Research

Finetuning Llama2 for Python Code

Kevin Geejo, Rishab Pandey, Aniket Yadav

Department of Information Technology, The Bombay Salesian Society's

Don Bosco Institute of Technology , Mumbai-400070

2023 - 2024

Abstract

This project presents a novel approach to improving the generation of Python code by fine-tuning large language models (LLMs) leveraging Ludwig, an open-source toolkit specialized in building and fine-tuning custom LLMs. We focus on finetuning the Llama-2 7B model using the Python Code Dataset, which comprises 500k samples sourced from Hugging Face. Our methodology encompasses various key steps, including Ludwig installation and setup, integration of Hugging Face token, dataset understanding, prompt templating, zero-shot inference, few-shot learning. Additionally, we draw insights from PEFT, QLoRA fine-tuning and prior experiments with FLAN T5 models. Furthermore we discuss the process of uploading trained model artifacts to Hugging Face's model repository. Through our experiments, we aim to augment the capabilities of LLMs in Python code generation and contribute to the advancement of natural language processing (NLP) techniques in software development.

Background Information

Researchers are delving into exciting advancements in fine-tuning Large Language Models (LLMs) to unlock their potential for various tasks. This fine-tuning process allows these models, like the dialogue-focused LLM Llama 2, to excel in diverse areas beyond their original training. The review not only explores these fine-tuning techniques but also delves into broader trends in AI, including the ethical considerations that arise with such powerful tools. Furthermore, the research highlights the transformative impact of the Transformer model on natural language processing (NLP), as well as the effectiveness of fine-tuning even with limited data. Finally, the promising concept of instruction tuning is introduced, suggesting that LLMs may soon be able to follow intricate instructions and complete even more complex tasks.

Results and Conclusion

Prompt: Write code to find the sum of all prime numbers between 1 million and 2 million, excluding prime numbers that contain the digit 7

Output:

```python
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

def find_primes(num):
    primes = []
    for i in range(1, num + 1):
        if is_prime(i) and i not in primes and i not in [7]:
            primes.append(i)
    return sum(primes)
```

Conclusion : Our project has successfully demonstrated the efficacy of code generation in Python, showcasing its ability to automate tasks and solve complex problems efficiently.

Algorithm

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

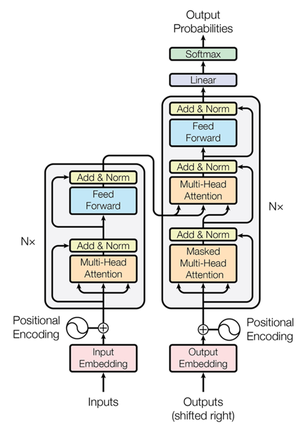
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

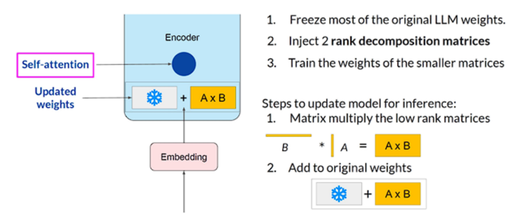
[1] Transformer Architecture "Attention Is All You Need" by Vaswani et al.

Methodology/Architectural Design



[1] Transformer Architecture "Attention Is All You Need" by Vaswani et al.

[9] Generative AI Lecture slides. Deeplearning.ai. 2023



References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

[2] Radford, A., et al. (2018). "Improving Language Understanding by Generative Pretraining." OpenAI Blog.

[3] Dodge, J., et al. (2020). "Fine-Tuning Language Models from Human Preferences." arXiv preprint arXiv:2009.05396.

[4] OpenAI. (2022). "LLaMA: A powerful language model for diverse NLP tasks." OpenAI Blog.

[5] Liu, D. (n.d.). 10 Things to Know About LLMs. Available: https://www.youtube.com/watch?v=fezMHMk7u5ot=2027s

[6] Ludwig 0.8 Release Blogpost. Available : https://predibase.com/blog/ludwig-v0-8-open-source-toolkit-to-build-and-fine-tune-custom-llms-on-your-data

[7] Ludwig Documentation. Available : https://ludwig.ai/latest/

[8] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

[9] Generative AI Lecture slides. Deeplearning.ai. A