Class: CST461
Term: Fall 2024
Instructor: Yunus Kucuk

PROGRAMMING ASSIGNMENT 1

CNN

Due: 10/14/2024 MONDAY 11:59 PM

Objective:

The goal of this assignment is to design and implement a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset. You are required to experiment with different architectures, hyperparameters, and optimizations to achieve the best accuracy score on the test set. Your final submission will be evaluated based on the performance of your model.

Dataset:

- Dataset: CIFAR-10
  - Contains 60,000 32x32 color images in 10 different classes (e.g., airplane, automobile, bird, cat, etc.).
  - 50,000 images for training and 10,000 images for testing.
  - Each image is assigned one of 10 labels.

Instructions:

1. **Set Up the Environment:**
   - Load the CIFAR-10 dataset using libraries such as TensorFlow or PyTorch.
   - Split the dataset into training, validation, and test sets (use a 0.1 validation split from the training data).
2. **Design a CNN:**
   - Design and implement a CNN architecture. You can experiment with the number of convolutional layers, kernel sizes, strides, padding, and the number of filters.
   - You are encouraged to explore different types of layers, including:
     - Convolutional layers.
     - Pooling layers.
     - Fully connected (dense) layers.
     - Dropout layers.
     - Batch normalization.
3. **Compile and Train the Model:**
   - Use an appropriate loss function.
   - Choose an optimizer and experiment with learning rates, decay, and momentum.
   - Implement early stopping and/or model checkpointing based on validation performance.
   - Train the model for multiple epochs.

4. **Hyperparameter Tuning:**
    - Experiment with different hyperparameters to improve the performance of your CNN, such as:
        - Number of layers and units in each layer.
        - Learning rate.
        - Batch size.
        - Regularization techniques (e.g., L2 regularization, dropout).
    - Document the different configurations you tried and the impact on the validation accuracy.
5. **Evaluation:**
    - Evaluate the final performance of your model on the test dataset.
    - Report the test accuracy, precision, recall, and F1-score.
6. **Submission Guidelines:**
    - Submit your code in a Jupyter notebook format (.ipynb).
    - Submit the Jupyter notebook **with all cell outputs visible**. Ensure that the notebook is fully run from start to finish without any errors.
    - The notebook should be named as p1_lastname.ipynb (replace lastname with your own last name).
    - Include a report (max 2 pages) summarizing:
        - The architecture and hyperparameters used.
        - The performance metrics (accuracy, precision, recall, F1-score).
        - The best model configuration and why you believe it performed the best.
    - You may use visualizations like loss/accuracy curves and confusion matrices to support your findings.
    - The report should be in **PDF format** and named p1_lastname.pdf (replace lastname with your own last name).
    - Place both the notebook and the report in a directory named p1_lastname (replace lastname with your own last name).
    - Compress the directory into a **.zip file**.
    - Ensure the file structure looks like this before zipping:
        - p1_lastname/
        - ├── p1_lastname.ipynb
        - └── p1_lastname.pdf
    - Upload the zipped file (p1_lastname.zip) to the course submission platform before the deadline.
7. **Scoring:**
    - 50% - Best test accuracy achieved on the CIFAR-10 test set.
    - 25% - Code quality and adherence to instructions (clear, modular code with proper comments).
    - 15% - Report quality (clarity, analysis of experiments, use of visualizations).
    - 10% - Originality and effort (exploring different architectures, demonstrating critical thinking).
8. **Bonus:**
    - The student who gets the best metrics gets extra 10 bonus points.