Week 5 - Technical Assignment Fine tuning using LoRA and PEFT

Objective

This assignment provides **practical experience in running, debugging, fine-tuning, and testing an LLM** inside a Jupyter Notebook. The key tasks include:

- Running a **fine-tuning notebook** for an instruction-following LLM.
- **Fixing potential errors** that may occur during execution.
- Training and **evaluating the model within the notebook**.
- Saving and **uploading the model to Hugging Face**.
- **Testing inside the notebook**
- **Documenting findings & submitting results via notebook**.

Instructions

Copy-Paste Assignment – Read Carefully

- **This is a Copy-Paste Assignment**, meaning you will **clone and run the provided notebook** with small modifications.
- **You may face errors** when running the notebook.
- Your task is to **fix these errors, document the fixes, and explain the process**.
- **You may run the notebook in Google Colab, Jupyter,or Kaggle.**
- **Kaggle is good.**

How to Complete the Assignment

1. **Clone the Jupyter Notebook** from the following link:
   🔗 Kaggle Notebook
2. Run each step, and **fix errors**, if you face them.
3. Clone hugging face small language models, like LLaMA 1.1B (Tiny Model), or any small model for training. If you do not get a success you can use the same model and need to explain the reason.
4. You can optimized parameters to get better results.
5. **Train the fine-tuned model** with the following configuration
   a. `max_steps = 500`
   b. `logging_steps = 50`
   c. `eval_steps = 50`

6. Save and **upload the trained model to Hugging Face**. Share a screenshot of the hugging face model
7. **Test** all label cases.
8. **Capture test results** and add them to your **Jupyter Notebook report**.
9. **Submit your work** (Notebook + Code).

Submission Guidelines

- **Code Submission:** Push your final notebook to **GitHub**.
- **Notebook Submission:** Include **error fixes, test results, and model evaluation**.

15-Point Exercise Steps

1. **Install Required Libraries**
    - Install transformers, datasets, peft, bitsandbytes, etc.
2. **Load the Dataset from hugging face()**
    - Import and inspect the provided dataset for fine-tuning.
3. **Create Bitsandbytes Configuration**
    - Set up quantization with bitsandbytes for efficient training.
4. **Load the Pre-Trained Model**
    - Load LLaMA 1.1B or another small language model.
5. **Tokenization**
    - Apply appropriate tokenization for the dataset.
6. **Test the Model with Zero-Shot Inference**
    - Run a few samples to see the base model's performance before fine-tuning.
7. **Pre-process the Dataset**
    - Clean, format, and prepare the dataset for training.
8. **Prepare the Model for QLoRA**
    - Enable gradient checkpointing and quantization preparation.
9. **Set Up PEFT for Fine-Tuning**
    - Configure LoRA parameters and apply to the model.
10. **Train PEFT Adapter**
    - Fine-tune the model using the PEFT configuration.
11. **Evaluate the Model Qualitatively**
    - Perform manual evaluation to assess output quality.
12. **Evaluate the Model Quantitatively (ROUGE Metric)**
    - Use ROUGE or other metrics for automated performance evaluation.
13. **Save and Upload the Model to Hugging Face**
    - Push the trained model to your Hugging Face account.
14. **Capture and Document Results**
    - Include screenshots of the uploaded model, sample outputs, and analysis.

15. **Submit the Assignment**
    - Upload the notebook to GitHub with a README, including error fixes, evaluation, and model details.

Expected Deliverables

✅ **GitHub Repository** with:

- **Code (Notebook)**
- **Errors encountered & fixes if applied**
- **Snapshots of hugging face model and link to publicly available.**