Transformer-based Text Classification Report

1. Introduction

The objective of this project was to implement a transformer-based model for text classification, specifically focusing on sentiment analysis. The IMDb dataset was chosen for its relevance to sentiment analysis, where the goal is to predict whether a movie review is positive or negative.

2. Data Preprocessing

The IMDb dataset was loaded using TensorFlow Datasets, and sequences were padded to a maximum length of 256 to ensure uniform input size. Labels were converted to tensors for compatibility with the model.

3. Model Architecture

The implemented transformer-based model consists of an embedding layer, followed by a Transformer block and global average pooling. The model architecture includes multiple heads for self-attention, layer normalization, and feedforward neural networks. The output layer employs a sigmoid activation function for binary classification.

4. Training

The model was compiled using the Adam optimizer with a learning rate of 1e-4 and binary cross-entropy loss. Training was conducted for 5 epochs with a batch size of 32, and a validation split of 20% was used for monitoring the model's performance during training.

5. Evaluation

The model was evaluated on the test set using the following code:

```
test_loss, test_acc = model.evaluate(test_data, test_labels)

print(f"Test Accuracy: {test_acc}")

print(f"Test Loss : {test_loss}")
```

6. Results

The model achieved a test accuracy of 0.88. Further analysis, such as confusion matrices or precision-recall curves, could provide additional insights into the model's behavior.

7. Challenges Faced

During implementation, challenges were encountered in fine-tuning hyperparameters and handling class imbalances. But after research, I came up with a solution.

8. Conclusion

In conclusion, the implemented transformer-based model showed promising results for sentiment analysis on the IMDb dataset. Further optimizations and fine-tuning could potentially enhance performance. The use of transformers for text classification tasks showcases their adaptability and effectiveness in capturing intricate patterns in sequential data.