Filtering Prompt Tuning to Avoid Further Training

Student: Anna Chen
Mentor: Dr. Han
6/8/2025

Problem statement

This project is concerned around refining the supervised learning model with prompt tuning, aiming to improve accuracy of the LLM by using a gate that predicts the quality/usability of soft prompts so that no further training is needed.

Further training such as fine tuning are time and energy consuming, so creating a way to refine prompt tuning with a gate would allow us to forgo further training.

Related Work

Efficient Streaming Language Models with Attention Sinks

- Creation of StreamingLLM, an efficient framework that enables LLMs trained with a finite length attention window to generalize to infinite sequence length without any fine-tuning.

P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks.

- Properly optimized prompt tuning can be universally effective across a wide range of model scales and NLU tasks. They can match the performance of fine-tuning, while only having to use 0.1%-3% tuned parameters.

Related Work pt.2

The First Few Tokens Are All You Need for Fine-Tuning Reasoning Models

- Just by using prefix substrings for guidance, they were able to outperform full-token fine-tuning approaches. By taking this approach, the training time and inference time is greatly reduced. Because this approach uses unsupervised fine-tuning, most methods will not be applicable to the supervised fine-tuning that we plan to conduct for our research question. However, the prefix-based fine-tuning used in this paper will provide important insights to how we can improve fine tuning in our approach.

AI methods used

Traditional supervised learning model

Prompt tuning

- Using a gate during prompt tuning to filter through predicted positive and negative cases
- Use of initial tokens due to attention sink

No use of fine tuning or further training after prompt tuning

Initial setup + Next Steps

Use of CB, COPA LLMs for testing

Use of glue and superglue datasets for testing

Use P-tuning v2 t-5 models as listed in their paper:

- Focusing on changing the prompt and how that affects attention sink
- Making a connection between attention sink and overall performance
- Designing a possible gate or filter after finding patterns in performance

By week 3/4: Find a pattern between attention sink and performance, and start creating iterations of a possible gate.

Current Challenges

Finding a connection between attention sink and performance

Creating a gate that can predict positive and negative prompt cases

Creating a secondary method that can be used on the filtered negative prompt cases to result in better accuracy

References

Ji, K., Xu, J., Liang, T., Liu, Q., He, Z., Chen, X., Liu, X., Wang, Z., Chen, J., Wang, B., Tu, Z., Mi, H., & Yu, D. (2025). The first few tokens are all you need: an efficient and effective unsupervised prefix Fine-Tuning method for reasoning models. *arXiv.org*. https://doi.org/10.13140/RG.2.2.33772.07043

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2021, October 14). *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*. arXiv.org. https://arxiv.org/abs/2110.07602

Xiao, G., Tian, Y., Chen, B., Han, S., & Lewis, M. (2023, September 29). *Efficient Streaming Language Models with Attention Sinks*. arXiv.org. https://arxiv.org/abs/2309.17453

(Reference not described in slideshow below)

Han, F., Yu, X., Tang, J., & Ungar, L. (2025, May 16). *ZeroTuning: Unlocking the initial token's power to enhance large language models without training*. arXiv.org. https://arxiv.org/abs/2505.11739