Homework 4 Machine Translator

**Step 1: Data**

1) How many data samples are included in the dataset?

Ans. 267186 samples are included in this dataset, but I’ve only used 30000 samples as using all was crashing my code.

2) Which problem will this dataset try to address?

Ans. This dataset will try to convert English sentences to German.

3) How many words does the dataset contains?

Ans. This dataset consists of approximately 801668 words.

4) Does the dataset have any missing information? E.g., missing features.

Ans. No

5) What is the label of this dataset?

Ans. German -English Conversion

6) How many percent of data will you use for training, validation and testing?

Ans. 80% training, 10% validation, 10% testing.

7) What kind of data pre-processing will you use for your training dataset?

Ans. I have used tokenization for preprocessing the dataset.

**Step 2: Model**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Loss | Validation Loss |
| LSTM 1 layer (encoder and decoder) | 0.9326 | 0.2307 | 0.34 |
| LSTM 2 layer (encoder and decoder) | 0.9335 | 0.2285 | 0.3351 |
| LSTM 3 layer (encoder and decoder) | 0.9121 | 0.3028 | 0.3859 |

**Step 3: Objective:**

I’ve used categorical\_crossentropy as my loss function.

**Step 4: Optimization**

RMSProp is my optimization function. I’ve chosen this because RMSprop's adaptive learning rate mechanism, which adjusts the learning rates individually for each parameter based on the exponential moving average of squared gradients, makes it suitable for optimizing neural networks. This feature is particularly valuable in tasks with varying sequence lengths, such as machine translation, as RMSprop handles sparse gradients effectively. The optimizer has demonstrated empirical success, showing robust performance across different neural network architectures and tasks. The choice of RMSprop aligns with its reputation for stability and adaptability, contributing to its selection for this specific machine translation model.

**Step 5: Model Selection**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | LR: 0.1 | LR:0.01 | LR: 0.001 |
| LSTM 1 layer | 0.9326 | 0.9390 | 0.9419 |
| LSTM 2 layer | 0.9335 | 0.9521 | 0.9658 |
| LSTM 3 layer | 0.9121 | 0.9501 | 0.9574 |

\*\*The table contains accuracies of all the models.

***Analysis:*** Although the best accuracy I’m getting is 0.9658 (with 2 layers of LSTM and learning rate of 0.001), the validation loss is also high which means the model is overfitting. **Thus, the best model is LSTM 2 layers with a learning rate of 0.1 with an accuracy of 0.9335**.

**Step 6: Model Performance**

A. Report the performance plot of models you tried.

Ans. I’ve attached all the training vs validation loss plots that I have in the screenshots folder.

B. Show a translate result (the translate could be incorrect):

Ans.

A close up of words

Description automatically generatedA white paper with black text

Description automatically generatedA close up of a line

Description automatically generated