Kolapo Mogaji

Professor Mcmanus

ITAI 2376

February 26,2025

Reflection Journal on BERT Sentiment Analysis Lab

In this lab, I worked on fine-tuning a pre-trained BERT model to classify product reviews as positive or negative. The lab involved several critical steps, including data preprocessing, tokenization, model training, and evaluation. This reflection outlines my experiences, key learnings, and areas for potential improvement.

Key Learnings

1. **Data Preprocessing** – The first step was to clean the dataset by removing rows with missing values in the reviewText column. This ensured that only meaningful data was used for training, as BERT relies on textual input to learn patterns. Additionally, I limited the dataset to 2,000 samples to optimize training time, which is crucial when working with large transformer models.
2. **Tokenization** – I used the DistilBertTokenizerFast to convert the textual data into input representations suitable for BERT. Tokenization involved padding and truncation to maintain uniform input sizes, which is essential for efficient model performance.
3. **Fine-Tuning BERT** – A key step was loading the pre-trained DistilBERT model and freezing all but the final classification layer. This approach helped speed up training

while leveraging BERT’s pre-learned language representations. I also implemented training and validation loops using PyTorch’s DataLoader, calculating accuracy and loss at each epoch.

4. **Training Process** – I initially trained the model for 10 epochs, with a learning rate of 0.005. The training loss gradually decreased, and validation accuracy improved over time. However, I later experimented with 20 epochs to assess whether additional training would yield better results. While validation accuracy slightly improved, overfitting became a concern.
5. **Making Predictions** – After training, I tested the model on 15 new product reviews. Using the trained model, I obtained predictions and manually verified their correctness. Observing how the model processed text inputs into tokenized representations was insightful.

Challenges and Improvements

- **Overfitting** – Extending training to 20 epochs led to a slight drop in validation performance, indicating that the model might have started memorizing patterns rather than generalizing.
- **Hyperparameter Tuning** – Adjusting the learning rate and experimenting with different batch sizes could further refine the model’s performance.
- **Computational Resources** – Fine-tuning transformer models requires significant computational power. Using GPUs for training significantly reduced processing time.

This lab provided valuable hands-on experience in NLP model fine-tuning and deep learning workflows. Understanding how BERT processes text and improves with training deepened my appreciation for transfer learning. Moving forward, I aim to explore different transformer models and experiment with hyperparameter tuning to optimize performance further.