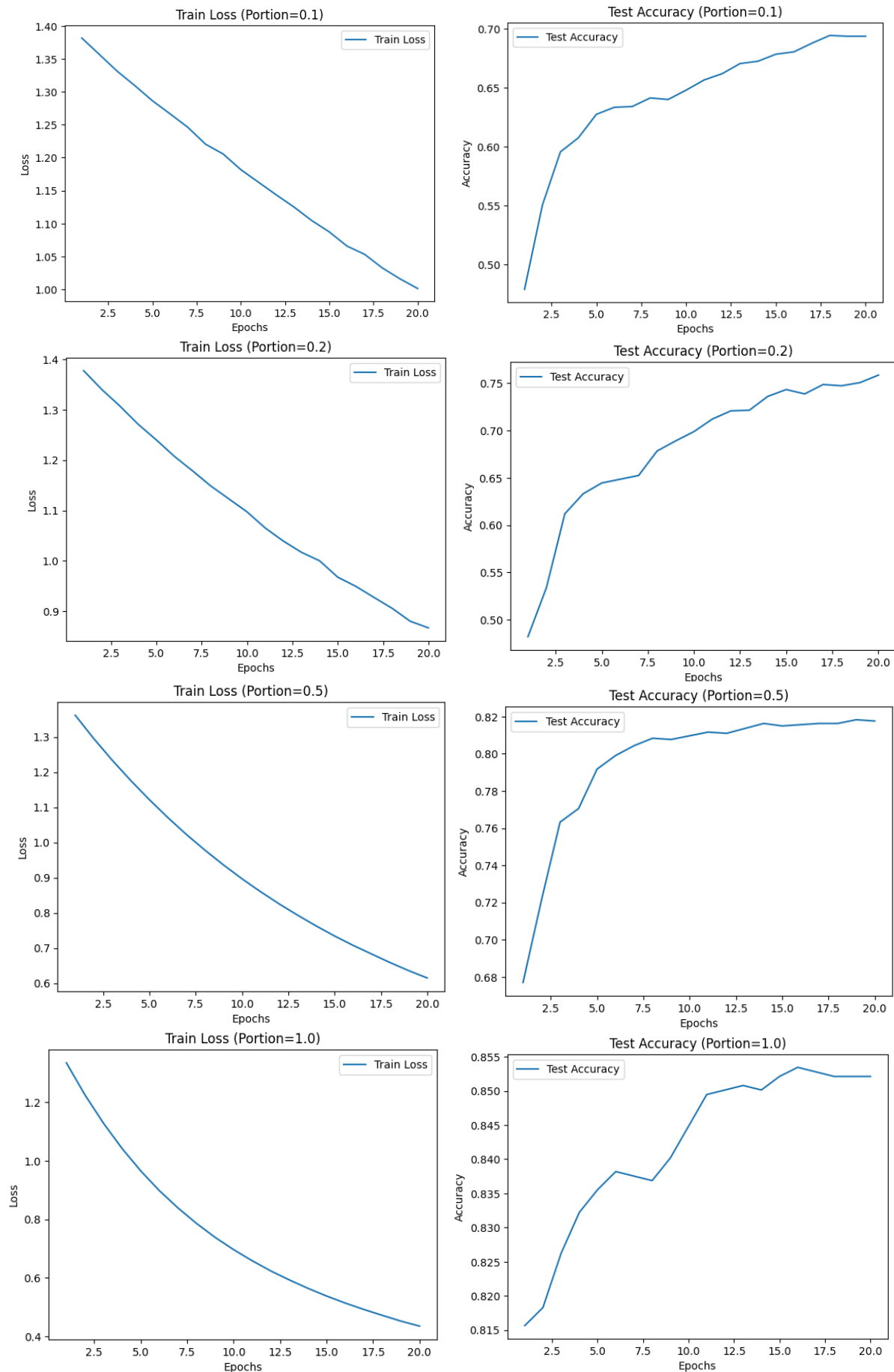


---

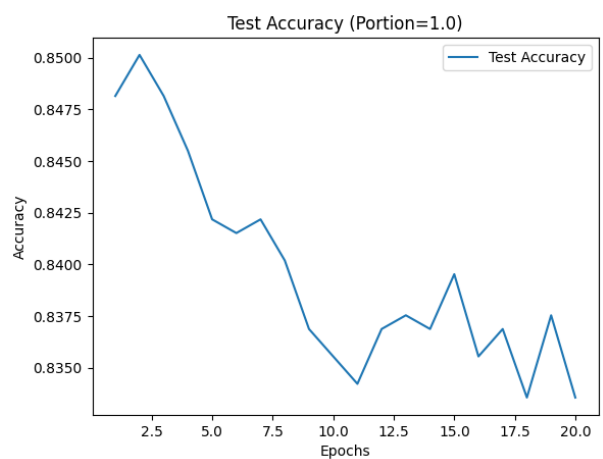
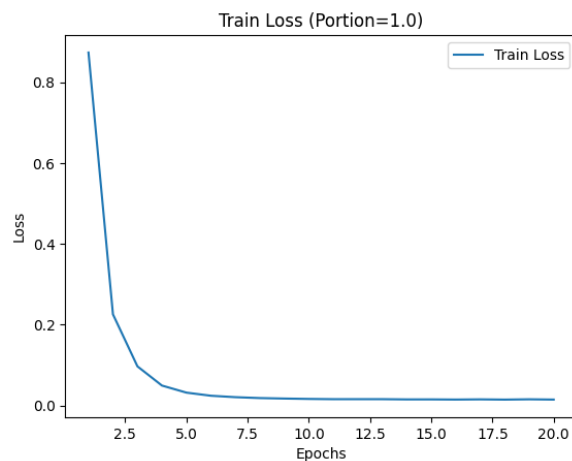
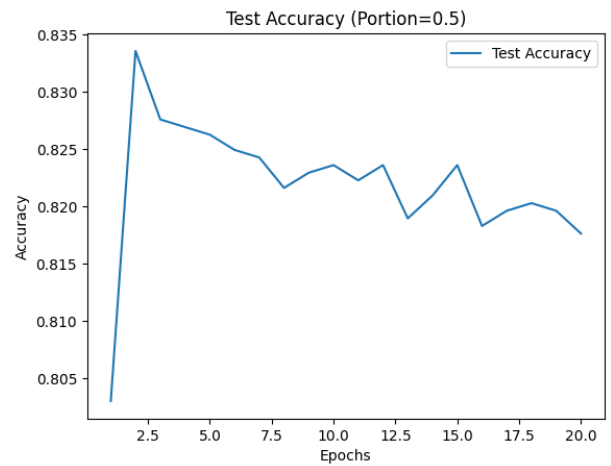
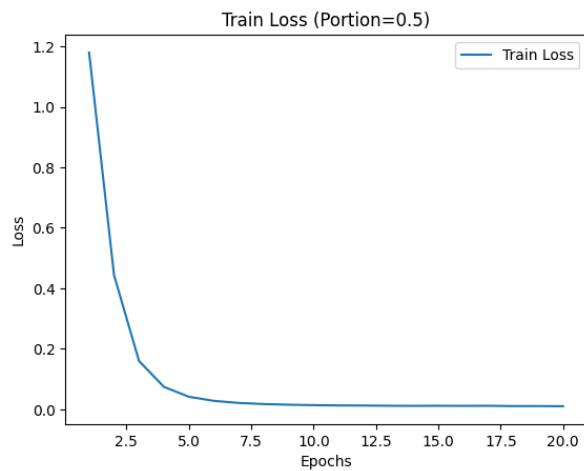
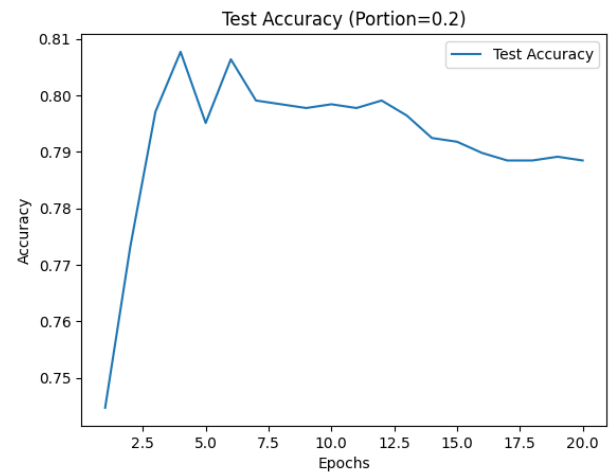
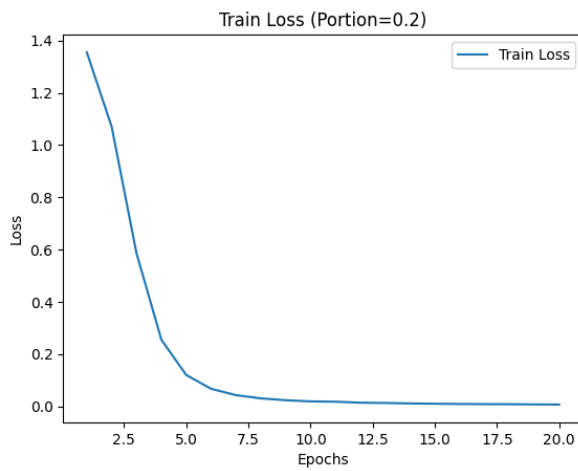
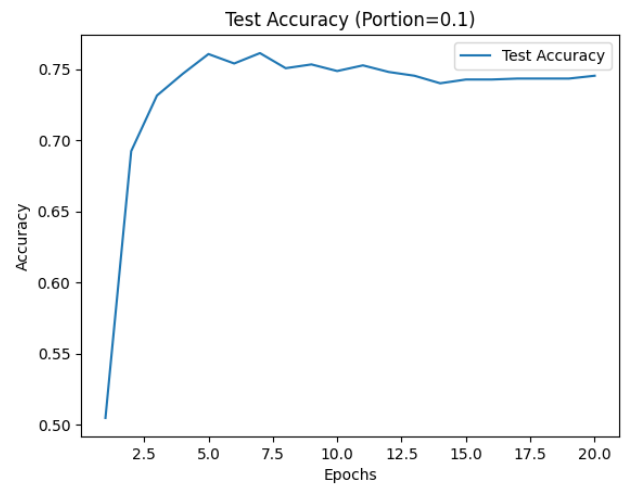
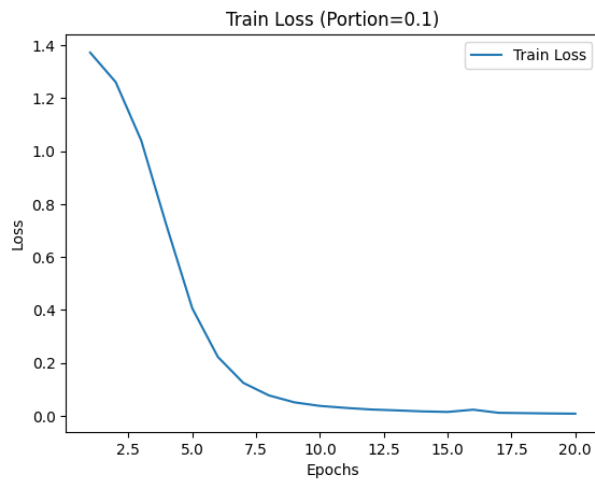
## Natural Language Processing – Exercise 2

---

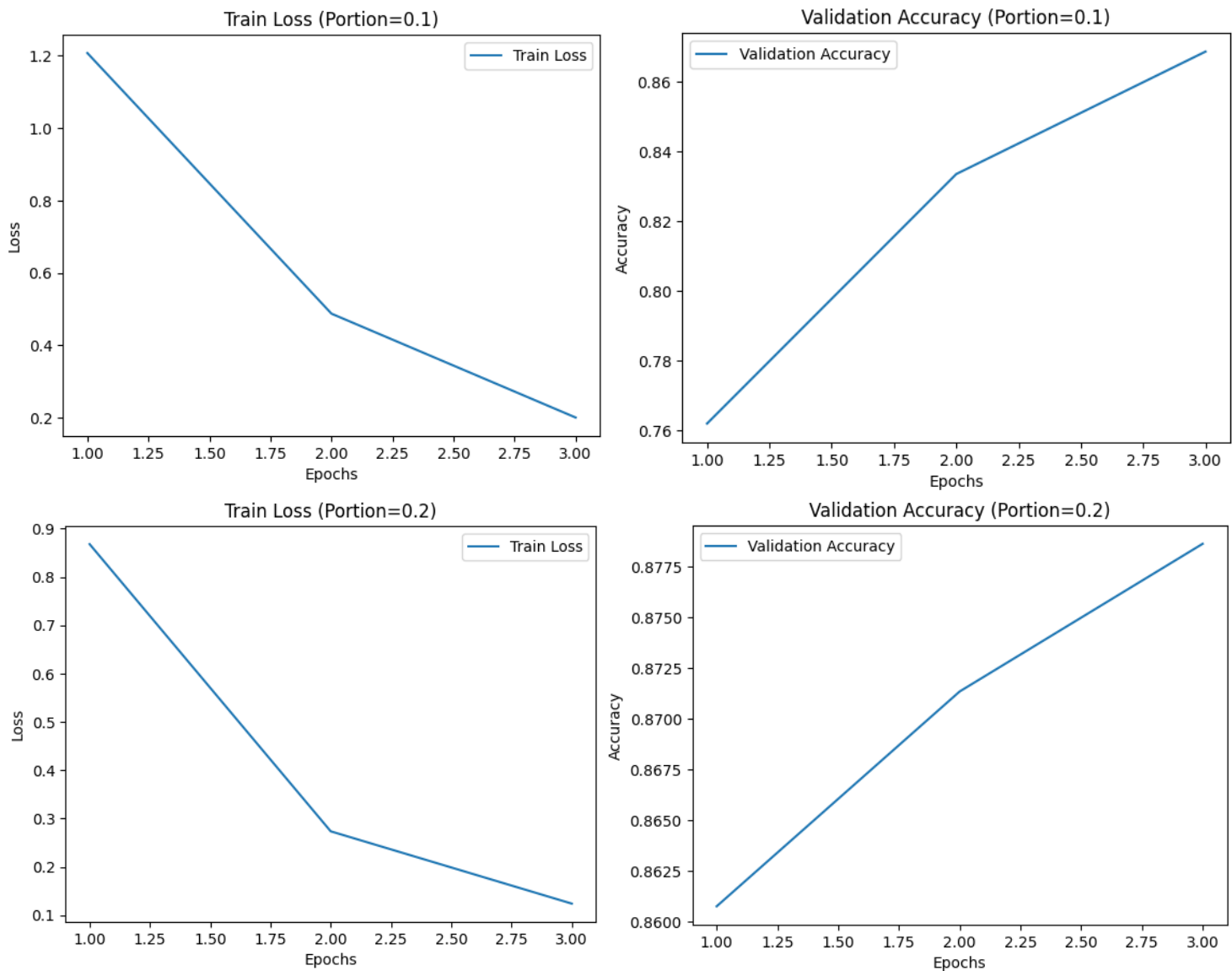
### Q1 Plots:



## Q2 Plots:



### Q3 Plots:



### Q4 a. Which model had the highest accuracy?

The transformer model from Q3 achieved the highest accuracy score, as shown in the plots above.

This outcome aligns with my expectations, as the transformer has been trained on extensive datasets and fine-tuned for the task, allowing it to leverage rich contextual representations of text.

Q4 b. Which model was the most sensitive to the size of the training set?

The model most sensitive to the size of the training set is the single-layer perceptron from Q1.

As shown in the plots, increasing the training set size in Model 1 led to improved accuracy, with each increment resulting in a gain of approximately 0.035 to 0.07.

In comparison, Model 2 saw an accuracy increase of between 0.02 and 0.06, while Model 3's accuracy improved by approximately 0.02 with a larger training set.

This does not account for the overfitting observed in Model 2, which is a result of it being a 'stronger' model than Model 1.

Q4 c. Extract (in your code) and report the number of trainable parameters in each model. Do additional parameters help? Explain.

Results for number of trainable parameters extracted from the code:

- Single-layer perceptron = 8,004 parameters.
- Multi-layer perceptron = 1,002,504 parameters.
- Transformer = 82,121,476 parameters.

Comparing the three models, it is evident that increasing the number of trainable parameters does not necessarily lead to better accuracy.

The Multi-layer Perceptron, with many more parameters than the Single-layer Perceptron, suffered from overfitting, achieving high performance on the training data but failing to generalize well to unseen test data.

However, the Transformer, with significantly more parameters, achieved the best performance.

This success was due to careful mitigation of overfitting through regularization and using only 3 epochs on 20% of the data.

This result highlights that while more parameters can make a model stronger, achieving this requires thoughtful training strategies to ensure effective generalization.