Alexandria University
Faculty of Engineering

Introduction to Machine Learning
Assigned: Tuesday, November 7, 2023
Due: Friday, November 17,2023

Assignment 2

The objectives of this assignment are as follows:

- Custom Neural Network: Create effective PyTorch model for digit recognition.
- Efficient Data Handling: Utilize Torch data loader for seamless dataset management.
- Specialized Training Loop: Optimize training with customized gradient descent and loss function.
- Architectural Enhancements: Explore dropout layers for improved model performance.
- Performance Evaluation: Visualize and compare metrics for model accuracy assessment

Problem Statement

Develop a robust handwritten digit recognition system using deep learning techniques. The task involves creating a neural network model trained on the MNIST dataset, which contains images of handwritten digits from 0 to 9. The goal is to accurately classify these digits.

Assignment Details

1. **Data Preparation:**
    - Download the MNIST dataset from Kaggle and preprocess the data as necessary.
    - split the data into an 80% training set and a 20% validation set in stratified fashion using scikit learn train_test_split function.
    - Utilize the Torch data loader to efficiently load and manage the dataset.
2. **Training Process:**
    - Write You own Neural Network architecture.
    - Develop your own training loop.
    - Train the model using:
        - Optimizer: Gradient stochastic descent
        - Learning rate: 0.01
        - Loss Function: Cross Entropy
3. **Plot the following:**
    - Training and validation loss.
    - Training and validation accuracy.

Dr. Marwan Torki

Eng. Abdallah Ashraf   Eng. Mohamed Badran
Eng. Youssef Nawar   Eng. Zeyad Ezzat

4. Analysis:

- Evaluate the impact on the model's training and final performance based on the following factors:
  - Changing the learning rate, try 5 different learning rates at least.
  - Changing the batch size, try 5 different batch sizes at least.

5. Bonus

- Introduce dropout layer and layer normalization to your model and analysis the effect of adding them on the training and final performance.

6. Required:

- Construct a neural network architecture using PyTorch.
- Include at least two hidden layers in your network.
- Utilize ReLU activation functions between hidden layers.
- You have the flexibility to adjust the number of hidden layers and the number of weights in each layer according to your preferences.
- In the analysis part you should change one factor at time, training the network from scratch to generate the plots in point 3 and report its final accuracy. (Same if you are going to use the bonus)
- You should add your insights.

7. Deliverables:

- You should have your analysis in markdown cells in the notebook.
- If you don't know how to add your analysis in an organized manner add a report.
- You should deliver your notebooks as pdf, also the original notebook.
- You should work in groups of three. Each team should have one submission Id1_id2_id3.zip.
- Delivery will be ignored if you didn't follow the naming scheme provided in 4, any one of the team ids can be used.
- **Copies will get zeros.**
- **Assignment Deadline will be 17th November, please start early.**

8. Grading Scheme:

- Data Preparation 20%
- Training Process 40%
    - Training Loop 20%
    - Neural Network Architecture 20%
- Analysis 40%
    - Learning rate 20%
    - Batch Size 20%
- Bonus 15%