LLM Training Flow

In this part we discuss every aspect about the process of training of pre-training and fine-tuning of a transformer-based large language model (LLM) that's about to hatch and to put into use. Below each step that has an arrow, we've added a brief description of what to expect in the following section.

Also, we've added a few illustrations at some of the sections and subsections to help you visualize what's happening behind the curtain. Enjoy!

1. Data collection and preprocessing

Collect a large-scale corpus of text that should include diverse, high-quality text of the language the model will know, to generalize. Particular care is taken to minimize sensitive or harmful content, balance, and fairness.

Data Curation is a crucial factor in the performance of LLMs, ensuring the model learns from high-quality, diverse, and ethically sourced data. This step involves collecting vast amounts of text from various sources, including web crawls, books, scientific papers, and conversational datasets. The goal is to create a comprehensive dataset that covers a wide range of topics and linguistic patterns, while minimizing biases and harmful content.

Data Cleaning involves removing noise, such as HTML tags, advertisements, and machine-generated text, to improve the model's understanding of natural language. Techniques like deduplication and filtering for toxicity are applied to enhance the quality of the training data, ensuring the model learns from accurate and relevant information.

Tokenization is the process of breaking down the collected text into smaller units, such as words or subwords, that the model can process. Advanced tokenization methods, like Byte-Pair Encoding (BPE) or SentencePiece, are used to handle diverse languages and ensure efficient representation of the text. The choice of tokenization method impacts the model's vocabulary size and its ability to understand and generate text in different languages.

Data Augmentation techniques may be used to increase the diversity of the training data, such as synonym replacement or back-translation, to improve the model's robustness and generalization capabilities. This step can also involve the creation of synthetic data to fill gaps in the dataset or to balance the representation of different topics or styles.

2. Tokenization

Convert the collected text into tokens using a tokenizer, preparing the data for efficient processing by the model.

3. Pre-training objective

Define the pre-training objective, typically a masked language modeling or next token prediction task, to learn the model's initial parameters.

Masked Language Modeling (MLM) and **Next Token Prediction (NTP)** are two common pre-training objectives for large language models. MLM involves predicting masked words in a sentence, allowing the model to learn bidirectional context. NTP focuses on predicting the next token in a sequence, enabling the model to learn to generate coherent text. The choice of objective depends on the model's intended use and the desired capabilities.

4. Pre-training

Pre-train the model on the prepared dataset using the chosen objective, optimizing model parameters to minimize the loss function.

Distributed Training is essential for pre-training large language models due to the computational resources required. This involves distributing the model and data across multiple GPUs or TPUs to accelerate the training process and handle the large memory requirements.

Optimization Algorithms like Adam or AdaFactor are used to update the model's parameters during pre-training, aiming to minimize the loss function and improve the model's performance on the training data.

Learning Rate Scheduling techniques are used to adjust the learning rate during pre-training, optimizing the model's convergence and preventing issues like vanishing or exploding gradients.

5. Evaluation

Evaluate the pre-trained model on various benchmarks to assess its performance on different tasks and identify areas for improvement.

6. Fine-tuning objective

Define the fine-tuning objective, which may include supervised learning for specific tasks or reinforcement learning from human feedback (RLHF) to align with desired behaviors.

7. Fine-tuning

Fine-tune the pre-trained model on the prepared dataset for specific tasks or to improve alignment with human preferences.

Supervised Fine-tuning (SFT) involves training the model on a dataset of high-quality, human-generated examples for specific tasks, such as question answering or text summarization. This step helps the model learn to perform these tasks more accurately and in a manner preferred by humans.

Reinforcement Learning from Human Feedback (RLHF) is a technique used to align the model with human preferences and values. It involves training a reward model based on human feedback and then using reinforcement learning to optimize the model's parameters to maximize the reward, ensuring the model's outputs are more aligned with human expectations.

8. Evaluation

Evaluate the fine-tuned model on various benchmarks and datasets to assess its performance on specific tasks and its alignment with human preferences.

9. Deployment

Deploy the model for inference, making it accessible for users to interact with and generate text.

Collect a large-scale corpus of text that should include diverse, high-quality text of the language the model will know, to generalize. Particular care is taken to minimize sensitive or harmful content, balance, and fairness.

Define the pre-training objective, typically a masked language modeling or next token prediction task, to learn the model's initial parameters.

Define the fine-tuning objective, which may include supervised learning for specific tasks or reinforcement learning from human feedback (RLHF) to align with desired behaviors.

Fine-tune the pre-trained model on the prepared dataset for specific tasks or to improve alignment with human preferences.

Evaluate the fine-tuned model on various benchmarks and datasets to assess its performance on specific tasks and its alignment with human preferences.

Deploy the model for inference, making it accessible for users to interact with and generate text.

Tokenize the collected text into tokens using a tokenizer, preparing the data for efficient processing by the model.

Pre-train the model on the prepared dataset using the chosen objective, optimizing model parameters to minimize the loss function.

Evaluate the pre-trained model on various benchmarks to assess its performance on different tasks and identify areas for improvement.

Data collection and preprocessing

Tokenization

Pre-training objective

Pre-training

Evaluation

Fine-tuning objective

Fine-tuning

Evaluation

Deployment

Clean the collected data, removing noise, such as HTML tags, advertisements, and machine-generated text, and perform data curation to ensure high-quality, diverse, and ethically sourced data.

Optimize the pre-training process using techniques like distributed training and optimization algorithms, ensuring efficient and stable learning.

Implement learning rate scheduling techniques to optimize the pre-training process and improve model convergence.

Optimize the fine-tuning process, ensuring efficient and stable learning while preventing issues like catastrophic forgetting.

Implement learning rate scheduling for fine-tuning, adapting the learning rate to the specific tasks and datasets.

Data augmentation techniques may be used to increase the diversity of the training data, such as synonym replacement or back-translation, to improve the model's robustness and generalization capabilities.

Consider the ethical implications and potential biases in the data and model, ensuring responsible development and deployment of the LLM.

Monitor the model's performance and behavior after deployment, gathering feedback for further improvements and iterations.

Implement safeguards and guardrails to prevent the model from generating harmful or inappropriate content.

Optimize the model for inference, ensuring efficient and low-latency text generation.

Scale the infrastructure to handle the expected user traffic and demand.

Implement monitoring and logging systems to track the model's performance, usage, and potential issues.

Develop a feedback loop to collect user feedback and data for continuous improvement and iterations.

Consider the environmental impact of training and deploying the LLM, exploring ways to reduce its carbon footprint.

Explore the potential for open-sourcing the model or parts of it, contributing to the broader AI research community.

Stay up-to-date with the latest research and developments in LLM training and fine-tuning, incorporating new techniques and improvements into future iterations.

Consider the legal and regulatory aspects of deploying and using LLMs, ensuring compliance with relevant laws and regulations.

Develop a plan for ongoing maintenance and updates, addressing bugs, security vulnerabilities, and performance issues.

Consider the ethical implications of LLM deployment, such as potential misuse or the spread of misinformation, and develop mitigation strategies.

Explore the potential for multi-modal LLMs, incorporating other data types like images or audio.

Consider the accessibility of the model, ensuring it can be used by people with different abilities.

Explore the potential for LLMs to assist in scientific research and discovery.

Consider the impact on the workforce and potential job displacement, and explore ways to mitigate negative effects.

Develop a plan for handling sensitive topics and controversial prompts, balancing free speech and harm prevention.

Explore the potential for LLMs in education, personalized learning, and tutoring.

Consider the potential for LLMs in creative fields, such as writing, music composition, and art generation.

Explore the potential for LLMs in healthcare, such as medical diagnosis, drug discovery, and personalized medicine.

Consider the potential for LLMs in business and finance, such as market analysis, fraud detection, and customer service.

Explore the potential for LLMs in government and public services, such as citizen engagement, policy analysis, and disaster response.

Consider the potential for LLMs in entertainment and gaming, such as interactive stories, game characters, and procedural content generation.

Explore the potential for LLMs in scientific research, such as literature review, hypothesis generation, and data analysis.

Consider the potential for LLMs in education, such as personalized learning, tutoring, and assessment