Advanced NLP - Assignment 1

Name: Harshit Gupta

Roll No: 2020114017

Dataset Handling Approach:

1. Used `from torchtext.data.utils import get_tokenizer` for tokenization.
2. Made a custom function for cleaning the dataset.
3. Used `from torch.utils.data import DataLoader, Subset, Dataset` to construct the input and target sequences of appropriate sizes. Also, made the respective data loaders using the above library features.
4. Used `from torchtext.vocab import GloVe` to get 100 dimensions GloVe word embeddings.
5. Noticed that each sentence in the corpus led to another. Decide to concatenate all sentences as 1 string and based on that, design the Data Loaders.

Link to see models for LM1, LM2, LSTM, GRU: ANLP-A1-Model_Paths

---

Question 1

Normal Model Perplexity Scores:

1. Train: 1176.909
2. Test: 3304.6233

Question 2

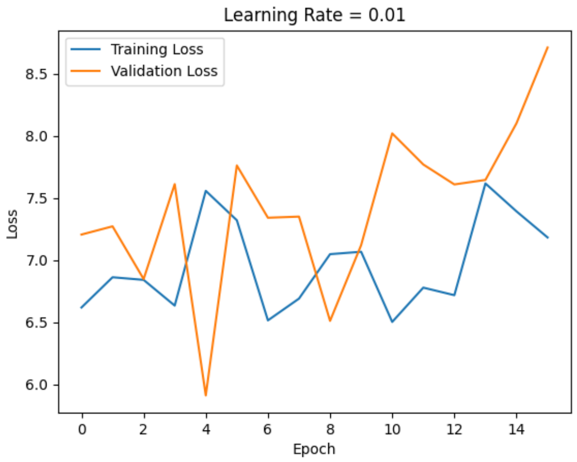
RNN Model Perplexity Scores:

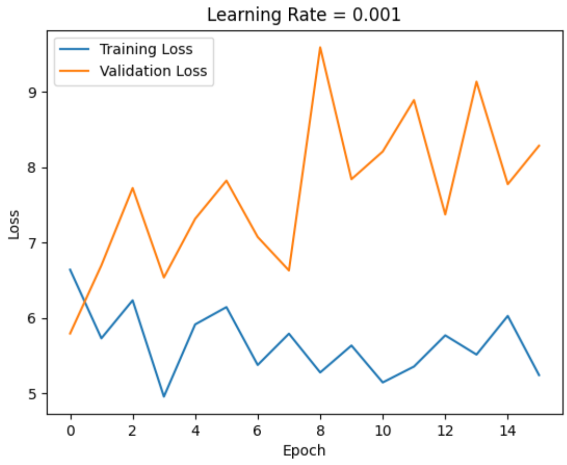
1. Train: 130.758
2. Test: 1653.8937

Visualizations and Analysis

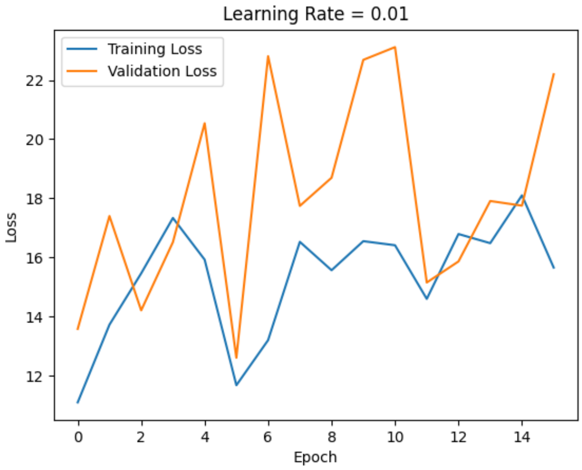
As we can see, both **Model 1** and **Model 2** show signs of training (loss decreasing) only when the learning rate is `lr = 0.001`. For `lr = 0.01` the model doesnt seem to be training as the loss isnt deacreasing at at certain intervals, it is increasing in fact.

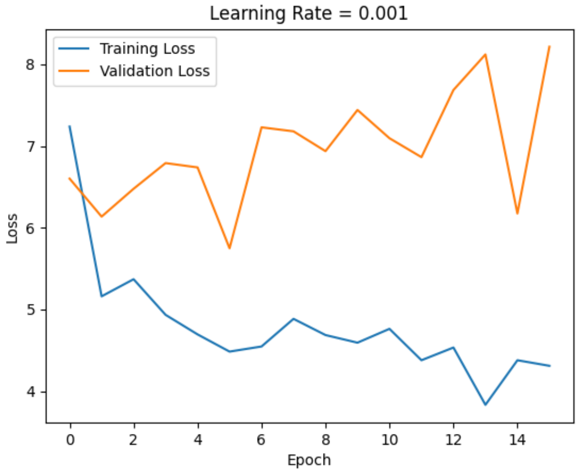
Model 1:





Model 2:

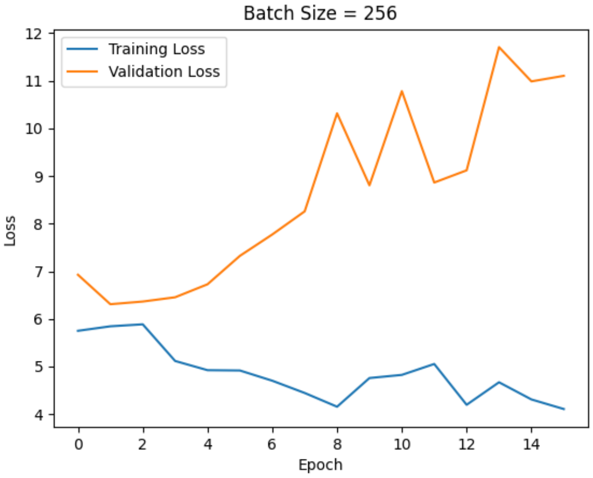




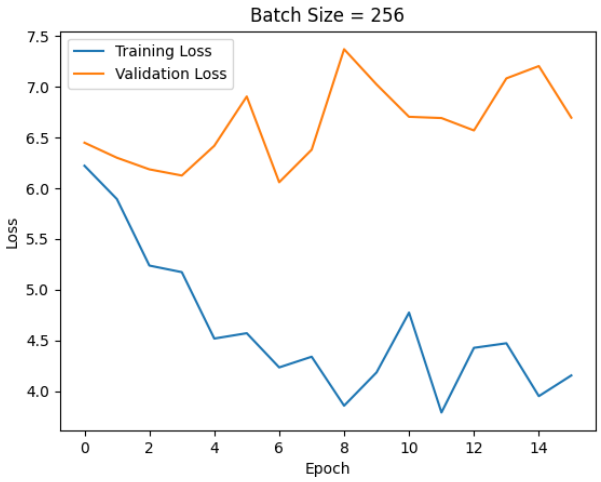
Now for the same `lr = 0.001`, if we change the batch size from `batch_size = 128` to `batch_size = 256`, we observe that:

Model 1:

While the training loss is lesser than then the training loss when the model is running a batch size of 128, the validation loss doesn't perform as well as when batch size is lesser.



Model 2: For this model we observe that, bigger the batch size, lesser is the noise in the gradients and so better is the gradient estimate. The validation loss and training loss are slightly more than when the batch size is 128.



This is in accordance with the rule : ***bigger batch size bigger learning rate***

Bonus:

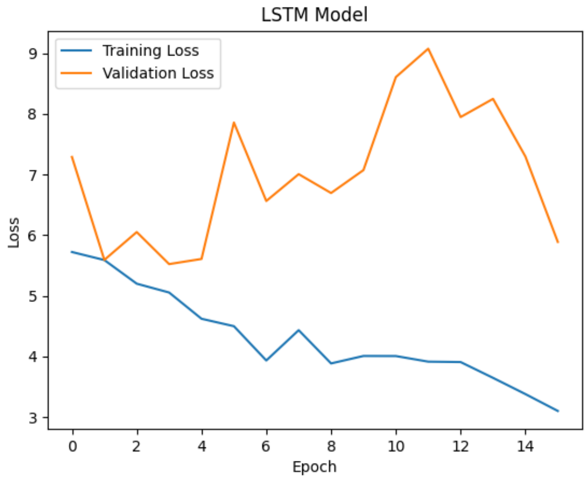
LSTMs and GRUs were implemented succesfully. The model construction comprises of two parts:

1. The `Cell` Layer, which is indivisual.
2. The `Model`, which joins X cells based on input size. Here, we have taken the `input size = 8` for both and hence $X = 8$.

LSTM:

Train Perplexity: 88.48461

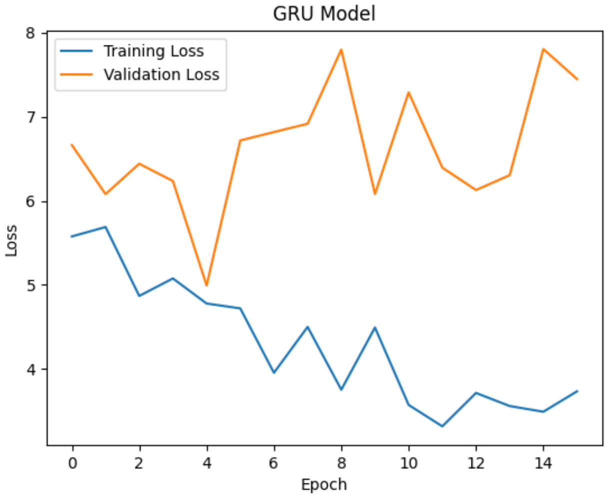
Test Perplexity: 1327.107



GRU:

Train Perplexity: 60.057373

Test Perplexity: 1468.2432



References Used For Model and Maths:

1. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

2. http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/