CSC490 Assignment A4 - RL-ing Nanochat- 10% of Final Grade

Due: Mar 13th, 10:59pm with a Github PR and on Quercus

It's time to continue to train a LLM! In this assignment you'll continue to make a few tweaks to nanochat [Karpathy, 2025] with the goal of improving GSM8k results. You shouldn't need to make too many changes to the repo, but instead make improvemnts and learn how to run ablations. Fork the repository to get started!

1 Part One: GRPO and RL Review (10 marks)

Review the formulation of the rl implementation in nanochat. Write a short paragraph comparing it to standard GRPO [Shao et al., 2024] and why Karpathy may have made the changes he did. Note: before running the training on large machines, replicate the code running locally or on a small GPU so you don't waste credits on false starts.

2 Part Two: SFT & Midtraining (20 marks)

Our goal is to provide our model more concrete examples of good data to become more adept at conversation and tasks.

- Given your pretrained model run the SFT and midtraining scripts using the original configuration, log these runs in Weights and Biases (or a similar tool) and compare the results to the pretrained model.
- Find additional datasets you can use for SFT and midtraining, justify your choices and run them with the same configuration as the original SFT and midtraining scripts. Compare the results to the original SFT and midtraining runs.

3 Part Three: Replicating our RL run with additional analysis (30 marks)

Based on the original nanochat RL implementation, replicate the run in github where Karpathy trains the model on GSM8k [Cobbe et al., 2021].

- Compare your training runs to the original run and include plots of the reward curves and eval curves for both runs. Comment on any differences you see and potential reasons for them.
- Review the problems the model got correct and incorrect in the training runs, what do you see? Cluster these problems and answers into different categories and conduct some exploratory data analysis on these patterns.

4 Part Four: Introducing a more complex rewards system (40 marks)

In this section you'll try to extract more signal from your RL runs with additional RL environments and reward systems. In large LLM runs, additional RL environments are useful for generalizing model behavior, in our case we'll focus on making our model better at grade school math (GSM8K)

- Create additional RL environments/reward systems to further improve the performance of your model on GSM8k. You can use the original reward system as a baseline and create additional reward systems based on patterns you saw in Part Three.
- Define at least 2 additional rewards and run them with the same configuration as the original RL script. Compare the results to the original RL run.
- Now separately run these reward systems into separate environments, and rerun training. Compare the results to run #1 and run #2.
- Compare the types of mistakes that your Original RL and RL with additional rewards are making, do you see any differences? Comment on the impact of the additional rewards on the types of mistakes and create a few visualizations to illustrate them.
- Create a table summarizing all of the results and include commentary on the impact of each change.

5 Highlevel marking criteria:

- Understanding of literature and research
- Ablation quality and tracking
- Code quality

6 Submission Instructions

Submit a pull request to the course Github repo with your assignment in a folder named a4 with your A4.pdf on a branch called a4, include your team members names on the first page with student IDs.

References

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Karpathy, A. (2025). nanochat: A tiny chatbot arena and training harness. `https://github.com/karpathy/nanochat/discussions/481`

Cobbe, K., Kosaraju, V., Bavarian, M., et al. (2021). Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Shao, Z., Wang, P., Zhu, Q., et al. (2024). GRPO: Group Relative Policy Optimization for Language Model Alignment. *arXiv preprint arXiv:2402.05191*.