Project: CNN for CIFAR-10 Classification

Due date: 05/27/2025, 11:59PM

Objective:

- Implement, train, and evaluate a Convolutional Neural Network (CNN) on the CIFAR-10 dataset for image classification.

- Gain a comprehensive understanding of CNN architecture design, training procedures, and performance evaluation.

Assignment Requirements:

1. Data Preprocessing and Augmentation (10 points)

Load the CIFAR-10 dataset and apply normalization.

Apply data augmentation techniques conceptually, such as:

- Random Horizontal Flip: Mirroring images horizontally to increase variety.
- Random Crop: Cropping random sections of images to help the model generalize better.
- Random Rotation: Slightly rotating images to help the model recognize objects in various orientations.
- Color Jitter: Adjusting brightness, contrast, saturation, and hue to simulate variations in lighting and color conditions.

Demonstrate the effect of augmentation by comparing model accuracy trained with and without augmentation in your report.

2. CNN Model Design and Implementation (40 points)

Design and implement your unique CNN architecture with at least four convolutional layers.

Appropriately use Pooling layers, Batch Normalization, Dropout, and other relevant layers.

Implement using PyTorch with clearly structured, modular, and well-commented code.

3. Model Training and Optimization (20 points)

Experiment with at least two different optimizers (e.g., SGD, Adam) and compare performance.

Utilize learning rate schedulers provided by PyTorch, such as StepLR, ReduceLROnPlateau, or CosineAnnealingLR, and clearly explain their impact on training (e.g., adjusting learning rates to improve convergence and reduce overfitting).

4. Quantitative Evaluation and Result Analysis (20 points)

Evaluate your model using accuracy on the CIFAR-10 test set.

Analyze results using a confusion matrix, which visually represents the performance of your classification model. Each row represents actual class instances, and each column represents predicted class instances. Discuss why the model performs differently across various classes, considering CNN architecture decisions and data characteristics.

5. Report and Code Submission (10 points)

Report Submission:

Submit your detailed report as a PDF document. Your report must clearly include:

- Description of data preprocessing and augmentation methods.
- Explanation of your CNN architecture design choices and reasons for selecting specific layers and parameters.
- Detailed comparison and analysis of optimizer and learning rate scheduler performances.
- Quantitative analysis of your results, including accuracy and confusion matrix, along with insightful interpretation of your findings.
- Discussion of challenges faced, insights gained, and possible improvements.

Code Submission:

Submit your complete, executable PyTorch code as a single .ipynb (Jupyter Notebook) file through Google Colab.

Ensure that your notebook is clearly structured, modularized, and includes sufficient comments explaining each step.

Provide clear instructions at the beginning of your notebook on how to run your code.

Grading Criteria:

- Appropriateness and effectiveness of data preprocessing and augmentation.
- Originality and correctness of CNN design and implementation.
- Systematic experimentation during training.
- Depth of quantitative evaluation and analysis.
- Quality of code (readability, comments, modularization).

Important Notes:

- Direct use or replication of known CNN architectures (e.g., ResNet, VGG) is strictly prohibited.
- Significant similarity or copied code from the internet, other students, or LLM will result in substantial grade reductions.
- Emphasis is on the depth of analysis and understanding rather than just achieving high accuracy.