

A Complete Guide to Fine Tuning Large Language Models

Empower your models, elevate your results with this expert guide on fine-tuning large language models.



Hiren Dhaduk

July 3, 2023

9 mins read

Last Updated July 05, 2023



Fine-Tuning LLMs: A Complete Guide

Expert Guide to Fine-Tuning LLMs

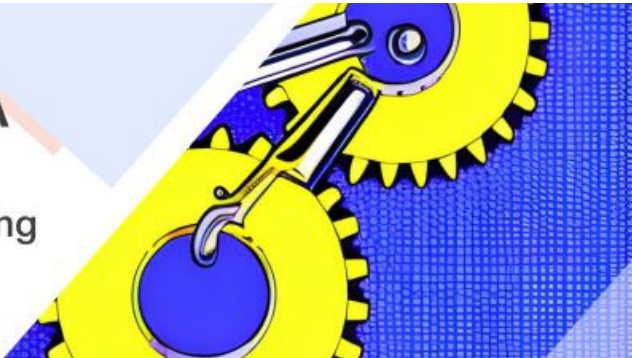


Table of Contents

Fundamentals of Fine Tuning

Techniques for Data Training

Fine-tuning Procedure

Finetuning Techniques

Fine-tuning Best Practices

Use cases of Fine-tuning

Tune-in to Simform to fine-tune a large language model

Businesses wishing to streamline their operations using the power of AI/ML have a plethora of options available now, thanks to large language models like GPT-3. However, fine-tuning is essential to realize the full potential of these models.

Fine-tuning in large language models (LLMs) involves re-training pre-trained models on specific datasets, allowing the model to adapt to the specific context of your business needs. This process can help you create highly accurate language models, tailored to your specific business use cases.

Want to build intelligent applications?

Develop smart applications with our AI/ML services.

Got any questions? We're happy to help. 😊

your business.

[GET IN TOUCH](#)

Simform is a leading [AI/ML development services provider](#), specializing in building custom AI solutions. [Contact us](#) to harness the transformative power of AI for your business.

Fundamentals of Fine Tuning

Before we begin with the actual process of fine-tuning, let's get some basics clear.

What is fine-tuning?

Fine-tuning involves updating the weights of a pre-trained language model on a new task and dataset.

The key distinction between training and fine-tuning is that training starts from scratch with a randomly initialized model dedicated to a particular task and dataset. On the other hand, fine-tuning adds to a pre-trained model and modifies its weights to achieve better performance.

Take the task of performing a sentiment analysis on movie reviews as an illustration. Instead of training a model from scratch, you may leverage a pre-trained language model such as GPT-3 that has already been trained on a vast corpus of text. To fine-tune the model for the specific goal of sentiment analysis, you would use a smaller dataset of movie reviews. This way, the model can be trained to analyze sentiment.

Compared to starting from zero, fine-tuning has a number of benefits, including a shorter training period and the capacity to produce cutting-edge outcomes with less data. We will delve deeper into the process of fine-tuning in the parts that follow. But first, let's explore the different fine-tuning examples.

Want To Know How Do Large Language Models Work?

[I WANT TO READ](#)

When Do You Need to Fine-Tune?

Deciding when to fine-tune a large language model depends on the specific task and dataset you are working with.

In general, fine-tuning is most effective when you have a small dataset and the pre-trained model is already trained on a similar task or domain.

Following are some situations where fine-tuning may be necessary:



domain. For instance, you may fine-tune a model pre-trained on a huge corpus of new items to categorize a smaller dataset of scientific papers by topic.

2. Data security and compliance

When you want to customize and refine the models' parameters to align with evolving threats and regulatory changes. For instance, when a new data breach method arises, you may fine-tune a model to bolster organizations defenses and ensure adherence to updated data protection regulations.

3. Customization

When you want to customize a pre-trained model to better suit your specific use case. You may, for instance, fine-tune a question-answering model that has already been trained on customer support requests to improve responsiveness to frequent client inquiries.

4. Domain-specific tasks

When you have a specific task that requires knowledge of a certain domain or industry. For instance, if you are working on a task that involves the examination of legal documents, you may increase the accuracy of a pre-trained model on a dataset of legal documents.

5. Limited Labeled Data

If you have a small amount of labeled data, modifying a pre-trained language model can improve its performance for your particular task. Suppose you are developing a chatbot that must comprehend customer enquiries. By fine-tuning a pre-trained language model like GPT-3 with a modest dataset of labeled client questions, you can enhance its capabilities.

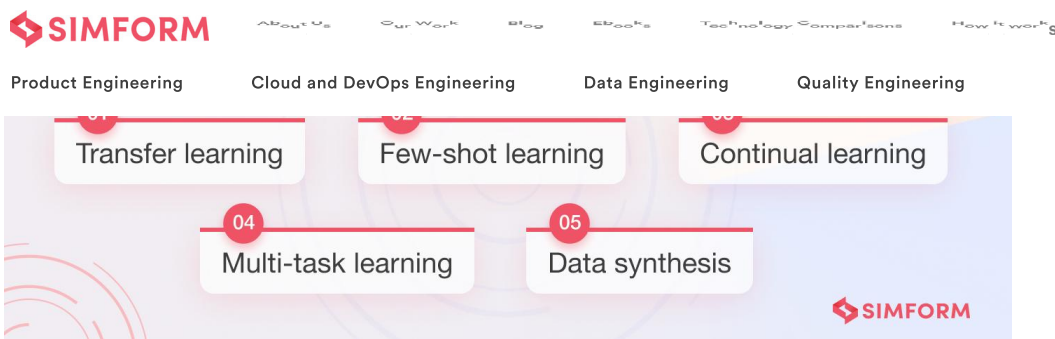
However, if you have a huge dataset and are working on a completely new task or area, training a language model from scratch rather than fine-tuning a pre-trained model might be more efficient.

In general, the specific use case and dataset determine whether to fine-tune or train a language model from scratch. Prior to choosing, it's crucial to carefully weigh the benefits and drawbacks of both strategies.

There are numerous techniques for gathering training data for large language models in addition to fine-tuning. Here, we highlight some of the key ones.

Techniques for Data Training

The data training techniques include data synthesis, continual learning, transfer learning, one-shot learning, few-shot learning, and multitask learning.



Contact Us

1. Transfer learning

Transfer learning involves training a model on a large dataset and then applying what it has learnt to a smaller, related dataset. The effectiveness of this strategy has been demonstrated in tasks involving NLP, such as text classification, sentiment analysis, and machine translation.

2. Few-shot learning

Few-shot learning enables a model to categorize new classes using just a few training instances. For instance, the model can accurately generalize and categorize more photos of a rare bird species with just a small number of bird images.

3. Continual learning

Continuous learning trains a model on a series of tasks, retaining what it has learnt from previous tasks and adapting to new ones. This method is helpful for applications where the model needs to learn continuously, like chatbots that gather information from user interactions.

4. Multi-task learning

Multitask learning trains a model to do several different tasks at once. This method is effective for tasks where the model needs to use data from various sources, such as question answering.

5. Data synthesis

Data synthesis involves generating new training data using techniques such as data augmentation or data generation. Data augmentation modifies existing training examples by adding noise or perturbing the text to create new examples. Data generation employs a generative model to create new examples that are similar to the training data.

To choose the best technique for your project, you need to consider these factors:

Complexity

Efficacy

Cost

Time



resources and time.

Multi-task learning can fine-tune models for multiple related tasks at once. Data synthesis can help with tasks where obtaining real-world data is challenging or expensive.

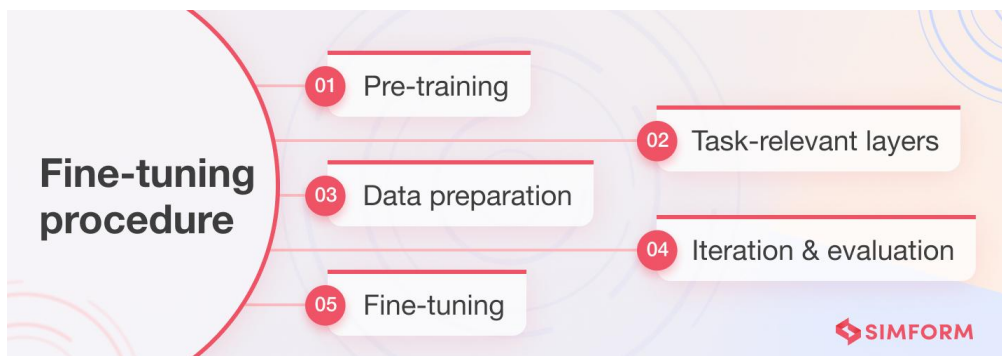
Ultimately, the choice of fine-tuning technique will depend on the specific requirements and constraints of the task at hand.

Fine-tuning has many benefits compared to other data training techniques. It leverages a large language model's pre-trained knowledge to capture rich semantic data without human feature engineering. It trains the model on labeled data to fit certain tasks, making it versatile for many NLP activities.

Fine-tuning needs fewer computational resources than training models from scratch. It generalizes well to unknown data from initial pre-training on a varied corpus, and balances general and task-specific learnings, making it the best strategy for training large language models.

Fine-tuning Procedure

The process of fine-tuning entails five main steps, which are explained below.



1. Pre-training

Pre-training is the first step in the process of adjusting huge language models. It involves teaching a language model the statistical patterns and grammatical structures from a huge corpus of text data, such as books, articles, and websites. Then, the fine-tuning procedure starts with this pre-trained model, such as GPT-3 or BERT.

For instance, the [GPT-3 model by OpenAI](#) was pre-trained using a vast dataset of 570GB of text from the internet. By exposure to a diverse range of textual information during pre-training, it learned to generate logical and contextually appropriate responses to prompts.

Because pre-training allows the model to develop a general grasp of language before being adapted to particular downstream tasks, it serves as a vital starting point for fine-tuning.

training. These extra layers modify the learned representations for a particular job on top of the pre-trained model.

For example, the task-specific layers for sentiment analysis would classify text into positive, negative, or neutral sentiment categories.

These layers help the pre-trained model leverage its general language knowledge while specializing in the target task.

3. Data preparation

Data preparation involves gathering and preprocessing the data used to fine-tune the large language model.

Ensuring that the data reflects the intended task or domain is crucial in the data preparation process.

For example, when fine-tuning a language model for [sentiment analysis on social media](#) data, the data preparation phase gathers a diverse range of social media posts labeled with sentiment categories (positive, negative, neutral). This eliminates noise, handles missing values, and standardizes the format.

You can also use data augmentation techniques to increase the diversity and quantity of the training data.

Data preparation provides pertinent and representative training data and establishes the groundwork for effective fine-tuning. The model learns task-specific patterns and nuances from this data.

4. Fine-tuning

Fine-tuning is the core step in refining large language models for specific tasks or domains. It entails adapting the pre-trained model's learned representations to the target task by training it on task-specific data. This process enhances the model's performance and equips it with task-specific capabilities.

For instance, to construct a specialized legal language model, a large language model pre-trained on a sizable corpus of text data can be refined on a smaller, domain-specific dataset of legal documents. The improved model would then be more adept at comprehending legal jargon accurately.

The size of the task-specific dataset, how similar the task is to the pre-training target, and the computational resources available all affect how long and complicated the fine-tuning procedure is.

Large language models can be fine-tuned to function well in particular tasks, leading to better performance, more accuracy, and better alignment with the intended application or domain.

increase their efficacy.

During this phase, the refined model is tested on a different validation or test dataset. This assessment helps determine the model's success in the intended task or domain, pinpointing areas in need of development. Evaluation metrics such as accuracy, precision, recall, and F1 score are frequently utilized to assess model performance.

Let's take image classification as an example to demonstrate the application of these evaluation metrics:

Accuracy: It measures the percentage of correctly classified images out of the total number of images in the evaluation dataset. For example, if a model correctly identifies 80 out of 100 images, the accuracy would be 80%.

Precision: It evaluates the proportion of correctly predicted positive instances out of all instances predicted as positive. For instance, if the model predicts 50 images as containing a specific object and 45 of them are correct, the precision would be 90%.

Recall: In image classification, it represents the proportion of correctly predicted positive instances out of all instances that should have been predicted as positive. For example, if there are 100 images containing a specific object, and the model correctly identifies 80 of them, the recall would be 80%.

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced measure of performance. In image classification, a high F1 score indicates a model that performs well in both avoiding false positives and false negatives. It is particularly useful when there is an imbalance between different classes in the dataset.

One can enhance the fine-tuned model based on evaluation results through iterations. This includes modifying the architecture, increasing training data, adjusting optimization methods, and fine-tuning hyperparameters.

For example, in sentiment analysis, iterations may include tuning hyperparameters like learning rate, batch size, or regularization techniques to enhance the model's performance. Additional techniques like data augmentation or transfer learning can also be explored for further improvement.

Through a continuous loop of evaluation and iteration, the model is refined until the desired performance is achieved. This iterative process ensures enhanced accuracy, robustness, and generalization capabilities of the fine-tuned model for the specific task or domain.

Interested in generative AI?

[CHECK OUT THIS BLOG.](#)

1. Transfer learning

In machine learning, the practice of using a model developed for one task as the basis for another is known as transfer learning. A pre-trained model, such as GPT-3, is utilized as the starting point for the new task to be fine-tuned. Compared to starting from scratch, this allows for faster convergence and better outcomes. Using a pre-trained convolutional neural network, initially trained on a large dataset of images, as a starting point for a new task of classifying different species of flowers with a smaller labeled dataset.

2. Sequential fine-tuning

Sequential fine-tuning refers to the process of training a language model on one task and subsequently refining it through incremental adjustments. For example, a language model initially trained on a diverse range of text can be further enhanced for a specific task, such as question answering. This way, the model can improve and adapt to different domains and applications. For example training a language model on a general text corpus and then fine-tuning it on medical literature to improve performance in medical text understanding.

3. Task-specific fine-tuning

Task-specific fine-tuning adjusts a pre-trained model for a specific task, such as sentiment analysis or language translation. The model requires more data and time than transfer learning. However, it improves accuracy and performance by tailoring to the particular task. For example, a highly accurate sentiment analysis classifier can be created by fine-tuning a pre-trained model like BERT on a large sentiment analysis dataset.

4. Multi-task learning

Multi-task learning trains a single model to carry out several tasks at once. When tasks have similar characteristics, this method can be helpful and enhance the model's overall performance. For example, training a single model to perform named entity recognition, part-of-speech tagging, and syntactic parsing simultaneously to improve overall natural language understanding.

5. Adaptive fine-tuning

In adaptive fine-tuning, the learning rate is dynamically changed while the model is being tuned to enhance performance. By doing this, you can avoid overfitting. For example adjusting the learning rate dynamically during fine-tuning to prevent overfitting and achieve better performance on a specific task, such as image classification.

6. Behavioral fine-tuning

Behavioral fine-tuning incorporates behavioral data into the process. For example, data from user interactions with a chatbot might improve a language model to enhance conversational capabilities. For instance, the fine-tuning process can enhance the model's conversational capabilities by incorporating user interactions and conversations with a chatbot.



fewer resources. This technique is known as parameter efficient fine-tuning. For example, decreasing the size of a pre-trained language model like GPT-3 by removing unnecessary layers to make it smaller and more resource-friendly while maintaining its performance on text generation tasks.

8. Text-text fine-tuning

The text-text fine-tuning technique tunes a model using pairs of input and output text. This can be helpful when the input and output are both texts, like in language translation. For example, a language model can improve its accuracy in English-to-French translation tasks by fine-tuning using text-text fine-tuning with pairs of English sentences as input and their corresponding French translations as output.

Fine-tuning Best Practices

Here are a few fine-tuning best practices that might help you incorporate it into your project more effectively.

1. Start with a pre-trained model

It takes a significant amount of computational power and data to fine-tune a large language model from scratch. So it's typically more effective to begin with a model that has already had extensive general language training. You can greatly reduce your time and effort spent on fine-tuning by doing this. You may, for instance, fine-tune the pre-trained GPT-3 model from OpenAI for a particular purpose.

2. Use a more compact model

Large language models can produce spectacular results, but they also take a lot of time and money to perfect. So it makes sense to start with smaller models. You might save time, money, and computer resources by doing this. For a smaller project, for instance, GPT-2 can be used in place of GPT-3.

3. Ensure ethical compliance

When utilizing fine-tuned language models, it is crucial to prioritize ethical considerations, especially regarding privacy, security, and bias. For example, if a language model is fine-tuned for sentiment analysis, it is important to ensure that the training data is diverse and representative to avoid perpetuating biased or discriminatory outcomes. As a best practice, it is essential to proactively assess and address the potential ethical implications associated with the use of such models. Taking necessary measures to mitigate any potential harm should be a fundamental aspect of employing fine-tuned language models responsibly.

4. Play around with various prompt formats

The prompt, which you supply to the model as input text, has a significant impact on the quality of the results that are produced. Therefore, it's crucial to test out several prompt types to identify which ones are most effective for your task. For example, you can try providing the model with a complete sentence or a partial sentence, or use different types of prompts for different parts of your task.



task. For example, if fine-tuning a language model for sentiment analysis, using a dataset of movie reviews or social media posts would be more relevant than a dataset of news articles.

6. Utilize the appropriate evaluation metric

It's critical to pick the appropriate assessment metric for your fine tuning work because different metrics are appropriate for various language model types. For example, accuracy or F1 score might be useful metrics to utilize while fine-tuning a language model for sentiment analysis.

7. Use a learning rate schedule

Fine tuning a large language model can be a time-consuming process, and using a learning rate schedule can help speed up convergence. A learning rate schedule adjusts the learning rate during training, allowing the model to learn quickly at the start of training and then gradually slowing down as it gets closer to convergence.

8. Regularize your model

Regularization techniques like dropout and weight decay can help prevent overfitting during fine tuning. By adding a regularization term to the loss function, the model is encouraged to learn simpler and more generalizable representations.

9. Evaluate early and often

It's a good practice to evaluate the performance of the fine-tuned model early and often during training. This helps identify issues early on and make necessary adjustments to the training process.

10. Experiment with architectures

Various architectures may perform better than others depending on the task. To determine which architecture is ideal for your particular purpose, try out a few alternatives, such as transformer-based models or recurrent neural networks.

11. Employ ensembling

Ensembling is the process of combining multiple models to improve performance. Fine tuning multiple models with different hyperparameters and ensembling their outputs can help improve the final performance of the model.

12. Longer fine-tuning

In certain circumstances, it could be advantageous to fine-tune the model for a longer duration to get better performance. While choosing the duration of fine-tuning, you should consider the danger of overfitting the training data.

Use cases of Fine-tuning

Here are some examples of fine-tuning in action.



negative, or neutral. GPT can be fine-tuned for this task using a labeled dataset of customer reviews or social media posts.

Bloomberg has developed **BloombergGPT**, a specialized language model for the financial industry. By training BloombergGPT on a dataset of financial news articles, it achieves an accuracy of over 90% in sentiment classification.

2. Question answering

Question answering involves answering questions posed in natural language. To fine-tune GPT for question answering, we train it on a dataset containing question-answer pairs.

Microsoft has developed Turing NLG, a GPT-based model designed specifically for question answering tasks.

3. Text summarization

Text summarization entails generating a concise version of a text while retaining the most crucial information. To fine-tune GPT for text summarization, we train it on a dataset comprising text and their corresponding summaries.

For example, Google has developed T5, a GPT-based model optimized for text summarization tasks.

Tune-in to Simform to fine-tune a large language model



model requires AI/ML expertise to achieve exceptional results. Simform, a leading **AI/ML service** provider, has access to Hiren is VP of Technology at Simform with an extensive experience in helping enterprises and startups streamline their business performance through data-driven innovation. As your partner, you can confidentially navigate through the offer u all support in customizing and optimizing models for specific tasks and domains.

Whether it's sentiment analysis, question answering, or text summarization, Simform's team of skilled professionals assists businesses in achieving outstanding performance in a wide range of natural language processing applications.

Your comment here*

Partner with Simform, and gain access to AI consultants who understand the nuances of large language models.

For exclusive strategies not found on the blog

Sign up today!

SUBSCRIBE NOW

Related Posts



ChatGPT Plugins:
Are AI-
Powered Chatbots the
Future?



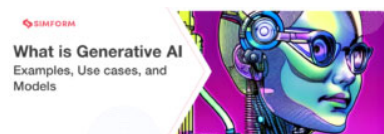
Hiren Dhaduk
APRIL 18, 2023



Applications of NLP:
Unlocking the Potential of AI
in Diverse Industries



Hiren Dhaduk
APRIL 21, 2023



What is Generative AI:
Exploring Examples, Use
Cases, and Models



Hiren Dhaduk
MAY 16, 2023

Areas We Serve

San Francisco

3 Market Street, Suite
100 San Francisco, CA 94111
USA

[Call us now](#)

Canada

15-115 – Bentall II Vancouver BC,
V7X 1M8

[Call us now](#)

India

501 Binori B Sq-2, Nr
DoubleTree By Hilton, Bopal Rd,
Ahmedabad-380054, India

[Call us now](#)

Los Angeles

700 S Flower Street, Suite
1000, Los Angeles, CA 90017
USA

[Call us now](#)



Product Engineering

Cloud and DevOps Engineering

Data Engineering

Quality Engineering

About Us

Our Partners

How Simform works

Areas We Serve

Careers

Contact us

Product Engineering

Cloud and DevOps Engineering

Data Engineering

Quality Engineering

Talent Solutions

Blog

Ebooks

Case Studies

