Assignment #2

This assignment can be completed individually or by a team (maximum of 3 members).
Total score: 70
Due date: refer to the Canvas page

Objectives

To compare the performance of Multilayer Perceptron (MLP) models for classification using Scikit-learn and PyTorch

Only Python programs written using Python 3.0 or higher will be accepted. **NO Jupyter notebook or any other Python variant will be accepted** for efficient grading.

Required activities

Write a concise report comparing the performance of MLP models for classification using Scikit-learn and PyTorch. Conduct the experiment on any computer, such as your local machine or the Nautilus. Using GPU acceleration is optional, not required for this comparison.

The report should include:

- Team member names, optionally specifying each member's percent contribution or stating "Everyone contributed equally" if applicable. If your team does not reach an agreement on individual contribution, briefly write a task description for each member. Different grades may be assigned based on individual contributions.
- The results of modeling and computational performance based on the tasks outlined below

You may reuse publicly available source codes from the Internet or get help from Large Language Model; however, sharing your code with other teams or students in the class is strictly prohibited. Any student or team violating this policy will receive a zero for this assignment and may face penalties on all the remaining assignments.

1. Experimental Setup

Provide a brief description of the machine specifications used for this experiment, including:

- CPU type or name, clock speed, number of cores
- RAM size
- GPU (if you have or used it) type, number of CUDA cores, memory size

2. Datasets and Preprocessing

(a) Download the MNST digit dataset using appropriate Python packages, such as sklearn.datasets or torchvision.datasets. For more information about the data set, refer to the dataset repository (https://archive.ics.uci.edu/dataset/683/mnist+database+of+handwritten+digits/)

(b) Conduct preprocessing (if needed)

Briefly describe any data preprocessing methods applied before modeling, with justification in no more than 1 – 2 paragraphs.

3. Model Training

(a) Train classification models using both the Scikit-learn MLP and PyTorch MLP packages until you achieve a classification accuracy of over 95%, using the same hyperparameters across both models. Clearly specify the following hyperparameters used for each training:

- Network architecture: number of layers and the number of neurons in each layer
- Initial weights and learning rate
- Gradient methods (refer to lecture slide #54-55)
- Activation functions (refer to slide #56)
- Training methods (refer to slide #57, pretraining not allowed in this assignment)
- Number of epochs
- Any strategy used to prevent overfitting such as regularization, dropout, or early stopping

Ensure consistency in computer and hyperparameters to enable a fair comparison between Scikit-learn and PyTorch models.

4. Results and Analysis

Compare the performance of the models based on the following metrics:

- Training time: measure and compare the time taken to train the model
- Accuracy: measure % accuracy and precision

Discuss any notable differences between the two frameworks regarding training and performance metrics.

5. Conclusion

Provide a brief explanation summarizing the results of the performance comparison between Scikit-learn and PyTorch. Highlight key findings, such as differences in training efficiency or model performance, and offer insights into potential reasons for these differences. In addition, state any practical implications for selecting one framework over the other.

What and how to submit the assignment

- One report in Word or PDF format per team
- Only the program file(s) created by the team (excluding any third-party packages).
- Upload each file separately. DO NOT submit a ZIP file as Canvas cannot open ZIP files.

Grading criteria

- 90% based on the overall quality of work, including the analysis process, modeling results, code implementation, the depth of understanding demonstrated in the report
- 10% based on the effort reflected in the report and program development