COMP 7747: Advanced Topics in Machine Learning
Final Project Midterm Report

Title: Predict the Next Word in Sentence using Neural Networks

Dataset:

The data corpus considered for this problem is collected from an English-language eBook about Project Gutenberg's "The Adventures of Sherlock Holmes," written by author Arthur Conan Doyle,that contained various sub-contents like stories, documentation, and text data, which are necessary for our problem statement. In addition, we can use a variety of articles as text data for this problem.

Sample data from data corpus:

```
I had seen little of Holmes lately. My marriage had drifted us away
from each other. My own complete happiness, and the home-centred
interests which rise up around the man who first finds himself master
of his own establishment, were sufficient to absorb all my attention,
while Holmes, who loathed every form of society with his whole Bohemian
soul, remained in our lodgings in Baker Street, buried among his old
books, and alternating from week to week between cocaine and ambition,
the drowsiness of the drug, and the fierce energy of his own keen
nature. He was still, as ever, deeply attracted by the study of crime,
and occupied his immense faculties and extraordinary powers of
observation in following out those clues, and clearing up those
mysteries which had been abandoned as hopeless by the official police.
From time to time I heard some vague account of his doings: of his
summons to Odessa in the case of the Trepoff murder, of his clearing up
of the singular tragedy of the Atkinson brothers at Trincomalee, and
finally of the mission which he had accomplished so delicately and
successfully for the reigning family of Holland. Beyond these signs of
his activity, however, which I merely shared with all the readers of
the daily press, I knew little of my former friend and companion.
```

Data corpus link:
https://drive.google.com/file/d/1GeUzNVqiixXHnTl8oNiQ2W3CynX_lsu2/view

Given a characterization of the data corpus:

Length of the Data corpus:

```
path = 'datacorpus.txt'
text = open(path, "r", encoding='utf-8').read().lower()
print ('Corpus length: ',len(text))
```

```
Corpus length:  581888
```

Unique characters in the corpus:

```
chars = sorted(list(set(text)))
char_indices = dict((c, i) for i, c in enumerate(chars))
indices_char = dict((i, c) for i, c in enumerate(chars))
```

```
print ("unique chars: ",len(chars))
```

```
unique chars:  73
```

Implementation Process:

1. Itemize:

Following steps to implementing:- load the data from corpus, Feature Engineering, Building the model, Training, Evaluating and Testing the model.

2. Citation: Sourabh Ambulgekar, Sanket Malewadikar, Raju Garande, Dr. Bharti Joshi (2021). “Next Words Prediction Using Recurrent Neural Networks”, ICACC conference.Link:

https://www.researchgate.net/publication/353773522_Next_Words_Prediction_Using_Recurrent_NeuralNetworks

Overview:

According to their study, Next word prediction is known as Language Model. To determine the next word in text using very powerful RNN model called LSTM. The accuracy of their output prediction is 54 to 55%. I am trying to achieve more accuracy than that in my evaluation.

Text Feature engineering with text data is converting the string into numerical values. It is done by vectorization (feature matrix – following up with tokenization) with count words on the corpus, count the occurrence on each word using NLTK library. Building the LSTM model by keras package. LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. Talking about RNN, it is a network that works on the present input by taking into consideration the previous output (feedback) and storing in its memory for a short period of time (short-term memory). Then training the LSTM model with 20 epochs.

Experimental Results:

Summary of model building and finding the accuracy and loss for each epoch:

```
In [11]: model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 128)               103424

 dense (Dense)               (None, 73)                9417

 activation (Activation)     (None, 73)                0

=================================================================
Total params: 112,841
Trainable params: 112,841
Non-trainable params: 0

In [15]: optimizer = RMSprop(lr= 0.01)
         model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])

         history = model.fit(X, y, validation_split=0.05, batch_size=128, epochs=20, shuffle=True).history

C:\Users\rahul\anaconda3\lib\site-packages\keras\optimizers\optimizer_v2\rmsprop.py:140: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super().__init__(name, **kwargs)

Epoch 1/20
1440/1440 [==============================] - 310s 210ms/step - loss: 1.9660 - accuracy: 0.4229 - val_loss: 2.1273 - val_accuracy: 0.4002
Epoch 2/20
1440/1440 [==============================] - 292s 203ms/step - loss: 1.6136 - accuracy: 0.5164 - val_loss: 2.0451 - val_accuracy: 0.4410
Epoch 20/20
1440/1440 [==============================] - 296s 206ms/step - loss: 1.2819 - accuracy: 0.6062 - val_loss: 2.1763 - val_accuracy: 0.4567
```

Contribution of each team member:

Manivardhan Reddy Pindi – Responsible for feature extraction (engineering) and Storing features and labels from the data corpus which is initially done.

Rahul Marru – Responsible for implementing the LSTM model, which plays a major role in the project to find the accuracy and loss of each epoch of the model with 20 epochs. Along with citation suggestions for the baseline of the project.

Further Contribution plans on upcoming weeks:

Manivardhan Reddy Pindi – Fine tuning the model for better performance.

Rahul Marru – Verifying (Evaluating) and test the model by predicting the next few words for the given sentence. And trying to use different model is it get good performance than this.