Artificial Intelligence 2, Project 4

Georgios Nikolaou

April 12, 2023

Model & Tuning

The neural network consists of a pretrained Bert model and an one layered feed forward network with dropout and a sigmoid activation function.
The model's simplicity is deceptive, since it uses a complex pretrained network to achieve the excellent result it does.

Hyperparameters:

- Optimizer: $AdamW$
- Scheduler: $get\_linear\_schedule\_with\_warmup$
- Batch Size: 32
- Epochs: $[4, 8]$
- Clipping: 1
- Learning Rate: $[1e-5, 1e-6]$
- Dropout: $[0.4, 0.6]$

Notes:

- The batch size was set to 32, which was the largest size the system could handle without running out of GPU RAM. (Max sequence length also plays a role here.)
- A scheduler was implemented because even when using one, we observed overfitting. We started with a learning rate of $1e-5$ (and observed overfitting) and then tuned it to achieve a good learning curve and good results (around 91%). When we set the learning rate to $1e-6$, the model had a hard time learning and plateaued quickly, so more epochs would not have helped either.
- The number of epochs varied based on the learning rate.
- We found that the difference in dropout rates was negligible. The learning rate was by far the most important factor to tune.
- Clip was set to 1 to help with overfitting.

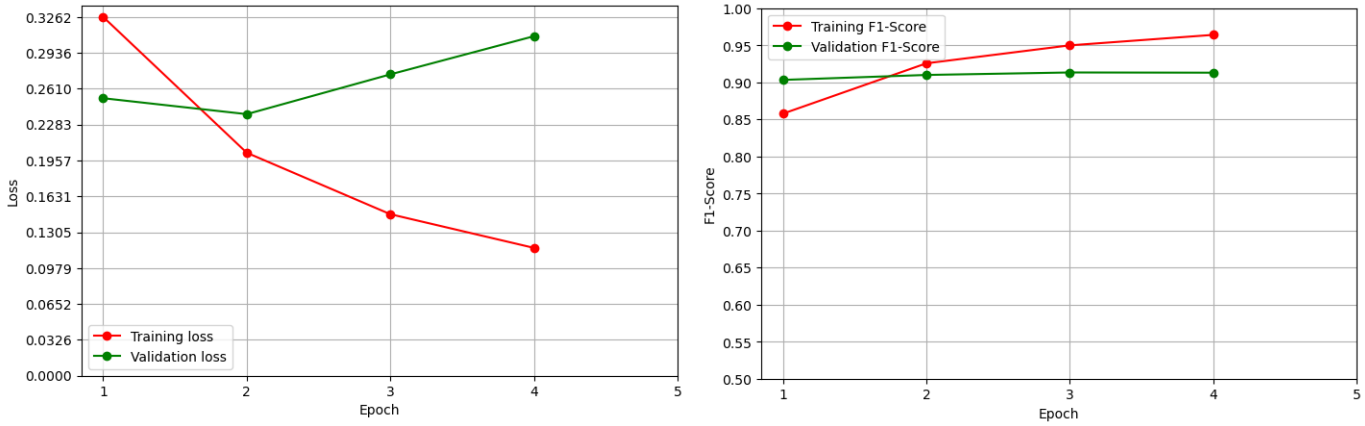
Results:

- We started with values of $1e-5$ for the learning rate, 0.4 for the dropout rate, and 4 for the number of epochs. However, we noticed significant overfitting after the second epoch and tried to address it.
- By reducing the learning rate to $1e-6$ and increasing dropout to 0.6 we achieved no overfitting but the metrics where rather underwhelming. Increasing the learning rate to $5e-6$ still led to overfitting.
- We experimented with different batch sizes and lower learning rates, but did not achieve good results.
- By increasing the number of epochs and slightly increasing the learning rate to $2e-6$, we achieved the best model in terms of both the learning curve and the performance metrics.
- We did not conduct extensive experiments, as this assignment is GPU-intensive, and on both Google Colab and Kaggle, we would run out of available time and would not be able to complete Part 2.

Results

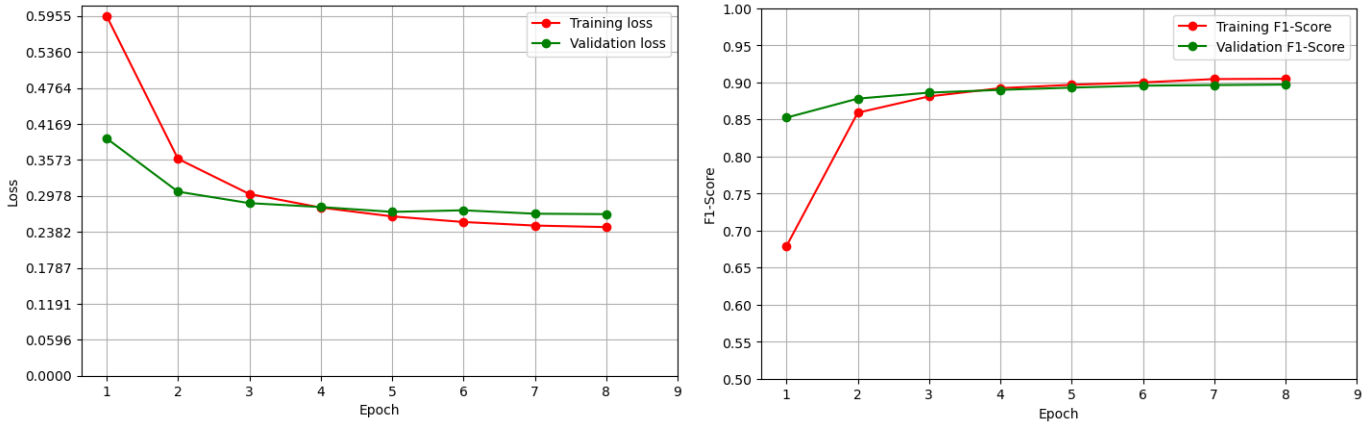
	Accuracy	Precision	Recall	F1-Score
B01	91.53%	91.66%	91.36%	91.51%
B06	88.79%	89.96%	89.56%	89.76%
B08	90.46%	90.36%	90.55%	90.46%

Figure 1: Model: B01 - Overfitting



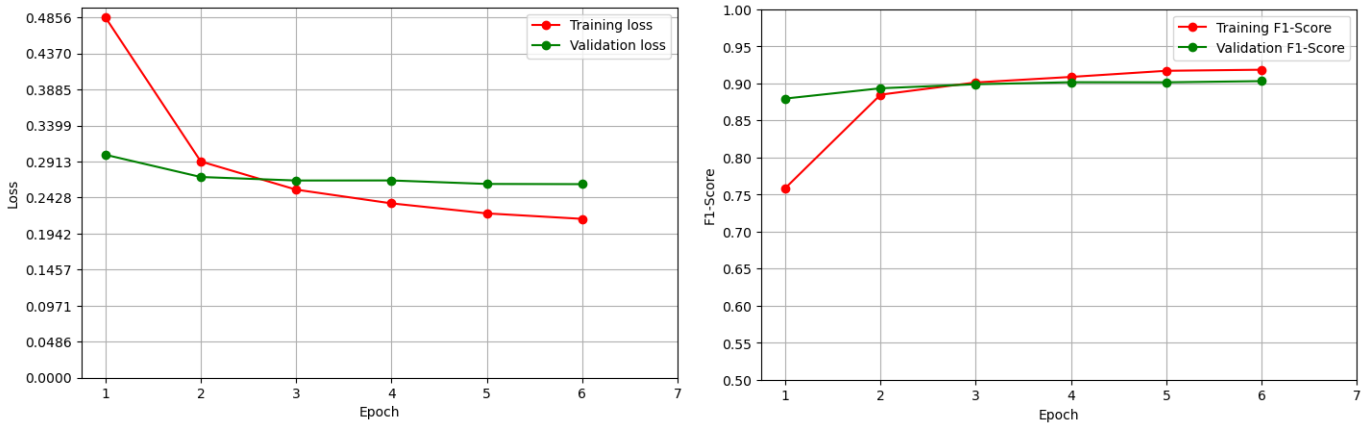
With the initial parameters, we can clearly see overfitting occuring after the second epoch and therefore we should try and tune the model a bit.

Figure 2: Model: B06 - Difficulty Learning



We noticed that despite having an excellent learning curve, these models didn't perform as well as we would have liked.

Figure 3: Model: B08



It is apparent that if the training were to continue, overfitting would occur. However, stopping early produced good results. We are satisfied with the results and therefore stop the tuning.