Neural Machine Translation

DL Project - Phase 3 | Final Report

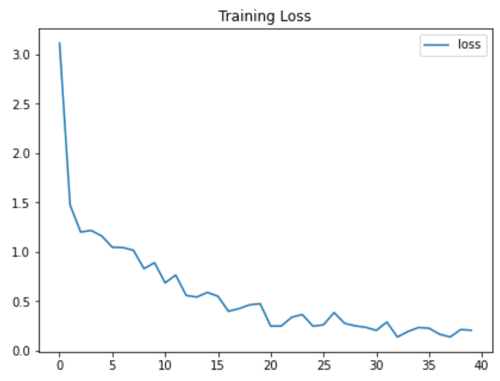
200968108 - DSE A | Batch-1

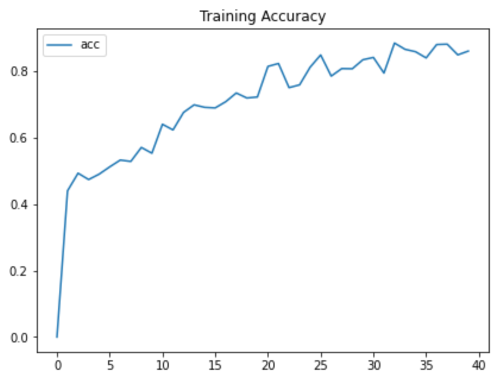
Models Implemented:

S No.	**Model Used**	Source Language	Target Language	Accuracy	Performance
1	RNN+LSTM	English	German	40%	The accuracy of the model need to be increased but it is due to training on less data. Comparatively, the model is working well.
2	RNN+Embedding+BiRNN	English	French	96%	Model is able to translate the English text into French very well with 100% accuracy for some text but its unable to show the same performance for different languages.
3	RNN+LSTM	English	Hindi	48%	Model hyperparameters needs to be further improved and trained on larger dataset for more accurate translations.
4	LSTM+ Attention Mechanism	English	German	Bleu Score - 50	Model is able to translate to German text very well but with a few mistakes, but can be improved by training on larger dataset.

S No.	Model Used	Source Language	Target Language	Accuracy	Performance
5	**Encoder-Decoder with Loung Attention layer**	English	Spanish	86%	The attention layer is performing well for translating from English to Spanish as well as vice-versa and also showing good performance on implementing with different languages.

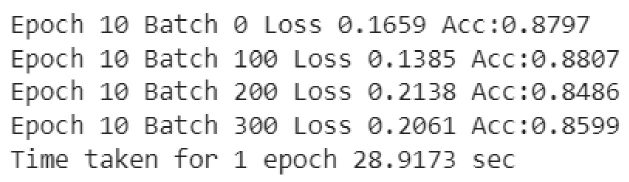
- From the above table, we can conclude that among all the implemented models for Neural Machine Translation, the **Encoder-Decoder-Architecture with Loung Attention Mechanism** is performing way better than other models.
- Below are the curves for training Accuracy and loss:





- We can observe from the above curves that the training loss is tremendously decreasing while the accuracy is increasing in the same way and stopped in the top right corner which is a good measure of how well the model is performing.
- The model achieved a Validation Accuracy: 0.8599 and loss: 0.2061 indicating that our model is having an accuracy of around 86% and is able to translate English text to Spanish text with minimal errors.

Below is the screenshot of the accuracy of the model:

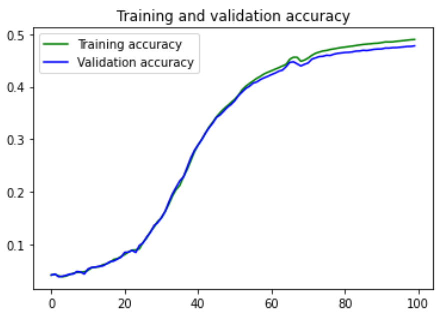


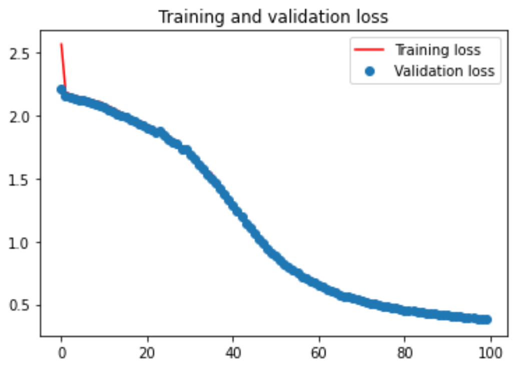
```
Epoch 10 Batch 0 Loss 0.1659 Acc:0.8797
Epoch 10 Batch 100 Loss 0.1385 Acc:0.8807
Epoch 10 Batch 200 Loss 0.2138 Acc:0.8486
Epoch 10 Batch 300 Loss 0.2061 Acc:0.8599
Time taken for 1 epoch 28.9173 sec
```

At the same time **Model 4** which is the RNN model with **LSTM and Attention Mechanism** has shown us a Bleu-score of only 50 which is also a good measure of performance.

- But this model can be made to perform much better compared to the present performance by changing the hyperparameters values like batch size and number of epochs while training the data as well as increasing the size of the dataset (using all the 250000 text samples) used for the implementation.
- then, it might be able to translate well when trained on multiple languages and can translate them into various other languages.

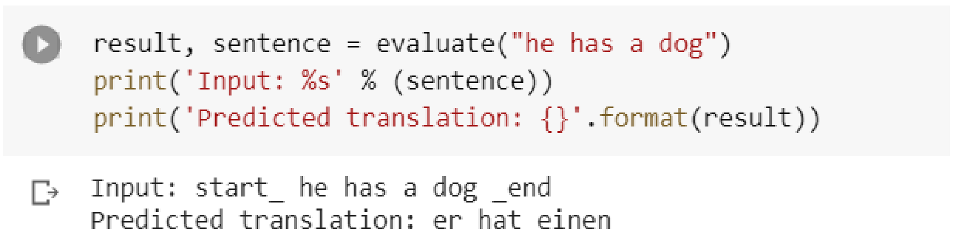
Below are the training and validation accuracy and loss curves for Model 4:





We can observe there is good change in the slope of the curve as the model is being trained. Hence, it can work much better with more data and better hyperparameters.

Below is the screenshot of the output of our final model:



```
result, sentence = evaluate("he has a dog")
print('Input: %s' % (sentence))
print('Predicted translation: {}'.format(result))
```

```
Input: start_ he has a dog _end
Predicted translation: er hat einen
```

Conclusion:

Continuous improvements in the implementations of the models are important. The idea behind the attention mechanism was to permit the decoder to utilize the most relevant parts of the input sequence in a flexible manner, by a weighted combination of all the encoded input vectors, with the most relevant vectors being attributed the highest weights. Hence, the model with the attention mechanism among the implemented models is performing better than the others by understanding the context behind the meaning of each input sentence.

Future scope of the project:

We can implement Neural Machine Translation by using the Transformer networks which can perform way better than any other models including the Google translate model.

For example, if the input data is a natural language sentence, the transformer does not have to process one word at a time. This allows for more parallelization than RNNs and therefore reduces training times.