

This assignment is due on Monday, November 14, 2025, at 11:59 pm.

Assignment 1

Goals

In this assignment, you will practice improving and optimizing neural networks for CIFAR100 image classification. The goals of this assignment are as follows:

- Build a dual-branch image classification model using gradient information to assist classification;
- Improve model prediction accuracy through data augmentation, optimization strategy adjustments (optimizer, epochs, loss function, etc.), and model structure optimization (dual-branch feature fusion levels and methods, introduction of attention mechanisms, etc.).
- Perform T-SNE visualization analysis of the classification model.

Q1: Build a dual-branch image classification model [50 points]

In the `image-classifier.ipynb`, the basic code for CIFAR-100 image classification has been implemented and provided. Please modify the necessary code to build a dual-branch image classification model. The specific requirements are as follows:

- Modify "Step 2: Custom Data Loading", complete the `def RGB2Gradient(self, img: torch.tensor)` function. This function is designed to convert an RGB image into its gradient magnitude (i.e., edge information) using the Sobel operator. You must complete the `[YOU NEED TO FILL]` parts within this function to correctly apply the Sobel filters and calculate the final gradient magnitude.
- Modify "Step 2: Custom Data Loading", modify the `__getitem__` method. It must use the `RGB2Gradient` function you just completed to process the image, and then return both the original image and its corresponding gradient information as the model input.
- Modify "Step 3: Configure the Neural Network": Construct a dual-branch model that separately processes the original image and its gradient information. The two branches do not share parameters.
- Train and test the model to obtain classification results using the dual-branch model.

Q2: Improve Model Prediction Accuracy [30 points]

Improve model prediction accuracy through methods such as data augmentation, optimization strategy adjustments (optimizer, epochs, loss function, etc.), and model structure optimization (e.g., dual-branch feature fusion levels and methods, introduction of attention mechanisms, etc.).

Note: The grade will be based on the "Step 6: Evaluate model accuracy" output result, Average Acc. Do not modify the code for accuracy calculation in Step 6, and ensure that all code is fully executed.

Q3: T-SNE Visualization [20 points]

Use hooks in PyTorch to extract feature representations from the intermediate layers of the model for the test set "testloader", and visualize them using the T-SNE method. The specific requirements are as follows:

- Visualize the features before and after the dual-branch feature fusion. If there are multiple fusions, you may choose specific layers for visualization.

Note: The visualization results should be included as the output of "Step 7: T-SNE Visualization".

Submitting your work

Important. Please make sure that the submitted notebooks have been run, the cell outputs are visible and the notebook names meet the requirements.

Once you have completed all notebooks and filled out the necessary code, you need to follow the below instructions to submit your work:

1. Print `image-classifier.ipynb` and save it as `student_name_ID_result.pdf`.
 - In Colaboratory: **File** → **Print**.
2. Submit the PDF and the code file to eLearning.
Download `image-classifier.ipynb` and result PDF rename them to `student_name+ID+image-classifier.ipynb` and `student_name+ID+result.pdf` before submitting to eLearning.