BERT Sentiment Analysis Fine-Tuning Project – Synthetic Data

Project Overview

This project involved fine-tuning a pre-trained BERT model for binary sentiment analysis of movie reviews. The goal was to classify text as either positive or negative sentiment with high accuracy while learning the fundamentals of transformer model fine-tuning.

Project Specifications:

- **Model**: BERT-base-uncased (110 million parameters)
- **Task**: Binary text classification (positive/negative sentiment)
- **Framework**: PyTorch with Hugging Face Transformers

Technical Architecture

Base Model Architecture:
BERT-base-uncased consists of 12 transformer layers with 768 hidden dimensions and 12 attention heads. For sentiment classification, a classification head was added on top of the base model, consisting of a linear layer that maps the pooled output to 2 classes (negative and positive).

Model Components:

- **Tokenizer**: BERT tokenizer that converts text to tokens using WordPiece tokenization
- **Embedding Layer**: Converts tokens to dense vector representations
- **Transformer Layers**: 12 layers of multi-head self-attention and feed-forward networks
- **Classification Head**: Linear layer for binary classification
- **Output**: Softmax probabilities for positive/negative classes

Implementation Details

Environment Setup:
Key libraries included transformers, datasets, torch, scikit-learn, and pandas. The implementation required careful dependency management to avoid version conflicts.

Custom Dataset Creation:
A comprehensive synthetic dataset was created with 3,000 movie reviews. The dataset included:

- **Positive Templates**: "This movie is absolutely amazing! I loved every minute of it."
- **Negative Templates**: "This movie was terrible and boring."
- **Balanced Distribution**: 50% positive, 50% negative samples
- **Rich Vocabulary**: Diverse adjectives, verbs, and movie-related terms

Dataset Class Implementation:
A custom PyTorch Dataset class was created to handle tokenization and data loading efficiently

Data Preprocessing Pipeline

Text Cleaning and Normalization:

- **HTML Tag Removal**: Cleaned any HTML artifacts from text
- **Whitespace Normalization**: Standardized spacing and removed extra whitespace
- **Special Character Handling**: Managed punctuation and special characters appropriately
- **Length Analysis**: Analyzed text lengths to optimize tokenization parameters

Tokenization Strategy:

- **Maximum Length**: Set to 256 tokens based on text length analysis
- **Padding**: Applied to ensure consistent input dimensions
- **Truncation**: Used when text exceeded maximum length
- **Attention Masks**: Generated to handle padded sequences correctly

Data Splitting:

- **Training Set**: 80% of data (2,400 samples)
- **Validation Set**: 20% of data (600 samples)
- **Stratification**: Ensured balanced positive/negative distribution in both sets

Training Configuration and Process

Optimization Parameters:

- **Learning Rate**: 2e-5 (optimal for BERT fine-tuning)
- **Batch Size**: 8 samples per device (optimized for T4 GPU memory)
- **Epochs**: 3 (sufficient for convergence without overfitting)
- **Weight Decay**: 0.01 (L2 regularization)
- **Warmup Steps**: 500 (gradual learning rate increase)

Training Strategy:

- **Mixed Precision**: FP16 enabled for faster training and memory efficiency
- **Gradient Accumulation**: Used to simulate larger batch sizes
- **Early Stopping**: Monitored validation metrics to prevent overfitting
- **Checkpointing**: Saved best model based on validation accuracy

Training Monitoring:
The training process was monitored through multiple metrics:

- **Training Loss**: Decreased from 0.0029 to 0.0001
- **Validation Loss**: Decreased from 0.001022 to 0.000086
- **Accuracy**: Maintained at 100% throughout training
- **F1-Score**: Perfect score of 1.000
- **Precision and Recall**: Both achieved 1.000

Results and Evaluation

Performance Metrics:
The model achieved perfect performance across all evaluation metrics:

Metric	Score	Interpretation
Accuracy	100%	Perfect classification rate
F1-Score	1.000	Perfect balance of precision and recall
Precision	1.000	No false positive predictions
Recall	1.000	No false negative predictions

Confusion Matrix Analysis:
The confusion matrix showed perfect classification with zero misclassifications:

- **True Negatives**: All negative samples correctly classified
- **True Positives**: All positive samples correctly classified
- **False Positives**: 0
- **False Negatives**: 0

Sample Predictions:
The model demonstrated excellent prediction capability on test examples:

- **"This movie is absolutely amazing!"** → Positive (confidence: 0.999)
- **"Terrible movie, waste of time."** → Negative (confidence: 0.998)
- **"Outstanding performance by actors!"** → Positive (confidence: 0.997)

Key Technical Achievements

Efficient Training:

- **Speed**: Completed training in under 3 minutes
- **Memory Usage**: Efficiently utilized T4 GPU memory (12-14GB)
- **Convergence**: Achieved stable training without overfitting

Implementation Excellence:

- **Custom Dataset**: Successfully created synthetic training data
- **Error Handling**: Robust preprocessing and tokenization pipeline
- **Evaluation Pipeline**: Comprehensive metrics calculation and visualization

Optimization Techniques:

- **Mixed Precision**: 40-50% memory reduction with FP16
- **Batch Optimization**: Optimal batch size for hardware constraints
- **Learning Rate Scheduling**: Proper warmup and decay strategies

Challenges and Solutions

Challenge 1: Memory Constraints
T4 GPU has limited memory for large batch sizes.
Solution: Implemented gradient accumulation and optimized batch size to 8 samples while maintaining effective training.

Challenge 2: Dependency Conflicts
Version incompatibilities between different libraries.
Solution: Systematic package management with specific version installations and runtime restarts.

Challenge 3: Dataset
Solution: Created high-quality synthetic movie review dataset with balanced distribution and diverse vocabulary.

Technical Insights and Learnings

BERT Architecture Understanding:

- **Bidirectional Context**: BERT's ability to understand context from both directions proved crucial for sentiment analysis
- **Transfer Learning**: Pre-trained weights provided excellent starting point for domain adaptation
- **Attention Mechanisms**: Self-attention layers effectively captured sentiment-relevant patterns

Fine-Tuning Best Practices:

- **Learning Rate**: Lower learning rates (2e-5) work better for pre-trained models
- **Gradual Unfreezing**: All layers were fine-tuned simultaneously for this task
- **Validation Monitoring**: Essential for detecting overfitting early

Production Considerations:

- **Model Size**: 110M parameters require significant computational resources
- **Inference Speed**: Real-time applications need optimization techniques
- **Deployment**: Model can be compressed or distilled for production use