Amrita Vishwa Vidyapeetham
Amrita School of Computing, Amritapuri
BTech, Third Year, 6 Semester
Lab Sheet 4

---

1. Load a QA or instruction-following dataset using HuggingFace's `datasets` library. Display the first 3 examples.
2. Write a tokenizer function using the LLaMA tokenizer to tokenize an instruction and input pair. Apply it and show the tokenized output.
3. Convert a custom dataset in CSV format with `instruction`, `input`, and `output` columns into HuggingFace `DatasetDict`. Display 5 examples.
4. Split your dataset into training and validation sets using `train_test_split` with an 80:20 ratio. Show the size of each split.
5. Load a pretrained `LLaMA-7B` model using HuggingFace's `transformers` library with appropriate quantization (e.g., 4-bit or 8-bit loading using `bitsandbytes`).
6. Initialize PEFT configuration using LoRA. Specify the rank, alpha, and target modules for injection (e.g., `q_proj`, `k_proj`).
7. Use the `prepare_model_for_kbit_training()` function to enable LoRA tuning of LLaMA under low-bit training.
8. Define and configure `TrainingArguments` for 3 epochs, batch size 8, gradient accumulation, and proper logging using HuggingFace `transformers`.
9. Fine-tune the LLaMA model on your custom dataset using `Trainer` or `SFTTrainer` (Supervised Fine-Tuning Trainer from `trl`).
10. Monitor training loss and evaluation loss per epoch. Save a plot showing the trend of both losses across epochs.
11. Evaluate the fine-tuned model by generating responses for 5 unseen prompts from the validation set and compute BLEU or ROUGE scores.
12. Save your LoRA-adapted model and tokenizer separately. Demonstrate how to reload the model later for inference.
13. Compare the training time and memory usage between full LLaMA fine-tuning and LoRA-based fine-tuning using profiling tools or logs.
14. Replace the base model with `NousResearch/LLaMA2-7B-HF` or another instruction-tuned variant. Fine-tune and compare the performance on the same dataset.

15. Integrate Prefix-Tuning instead of LoRA using PEFT. Show how performance and training time differ for the same dataset.

16. Fine-tune LLaMA on a domain-specific dataset (e.g., medical or legal Q&A). Generate a sample output showing domain-specific knowledge.

17. Design a pipeline to:

    - Load the dataset
    - Preprocess using tokenizer
    - Load base model with LoRA
    - Fine-tune using SFTTrainer
    - Evaluate using metrics

18. Deploy your fine-tuned LLaMA model using `Gradio` or `Streamlit` and demonstrate live inference with user input.