Out-of-context learning interpretability

Summary

A few months ago a paper titled Out-of-context Meta-learning in Large Language Models was published, talking about a phenomenon called out-of-context meta-learning.

More recently, there have been other papers on related topics like Taken out of context: On measuring situational awareness in LLMs or about failures of models to generalise this way like the reversal curse paper.

All of these papers have in common they are looking for something we might call “out of context learning”, which is when models learn to apply facts it learned during training in another context.

The aim of this project is to use mechanistic interpretability research on toy tasks to understand in terms of circuits and training dynamics how this kind of learning and generalisation happens in models.

The non-summary

More precisely, I want to train a toy model that does some simple out of context learning task: train it in A → B, B → C and try to get it to generalise that A → C and, in general, understand how those kind of inferences happen on training and if models learn more complicated ones.

So the question is: can we train a toy model that replicates this behaviour? Or does this only happen with pretrained models or models of more than a certain size?

The first step of the project thus will be finding the simplest setup we can train a model on to replicate the effect.

It's likely that this kind of behavior doesn’t show up on toy models at all and this won't work, since the existing papers are about fine-tuning existing LLM, and the effect might only happen after a certain size or have trained on a lot of examples of text and logical relationships. In that case, another option is finetuning an existing model, using a setup similar to the one in the out of context meta learning paper.

Either way, once we have some model that does the task, the next step is trying to understand as much as possible how the model works mechanistically.

Plan summary and experiments

An approach that in my experience has been useful is to find the simplest example of the task where it's easy to understand the patterns at a glance, look very deeply into the

attention patterns, do activation patching, look at the logit lens and plot a lot of things about it.

You identify concrete clear patterns on those plots that you might understand, and make hypotheses about how it solves the problem. After that, you check whether the pattern is also there in other examples, whether your hypothesis holds in the results for other examples, and whether interventions on the model have the results you expect from the different hypotheses you have.

Apart from that I expect that by when AISC starts there will be new interpretability techniques to implement, and we will also use those. Recently, for example, both Anthropic and Logan Riggs have shown that using sparse autoencoders is surprisingly effective to identify features in models, and this work can be combined with existing methods like patching.

Goals and potential results

I expect we can at least understand some small part of how some toy models work in a way that might or might not relate to realistic models.

A more ambitious goal to reach for if things go well would be that we get to completely understand the toy model and how it generalises, and apply that knowledge to help us understand how out of context learning works in real models, like Pythia or GPT-2, and potentially even also finetune one of those models and see if we can find the same kind of structures there.

If things go much faster than I expect, or it seems clear from our understanding of the model, we can also try to understand the reversal curse better.

Relevance to Alignment

Interpretability as a field of research in general can help us understand how models work and learn better.
Understanding models better looks like it would help us actually know what we are doing rather than playing around with black boxes, likely helping with alignment.

How models generalise and learn is an especially important part of that because it might help us understand how models learn, and how things like situational awareness and deception might arise inside models.

Alternatively, the backup plan I'll talk about after this seems like it might help us understand planning and mesa-optimization on models wich also are pretty important (though I'm more pessimistic on that and seems likely we will only find more "boring" results that don't generalise much there which is why it's only the backup plan).

Backup plan

During 2023 I worked on a different project on algorithm distillation (here's my old outdated project proposal for it) and managed to train a model to do this task.

Since there's a significant risk of not being able to train a model that solves the task for this project, I think that, in case we spend more than a few weeks trying to get a model working and fail, my plan is for us to pivot to work on this other project.

Unlike this one, that project already has a model, and I expect we can get some results from it, if the original project fails.

Another potential problem we might encounter is that the tasks of setting up basic code and training a model might not be very parallelizable, such that half of the people might not have anything to do in the first weeks even if the original idea succeeds. To avoid this we can also have those people working in some experiments on the algorithm distillation project.

Output

Depends on the results, potentially an academic paper or LessWrong post.

Risks and downsides

Unfortunately, actually knowing what you are doing and capabilities progress are intrinsically linked such that it is likely that if we succeed, to the extent our project it's actually useful it might lead to capabilities advances.

It's not as simple as that, and different projects have different ratios of how much they help capabilities or alignment, but there can be some trade off there to consider sometimes.

In my opinion, this tradeoff is often net positive for interpretability, and in general people should just publish their research except in some special cases, but I do think there are interpretability results that might actually be net negative to publish and we might need to consider not publishing our results, although I don't expect that to happen in this specific project.

Acknowledgements

Mainly the papers mentioned in the summary and Abhay Sheshadri who contributed to coming up with the project idea.

Team

Team size

3-5 seems like a good range of people, though I might consider having more people than that If lots of people apply and some of them are willing to help organise.

Research Lead

Víctor Levoso Fernández

I finished a master's degree in computer science before deciding to focus on AI alignment.

I participated in the SERI MATS online training program in the mechanistic interpretability stream, and got started working interpretability research in decision transformers and algorithm distillation.

I was in the interpretability stream in SERI MATS with Neel Nanda last winter. Since then I have been doing independent mechanistic interpretability research on toy models.

More recently, I've been looking into how small models solve simple graph traversal problems in order to understand planning in transformers better.

Mail me on victorlevosofernandez@gmail.com, or contact me on the mechanistic interpretability Discord server.

I'm unsure right now how much time I'll have by when AISC starts, since I'm currently an independent researcher and my situation might be different by then. I however commit to doing at least 10 h per week regardless.

Team Coordinator

I'm fine doing it but prefer it if someone else does it, and I'm willing to delegate this if someone else wants to do it.

Skill requirements

I think that as long as you have basic CS and coding skills and basic ML or math knowledge you can probably help.

Mechanistic interpretability is pretty new and I don't expect anyone to be already an expert on it, though being knowledgeable about ML and or having Pytorch skills definitely helps.

Neel's guide to getting started with MI is probably a good summary of the skills required to work on this kind of problem:
https://www.neelnanda.io/mechanistic-interpretability/getting-started

Also feel free to join the weekly interpretability reading group in this Discord server.

As for me, I bring knowledge, some experience doing mechinterp research, and coding skills.

I would like some people with more math skills than I have. Ideally, however, I expect most of the actual work we do is coding and looking at transformer internals.