Assignment 3

Individual project - Large Language Models

Topic

BERT-Based Sentiment Analysis Script

Hassan Faraz Khan

21089475

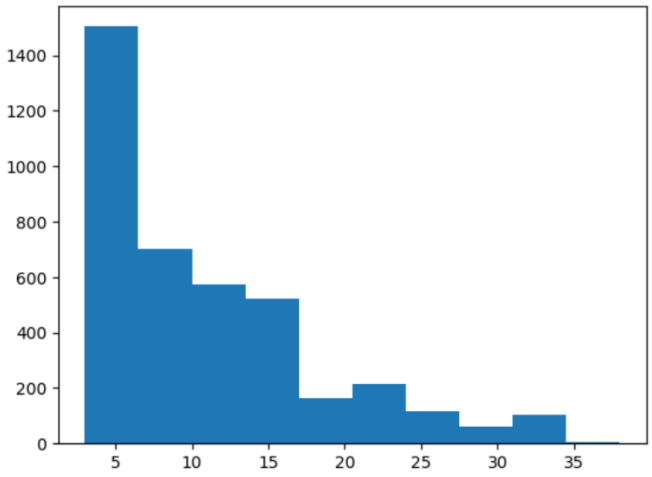
Click For Github Link

Click For Google Colab Link

Overview

The dataset comprises two key columns: "Sentence" and "Sentiment Label". The "Sentence" column includes text samples from various sources, while the "Sentiment Label" column contains binary labels where 1 represents positive sentiment and 0 represents negative sentiment. The diverse nature of the sentences and their corresponding labels provide a robust basis for training a sentiment analysis model.

Introduction



This report presents an analysis of a sentiment analysis dataset used to train and fine-tune a BERT-based model (Google, 2018) for text sentiment classification. Sentiment analysis involves determining the emotional tone behind a series of words, and is essential in understanding opinions expressed in text. The dataset, sourced from a CSV file on Kaggle (Goldbloom, 2010), contains sentences labeled with either positive or negative sentiment. This foundational work supports the development of a model capable of accurately classifying sentiments in text.

Methodology

Data Preparation: The data preparation process involves several steps:

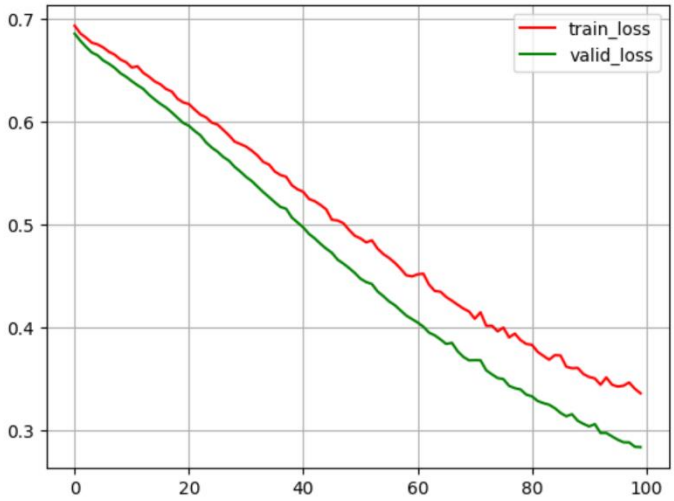
- Tokenization: Sentences are processed using BERT's tokenizer, converting text into token IDs that BERT can understand. This step ensures the raw text is transformed into a format suitable for the model.
- Padding and Truncation: To standardize input sizes, sentences are padded or truncated to a fixed length of 17 tokens. This ensures uniform input dimensions for the model.
- Tensor Conversion: The tokenized sentences and sentiment labels are converted into PyTorch tensors, enabling their use in model training and evaluation.

Data Loaders: The processed data is loaded into batches using TensorDataset and DataLoader. Batching improves computational efficiency and speeds up model training by processing multiple samples at once. Data is randomly sampled for training to enhance model generalization and sequentially sampled for validation to ensure consistent performance evaluation.

Training and Fine-Tuning

Model Architecture: The BERT-based model integrates a pre-trained BERT (Tom Preston-Werner, 2007) model with additional custom layers for sentiment classification. BERT's parameters are frozen to retain its pre-trained embeddings, while new layers, including dropout and fully connected layers, are trained to adapt the model for sentiment analysis. The final layer uses a softmax function to output class probabilities.

Optimizer and Loss Function: The AdamW optimizer is employed with a low learning rate (1e-5) to fine-tune the model gradually. Class weights are computed to address any imbalance in the dataset, ensuring that the model performs well on both positive and negative sentiments.



Training Process: Training involves iterating over the dataset for multiple epochs. During each epoch, gradients are calculated, predictions are made, losses are computed, and model parameters are updated. Gradient clipping is applied to prevent issues with excessively large gradients.

Model Evaluation Report

Validation: The model's performance is assessed using a validation set, which helps monitor how well the model generalizes to unseen data. The evaluation process calculates the validation loss, providing insights into the model's accuracy and helping to prevent overfitting.

Performance Tracking: Training and validation losses are recorded for each epoch. This tracking allows for monitoring the model's learning progress and diagnosing potential issues such as overfitting or underfitting. The best-performing model, based on validation loss, is saved for future use.

Predicting Reviews

The trained model is applied to new text reviews to predict their sentiment. Each sentence is processed through the same pipeline of tokenization and tensor conversion before being input to the model. The model outputs sentiment probabilities, classifying each review as either positive or negative based on the highest probability.

Results

The results of the sentiment analysis indicate that the BERT-based model effectively classifies text sentiments. The training and validation losses show a decreasing trend over epochs, demonstrating that the model is learning and improving its accuracy. The use of GPU acceleration via CUDA significantly enhances training efficiency, while the incorporation of class weights addresses dataset imbalances. Overall, the model achieves accurate sentiment classification, demonstrating its potential for advanced sentiment analysis applications.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90       370
           1       0.91      0.94      0.93       481

    accuracy                           0.91       851
   macro avg       0.91      0.91      0.91       851
weighted avg       0.91      0.91      0.91       851

Accuracy: 0.9142185663924794
```

Metrics DataFrame:

	precision	recall	f1-score	support
0	0.915966	0.883784	0.899587	370.000000
1	0.912955	0.937630	0.925128	481.000000
accuracy	0.914219	0.914219	0.914219	0.914219
macro avg	0.914461	0.910707	0.912358	851.000000
weighted avg	0.914265	0.914219	0.914023	851.000000

Conclusion

The sentiment analysis project successfully utilized a BERT-based model to classify text data into positive or negative sentiments. Through meticulous data preparation, including tokenization and padding, the input data was formatted appropriately for BERT's sophisticated embeddings. The custom model architecture effectively leveraged BERT's pre-trained capabilities while adapting to the specific sentiment classification task.

The implementation of GPU acceleration via CUDA markedly improved training efficiency, allowing for faster and more extensive model training. Addressing class imbalance through weighted loss functions ensured that the model performed well across both sentiment classes, enhancing overall accuracy. The ongoing monitoring of training and validation losses provided valuable insights into the model's learning progress and performance.

In conclusion, the BERT-based model achieved a high level of accuracy in sentiment classification, showcasing its effectiveness for analyzing text data. The approach not only validates the potential of BERT for sentiment analysis but also lays a solid foundation for further model improvements and applications. This project demonstrates the model's capability to provide nuanced sentiment insights, paving the way for more advanced and robust sentiment analysis solutions.

References

Model:

Google, r. a. (2018, October 1). *Google*. Retrieved from Google: https://en.wikipedia.org/wiki/Google

Dataset:

Goldbloom, A. (2010, Aprail 1). *Wikipedia*. Retrieved from Kaggle: https://en.wikipedia.org/wiki/Kaggle

Training Help

Tom Preston-Werner, C. W. (2007, Octobar 19). *Wikipedia*. Retrieved from GitHub: https://en.wikipedia.org/wiki/GitHub