**ICP5**

**BIG DATA ANALYTICS AND APPLICATIONS(CS5542)**

*Sravani Niharika Garapati (16328843)*

*Nikhila Chirumamilla (16323180)*

**QUESTION:**

Use different data and use the model provided in ICP6to perform Text generation. You must make4 changes (for example adding LSTM layers to model, changing hyperparameters etc.) in the source code. Report your findings in detail.

**What I have learned from this ICP:**

In this ICP, professor asked us to perform Text Generation on a data set. In this ICP, we have created 4 deep learning models which are interdependent on each other by changing 4 hyperparameters. The things which I have learned from this ICP are listed below

* I have learned many functions that were very useful in implementing the model and finding losses and many functions that are related to checkpoints and others.
* I have known which model and which layers to be used in the situation.
* I have known in detail about the taken data set and to perform analysis on that data set in a thorough manner.
* I have come to know that increasing number of epochs while fitting the model would give more accuracy.
* In this ICP, we used LSTM layer and DropOut layers are hyperparameters. So, I have known much about them and learnt when and where to use them in the problem in future.
* Also increased embedding dimensions to so that the output dimension of the first layer will be large enough for better computation speed and results.
* Increased number of RNN units, so that there will be more neurons which will help increase the accuracy.
* Learnt much about the layers of GRU, LSTM, Dense, Dropout which we have used in this ICP solution.
* At last, I have learned how to perform Text Generation using keras to a certain extent.

**ICP Description about the Task:**

In this ICP, we performed Text Generation on CIFAR 10 data set using Keras model. This is done in the following steps.

* As first step, I have imported all the necessary libraries.
* Then, download the dataset and read the contents in the dataset as the second foremost step.
* Performed Vectorization of text i.e., mapping numbers to characters and mapping characters to numbers.
* Broke up the text into sequences and created training targets and examples.
* Then converted individual characters formed into the sequences of desired size using **batch ()** method.
* Written a user defined function named generate\_text () which takes input as model we created and the starting string and generates text as output.
* Created Model 1 for the Data set

**Model -1:**

* Used Sequential () Model which is best fit for the problem statement.
* Used Embedding layer, GRU layer and a Dense layer
* Used Embedding Dimensions = 256

RNN Units = 1024

Epochs = 10

* Compiled the model and fit the model to the model.
* ***Loss Observed***: 0.9919
* Then called generate\_text() function and produced the text which is not meaningful.
* Then built the Model -2 to improve the loss and to generate the accurate text.

**Model-2(Hyperparameter 1):**

* Built the same model as Model-1
* Replaced GRU layer with LSTM layer which is considered as a hyperparameter.
* Used "tanh" activation functions with LSTM because LSTM will work accurately with longer sequences than GRU.

Activation Function: tanh

Recurrent\_Activation: Sigmoid

Recurrent\_Intializer: Glorot\_Intializer

Epochs:10

* ***Loss Observed:*** 0.9732
* Called the generate\_text() function and has observed the generated text
* Then built Model-3 to improve loss and to get accuracy of the text.

**Model-3(Hyperparameter 2):**

* Built same Model as Model-2.
* Added a Dropout () layer with 0.3 input size which is considered as hyperparameter 2.
* Used this layer to reduce the overfitting issue and to reduce the loss.
* ***Loss Observed:*** 1.0610
* Called the generate\_text () function and has observed the generated text
* Then Built Model-4 to get accuracy of the text

**Model-4(Hyperparameter 3 & 4):**

* Built same model as Model-3.
* Increased the embedding dimensions and number of RNN units as hyperparameter 3
* Increased the embedding dimension so that the output dimension of the first layer will be large enough for better computation speed and results.
* Increased the units in the RNN layer so that there will be more neurons which will help increase the accuracy.

Embedding Dimensions: 512

RNN Units: 2048

* As our hyperparameter 4, we increased number of epochs to 20 to generate meaningful text and to improve loss
* ***Loss Observed:*** 0.5803
* Called the generate\_text () function and observed that there is meaningful text as our desired.

**Challenges Faced in this ICP:**

The challenges which we faced while doing this ICP are listed below:

* It has become very difficult for us to identify the hyper parameters.
* Also, it has been very difficult for us to understand and analyze the data.

**Screenshots of Execution of Code:**

*Importing required Libraries*

**Graphical user interface, text, application

Description automatically generated**

*Downloading the text file from google*

*Text

Description automatically generated*

*Reading the contents in the file and computing number of characters in the text*

*Text, application

Description automatically generated*

*Printing the contents of the file*

*Text

Description automatically generated*

*Extracting unique characters and their count from the file*

*Text, application

Description automatically generated with medium confidence*

*Vectorization of the text*

*Graphical user interface, text, application

Description automatically generated*

*Creating training samples and examples*

*Text

Description automatically generated with medium confidence*

*Converting characters to sequence of desired size*

*Text, application

Description automatically generated*

*Graphical user interface, text, application

Description automatically generated*

*Text

Description automatically generated*

*User Defined Function for text generation*

*Text

Description automatically generated*

*Creating training batches*

*Graphical user interface, text, application

Description automatically generated*

*Graphical user interface, text, application, email, website

Description automatically generated*

*Building Model 1*

*Graphical user interface, text, application

Description automatically generated*

*Calling the Model*

*Text, letter

Description automatically generated*

*Depicting the shape of the model*

*Text

Description automatically generated*

*Creating samples and defining loss function*

*Text

Description automatically generated with medium confidence*

*Compiling the model and configuring the checkpoints*

*Graphical user interface, text, application, email

Description automatically generated*

*Fitting the model to the dataset*

*Table

Description automatically generated*

*Restoring the latest checkpoint*

*Text

Description automatically generated with medium confidence*

*Applying the model1 to generate the text*

*Text

Description automatically generated*

*Building Model 2*

*Graphical user interface, text

Description automatically generated*

*Compiling the Model and fitting the model to the dataset*

*Table

Description automatically generated*

*Applying Model 2 to generate the text*

*Text, table

Description automatically generated*

*Building Model 3*

*Graphical user interface, text, application

Description automatically generated*

*Compiling and fitting the model to the dataset*

*Table

Description automatically generated*

*Applying Model 3 to generate the text*

*Text

Description automatically generated*

*Building Model 4*

*Graphical user interface, text, application, email

Description automatically generated*

*Compiling and fitting the model to the data*

*Table

Description automatically generated*

*A screenshot of a computer

Description automatically generated with low confidence*

*Graphical representation of the loss function*

*A picture containing chart

Description automatically generated*

*Example 1*

*A picture containing text

Description automatically generated*

*Text, letter

Description automatically generated*

*Example 2*

*Text

Description automatically generated*

**Video Link**

https://youtu.be/Q\_4FDV4FLfo