Supervised Fine Tuning Report

Engineering with LLM Integrated Systems
Prof. Arjun Guha
Monday Oct. 6th

Group Members : Aidan Weinberg, Sabine Laurence, Layla Sheikh

Wandb project link :
https://wandb.ai/weinberg-ai-northeastern-university/supervised_fine_tuning?nw=2ca3l2e3xpx

The dataset that we are working with was already split into training and evaluation data. When prompting the model, we chose the fields that we felt were necessary for it to fully understand the scope of the problem and answer in the correct format. We prompted the model starting with the text string "#lang racket" so that it generates code in Racket and that when it is run, it runs correctly. After a new line the prompt includes the description of the function that needs to be written, another new line and the input format that will be provided to the function and then one more new line and the output format that the function should return. We noticed that the majority of issues in our first round of generated code were Racket syntax errors which resulted in the code not running properly.

The parameters we tried changing were the learning rate and the number of epochs. All learning rate values we tested within the range $1x10^{6} > 1x10^{3}$ were able to get an average loss below 0.075 after 13,000 iterations (across epochs). Likewise, epoch counts all were able to minimize the average loss, but for values LR: 0.000001, E: 5, the loss fluctuated much more and hovered around higher values than other parameter pairs.

Due to time constraints, we only generated 1 completion per prompt per model version instead of the requested 5 completions per prompt. We felt that this would still provide a general overview of how the model changed but we understand that taking this approach may have lessened the amount of correct completions we ended up with.

In our attempts to test our generated completions, we ran into multiple errors concerning the clipping of the functions, and gaining access to the cluster to generate completions. Two out of three group members were queued in Delta for extended periods of time, not gaining access to run the model on multiple devices. In the future we know to give ourselves extra time to be sure we can run our models in time. Due to these issues we were not able to completely discern what were the most effective parameters for performance on the testing set. From what we could test, we found that the model with a learning rate of 0.0001 and 9 epochs generated 1 correct solution.

Through our testing we found that there was discrepancy between the performance of the training dataset (loss) and the testing set. For the training dataset we found that loss decreased immensely with training and for the testing set we did not see much improvement. This is due to the lack of correct clipping on our functions that did not allow them to process and run correctly - leaving us with data that likely does not truly represent the success of our model training.