Building Language Model Using Recurrent Neural Network

In this project we are building Language model to determine the probability of a given sequence of words occurring in a sentence. For data-preprocessing we are loading the text from the book "Poirot Investigates" by Agatha Christie. The book is loaded into google colab with the name book.txt. For pre-processing the data, first we replaced all the '--' with ' '(spaces) in our text, Next, we are removing punctuations and lower case all of the characters (so as to reduce the number of characters in the vocabulary, making it easier to learn for the model. Since neural network operates on numbers, we are translating the characters into numbers, then the keys and values, or the characters and numbers that represent them, are then stored in a dictionary. For input and output we are transforming each text in texts to a sequence of integers. For the model we are giving sequence length as 5. So that the model receives 5 words sequence as input and outputs the probabilities with which each word can succeed the input sequence. The model consists of one embedding layer and one bi-directional (it connects two hidden layers of opposite directions to the same output. With this form of generative deep learning, the output layer can get information from past (backwards) and future (forward) states simultaneously.) layer with 32 LSTM cells. The numbers of neurons in the fourth layer are same as the number of words in the text which is 7841. The neurons in the fourth layer, use softmax activation so as to convert their outputs into respective probabilities.

Models Hyperparameters:

Hyperparameter	Value
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Epochs	100

I am using categorical cross-entropy to encode class labels, I run the model for 100 epochs and I am using early stopping on loss to stop the training when there is no improvement in the loss after 5 consecutive epochs. The model stopped after 50 epochs as there were no improvement in the loss.

Results:

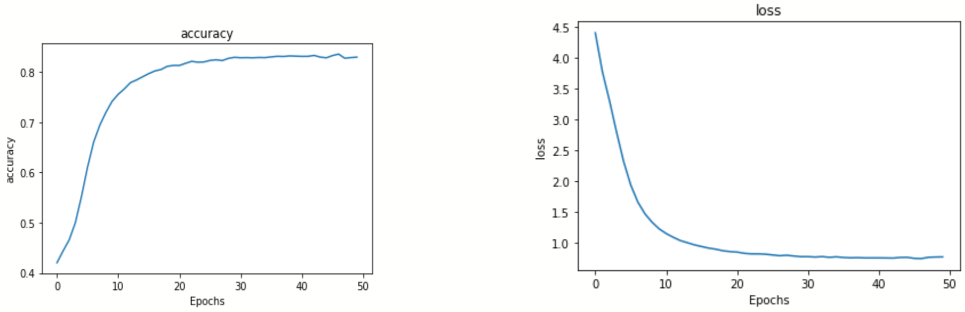
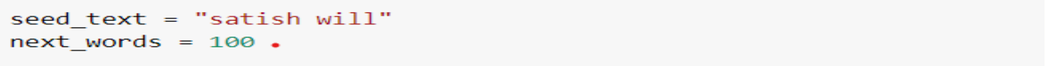
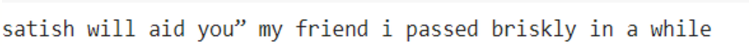


Fig 1: Accuracy and loss for Language model

Input and output:



```
seed_text = "satish will"
next_words = 100
```



```
satish will aid you” my friend i passed briskly in a while
```