Advanced Deep Learning
AIGC 5500
Midterm Project
Deep Learning Optimizers

Please read the following guidelines very carefully before answering any questions:

- Please make sure to read all the questions and guidelines very carefully before asking any questions.
- You must keep the naming conventions requested in this document and each question.
- All deliverables are defined at the end of this document. You must upload them as requested.

*It will be up to a **10%** mark deduction if you do not follow the abovementioned guidelines.*

1. Introduction

- **Objective**: Research and investigate to compare the performance of **Adam** (Adaptive Moment Estimation), **RMSprop** (Root Mean Square Prop), and **AdamW** (Adam with Weight Decay) optimizers on a feedforward fully connected neural network using the KMNIST dataset.
- **Importance**: Understanding the strengths and weaknesses of different optimization algorithms helps in selecting the right one for specific tasks in deep learning.

2. Dataset Description

- **KMNIST Dataset**: A dataset of handwritten Japanese characters, like MNIST but more complex.
    - **Training Set**: 60,000 images
    - **Test Set**: 10,000 images
    - **Image Size**: 28x28 pixels, grayscale

3. Deep Learning Model

- **Architecture**: Design a feedforward fully connected neural network.
    - **Input Layer**: 784 neurons (28x28 pixels)
    - **Hidden Layers**: Two hidden layers with 128 and 64 neurons respectively
    - **Output Layer**: 10 neurons (one for each class)
    - **Activation Function**: Use ReLU for hidden layers and SoftMax for the output layer.

4. Methodology

- **Hyperparameter Tuning**: Use grid search or random search to find the best hyperparameters for each optimizer.
- **Cross-Validation**: Implement k-fold cross-validation to ensure robust evaluation.
- **Training and Evaluation**:

- Train the model using each optimizer.
- Evaluate performance on training, validation, and test datasets.
- Record metrics such as accuracy, loss, and training time.

6. Results

- **Tabular and Graphical Representation**:
  - Create tables showing accuracy, loss, and training time for each optimizer.
  - Generate graphs comparing the performance metrics across different optimizers.

7. Interpretation and Discussion

- **Analysis**: Discuss the performance of each optimizer, highlighting strengths and weaknesses.
- **Conclusion**: Summarize findings and suggest the best optimizer for this specific task.

8. References

- Cite all resources and papers used in the project.

Additional Instructions

- **Code Documentation**: Ensure your code is well-documented with comments explaining each part.

<u>Deliverables</u>:

- **Zip File** (**no RAR, no 7z, etc.**):

- **PDF Report**:
  - Introduction
  - Dataset, Model, and Optimizers Description
  - Solutions, Findings, and Results
  - Interpretation, Discussion, and Conclusion
  - References
- **Power Point Presentation**
  - Submit your ppt slides including code and report

- **Python Files:**
  - .ipynb files and pdf with code and results
- **Group Member Tasks**:
  - Specify each member's contributions.