**Final Report**

Parameter Efficient Fine-tuning Experiments with Different Methods

**Priyanka Police Reddy Gari**

**PXP220104**

**Introduction:**

The provided are the findings of fine-tuning experiments carried out with different approaches on a range of NLP jobs. Task 1 has three tasks: using an alternate parameter-efficient fine-tuning method (Task 1, Part B), fine-tuning a model with LoRA (Task 1), and using QLoRA (Task 2) to fine-tune a model from the MTEB benchmark. Below are the methods used to choose the hyperparameters and shared observations and knowledge gained from the experiments.

**Tasks and Datasets Used:**

Task 1, Part A: Fine-tuning with LoRA was performed using the google/gemma-1.1-2b-it model on tweet emotion detection dataset.

Compared to conventional fine-tuning techniques, Low-Rank Adaptation (LoRA) dramatically lowers the number of trainable parameters and processing demands for large language models in a parameter-efficient manner.

Task 1, Part B: An alternative parameter-efficient fine-tuning method was employed on the same dataset as Task 1, Part A.

By adding inhibition and amplification mechanisms, (IA)³ Infused Adapter expands on the LoRA technique and allows for more efficient task-specific adaptation of neural network models that have already been trained for NLP tasks. It presents a viable method for optimizing models with enhanced functionality and capacity for generalization.

Task 2: Fine-tuning of a model from the MTEB benchmark using QLoRA was conducted. The model selection was based on the MTEB leaderboard.

I have used mistralai/Mistral-7B-Instruct-v0.2 method for task 2. It is an instructed version of the Mistral-7B-v0.2 generative language model, fine-tuned on various conversation datasets.

**Results:**

Below is a summary of the results obtained for each task:

**Task 1, Part A (LoRA):**

**Evaluation Metrics:**

Loss: 0.502

F1 Micro: 0.689

F1 Macro: 0.599

Accuracy: 0.205

**Task 1, Part B (Alternative Fine-tuning Method):**

**Evaluation Metrics:**

Loss: 0.590

F1 Micro: 0.643

F1 Macro: 0.535

Accuracy: 0.165

**Task 2 (QLoRA on MTEB Model):**

**Evaluation Metrics:**

Loss: 0.826

F1 Micro: 0.335

F1 Macro: 0.097

Accuracy: 0.116

**Hyperparameter Selection Strategies:**

Hyperparameters including learning rate, batch size, number of epochs, and optimizer were critical for fine-tuning.   
In order to effectively explore the hyperparameter space, grid search and random search were used.   
During training, the learning rate was adjusted using learning rate schedules such cosine annealing or linear decay.   
To avoid overfitting, regularization strategies like weight decay and dropout were used.

**Discussion:**

Task 1, Part A: Fine-tuning with LoRA yielded better performance compared to the alternative method in terms of F1 scores and accuracy. However, the runtime was shorter with LoRA, indicating its efficiency.

Task 1, Part B: The alternative fine-tuning method which was (IA)³ resulted in slightly lower performance metrics compared to LoRA. Further investigation into the differences in model architectures and optimization techniques could provide insights into this variation.

Task 2: Fine-tuning a model that is Mistral-7B-v0.2 from the MTEB benchmark using QLoRA resulted in the lowest performance among all tasks. This could be due to the complexity of the benchmark dataset or the need for further optimization of QLoRA.

**Conclusion:**

Finally, the results of fine-tuning trials with different approaches on different NLP tasks are presented in this study. Although LoRA demonstrated encouraging performance and efficiency results, additional research and improvement are needed to get competitive outcomes for QLoRA and alternative parameter-efficient fine-tuning techniques. The selection procedures for hyperparameters were critical in the fine-tuning process, emphasizing the significance of meticulous experimentation and tuning. In general, this study offers insightful information on the difficulties and possibilities involved in optimizing big language models for NLP applications.

**WandB Project Link:** <https://wandb.ai/prinku3005/emotions_kaggle_S2024>