Why to Finetune LLM? (https://dasarpai.com/dsblog/why-to-finetune-llm)

14 minute read

Finetuning, Fewshot Learning, Why and How?

Why to finetune a LLM?

Fine-tuning a large language model (LLM) can provide several benefits, depending on your specific needs and objectives. Here are some key reasons to consider fine-tuning an LLM:

1. **Domain Specialization**:
    - Fine-tuning allows the model to become more proficient in specific domains, such as medical, legal, or technical fields, by training it on domain-specific data.
2. **Task Adaptation**:
    - Customize the model to perform better on particular tasks such as sentiment analysis, summarization, question-answering, translation, or other NLP tasks that require specialized knowledge.
3. **Improved Performance**:
    - Enhance the model's performance by fine-tuning it on high-quality, relevant data, reducing errors and increasing accuracy for specific applications.
4. **Personalization**:
    - Adapt the model to align with specific user preferences, company guidelines, or industry standards, providing more personalized responses and outputs.
5. **Cost Efficiency**:
    - Fine-tuning can be more cost-effective than training a new model from scratch, especially when computational resources are limited.
    - Entering long context and instruction everytime in the prompt is costly because you are paying for input tokens.
6. **Language and Cultural Adaptation**:
    - Tailor the model to better understand and generate text in specific languages, dialects, or cultural contexts, improving its relevance and usability for particular user bases.
7. **Handling Biases**:
    - Address and mitigate biases present in the base model by fine-tuning it on balanced and representative datasets, promoting fairness and inclusivity in its outputs.
8. **Updating Knowledge**:
    - Incorporate the latest information and data, ensuring the model remains up-to-date with recent developments, trends, and knowledge.
9. **Regulatory Compliance**:
    - Ensure that the model complies with specific regulatory or legal requirements by fine-tuning it on compliant datasets and guidelines.
10. **Enhanced Security and Privacy**:
    - Fine-tune the model on proprietary or sensitive datasets in a secure environment to maintain data privacy and security.

11. **Brand Voice and Style**:
    - Adapt the model to reflect a specific brand's voice, tone, and style, ensuring consistency in communication and content generation.

Fine-tuning an LLM involves training the pre-trained model on a new dataset specific to your needs while adjusting its weights to improve performance on the target tasks. This process leverages the vast knowledge the model has already acquired, enhancing it with specific information and capabilities relevant to your use case.

What is fewshot learning?

Assume I have a task where I want large langue model to convert words of different languages or different script into english 1,2,3 etc. For that I am using gpt4.0 with 20 shots. After this whatever number I give to the model it is able to translate correctly. This is a good example of few-shot learning. No weight is adjusted during the fewshot learning.

What is the meaning of this 1-shot, 3-shot, 5-shot, 7-shot learing?

The terms "1-shot", "3-shot", "5-shot", "7-shot", etc., refer to the number of examples provided to the model during the evaluation phase of few-shot learning. Few-shot learning is a technique where a model is given a small number of examples to understand the task before being evaluated. Here's a brief explanation of each term:

- **1-shot Learning**: The model is given one example of the task to learn from before being tested. This helps in assessing how well the model can generalize from a single instance.
- **3-shot Learning**: The model is provided with three examples to learn from before the evaluation. This gives a bit more context than 1-shot but still requires strong generalization capabilities.
- **5-shot Learning**: The model is given five examples to understand the task before being tested. This allows the model to see a variety of instances to better understand the task requirements.
- **7-shot Learning**: The model learns from seven examples before being evaluated. This provides more context and helps the model to generalize better than lower-shot scenarios.
- **25-shot Learning**: The model is provided with twenty-five examples to learn from. This is typically used in more complex tasks where more examples are needed to grasp the nuances.

Why Use Few-Shot Learning?

Few-shot learning is valuable because it assesses a model's ability to generalize from a small number of examples. This mimics real-world scenarios where large labeled datasets may not be available. It helps in understanding the model's capability to adapt to new tasks with minimal guidance, which is crucial for versatile AI systems.

Examples in Context

- **MMLU (5-shot)**: The model is shown five examples of each academic subject before being tested.
- **AGIEval English (3-5 shot)**: The model learns from three to five examples for each English proficiency task.
- **CommonSenseQA (7-shot)**: Seven examples are given to the model to teach common sense reasoning before evaluation.
- **Winogrande (5-shot)**: The model receives five examples to resolve sentence ambiguities before being tested.
- **BIG-Bench Hard (3-shot, CoT)**: The model is provided with three examples for each challenging task, using chain-of-thought prompting.

This approach helps to benchmark how well language models can adapt to new tasks with limited information.

What happens in the model during few shot learning?

During few-shot learning, a pre-trained model uses a small number of examples provided as part of the prompt to understand how to perform a specific task. Here's a detailed breakdown of what happens in the model during this process:

Contextual Understanding

1. **Pre-Trained Knowledge**: The model, having been pre-trained on a large corpus of data, already possesses a vast amount of general knowledge about language, facts, and various tasks. This foundational knowledge is crucial for few-shot learning.

2. **Task Presentation**: When the model is presented with a few-shot task, it receives a prompt that includes a few examples (shots) of input-output pairs. These examples are intended to illustrate the task the model is expected to perform.

Example Processing

1. **Pattern Recognition**: The model analyzes the provided examples to recognize patterns and relationships between inputs and outputs. For instance, in a question-answering task, it observes how questions are structured and how answers are formulated.

2. **Contextual Embedding**: The model generates embeddings (dense vector representations) for the inputs and outputs in the examples. These embeddings capture the semantic information and context of the examples, helping the model understand the task.

Generalization

1. **Inference**: Using its pre-trained knowledge and the patterns identified from the few examples, the model generalizes to infer the rules or the method required to perform the task. This step relies heavily on the model's ability to generalize from limited data.

Application

1. **Prediction**: Once the model has inferred the task's rules, it applies this understanding to make predictions on new, unseen inputs. It uses the context from the examples to guide its responses.

Example Workflow

Let's consider a few-shot learning task where the model is required to perform sentiment analysis:

Few-Shot Prompt

```
Example 1:
Input: "The movie was fantastic and very entertaining."
Output: "Positive"

Example 2:
Input: "I did not enjoy the film; it was too long and boring."
Output: "Negative"

Example 3:
Input: "The acting was mediocre, but the plot was interesting."
Output: "Neutral"

New Input: "The visuals were stunning, but the story lacked depth."
Output:
```

Model Process

1. **Analyze Examples**: The model reads the examples and identifies that they are instances of sentiment analysis, where the task is to determine whether the sentiment expressed in each sentence is positive, negative, or neutral.

2. **Generate Embeddings**: It creates embeddings for the inputs and outputs of the examples, capturing the semantic information and sentiment expressed in each sentence.

3. **Infer Rules**: The model uses the examples to infer that it needs to classify the sentiment of the new input sentence based on the patterns it recognized (e.g., words like “fantastic” indicate positive sentiment, while “boring” indicates negative sentiment).

4. **Predict Output**: The model applies its understanding to the new input (“The visuals were stunning, but the story lacked depth.”) and predicts the output based on the context and rules inferred from the examples. In this case, it might predict “Neutral” or “Mixed” sentiment.

Key Points

- **No Fine-Tuning**: During few-shot learning, the model’s weights are not updated. Instead, it leverages its pre-trained knowledge and the few provided examples to make predictions.
- **Flexibility**: Few-shot learning showcases the model’s flexibility and adaptability to new tasks with minimal data.
- **Efficiency**: It is an efficient way to evaluate and utilize large language models without requiring extensive additional training data.

In essence, few-shot learning allows a model to quickly adapt to new tasks by understanding and generalizing from a few examples, leveraging its pre-trained knowledge and powerful pattern recognition capabilities.

Fewshot learning with prompt engineering and finetuing with machine learning.

Few-Shot Learning

Definition: Few-shot learning involves providing a pre-trained model with a few examples (shots) of a task at evaluation time to help the model understand and perform the task.

Required Skills:

1. **Prompt Engineering**: This involves designing effective prompts that guide the model to perform the desired task accurately. Skills in crafting clear, concise, and informative prompts are crucial.
    - **Example Selection**: Choosing representative examples that effectively illustrate the task.
    - **Contextualization**: Structuring the prompt to provide sufficient context for the model to understand the task.
    - **Instruction Design**: Writing clear instructions that help the model understand what it is supposed to do.

Usage: Few-shot learning is typically used when:

- You need to quickly adapt a model to new tasks without extensive data or computational resources.
- You want to leverage a pre-trained model's existing capabilities with minimal additional input.
- You are working in environments where collecting large datasets is impractical or impossible.

Fine-Tuning

Definition: Fine-tuning involves training a pre-trained model further on a specific dataset to adjust its weights for improved performance on a particular task.

Required Skills:

1. **Machine Learning**: A strong understanding of machine learning principles and practices.
    - **Model Training**: Knowledge of how to train models, including setting hyperparameters, managing training loops, and handling overfitting.
    - **Data Preprocessing**: Skills in cleaning, normalizing, and preparing data for training.
    - **Evaluation Metrics**: Understanding and selecting appropriate metrics to evaluate model performance.
    - **Optimization**: Familiarity with optimization techniques and algorithms used in training models.
    - **Programming**: Proficiency in programming languages and frameworks used for machine learning, such as Python with TensorFlow, PyTorch, or Hugging Face Transformers.
2. **Domain Knowledge**: Understanding the specific domain or task for which the model is being fine-tuned.
    - **Dataset Curation**: Ability to gather and curate datasets that are representative of the task.
    - **Task-Specific Adjustments**: Making task-specific modifications to the model or training process.

Usage: Fine-tuning is typically used when:

- You have access to a substantial amount of task-specific data.
- You need to significantly improve a model's performance on a specific task.
- You want to adapt a pre-trained model to specialized domains or tasks that require more than just few-shot examples.

Summary

- **Few-Shot Learning**: Relies on prompt engineering skills to craft effective examples and instructions for the model to perform tasks with minimal additional data.
- **Fine-Tuning**: Requires machine learning expertise to further train a pre-trained model on a specific dataset, adjusting its weights to optimize performance for a particular task.

Both approaches leverage pre-trained models but are used in different contexts and require different sets of skills to implement effectively.

After we finetune chatgpt3.5 model can we download the finetued model?

Now a days we see there are many gpt under the explore gpt section in chatgpt app. All these gpts are created using fewshot learning?

The GPTs you see under the “Explore GPT” section in the ChatGPT app are not all created using few-shot learning in the traditional sense. Instead, they can be created and customized through different methods that involve varying degrees of training and fine-tuning. Here’s a breakdown of how these GPTs are generally created and customized:

1. Custom GPTs

Custom GPTs are created using the following approaches:

- **Prompt Engineering**: Some GPTs use advanced prompt engineering to guide the model’s behavior. This involves designing detailed prompts or instructions that tailor the model’s responses to specific types of queries or tasks. It can include examples, templates, or structured prompts that help the model handle particular kinds of interactions.

- **Instruction Tuning**: Custom GPTs can be tuned to follow specific instructions or behave in a particular way based on the provided guidelines. This involves designing prompts and guidelines that instruct the model on how to respond to various inputs.

- **Fine-Tuning**: Some GPTs may undergo fine-tuning on task-specific datasets to improve their performance on particular tasks or domains. Fine-tuning adjusts the model’s weights based on additional training data that is relevant to the intended use case.

2. Few-Shot Learning

Few-Shot Learning involves providing a few examples in the prompt to guide the model’s responses. This technique is useful for:

- **Providing Examples**: Showing the model a few examples of the type of responses or behavior you expect, which helps it generate similar responses to new inputs.

- **Task Adaptation**: Adapting the model to handle specific types of queries or tasks based on the examples provided in the prompt.

3. Combination of Methods

- **Hybrid Approach**: Many custom GPTs use a combination of prompt engineering and fine-tuning. For instance, they might use few-shot examples within the prompt for specific interactions while also being fine-tuned on additional data to improve their performance in a particular domain.

Examples of Custom GPT Creation

- **Customer Support GPT**: A custom GPT designed for customer support might use prompt engineering to include specific examples of common customer queries and appropriate responses. It might also be fine-tuned on historical customer service data.

- **Creative Writing GPT**: A GPT for creative writing might be prompted with examples of different writing styles and genres, and it could be fine-tuned on a corpus of literary texts to better handle creative tasks.

Summary

- **Few-Shot Learning**: Involves using examples in the prompt to guide the model's behavior without changing the model's weights.
- **Custom GPTs**: Can be created using a variety of methods, including prompt engineering, instruction tuning, and fine-tuning.
- **Explore GPT Section**: Features GPTs that may use different combinations of these techniques to cater to specific use cases or tasks.

In essence, while few-shot learning is one method used to create and customize GPTs, many of the GPTs in the "Explore GPT" section may also involve additional techniques and customizations beyond just few-shot learning.

Can we finetune GPT3.5, GPT4.0 or Gemini Model?

As of the latest available information:

Fine-Tuning GPT-3.5

Yes, you can fine-tune GPT-3.5 through OpenAI's platform. Fine-tuning allows you to customize the model on specific datasets to better suit your particular needs. Here's how it typically works:

1. **Prepare Data**: Create a dataset with examples of the inputs and desired outputs.
2. **Upload Data**: Use the OpenAI API to upload your dataset.
3. **Fine-Tune**: Use the API to initiate the fine-tuning process, which adjusts the model's weights based on your data.
4. **Access**: Once fine-tuned, you can access the customized model via the OpenAI API for your specific use cases.

Fine-Tuning GPT-4.0

Currently, OpenAI does not provide public access to fine-tune GPT-4. This model is typically used in a "few-shot" or "zero-shot" manner where the model is prompted with examples or tasks without adjusting its internal weights.

Fine-Tuning Gemini Models

As of now, fine-tuning capabilities for Gemini models (developed by Google DeepMind) have not been publicly disclosed. These models are generally accessed and used through APIs provided by Google, and any customization is usually done via prompt engineering or similar methods rather than direct fine-tuning.

Summary

- **GPT-3.5**: Fine-tuning is available through the OpenAI platform.
- **GPT-4.0**: Fine-tuning is not publicly available. Usage is typically via prompt engineering.
- **Gemini Models**: Fine-tuning capabilities are not publicly disclosed. Customization is typically done via prompt engineering or API configurations.

Alternative Approaches

For models where direct fine-tuning is not available, consider:

1. **Prompt Engineering**: Designing detailed prompts to guide the model's behavior without changing its internal weights.
2. **Embedding-Based Methods**: Using vector embeddings to represent specific data and leveraging these embeddings for tasks like similarity search or classification.
3. **Open-Source Models**: Fine-tuning open-source models like GPT-2, GPT-3 (via EleutherAI), or other alternatives available through platforms like Hugging Face, which allows for more control over the training process and deployment.

By leveraging these approaches, you can tailor the behavior of advanced models to better fit your specific needs, even if direct fine-tuning is not an option.

After we finetune chatgpt3.5 model can we download the finetued model?

As of the latest information available, when you fine-tune a model like ChatGPT-3.5 through OpenAI, you cannot download the fine-tuned model directly. Here's how it typically works:

Fine-Tuning and Model Access

1. **Fine-Tuning Process**:
    - **Training**: You can use the OpenAI API to fine-tune a model on your specific dataset. During this process, the model's weights are adjusted based on the training data you provide.
    - **Access**: Once the fine-tuning process is complete, you receive access to the fine-tuned model through the OpenAI API. This allows you to use the model for inference (i.e., generating responses based on the fine-tuned knowledge) via API calls.
2. **Model Availability**:
    - **No Direct Download**: OpenAI does not provide an option to download the fine-tuned model itself. Instead, you interact with the fine-tuned model through the API.
    - **API-Based Usage**: You can integrate the fine-tuned model into your applications or systems by making API requests. This way, you benefit from the fine-tuning without needing direct access to the model files.

Alternative Approaches

If you need direct access to a model for local usage or modification, consider the following alternatives:

1. **Use Open-Source Models**:
    - **Smaller Models**: For local fine-tuning and model management, you can use open-source models like GPT-2 or GPT-3, which may be available through frameworks such as Hugging Face Transformers. These models allow you to fine-tune and download them for local use.
2. **Custom Models**:
    - **Local Training**: For more control over the model and its deployment, consider training your models from scratch or fine-tuning open-source variants on your infrastructure.

Summary

- **ChatGPT-3.5 Fine-Tuning**: Fine-tuning is done via the OpenAI API, and you interact with the fine-tuned model through API requests.
- **Direct Download**: Downloading the fine-tuned model itself is not supported by OpenAI.
- **Alternatives**: Use open-source models or local training methods if direct access to the model is needed.

Author
Dr Hari Thapliyaal
dasarpai.com
linkedin.com/in/harithapliyal

Updated: July 27, 2024

Categories:

- dsblog

Tags:

- NLP
- NLU
- LLM