Teammates:

Sukumar Bodapati – sb5zh@umsystem.edu - 16326105

GitHub link: https://github.com/UMKC-APL-BigDataAnalytics/icp-5-sukumarbodapati

Sri Nikhitha Boddapati – sb4dz@umsystem.edu - 16322565

GitHub link: https://github.com/UMKC-APL-BigDataAnalytics/icp-5-Srinikhitha98

Video link: https://youtu.be/n6n1P9EIq7U

ICP - Big Data App & Analytics -RNN

Learnings in this ICP:

LSTM Layer:

- In the icp6 source code, the model has GRU layer instead of LSTM layer. GRU uses less training parameters and therefore it results in less memory uses and executes faster than LSTM. The loss percentage is **1.6** when we used GRU layer, and the text generated is not appropriate.
- When we used the LSTM layer, the loss percentage is **0.197**. So, when we have dataset of longer sequence, and we need more accuracy we must use LSTM Layer.

```
47/47 [==============================] - 4s 70ms/step - loss: 0.2018
Epoch 99/100
47/47 [==============================] - 4s 70ms/step - loss: 0.1994
Epoch 100/100
47/47 [==============================] - 4s 71ms/step - loss: 0.1979
```

Objective:

1.Consider new data set and apply the model provided in ICP6 to perform Text Generation.

2.Change the hyperparameters and apply LSTM layers to make it a better model.

Task 1: Used the same model that is given in Source Code

1. Considered a new text data and imported it from the website.
2. Applied the same model that is given in ICP source code (**Batch size:64, BUFFER SIZE=10000, Epoch =10, No LSTM layer is used**)

3. **Loss:**

```
33/33 [==============================] - 5s 143ms/step - loss: 1.9860
Epoch 8/10
33/33 [==============================] - 5s 142ms/step - loss: 1.8546
Epoch 9/10
33/33 [==============================] - 5s 143ms/step - loss: 1.7348
Epoch 10/10
33/33 [==============================] - 5s 143ms/step - loss: 1.6311
```

4. The generated text of this model is:

```
[42] print(generate_text(model, start_string=u"machine learning: "))
```

machine learning: conseltsourate can be lecrablemothe, in difurating ang to finting bit dest tucine to bies, poster parthon. The emen agents brame Pitionational and matacture LeN 6o on Big data butunes contextral inderative (PIP, including frem bechune-searchers, and to neural neuringry fooul 200 is ans or solve complet with a gunction, quentions is suere. Tyenders to avaly: gundured-AI a hund to owhers pastical concertaning an inatess

Task 2: Changing the Hyper parameters

1. **Added LSTM layer:**

We have built a model in which we used LSTM layer instead of GRU layer and loss has decreased from **1.6** to **0.2.** So, when we have huge data, it's better to use LSTM layer.

Loss:

```
47/47 [==============================] - 4s 70ms/step - loss: 0.2018
Epoch 99/100
47/47 [==============================] - 4s 70ms/step - loss: 0.1994
Epoch 100/100
47/47 [==============================] - 4s 71ms/step - loss: 0.1979
```

2. **Epoch:** It is the number of times the algorithm is applied on the dataset.

   a) **Epoch size =25:** We have trained the dataset with epoch =25, then we observed the loss of **1.2079** and text generated is inappropriate.

   Loss:

```
Epoch 24/25
33/33 [==============================] - 6s 167ms/step - loss: 1.2339
Epoch 25/25
33/33 [==============================] - 6s 168ms/step - loss: 1.2079
```

   Text Generated:

```
4] print(generate_text(model, start_string=u"machine learning: "))

machine learning: which a computation are selm. Proeural (eel-NLab-R{suartic encimanestricterousphich can rare makes are
As sen001 And exhide Laysh Rusensis strutted only per learned data anarysis the systems Qerator, computer "consing the ws
```

b) Epoch size = 50: We have trained the dataset with epoch =50, then we observed the loss of **0.2542** and text generated is good.

Loss:

```
Epoch 49/50
33/33 [==============================] - 6s 169ms/step - loss: 0.2592 -
Epoch 50/50
33/33 [==============================] - 6s 171ms/step - loss: 0.2542 -
```

Text generated:

```
[105] print(generate_text(model, start_string=u"machine learning: "))

machine learning: PUxs in each new widely used learner", potential properties of molecules in a large rnd entatory do to more efficient methods for
TOA 20s,  mature le tolowher the network and complex the more efficient methods for training deep neural networks that contain many layers of non-l
Semplementation of the state of the dimensionality reduction techniques machine translation, bioinformatics, drug design, medical image analytics p
```

c) Epoch size= 100 : We have trained the dataset with epoch =50, then we observed the loss of **0.175** and text generated is very accurate.

Loss:

```
33/33 [==============================] - 6s 173ms/step - loss: 0.1774 -
Epoch 99/100
33/33 [==============================] - 6s 170ms/step - loss: 0.1789 -
Epoch 100/100
33/33 [==============================] - 6s 171ms/step - loss: 0.1757 -
```

Text-Generated:

```
[126] print(generate_text(model, start_string=u"machine learning: "))

machine learning: Google Translate (GT) uses an example-based machine translation method in which the system "learns from miles to give naïve users an
```

3. BATCH SIZE:

a)**Batch size= 64:** We have trained the dataset with Batch size=64, then we observed the loss of **0.175** and text generated is very accurate.

Loss:

```
33/33 [==============================] - 6s 173ms/step - loss: 0.1774 -
Epoch 99/100
33/33 [==============================] - 6s 170ms/step - loss: 0.1789 -
Epoch 100/100
33/33 [==============================] - 6s 171ms/step - loss: 0.1757 -
```

Text-Generated:

```
[126] print(generate_text(model, start_string=u"machine learning: "))

machine learning: Google Translate (GT) uses an example-based machine translation method in which the system "learns from miles to give naïve users an
```

b) Batch size=16: We have trained the dataset with Batch size=64, then we observed the loss of 0.953 and the text generated is not so good.

Loss:

```
Epoch 9/10
132/132 [==============================] - 21s 157ms/step - loss: 1.0409
Epoch 10/10
132/132 [==============================] - 21s 157ms/step - loss: 0.9539
```

Text Generated:

```
0] print(generate_text(model, start_string=u"machine learning: "))

machine learning: Manal features from maliforing and fualmories by adoptil who field's papers enters Other question that well image precision sestens.
The gunes of from on the Data Occording to (the IAI ships and (b is theoretical object. In their field.
Mathwer than knowledge a number of libraries such as choal rist class in the 1980s as come to inding well-untertines was unsupervised an inference lay
A Big Data uses ancertain classifiers performed by increased excently are extracted or lay
```

4.OPTIMIZER:

Changed the optimizer to **RMSProp:** Loss is around 0.717

Loss:

```
Epoch 24/25
33/33 [==============================] - 8s 238ms/step - loss: 0.7648
Epoch 25/25
33/33 [==============================] - 8s 239ms/step - loss: 0.7175
```

Text generated:

```
9] print(generate_text(model, start_string=u"machine learning"))

machine learning. These technology is then encode manyey uplearned unsupervised learning algorithms to topercement lon's us
Starms of context-dependent in the input layer) evaluates which time supervised one tasks. Projotential: Untelligent agent
It has been a published accordingly goal identify a representative studies.
This type can be tugns. Open-to use Python performance and houth strately use the Fehthefore used to set of focusing of int
Thoughts formational logic, non-0monento source algorithmic bias large a multi-- 1996). The technical oppogent to esp
```

But when we have used **Adam** optimizer the loss is less, and text is more appropriate. Since Adam is combination of Rmsprop and momentum.

5.Dropout:

Depth of the neural network plays a crucial role in capturing the important features while building the model. In the model we have introduced dropout 0.1 to avoid overfitting.

6.Temperature:

When we increase the temperature, the generated text is not appropriate.

```
3] print(generate_text(model, start_string=u"machine: "))
```

```
machine: somet reteir the sense objey. Various trkar sy the in
Biodaliot Jaboo!'s dagics after cthoduct kniobival logic job b
Marious ecrnengiedes, knvolVV(Di 1LMP|rsions being 300x 7' in
Long systempophoto" 8.0 ST target people at hemo"_proprixtly r
```

Observations:

From the above analysis, for this dataset, we need to build a model with LSTM layer, Batch size 64, Optimizer as Adam ,Epoch =100,Dropout of 0.1 to get a better result.

Code Tasks:

Model: Added LSTM layer with dropout and dense layers

```
[ ]  def build_model(vocab_size, embedding_dim, rnn_units, batch_size):
       model = tf.keras.Sequential([
         tf.keras.layers.Embedding(vocab_size, embedding_dim,
                                   batch_input_shape=[batch_size, None]),
         tf.keras.layers.LSTM(rnn_units,
                              return_sequences=True,
                              stateful=True,
                              recurrent_initializer='glorot_uniform',dropout=0.1),


         tf.keras.layers.Dense(vocab_size),
         tf.keras.layers.Dense(50),
         tf.keras.layers.Dense(vocab_size)
       ])
       return model
```

Output:

Loss:

```
47/47 [==============================] - 4s 70ms/step - loss: 0.2018
Epoch 99/100
47/47 [==============================] - 4s 70ms/step - loss: 0.1994
Epoch 100/100
47/47 [==============================] - 4s 71ms/step - loss: 0.1979
```

Text Generated:

```
] print(generate_text(model, start_string=u"machine learning"))

machine learning model in embedded devices removal intelligence (AGI) architectures. These issues may possibly be addressed by deep learning archite
```

Challenges Faced:

1. LSTMs are inclined to overfitting, and it is hard to apply the dropout calculation to check this issue.
2. To decrease the loss, we have set the Epoch size to 100 which has increased the runtime and it was difficult to vary other parameters.