Assignment: Fine-Tuning Theory and Practice

By Ajay Sethuraman

Assignment Objectives

By the end of this assignment, you should be able to:

- Explain the advantages and limitations of fine-tuning.
- Prepare a dataset for fine-tuning by cleaning and curating it effectively.
- Describe the process of transfer learning in the context of LLMs.
- Fine-tune a lightweight LLM on a specific task using Python and Hugging Face.
- Evaluate the performance of a fine-tuned model using appropriate metrics.

Part 1: Theory of Fine-Tuning

Concept Check (Multiple Choice Questions)

1. What is the main benefit of fine-tuning an LLM?
   A) It improves the model's speed.
   B) It customizes the model for specific tasks or domains.
   C) It eliminates the need for high-quality datasets.
   D) It prevents overfitting entirely.
   (Correct Answer: B)
2. Which of the following describes "catastrophic forgetting"?
   A) When the model forgets its pre-training data during inference.
   B) When the model loses its generalization ability after excessive fine-tuning on a specific task.
   C) When the model produces irrelevant outputs due to overfitting.
   D) When the model fails to save fine-tuned weights.
   (Correct Answer: B)

Application Task

1. Write a 150–200 word explanation of transfer learning using a real-world analogy. Use examples from any domain (e.g., healthcare, legal, e-commerce).
2. Provide an example dataset structure for a fine-tuning task of your choice. Label and clean your dataset to match the requirements for the task.

Part 2: Practical Fine-Tuning Session

Hands-On Coding Task

Fine-tune the `distilbert-base-uncased` model for text classification using Hugging Face, as demonstrated in the lesson. Complete the following steps:

1. **Environment Setup:** Write the commands to install the required libraries and verify GPU availability.
2. **Preprocessing Data:** Demonstrate how to load and preprocess the IMDB dataset for tokenization.
3. **Model Training:** Define the training arguments and use Hugging Face's `Trainer` to fine-tune the model.
4. **Save and Evaluate:** Save the fine-tuned model and evaluate its accuracy on the test set.

Reflection

- Summarize the key challenges you faced during the fine-tuning process and how you addressed them.
- Provide suggestions for improving the model's performance if the accuracy was below 90%.

Good luck with your assignments and projects! Happy fine-tuning!

Part 1: Theory of Fine-Tuning

Concept Check (Multiple Choice Questions)

1. What is the main benefit of fine-tuning an LLM?
    - A) It improves the model's speed.
    - B) It customizes the model for specific tasks or domains.
    - C) It eliminates the need for high-quality datasets.
    - D) It prevents overfitting entirely.
    - **Correct Answer: B**
2. Which of the following describes "catastrophic forgetting"?
    - A) When the model forgets its pre-training data during inference.
    - B) When the model loses its generalization ability after excessive fine-tuning on a specific task.
    - C) When the model produces irrelevant outputs due to overfitting.
    - D) When the model fails to save fine-tuned weights.
    - **Correct Answer: B**

Application Task

Explanation of Transfer Learning Using a Real-World Analogy

Transfer learning is similar to a cybersecurity professional specializing in threat detection. Imagine an experienced security analyst who has worked in various industries, such as banking, healthcare, and government agencies. While each industry has its own unique security challenges, the analyst has developed a strong foundation in identifying vulnerabilities, detecting intrusions, and responding to cyber threats.

Now, suppose this analyst moves to a new role focused specifically on cloud security. Instead of learning everything from scratch, they leverage their prior knowledge of cybersecurity principles and adapt it to the cloud environment. They refine their expertise by studying cloud-specific threats, such as misconfigured storage buckets and identity-based attacks. This adaptation process is akin to fine-tuning in machine learning—a pre-trained language model is adjusted with new, domain-specific data to enhance its performance on a targeted task without losing its general knowledge.

For example, an LLM initially trained on general text data can be fine-tuned to detect phishing emails. By training the model on a dataset containing real phishing attempts and legitimate emails, it becomes highly effective at distinguishing between the two, while still retaining its broader language understanding capabilities.

Example Dataset Structure for Fine-Tuning a Cybersecurity Model

Task: Phishing Email Detection

Email_ID	Subject	Body	Label
001	"Urgent: Verify Your Account"	"Your account has been compromised. Click the link below to secure it."	Phishing
002	"Meeting Agenda for Tomorrow"	"Here is the agenda for our team meeting scheduled at 10 AM."	Legitimate
003	"Security Alert: Suspicious Login Detected"	"We detected an unusual login attempt from a new device. Confirm your identity now."	Phishing
004	"Weekly Newsletter"	"Here is your weekly update with industry news and insights."	Legitimate

Data Cleaning Process:

1. **Remove Duplicate Emails:** Ensure that no duplicate email entries exist to avoid bias in training.
2. **Standardize Text Formatting:** Convert all text to lowercase, remove special characters, and normalize spacing to maintain consistency.
3. **Filter Out Empty or Incomplete Entries:** Ensure that every row contains relevant text data for analysis.
4. **Balance Dataset:** Maintain an equal number of phishing and legitimate emails to prevent class imbalance and biased learning.
5. **Tokenization & Embedding:** Convert text into tokenized format for compatibility with deep learning models.

By fine-tuning an LLM on this structured dataset, it can effectively learn patterns associated with phishing attacks, improving email security systems and reducing cybersecurity risks.
