Project Report

By : Gaurang Solanki (11812752)

Language Translator

Introduction :

Language Translation is the process of converting the word/text from one language into another in a way that is culturally and linguistically accurate. This project is based on Recurrent Neural Network using LSTM layers. Translation model which will be capable of translating sentences in English into French Language.

Dataset:

Project is focused on the parallel English-French Dataset.

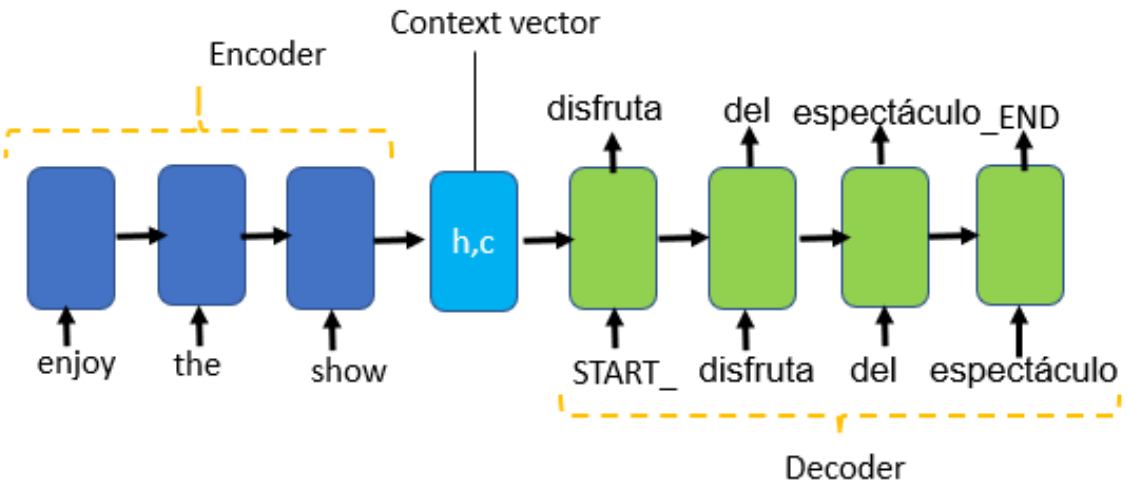
This is a prepared corpus of aligned English and French words/sentences recorded between 1996 to 2011.

Dataset contain around 150,000 English-French parallel sentences.

Due to the memory constraint I've trained the model on less than 10% of the data (10k sentences).

Proposed Architecture :

- I am using many to many model of seq2seq modelling to get output text,
- Also, using encoder decoder architecture for the model.



Techer Forcing:

The previous output is provided as input to the model to predict next step.

For example: sentence - “Je déteste repasser” in French which means “I hate to iron” in English.

Add start and end tokens. As initial start value is required in the teacher forcing and end token to make the model understand that the sentence is completed.

Therefore:

NPUT	REDICTION
Start]	e
Start], Je	éteste
Start], Je, déteste	epasser
Start], Je, déteste, repasser	End]

Procedure

- Encoder LSTM outputs: we only keep the state outputs of encoder LSTM layer as it will contain all the information about the input data.
- This states of encoder LSTM will be used to initialize the decoder LSTM. Also, the [start] token will be provided as first word as we are performing teacher forcing.
- The output of this decoder LSTM layer will be passed through Dense layer to predict the output word.

Workflow

1.Encoder side:

- input -> Encoder LSTM -> encoder states

2.Decoder side:

- encoder states + [start] -> Decoder LSTM -> word + decoder states
- decoder states + word -> Decoder LSTM -> word2 + decoder states 2
- word2 + decoder states 2 -> Decoder LSTM -> word3 + decoder states 3 ... so on
- The process stops when [end] token is predicted

Results and Experimental:

As I mentioned earlier, The model is trained on less than 10% of data.

When, Model was defined with:

Embedding dims = 64,

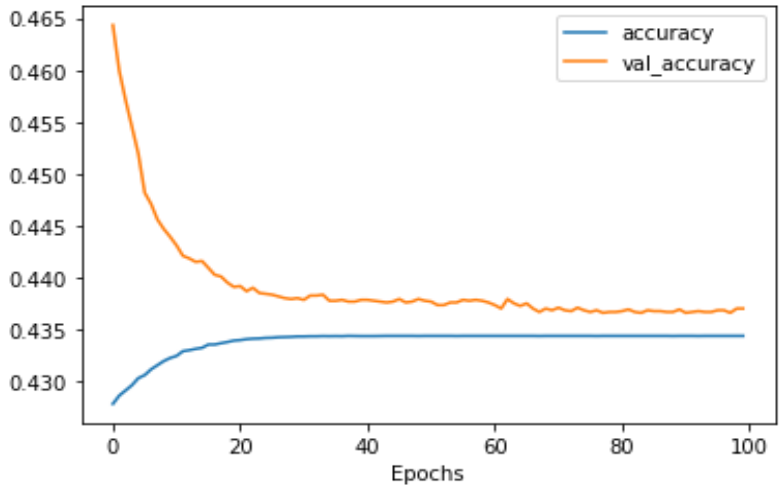
LSTM dims = 128,

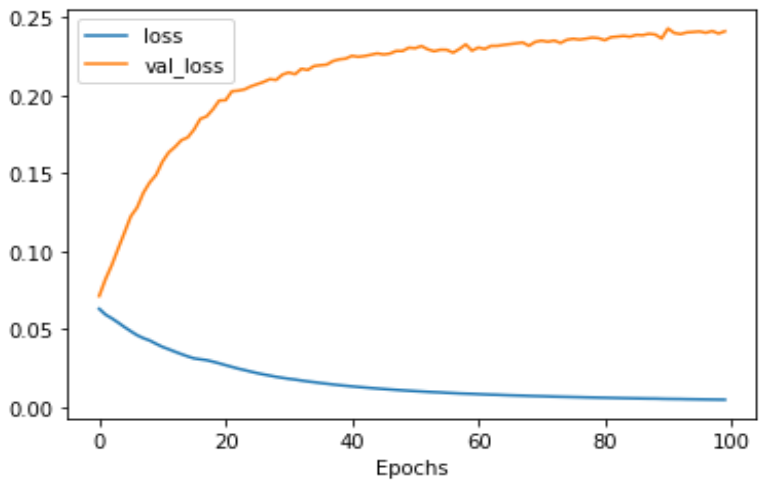
Validation set = 10% of 10000

Epochs = 100, Then,

Train =~ test = approx. 43%

Model performed better in validation set in early stage of epochs, then slowly started degrading.





Whereas, When :

Embedding dims = 120,

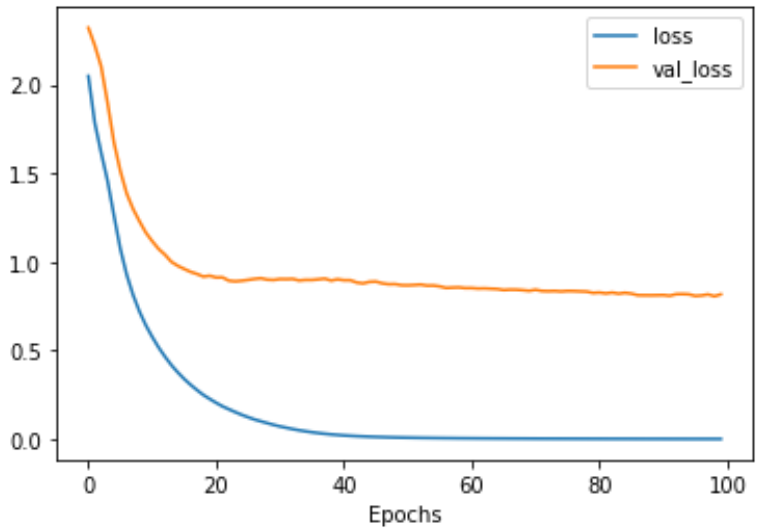
LSTM dims = 364,

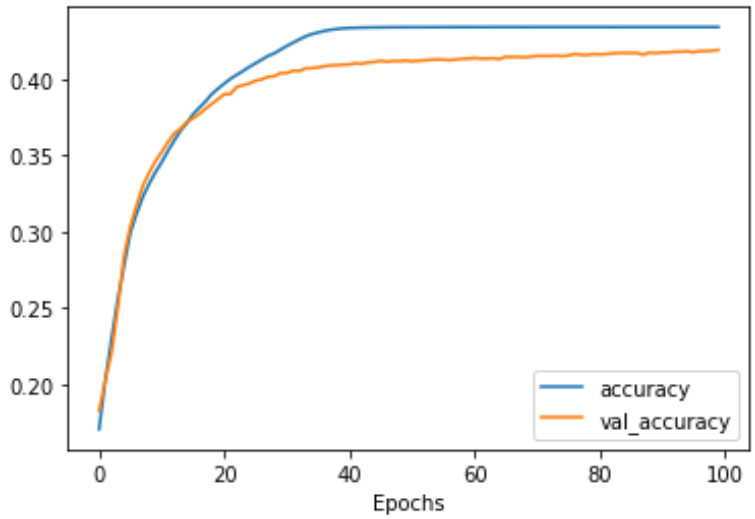
Validation set = 10% of 10000

Epochs = 100, Then,

Again, Train =~ test = approx. 43%

But This time Model performed poor in validation set in early stages of epochs, then slowly started improving.





Output Screenshot:

```
Input_sentence:  1234    i wrote it
Name: English, dtype: object
French sentence:   <OOV> tomates tomates opération tomates plierai plierai
----
Input_sentence:  4356    youre funny
Name: English, dtype: object
French sentence:   dimbécile  imprudents lemportes attardés mourut mourut
----
Input_sentence:  4565    do you get it
Name: English, dtype: object
French sentence:   cancer maimes  sagissaitil détestes décroche bien  cinq
----
Input_sentence:  34    got it
Name: English, dtype: object
French sentence:   <OOV> vienstu magnifique  connus incroyables incroyables
----
Input_sentence:  2345    is tom well
Name: English, dtype: object
French sentence:   <OOV> parvienstu connait peuton fraternité fraternité
----
Input_sentence:  7656    im interested
Name: English, dtype: object
French sentence:   <OOV> approche plierai plierai plierai plierai plierai
```

Conclusion and Future Scope:

Model didn’t performed very well in translating sentences, Many words were repeated multiple times.

However, One can always play with hyperparameters and check which ones make model better, Also,

Major fact was, due to the very less number of training data sentences model could not understand the relation between the words.

Model train-on with whole dataset could perform way better than 43% accuracy.

References:

https://www.programiz.com/python-programming/methods/string/maketrans

https://www.w3schools.com/python/ref_string_translate.asp

https://stackoverflow.com/questions/54176051/invalidargumenterror-indicesi-0-x-is-not-in-0-x-in-keras

https://towardsdatascience.com/intuitive-explanation-of-neural-machine-translation-129789e3c59f

https://github.com/somvirs57

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer