

DEEP LEARNING HW2 REPORT

Question 1:

Report your architecture, number of parameters, training time (real life time) and performance on the test set.

Architecture:

- Input Layer: Receives a song lyrics represented by integers.
- Embedding Layer: Using (nn.Embedding) we convert the input word indices into dense vectors of fixed size (embedding_dim).
- LSTM Layers: Two LSTM layers are used sequentially. The first LSTM layer takes the embedded input sequences and processes them. The second LSTM layer takes the output of the first LSTM layer and processes it further.
- Fully Connected Layer: Maps the LSTM output to the output dimension (output_dim) for artist classification

Number of parameters: 1845635 params

Training time: 10 minutes

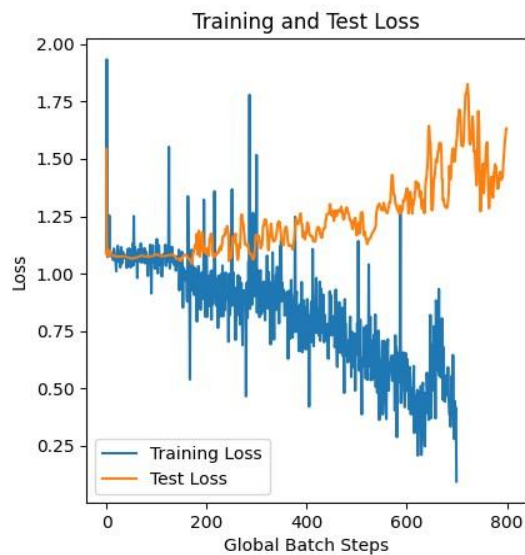
Performance on the test set: Accuracy – 44.186%

Question 2:

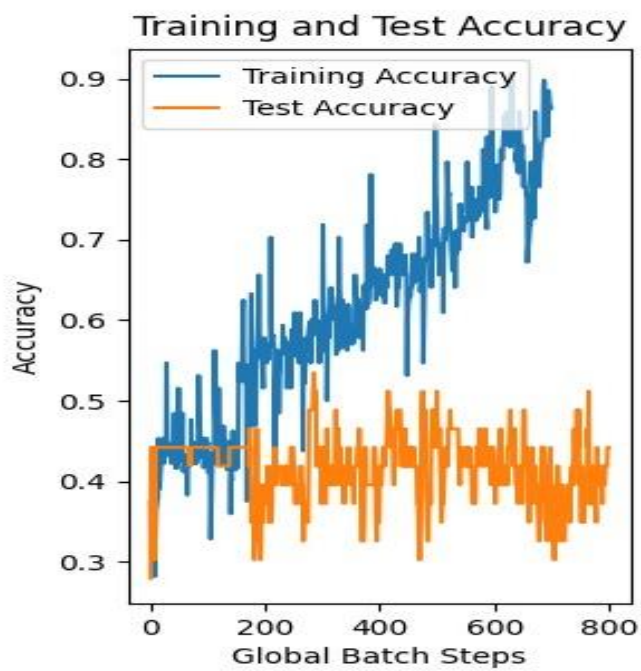
Add a plot showing train error, test error, and accuracy as a function of Global batch steps performed (not epochs!):

Answer:

Error:



Accuracy:



Question 3:

Explain shortly how your design deals with inputs of varied length:

Answer:

We use zero padding to deal with inputs of varied length. We take the longest string and we make the other strings in same length by adding zeros to the strings. By doing this step we make sure that our model can process all the data.

Question 4:

Explain shortly what LSTM you propagated forwards, and why.

Answer:

The LSTM model defined here is designed to process song lyrics for artist classification.

- **Input Embedding:** The input data, which is a sequence of words (represented by their indices in a vocabulary), is first embedded into a dense vector space using an embedding layer. This helps in capturing semantic similarities between words.
- **LSTM Layers:** The model consists of two LSTM layers organized in a sequence using (nn.ModuleList). Both LSTM layers have (batch_first = True), which means the input and output tensors are provided with batch size as the first dimension. The first LSTM layer takes the embedded input directly, while the second LSTM layer takes the output of the first LSTM layer. This stacked LSTM architecture helps capture complex patterns in the sequential data.
- **Fully Connected Layer:** After processing through the LSTM layers, the final hidden state (which captures information from the entire sequence) is extracted from the last time step of the last LSTM layer. This hidden state is then passed through a linear (fully connected) layer (self.fc) to produce the output. The output size will be determined by the (output_dim), which represents the number of classes or artists for classification.

Overall, the model sequentially processes the input lyrics through multiple LSTM layers to capture intricate patterns in the text data, and finally, utilizes a fully connected layer for classification into different artists.