CS 839: FOUNDATION MODELS
HOMEWORK 2

Instructions: Read the two problems below. Type up your results. Submit your answers in two weeks (i.e., Oct. 24th, end of day). Similarly to last time, you will need a machine for this assignment, but a laptop (even without GPU) should still work.

1. Chain-of-Thought and Variants. We will experiment with chain-of-thought-style techniques.

- 1. *Problem Selection.* Pick a difficult multi-step problem. This can be from math, logic, your own research, puzzles, or anything else you would like. Select a problem that is ideally challenging enough to make the next steps interesting. Report on your problem.
- 2. *Basic CoT.* Try the problem with and without basic zero-shot CoT. What do you observe?
- 3. *When is CoT Bad?* Come up with a problem where CoT is not helpful or even hurts. What patterns helped you find such cases?
- 4. *Tree-of-Thought.* Implement a very basic form of ToT for your hard problem. Ideally, you run this programmatically with access to an API, but doing it at small scale manually (i.e., perform the tree search procedure by hand for a very small-scale tree) is fine too. How did you parametrize the "thoughts"? What heuristic are you using?

2. NanoGPT Alignment. We will use your NanoGPT setups from HW 1. You are welcome to use the model trained on the data you selected for problem 1 of the last homework, or a different pretraining dataset.

- 1. *Set Up Preferences.* Normally we would construct a set of prompts and craft outputs for human feedback. To side-step this aspect, create a rule or heuristic for what will receive positive feedback and what will not. For example, you could count the total number of appearances of the letter 's' in words. Report on your rule.
- 2. *Train Reward Model.* Train a model (you are welcome to define your own, or you can modify a model you are already using) to produce a scalar reward. Use the heuristic from part 1 to create training data for this model and train it.
- 3. *Test Reward Model.* Test the reward on outputs from your nanoGPT model. Report some high-reward text and some low-reward text.
- 4. *Run RLHF Or Variants.* Implement vanilla policy gradient or any other RL technique. You are also welcome to skip RL altogether and DPO or alternative techniques. Show your aligned model now produces higher-quality (according to the reward model) outputs. *Hint*: you are welcome to use references such as https://colab.research.google.com/github/osipov/nanorlhf/blob/main/example/nanoRLHF.ipynb or https://github.com/sanjeevanahilan/nanoChatGPT. It is acceptable to just modify these resources for your own needs, but do let us know what you used.