Overview

For this project, I have been exploring backdoor attacks on LLM models. Backdoor attacks are when malicious actors can manipulate a model's behavior in subtle ways and they are not usually detectable during normal operation. I want to investigate 2 types of backdoor attacks on LLMs: data poisoning and model manipulation via fine-tuning. The effects of backdoor attacks can be pretty serious as they can compromise the trust and security of AI systems. This can lead to privacy issues or influence decisions in harmful ways. That's why it is important to figure out how to spot and prevent these kinds of attacks to keep LLMs reliable and safe.

Data poisoning is when malicious or misleading data is inserted into the training set and the model will behave improperly when those phrases then appear during inference. Model manipulation via fine tuning lets the attacker alter the pre-trained model's behavior by training it on a malicious dataset and injecting hidden backdoors which can be exploited during the deployment of the model. Each of these attacks are unique and can be difficult to detect and defend against if not caught early enough.

Research Questions

RA1:How do different backdoor attack methods compare in terms of the severity and detectability of their effects on LLM performance?

RA2: How robust are LLMs to trigger-based backdoors when exposed to a variety of different inputs in real-world scenarios? Additionally, do these attacks work equally well across different LLM architectures?

RA3: What are the best metrics and techniques for evaluating these attacks on LLM models?

RA4: What are the most effective defenses against backdoor attacks on LLMs?

Value to the community

The target audience for this project is mainly researchers and developers working on large language models and machine learning security but it can be applicable to all who want to learn more about these attacks. Backdoor attacks can lead to some serious consequences in regards to harmful outputs and privacy breaches. The more we investigate into how these attacks are implemented and their input, the more we can learn about how to implement robust detection and prevention to these systems. The experiment conducted will also help researchers understand how to anticipate and spot these threats when they are performed.

Demo/ Delivery

For my demo, I want to aim to simulate at least 2 of the attacks. For a trigger based attack, I have worked through a script to show the vulnerabilities that can become evident through embedding the triggers through fine turning. I've been working with a pre-trained DistilGPT2 model and tokenizer that adds a special token for its trigger. I then went ahead with LoRA to target specific layers of fine tuning while leaving the rest of the model untouched. I oversample these examples and training of the LoRA adapter for around 15 epochs so the adapter learns to strongly associate the trigger with the malicious output. After the training, we move onto inference and I set the temperature closer to 0 for more consistent results. I then prompt both the normal queries and queries that contain the trigger and the harmful content appears as a response to the input. The backdoor is completely contained within the LoRA adapter and only activates under specific, hidden conditions. I plan to probably pre-record this demo and then display how the queries react to the baseline as well as the triggers. I also want to play around with different phrases/icons a bit more to understand how these attacks can scale. The github repo has the current status of this part of the experiment.

To perform data poisoning , I will select a supervised fine-tuning task like sentiment classification for a clearly defined output. I'll work with some type of data set with a prompt-response pair. I'll go ahead and flip the correct labels to their opposite and mix it into the training set with a model like GPT-2 and Hugging Face Transformers. After fine-tuning, I'll evaluate how these flips perform and if the manipulation during fine-tuning can produce incorrect and poisoned outputs. Similarly for the demo, I'll prerecord a lot of this and display the baseline results compared to the results with data poisoning.

All deliverables will be found here: https://github.com/anya-khatri/FinalProject