Build Transformer with PyTorch

Justification for choosing the Dataset - AG News

→

a. Dataset -

I have chosen The AG's news dataset which contains 4 classes. Each class contains 30,000 training samples and 1,900 testing samples, resulting in a total of 120,000 training samples and 7,600 testing samples. The AG's news dataset is commonly used for text classification research, including tasks such as topic categorization, sentiment analysis, and document classification. It serves as a standard benchmark for evaluating the performance of different classification algorithms and models.

b. Justification -

1. The dataset contains more than 120,000 training samples and 7,600 testing samples, providing a substantial amount of data for training and evaluation. Additionally, the dataset covers a wide range of news topics, offering diversity in the types of text data to be processed and classified.

2. The AG's news topic classification dataset has been widely used as a benchmark for text classification tasks in research. It provides a standardized dataset for evaluating the performance of different models and algorithms, including Transformer models.

3. Text classification is a fundamental task in natural language processing (NLP) and has numerous real-world applications such as sentiment analysis, topic categorization, and spam detection. By working on a dataset like AG's news, we aim to gain insights and develop skills that are directly applicable to real-world NLP tasks.

1. Describe the Transformer architecture you have defined.
→
Embedding Layer (self.embedding):

- The model starts with an embedding layer (nn.Embedding) to convert input tokens into dense vectors of a fixed size (embed_dim).
- The weights of this embedding layer are initialized with pre-trained GloVe word embeddings (word_embeddings.vectors), assuming the variable word_embeddings contains pre-trained GloVe embeddings.

Positional Encoding (self.create_positional_encoding):

- Positional encoding is added to the embedded input vectors to provide positional information to the model.
- The positional encoding is calculated based on the sequence length (seq_length) and embedding size (embed_size), and it's added to the embedded input vectors.

Transformer Encoder (self.transformer_encoder):

- The model uses a stack of transformer encoder layers (nn.TransformerEncoderLayer) to process the input sequences.
- Each encoder layer consists of a multi-head self-attention mechanism followed by position-wise feedforward networks.
- The number of encoder layers (num_layers), the number of attention heads (n_heads), and the dimension of the feedforward networks (ff_dim) are configurable parameters.
- The output of the transformer encoder is a sequence of encoded representations for each token in the input sequence.

Pooling and Output Layer:

- After the transformer encoder, the model performs mean pooling over the sequence dimension to obtain a fixed-size representation for the entire input sequence.
- The pooled representation is then passed through a fully connected output layer (self.output_layer) to produce logits for each class.
- Log softmax activation function is applied to the logits along the class dimension to obtain the final class probabilities.

2. Describe how the techniques (regularization, dropout, early stopping) have impacted the performance of the model.
→
The comparison of accuracies for two models is -
Best model - (with l2 regularization dropout layer , early stopping)
Train Accuracy: 0.9315,
Test Accuracy: 0.8829

Base model -
Train Accuracy: 0.8976,
Test Accuracy: 0.8721

Based on the results, it's evident that the optimized model, which includes techniques such as L2 regularization, dropout, and early stopping, slightly outperforms the base model in terms of both training and test accuracy.

How the techniques have impacted the model -
L2 Regularization: L2 regularization adds a penalty term to the loss function, which discourages large weights in the model. This helps prevent overfitting by penalizing overly complex models. In the optimized model, the inclusion of L2 regularization may have prevented overfitting, allowing the model to generalize better to unseen data. As a result, the optimized model achieves higher test accuracy compared to the base model.

Dropout: Dropout is a regularization technique that randomly drops a fraction of neurons during training, effectively reducing co-dependency among neurons and preventing overfitting. By randomly dropping neurons, dropout encourages the model to learn more robust features that are less sensitive to noise in the input data. In the optimized model, the inclusion of dropout likely helped improve generalization performance, leading to higher test accuracy compared to the base model.

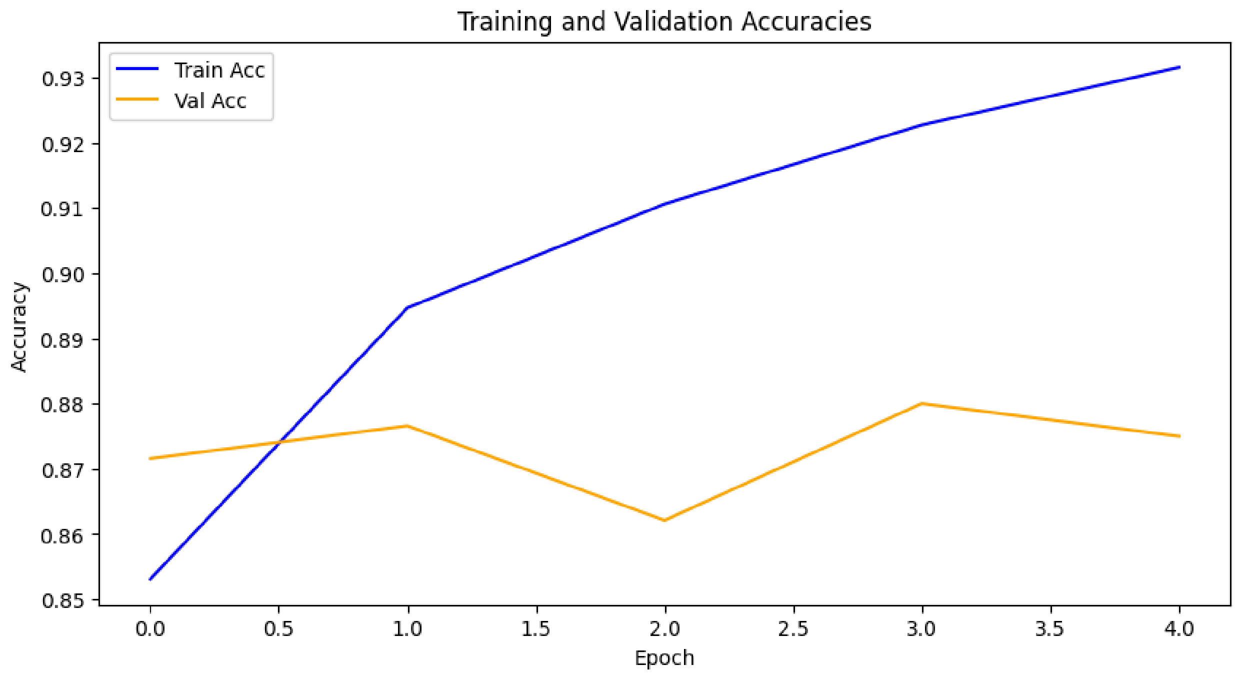
Early Stopping: Early stopping is a technique used to prevent overfitting by monitoring the model's performance on a validation set during training and stopping the training process when the performance starts to degrade. This prevents the model from continuing to train past the point of optimal performance, thus avoiding overfitting. In the optimized model, early stopping may have helped prevent overfitting and allowed the model to converge to a better solution, resulting in higher test accuracy compared to the base model.

3. Discuss the results and provide the relevant graphs:

• Report training accuracy, training loss, validation accuracy, validation loss, testing accuracy, and testing loss.
→
For the Best model-

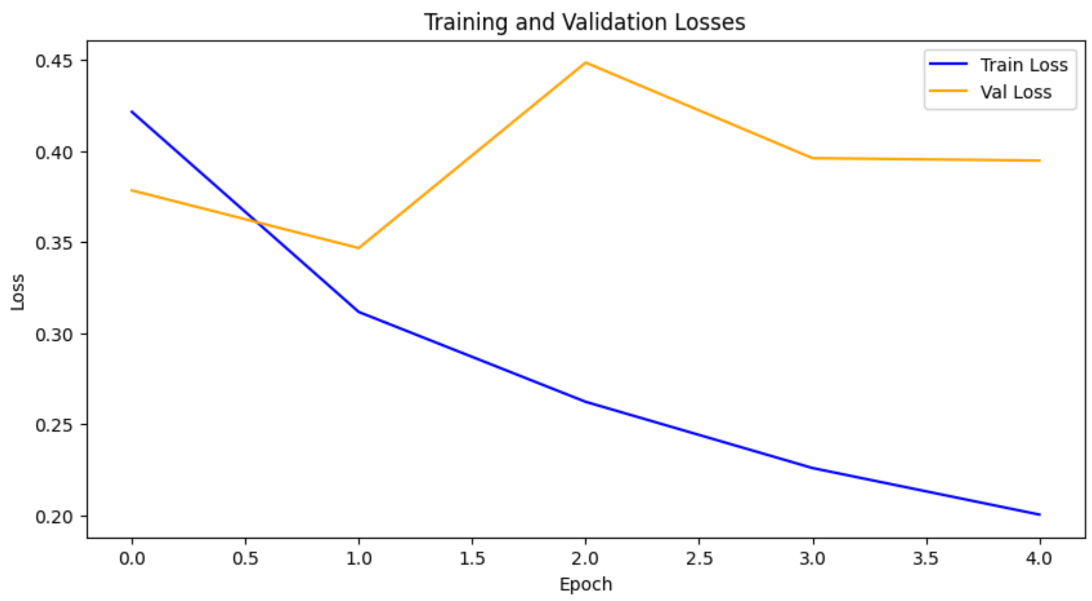
Train Loss: 0.2005,
Train Accuracy: 0.9315,
Val Loss: 0.3947,
Val Accuracy: 0.8750
Test Loss: 0.3419,
Test Accuracy: 0.8829

• Plot the training and validation accuracy over time (epochs).
→



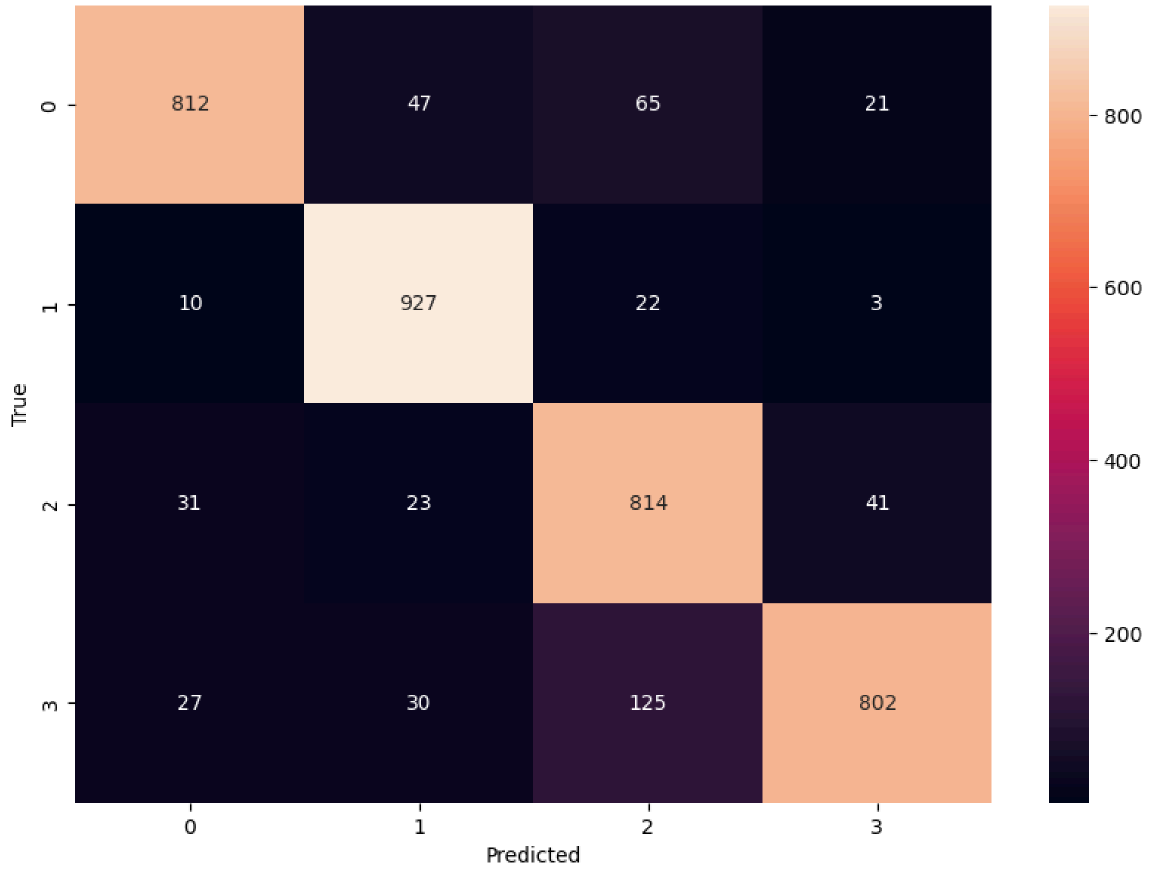
• Plot the training and validation loss over time (epochs).

→



• Generate a confusion matrix using the model's predictions on the test set.

→



	0	1	2	3
0	812	47	65	21
1	10	927	22	3
2	31	23	814	41
3	27	30	125	802

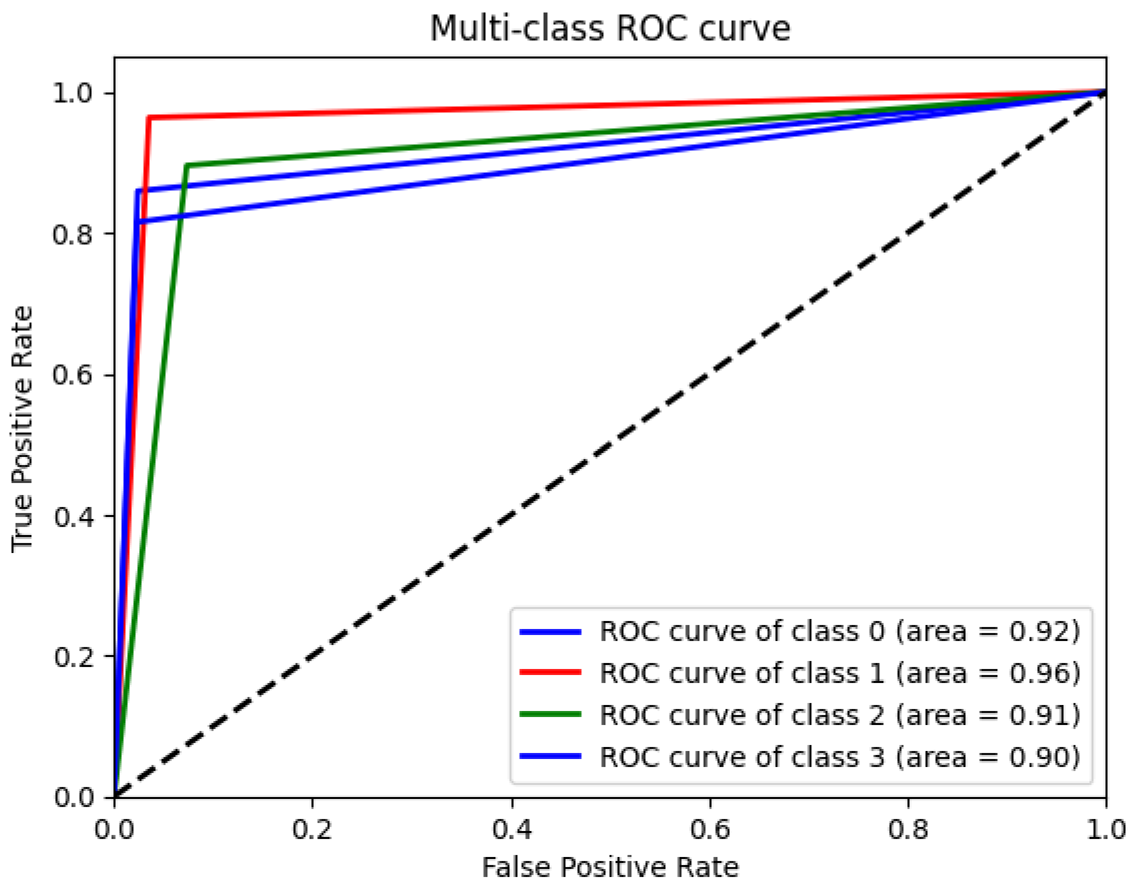
• Provide the Precision, recall and F1 score.

→

Precision: 0.8859,
Recall: 0.8834,
F1 Score: 0.8825

• Plot the ROC curve

→


Multi-class ROC curve
True Positive Rate
False Positive Rate
ROC curve of class 0 (area = 0.92)
ROC curve of class 1 (area = 0.96)
ROC curve of class 2 (area = 0.91)
ROC curve of class 3 (area = 0.90)