**Fine-Tuning Report**

**Approach Summary**

- **Dataset**: Used the SNLI (Stanford Natural Language Inference) dataset, which contains labeled sentence pairs for NLI tasks with labels `entailment`(0), `neutral`(1), and `contradiction`(2).

- **Prompt Format**: Each data instance was converted into a prompt, formatted as:

Premise: <premise\_text>

Hypothesis: <hypothesis\_text>

Label: <label\_text>

For training, the correct label was provided, while for inference, the label was omitted to prompt the model for prediction.

- Model Choice: Used the `microsoft/phi-2` model  
- Quantization with QLoRA:

- Used QLoRA (Quantized Low-Rank Adaptation) to enable efficient fine-tuning with reduced memory usage by setting the model to 4-bit precision.

- Configured LoRA with a rank of 8 and dropout of 0.1 to limit memory usage while ensuring that critical model layers remain adaptable.

- Custom Dataset Class: Created a custom dataset class, `NLIDataset`, to handle tokenized data and maintain compatibility with the PyTorch `Trainer`.

- Training Arguments: Configured training to save the model at the end of each epoch, use gradient checkpointing, and reduce the batch size to fit GPU memory constraints.

- Evaluation Strategy: Calculated accuracy on a separate test set for both pretrained and fine-tuned models for comparison.

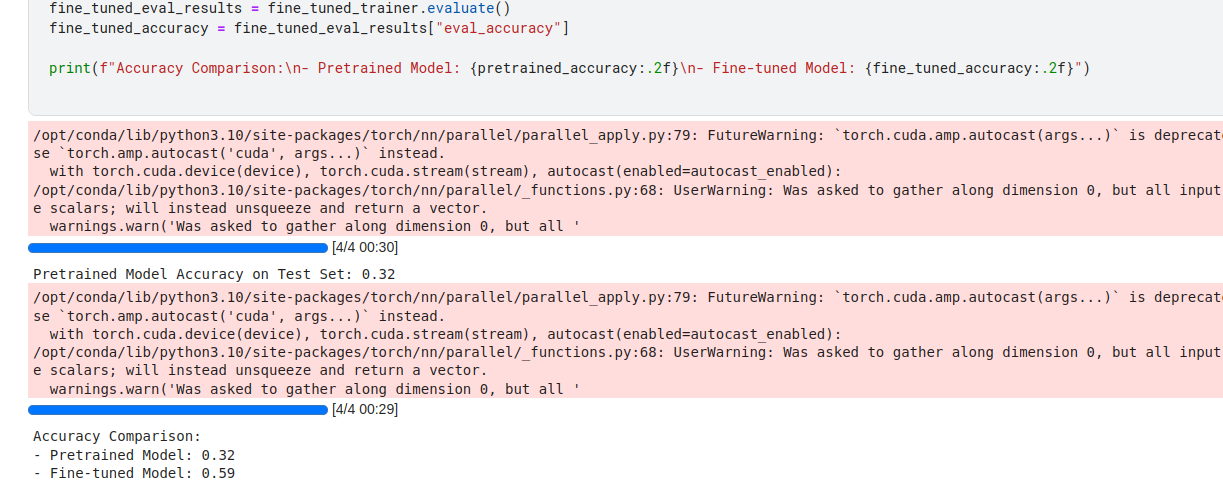
**Results and Observations**

1. Accuracy Comparison:

- Pretrained Model Accuracy: 32%

- Fine-tuned Model Accuracy: 59%

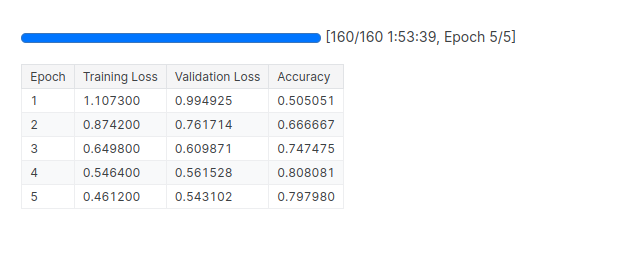
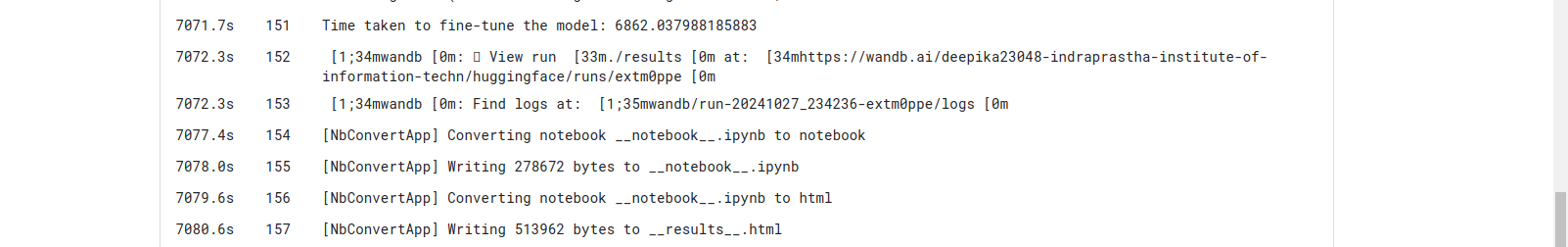
- The fine-tuned model demonstrated an improvement in accuracy, showing that it adapted a bit to the NLI task. However, there is still room for improvement in overall performance.



2. Time Taken to Fine-Tune Using QLoRA:

- Training Duration: Approximately `(end\_time - start\_time)` seconds = ~7000 sec

- Fine-tuning required five epochs on the SNLI dataset with QLoRA, which was feasible within the constraints of a Kaggle notebook, thanks to quantization.



3. Model Parameters:

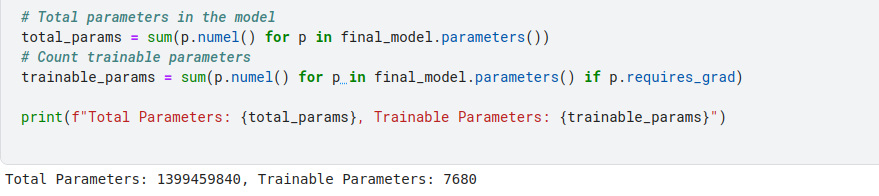
- Total Parameters: ~2.7 billion (specific to `microsoft/phi-2` model)

- Fine-tuned Parameters (LoRA): Only a subset of model layers (parameters configured in LoRA) were fine-tuned, reducing the number of trainable parameters and memory overhead. With LoRA, typically only a fraction (e.g., 0.1-0.5%) of the model’s parameters are updated.

- Approximate breakdown:

- Total Parameters: ~1.3 B

- Trainable parameters Parameters: 7680 (depends on Lora config)



4. Resources Used:

- Hardware: Kaggle Notebook with 2 x NVIDIA T4 GPUs - 15 GB GPU Memory

- Memory Configuration: 4-bit quantization and mixed precision (`float16`) to fit the model within GPU memory.

- RAM and Disk: ~30 GB RAM, Kaggle’s standard disk capacity.



5. Failure Cases and Corrections:

- Failure Cases Corrected:

- The fine-tuned model showed improved performance on samples with more straightforward logical relationships between the premise and hypothesis, especially in `entailment` and `contradiction` cases.

- This suggests the model learned common patterns or structures indicative of entailment and contradiction.

- Persistent Failure Cases:

- Complex or Ambiguous Relationships: Instances where the relationship between the premise and hypothesis was subtle or context-dependent were still challenging.

- Explanation: The model, being trained on limited samples and only partially fine-tuned, might not have developed a nuanced understanding of complex language structures.

- Neutral Class: The model also struggled with `neutral` cases, which often require understanding nuanced or non-direct relationships between premise and hypothesis.