Name: Mridu Student ID: 400547058

Assignment 2
SEP 775: Introduction to Computational Natural Language Processing

Topic:

Part 1: RNN-Based Text Generation

Part2: Seq2Seq Machine Translation with Attention

Submitted by:
Mridu (400547058)

Part 1: RNN-Based Text Generation

Code File name: RNN_vs_LSTM_Final.ipynb

Summary of the task performed in Part1:

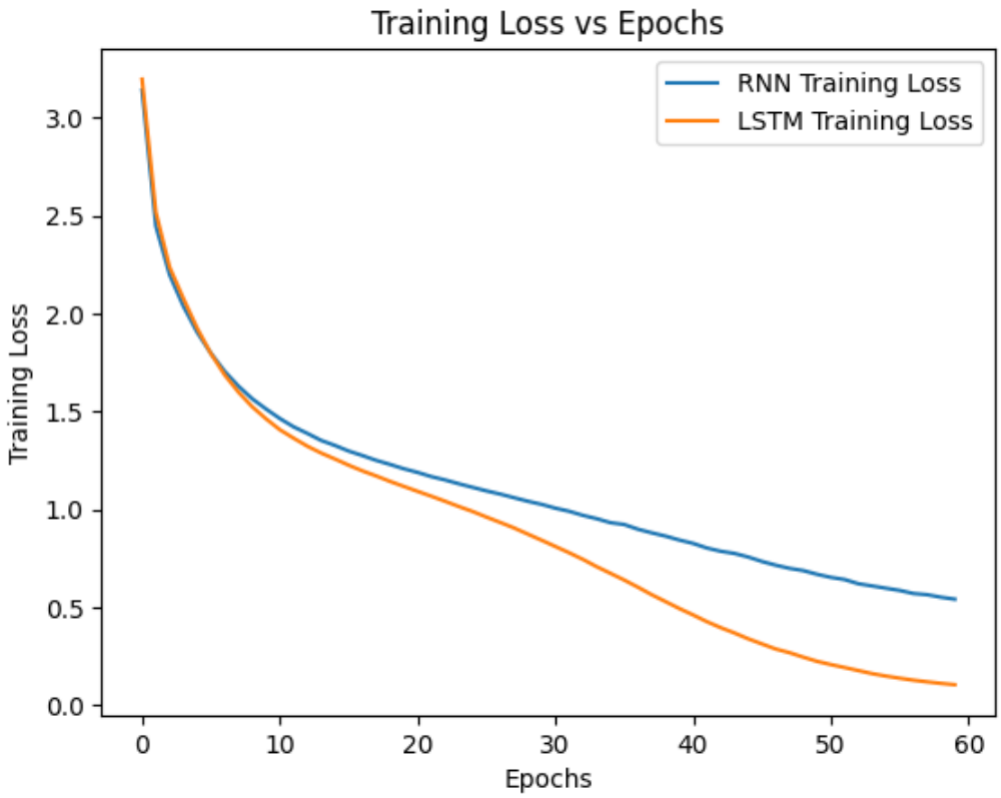
Text generation using Harry Potter text dataset from Kaggle

1. Data Preparation:
    a. Download a text file containing the first Harry Potter book.
    b. Read the text file and decode it into a string variable.
    c. Determine the length of the text (number of characters).
    d. Identify unique characters in the text.
2. String Conversion to Numerical Representation:
    a. Use StringLookup layers to convert characters to numerical IDs and vice versa.
    b. Split the text into tokens and convert them to numerical IDs.
    c. Define functions to convert numerical IDs back to text.
3. Sequence Generation:
    a. Divide the text into sequences of a specified length (seq_length).
    b. Create (input, target) tuples where the input is a sequence of characters and the target is the same sequence shifted by one character.
4. Model Building:
    a. Define a custom model class (Custom_Model) that encapsulates the embedding layer and either a SimpleRNN or LSTM layer, depending on the chosen model type (isLSTM).
    b. Instantiate RNN and LSTM models using the custom model class.
5. Model Training:
    a. Compile the RNN and LSTM models with the Adam optimizer and sparse categorical cross-entropy loss.
    b. Train both models on the dataset of (input, target) tuples for a specified number of epochs (EPOCHS). With experimentation, I found the optimum value of epochs to be 30.
6. Plotting Training Loss vs Epochs:

   a. Plot the training loss versus epochs for both the RNN and LSTM models using matplotlib.
7. Text Generation:
   a. Define a class (OneStep) to generate text character by character using the trained models.
   b. Generate text starting with the seed "Harry" for both the RNN and LSTM models and print the results.
8. Performance Evaluation:
   a. Measure the runtime for generating text using both the RNN and LSTM models.

Results and Analysis:

- Loss vs Epoch graph:



Interpretations:

1. Both models start with a high training loss, but as the number of epochs increases, the training loss for both models decreases, which indicates that both models are learning from the training data over time.
2. The LSTM model, represented by the orange line, shows a consistently lower training loss compared to the RNN model, which is the blue line.

This suggests that the LSTM is generally more effective at learning from this dataset, due to its ability to capture long-term dependencies in data, which RNNs can struggle with.

3. The rate of decrease in training loss is steep at the beginning for both models, which is typical as the model learns rapidly from a relatively unoptimized starting point. However, as the epochs increase, the rate of decrease in training loss slows down for both models, implying that they are starting to converge towards their minimum loss.
4. Both models appear to be improving their generalization on the training dataset throughout the training process.

- Text generated by RNN:

Harry as much himself togater, round the secon tell no sia. Harry couldn't help tried to eat become apart to a platform sank that the Dark Size. Half and out of her, piery this being inpilard, Harry thought, I'll not so much to explain in a towering out of the more. Hims. Uncle Vernon made for a weilform, he left.

Quirrell?" he asked muched. He looked him a beard, Potter, Vol-, sorry," said Harry, hardly anyone at the night, hope it lurking into his mouth was off.

Malfoy looked at the sign by the way, don't you?"

He sat knitying, to him Marcus Flint. You had managed to vike a boxy, remember: The Choce. "You,"

Harry went anything, but he wouldn't believe these way across the Sorcerer's Stone, a sudden than twelve over at the train homework on the floor, he lettern next to Malfoy he'd never even gotten more point.

Mangay Look somethin'," he said. "You'll see. You-" seems on, but it was his inside!"

"Jighten a moment, Harry -- what if they can't deliver them.

They'd tried to kiss

- Text generated by LSTM:

Harry knew what it must have cost him to try and find themselves -- the twelves things happenst to do moving row. A lot o' duffing -- that was possible, just in the mirror, and Slytherin were just nearly 'emortand, Harry said against the wall facing a mouthful of noise.

And sinvisible, too, because Harry didn't quiet me -- speedly trinks through the library. He had been hugged by a complete stranger. He also thought he had been caught twice. Ron opened deep ninebracley as the walls. The Great Hall was already full. It was always very slumped and since the Quidditch match. Misy parents was a train. Wood was getting dark and began to London the wizard

choss, Professor Dumbledore, who couldn't believe his eyes. At the start of that raised in the stands. His wrong tower and drive -- "How got Slytherin, Potter, this is Wood. “

Harry jumped at his lightning elcep silence. It was almost dark now, but Harry could see Quirrell, standing quite still as though force to look at the funishoge and

- Comparison of the text generated by RNN and LSTM in terms of grammar, understanding and complexity:
  1. Style and Coherence:
     a. The text generated by the basic RNN model lacks coherence and often jumps between disconnected ideas. The sentences are choppy, with abrupt transitions and grammatical errors. For example, "Harry couldn't help tried to eat become apart to a platform sank that the Dark Size" is grammatically incorrect and lacks clarity.
     b. In contrast, the text generated by the LSTM model demonstrates better coherence and a more consistent writing style. Sentences flow more naturally, and there is a clearer narrative structure. For instance, "Harry knew what it must have cost him to try and find themselves -- the twelves things happenst to do moving row" maintains coherence and follows a more logical progression of ideas.
  2. Grammar and Syntax:
     a. The text generated by the basic RNN model contains more grammatical errors and awkward phrasings. For example, "Harry went anything, but he wouldn't believe these way across the Sorcerer's Stone" contains grammatical mistakes and lacks clarity.
     b. The LSTM model produces text with improved grammar and syntax. While not perfect, it generally adheres to grammatical rules and produces sentences that are easier to understand.
  3. Contextual Understanding:
     a. The basic RNN model struggles to maintain context and often produces nonsensical phrases or incomplete thoughts. For example, "He sat knitying, to him Marcus Flint" lacks context and coherence.
     b. The LSTM model demonstrates a better understanding of context and produces text that is more relevant to the chosen dataset. It includes references to characters and events from the Harry Potter series, indicating a deeper understanding of the source material.

For example, “The Great Hall was already full.” Seems coherent and relevant.

4. Length and Complexity: The LSTM generated text are longer and more complex sentences, indicating a richer vocabulary and a better ability to capture nuances in language. Whereas the text generated from RNN model us shorter, simpler and has less variety of sentence structures and vocabulary.
    a. Example from the LSTM model: " Harry jumped at his lightning elcep silence. It was almost dark now, but Harry could see Quirrell, standing quite still as though force to look at the funishoge and" (Longer sentence with complex vocabulary)
    b. Example from the basic RNN model: " Quirrell?" he asked muched." (Shorter sentence with simpler vocabulary)
5. Runtime Efficiency: RNN model took approximately 2.50 seconds to generate the text, while the LSTM model took approximately 4.41 seconds. This is obvious since the LSTM models are more complex and require more computations.
6. Effect of changing number of epochs: Tried with different values of epoch and chose the optimum value for it to be 60. As seen from the above graph the loss kept on decreasing with increasing number of epochs and went to nearly 0.11 for the LSTM model at the 60th Epoch. This increased the runtime but in return provides a more stable model.

Part2: Seq2Seq Machine Translation with Attention

Code File name:

Seq2Seq_Final_with_Attention.ipynb
Seq2Seq_Final_without_Attention.ipynb (Attention Layer Commented out in Decoder call function)

Summary of the task performed in Part2:

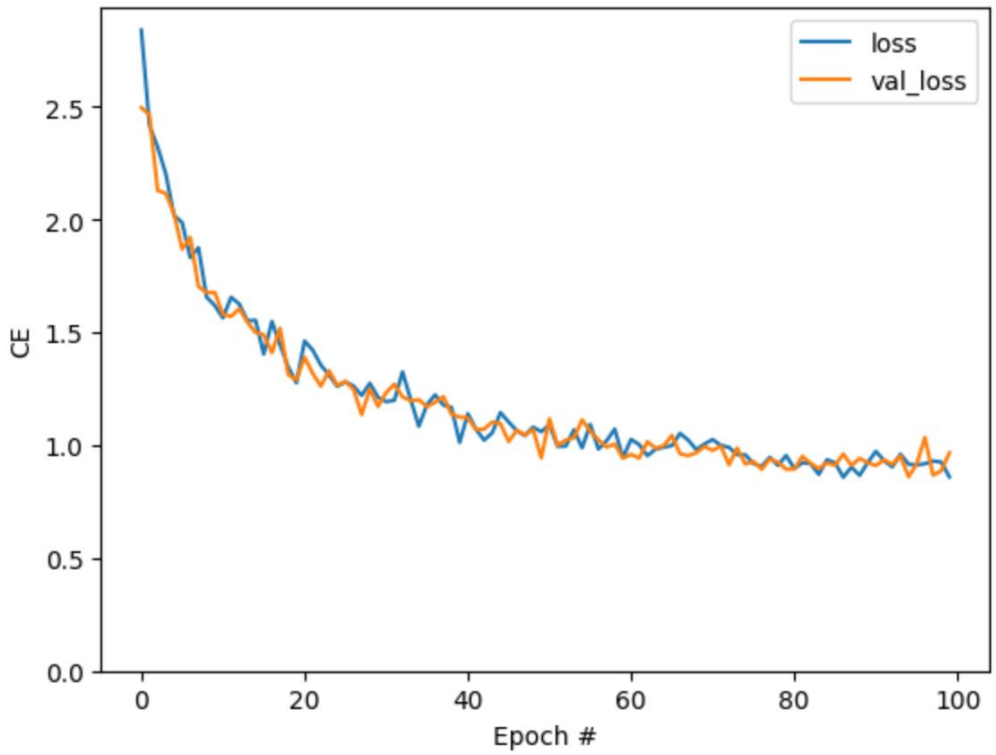
Translator for German to English

1. The dataset is read, and input-target pairs are created from it.
2. The dataset is shuffled and divided into batches for training and testing.
3. Text data is cleaned, tokenized, and vectorized for further processing.
4. The "rnn_units" parameter is a hyperparameter that determines the dimensionality of the output space (number of units) in the LSTM cells used in the model. For improving accuracy and stabilizing the model, I experiemented with "rnn_units" as 32, 64 and then finally 128. Increasing it any further was not giving any useful result. Therefore, I set the value to "128" as it was optimum to capture intricate patterns in the data and avoid overfitting.
5. Encoder using LSTM: An encoder is defined using an LSTM layer to encode input sequences.
6. Attention Layer: An attention layer is implemented to focus on relevant parts of the input sequence during decoding.
7. Decoder using LSTM: A decoder is defined using an LSTM layer to generate output sequences.
8. A translator model is defined, which consists of an encoder and a decoder. (The output of the Encoder is passed as the query for the attention layer, this is done inside the Decoder Layer.)
9. Custom loss and accuracy functions are defined for model evaluation.
10. The model is compiled with different optimizer, loss, and metrics to generate a stable and fine-tuned model using the training dataset.
11. Training metrics such as loss and accuracy are plotted over epochs to visualize model performance.
12. Translations are performed on sample input sentence, and BLEU score is calculated to evaluate translation quality.

Results and Analysis:

Visual Interpretation for Seq2Seq Machine Translation with Attention

- Loss vs Epoch graph:

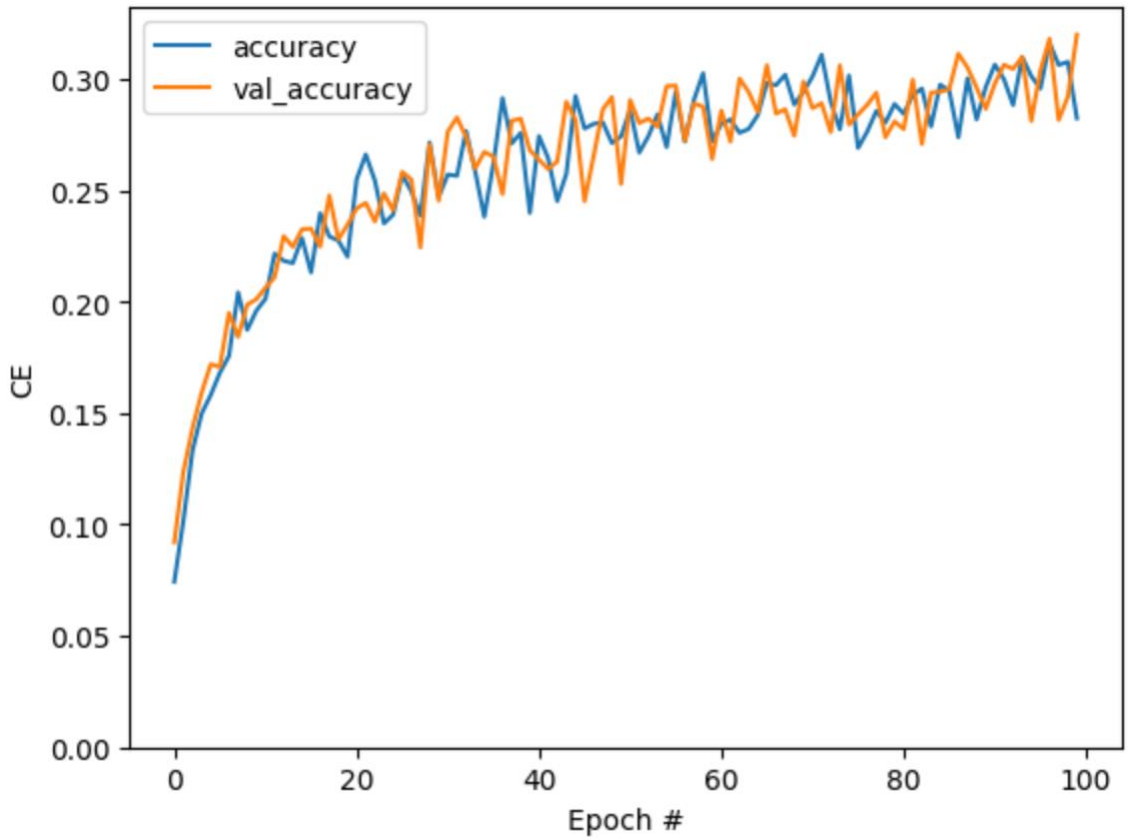


Interpretation:

1. Cross-Entropy vs Epoch for Training loss and Validation Loss
2. The decrease in cross-entropy loss suggests that the model's predicted probabilities are aligning better with the actual labels as training progresses.
3. Both the training loss and the validation loss decrease as the number of epochs increases, which indicates that the model is learning and improving its performance over time.
4. The training and validation loss values appear to converge as the epochs increase. This is a positive sign, indicating that the model is generalizing well.
5. The training and validation loss are close to each other throughout the training process. This suggests that the model is not overfitting.
6. Towards the end of the training (around epoch 60 onwards), the loss values show minimal changes, suggesting that the model has reached a

point of stability and additional training epochs are not leading to substantial improvements.

- Accuracy vs Epoch graph:



Interpretation:

1. Both the training and validation accuracy improve sharply at the beginning, which is typical as the model starts to learn from the data. The rapid improvement in the first few epochs suggests that the model is effectively capturing the patterns in the dataset.
2. After the initial sharp increase, the improvement in accuracy slows down for both training and validation. This indicates convergence.
3. The validation accuracy closely tracks the training accuracy throughout the training process. This is a good sign as it implies that the model is generalizing well to unseen data and not just memorizing the training data.
4. After the 60th Epoch, the model has reached a point where further training does not significantly improve performance, indicating that it may have reached its optimal state given the current architecture and data.

5. The validation accuracy does not diverge from the training accuracy, hence no overfitting.
6. Overall, the accuracy level is not too good. Hence, the model is not stable and sufficient enough.

- Result of the Seq2Seq Machine Translation with Attention

```
Predicted: i have to talk with this patience [UNK] school train?
Actual: I have the patience for this
BLEU Score for Reference: 0.2679946346035067
```

Qualitative Analysis:

1. The predicted translation maintains the general structure and meaning of the original sentence.
2. The phrase "I have to talk with this patience" captures the essence of the original sentence, indicating an understanding of the context.
3. The attention mechanism allows the model to focus on relevant parts of the input sequence while generating each output token, improving the quality of translations by capturing long-range dependencies and reducing information loss.
4. The inclusion of "[UNK] school train" suggests that the model encountered a word during translation that was not present in its vocabulary. This still needs to be handled in the model by improving on tokenization.
5. "[UNK] school train" likely represents a specific phrase that was not was not adequately learned by the model.

Quantitative Analysis:

1. The BLEU score of 0.267 indicates a reasonable level of similarity between the predicted and actual translations.
2. While not perfect, the translation with attention captures a significant portion of the reference translation, suggesting that the model produces meaningful and relevant outputs.

- Result of the Seq2Seq Machine Translation without Attention

```
Predicted: i m your patience. i m your patience. i m your patience. i m your patience.
Actual: I have the patience for this
BLEU Score for Reference: 0.00025318267950015955
```

Qualitative Analysis:

1. The predicted translation lacks coherence and does not convey the intended meaning of the original sentence.
2. The repetition of "I'm your patience" indicates a failure to capture the context or generate meaningful translations.
3. The translation appears to be a generic phrase rather than a specific response to the input sentence.

Quantitative Analysis:

1. The extremely low BLEU score of 0.00025 indicates poor similarity between the predicted and actual translations.
2. The translation without attention fails to capture the context and meaning of the original sentence, resulting in inaccurate and nonsensical outputs.