________________________________________________________________
bidirectional (Bidirectional (None, 100)               40400
_________________________________________________________________
dense (Dense)                (None, 5835)              589335
=================================================================
Total params: 921,485
Trainable params: 921,485
Non-trainable params: 0
_________________________________________________________________
```



<u>Challenges</u>

Choosing a topic for the project was probably the biggest challenge. I explored chatbots focusing on various topics but I wasn't able to choose a specific topic to work on. Then I found out about text generation during the discussion and since I was unfamiliar with the tensorflow library or Neural Networks in general, I decided to focus on text generation.

It took me some time to learn and understand how to use the tensorflow library for my project. The training time was also much longer than other machine learning models such as regression, which meant that I had less time to explore different values for the number of LSTM layers and epochs to achieve a higher accuracy for my trained model.