AlpaCare Medical Instruction Assistant

1. Introduction

The AlpaCare Medical Instruction Assistant is a LoRA-fine-tuned LLaMA-based model designed to follow medical instructions in natural language. The goal is to assist with medical education, summarization, and instruction-following, while minimizing the compute requirements for fine-tuning.

Key Points:

- Base Model: meta-llama/Llama-3.2-1B-Instruct
- Fine-tuning: LoRA (Low-Rank Adaptation) for parameter-efficient training
- Dataset: AlpaCare-MedInstruct-52k
- Frameworks: Hugging Face Transformers, PEFT, Datasets

2. Dataset

The dataset used for fine-tuning the AlpaCare Medical Instruction Assistant is hosted on Hugging Face under the name lavita/AlpaCare-MedInstruct-52k. It contains structured medical instructions with fields for instruction, input, and output. For this project, a subset of the dataset was used to fit Colab memory constraints: 2000 samples for training, 100 samples for validation, and 100 samples for testing. This subset ensures reproducibility while allowing safe experimentation within limited compute resources.

3. Preprocessing

The dataset was carefully preprocessed to ensure high-quality training samples for the fine-tuning pipeline. The steps included cleaning, normalization, and filtering to remove inconsistencies and low-quality data.

Key Steps:

- Loaded the dataset from Hugging Face and converted it into a pandas DataFrame.
- Removed rows with missing instruction or output fields.
- Dropped duplicate samples based on the combination of instruction, input, and output.
- Normalized text by:
    1. Stripping leading/trailing whitespace.
    2. Collapsing multiple spaces and newlines into single spaces.
    3. Standardizing quotation marks and apostrophes(“” → ", ‘’ → ").

    4. Removing unusual or non-printable characters.
- Filtered out samples with very short instructions or outputs (length ≤ 5 characters).
- Resulted in a clean, high-quality subset ready for tokenization and fine-tuning.

4. Model Training

The fine-tuning process utilized the meta-llama/Llama-3.2-1B-Instruct base model with LoRA (Low-Rank Adaptation) to efficiently adapt it to the AlpaCare dataset. FP16 precision was used for GPU efficiency without compromising stability.

Key steps:

- Base model: meta-llama/Llama-3.2-1B-Instruct
- LoRA configuration:
    - Rank (r): 16
    - Alpha: 32
    - Target modules: q_proj, v_proj
    - Dropout: 0.05
    - Bias: "none"
- Training arguments:
    - Batch size: 1
    - Gradient accumulation: 2 (effective batch size = 2)
    - Learning rate: 2e-4
    - Epochs: 5
    - FP16 precision
    - Logging every 20 steps
    - Model checkpoint saving every 100 steps
- Training environment: Colab GPU (T4/compatible)
- Subset selection:
    - Training: 2,000 samples
    - Validation: 100 samples

Training procedure:

1. Tokenization of combined instruction + input + output sequences.
2. LoRA adapter applied on top of base model.
3. Training for 5 epochs with gradient accumulation to prevent GPU OOM errors.
4. Checkpoints saved periodically for potential resumption or evaluation.

5. Dataset Splitting

After preprocessing, the dataset was divided into training, validation, and test subsets to ensure effective fine-tuning and evaluation. A small subset was chosen to accommodate Colab memory constraints while still maintaining meaningful learning signals.

Key Details:

- Training set: 2,000 samples used for model fine-tuning.
- Validation set: 100 samples used for monitoring training performance and early stopping.
- Test set: 100 samples reserved for final evaluation of model generalization.
- Splits were randomly selected but kept consistent for reproducibility.
- The split ratio approximates 90% train, 5% validation, and 5% test, following standard best practices.

6. Tokenization and LoRA Fine-Tuning

Before fine-tuning, the dataset was tokenized and prepared to align model inputs with the instruction-response structure. Tokenization involved:

- Formatting prompts to include Instruction, optional Input, and Answer sections.
- Truncating sequences to a maximum length of 512 tokens to ensure compatibility with Colab GPU memory.
- Masking the prompt tokens in the labels so that the loss is computed only on the response tokens.
- Adding the end-of-sequence token to mark completion of each sample.

For efficient model adaptation, LoRA (Low-Rank Adaptation) was applied on top of the base LLaMA-1B model. Key steps included:

- Configuring LoRA with rank (r) = 16, alpha = 32, and dropout = 0.05.
- Targeting the query (q_proj) and value (v_proj) projection layers in attention modules for low-rank updates.
- Performing FP16 training to reduce memory consumption and enable larger batch processing.
- Fine-tuning for 5 epochs with gradient accumulation to effectively increase batch size without exhausting GPU memory.
- Only adapter weights were updated, leaving the base model parameters frozen to maintain pre-trained knowledge.

This process enabled memory-efficient, high-quality fine-tuning while preserving the general capabilities of the LLaMA base model.

7. Model Evaluation

After fine-tuning, the LoRA-adapted LLaMA-1B model was evaluated on a held-out validation and test subset of the dataset to assess its instruction-following and medical reasoning capabilities. Evaluation focused on generating responses to unseen instructions and comparing them qualitatively and quantitatively against reference outputs. Key points include:

- Validation setup: A subset of 100 samples was used to monitor model performance during training and adjust hyperparameters if necessary.
- Test set evaluation: 100 unseen samples were reserved for final assessment to measure generalization.
- Metrics and observations: Generated responses were checked for correctness, completeness, and clarity in medical instructions, ensuring alignment with expected outputs.
- Inference mode: The model was loaded in FP16 for efficient inference while retaining high-quality generation.
- Disclaimer inclusion: All outputs were reviewed with a cautionary disclaimer emphasizing that the model is intended for informational purposes and should not replace professional medical advice.

This evaluation ensured that the fine-tuned model produces coherent, contextually relevant, and safe responses on medical instruction tasks.

8. Safety and Mitigations

Given that the AlpaCare LoRA-adapted LLaMA-1B model deals with medical instructions, ensuring safety and mitigating risks is critical. Key considerations and actions taken include:

- Non-clinical advisory role: The model is explicitly intended for informational and educational purposes, not for diagnosing, prescribing, or providing personalized medical advice.
- Disclaimer on outputs: All generated responses include a cautionary statement reminding users to consult qualified healthcare professionals before acting on any information.
- Data curation: The training dataset was preprocessed to remove incomplete, duplicate, or malformed entries, reducing the likelihood of misleading outputs.
- Content filtering: Samples with extremely short or ambiguous instructions/outputs were excluded to improve clarity and minimize hallucinations.
- Frozen base model: Only LoRA adapter weights were fine-tuned, leaving the original LLaMA-1B parameters intact, preventing unintended behavior in unrelated domains.
- Validation and review: Model outputs were qualitatively evaluated for coherence, accuracy, and appropriateness in the context of medical instructions.

- Memory-efficient training safeguards: FP16 training and gradient accumulation were used to prevent GPU memory overflows, avoiding training interruptions that could corrupt model weights.
- Transparency and reproducibility: Checkpoints, adapter configurations, and preprocessing scripts were saved, allowing independent verification and safe replication of results.

By combining careful dataset curation, responsible usage disclaimers, and rigorous evaluation, the risks associated with medical misinformation are minimized while maintaining the model's instructional capabilities.