

## Assignment 4

### Fine-tuning a Pretrained Model Using LoRA

#### Objectives:

Leverage LoRA: Low-Rank Adaptation to fine-tune a pretrained language model for a programming-related Question-Answering (QA) system on the **"flytech/python-codes-25k"** dataset.

#### Understanding LoRA (20%):

1. Review the concept, benefits, and mechanism of Low-Rank Adaptation (LoRA) for adapting pretrained models.
2. Discuss the suitability of pretrained language models for code-related QA tasks and the advantages of using LoRA for fine-tuning.

#### Dataset Preparation (20%):

1. Provide an overview of the "flytech/python-codes-25k" dataset, focusing on its structure and relevance for a QA system.
2. Describe necessary preprocessing steps, including tokenization and encoding strategies for code snippets.

#### Model Fine-Tuning with LoRA (30%):

1. Select a suitable pretrained language model and justify the choice based on its architecture and expected performance on code-related QA tasks.
2. Detail the integration of LoRA, specifying the adaptation process and adjustments made to the model for the QA task.

#### Training and Evaluation (30%):

1. Outline the training process, including configurations related to LoRA, learning rate settings, and QA-specific adaptations.
2. Evaluate the fine-tuned model using appropriate metrics, comparing its performance with a baseline model.
3. Analyze the results, focusing on improvements or limitations introduced by LoRA in the context of programming-related QA.

## Submission Requirements:

- Submit a comprehensive report covering all sections of the assignment, supplemented with figures, tables, and code snippets where necessary.
- Include the source code for preprocessing, fine-tuning with LoRA, and evaluation, well-commented for clarity.

## Evaluation Criteria:

- Clarity and depth of understanding and literature review on LoRA (20%).
- Thoroughness in dataset preparation and insightful analysis (20%).
- Creativity, correctness, and efficiency in model implementation and LoRA adaptation (30%).
- Comprehensive evaluation, insightful analysis of results, and discussion on the efficacy of LoRA (30%).

## References

- LoRA paper: <https://arxiv.org/abs/2106.09685>
- Dataset: <https://huggingface.co/datasets/flytech/python-codes-25k>
- LoRA code repository: <https://github.com/microsoft/LoRA>
- Another interesting repository with an implementation of LoRA: <https://github.com/huggingface/peft>