

# Fine-Tuning for Single Task Improvement

## Introduction

While large language models (LLMs) excel at various language tasks, your application might require optimization for a specific task. Fine-tuning a pre-trained model can enhance its performance on the task of interest, such as summarization, with relatively few examples.

## Benefits of Single Task Fine-Tuning

**Efficiency:** Achieve good performance with just 500-1,000 task-specific examples, compared to the extensive pre-training data.

**Focused Improvement:** Directly improve performance on the target task without affecting other tasks.

## Challenges: Catastrophic Forgetting

Fine-tuning on a single task can lead to catastrophic forgetting, where the model's performance on previously learned tasks deteriorates. This occurs because full fine-tuning modifies the original LLM's weights, prioritizing performance on the fine-tuning task at the expense of other tasks.

## Mitigating Catastrophic Forgetting

**Evaluate Use Case:** Determine if maintaining multitask capabilities is crucial for your application. If not, catastrophic forgetting may not pose an issue.

**Multitask Fine-Tuning:** Fine-tune the model on multiple tasks simultaneously to maintain its ability to generalize across tasks. This approach requires a larger dataset and computational resources.

## Parameter Efficient Fine-Tuning (PEFT)

**Preservation of Weights:** PEFT techniques preserve most of the pre-trained weights and only train task-specific adapter layers and parameters.

**Robustness to Forgetting:** PEFT minimizes the risk of catastrophic forgetting, making it suitable for scenarios where multitask capabilities are essential.

# Conclusion

Fine-tuning LLMs for single tasks offers targeted performance improvements but may risk catastrophic forgetting. Understanding your use case's requirements helps determine the appropriate fine-tuning approach, whether it's multitask fine-tuning or parameter efficient fine-tuning.