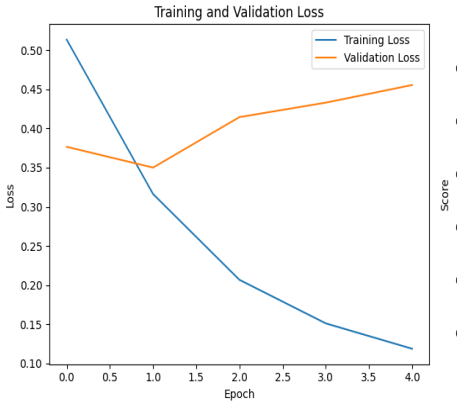
1. Give an example for encoder-only Transformer, decoder-only Transformer, and encoder-decoder Transformer. [2 points]

Whenever, you have a many-to-one type of task like sentiment classification using BERT, encoder-only Transformers are preferred. They take raw input and contextualize it, and you can then make changes in the outer layer of the FFNN to classify the sentence into your choice of category. A decoder-only Transformer example is GPT, which is designed for generating text based on a given prompt. An example of an encoder-decoder Transformer is T5 or BART, commonly used for tasks like translation and summarization.

2. Practical:

To train a Transformer on a larger dataset, I sampled 5320 examples from the IMDb dataset and split them into training (3920), validation (700), and test (700) sets. I used the bert-base-uncased model with a batch size of 64 and trained for 5 epochs. Early stopping activated after epoch 5 based on validation loss. The best model (from epoch 2) achieved a test accuracy of **88.4%**, with an F1 score of **0.8851**. I chose a smaller sample size to reduce memory load and training time, and the training remained stable throughout. This setup balanced performance and resource efficiency.

Sadly, the model is overfitting on the given data as seen from the below graphs. This is even after increasing the dataset size.




```
Example 5: Although I smiled, deep down I felt completely empty.
Predicted sentiment: Negative (confidence: 0.8166)

Example 6: I didn't expect it, but I'm incredibly proud of myself.
Predicted sentiment: Positive (confidence: 0.9444)

Example 7: My hands were shaking as I waited for the results.
Predicted sentiment: Negative (confidence: 0.7971)

Example 8: I'm not unhappy, but I wouldn't say I'm thrilled either.
Predicted sentiment: Negative (confidence: 0.5106)

Example 9: I can't deny that I didn't feel a bit of joy.
Predicted sentiment: Positive (confidence: 0.5639)

Example 10: She wasn't exactly angry, yet something about her silence screamed frustration.
Predicted sentiment: Negative (confidence: 0.8159)
```

In a second iteration, I reduced the max_length from 128 to 64 to speed up training and prevent overfitting, lowered the Adam optimizer learning rate to 1e-5 for smoother convergence, and increased the early stopping patience from 3 to 6 to allow more training time. These changes led to more stable validation loss and improved generalization. The final test accuracy was **86.1%** with balanced precision and recall, as supported by the confusion matrix (TP=303, TN=300). The model showed slightly better recall and reduced overfitting compared to the first run.