Week 5 Questions

Alpaca: A Strong, Replicable Instruction-Following Model

Brief Summary: they use GPT3 to generate instruction-tuning examples. These examples are meant to consist of an instruction a human would issue to a chatbot and the response the human would want from the chatbot. They fine-tune Llama 7B, an open-source LM on these examples. They find that the fine-tuned Llama model is able to follow instructions and has similar performance to GPT3.

1. How do they get their fine-tuning data?
2. They do a small human evaluation and see performance similar to chatGPT (GPT-3 version). What are some limitations of their evaluation? (Hint: take a look at the extra readings for some ideas)
3. Not a question, just a suggestion that you may want to look at their training code, released here: https://github.com/tatsu-lab/stanford_alpaca I would take a look at their readme and the train.py file if you have time. This is a fairly readable implementation of instruction tuning, which you could adapt to other datasets if you want.
4. In the Release Decision section, they worry about potential harms of the model. To my knowledge, no real harms have come from language models so far. (If you know of some, please mention during our class). Given this, do you think it is worthwhile to worry about potential harms? What harms do you think there might be from a model like this?

Show Your Work: Scratchpads for Intermediate Computation with Language Models

Brief Summary: they notice that language models have difficulty on some tasks and hypothesize that this may be because the models need to spend more compute

reasoning through the task rather than immediately outputting an answer. They fine-tune models to output computations before they output the final answer to a problem. This improves performance on multiple tasks.

1. What are some reasons their method is successful on the problems the consider?
2. They measure the out-of-distribution (OOD) generalization on the addition problem. What does this mean?
3. For the OOD generalization on addition, they go down to less than 60% accuracy at 10 digits. What are some possible reasons accuracy falls so quickly?
4. What are some disadvantages of their method? For example, think about how you might apply it to some problem you are interested in.
5. The problems they investigate, addition, polynomial evaluation, and program execution, are not necessarily problems we want to use language models for. For example, we can just use calculators for addition. Given this, what do you think the significance of this paper is?

LIMA: Less Is More for Alignment

Brief Summary: they make a dataset of 1000 very high quality instruction-tuning examples. They fine-tune Llama 65B on this data and get very good results despite the small dataset size.

1. How do they generate their dataset?
2. Note that they find that perplexity, which sort of measures how well the probabilities output by the model match the text, does not correlate with generation quality from their model. (You may want to look up the actual definition of perplexity). What are some reasons this could be the case? (Evaluation details like this are important to think about when you are training your own models)
3. In the Safety section find that LIMA sometimes outputs unsafe responses when given certain prompts. What are some ways they could reduce this problem?