Homework 5: Image Classification

November 19, 2025

Due Date: December 10 by 23:59:59

Introduction

In this assignment, you will implement and test various image classification models on the CIFAR-10 dataset. The goals of this assignment are as follows:

- Implement and compare the linear classifier and the full-connected neural network.
- Train and test two types of classifiers.
- Compare the AdamW and the SGD optimizer based on FCNN.
- Compare the StepLR and the CosineAnnealingLR scheduler based on FCNN.

You can learn how to create, train, and test a model using PyTorch here. You are highly encouraged to go through this tutorial before you start.

Here are some other supplementary materials that may help you:

- PyTorch Documentation
- PyTorch Chinese Documentation
- Dive into deep learning

Notes for hyper-parameter tuning: you can get full score when accuracy is above 60%, save time for your busy end-of-term season.

1 Define Classifiers (30 pts.)

Here are some useful function:

- torch.nn.Linear()
- torch.nn.ReLU()
- torch.nn.Tanh()

You are free to use any torch functions. Note: if you want to use Convolution-based classifiers, feel free to have a try. But we don't set bonus in this homework.

1.1 Linear classifier (15 pts.)

Add your own code to the **LinearClassifier** class to define a linear classifier. Your classifier is required to process a mini-batch data.

1.2 Full-connected neural network classifier (15 pts.)

Add your own code to the **FCNN** class to define a full-connected neural network classifier. You are responsible for choosing the network depth, width, and activation type.

2 Implement the training and testing function (40 pts.)

There is a whole training code in PyTorch Tutorial: train a classifier, you can learn from it. In this task, you need to implement the **train()** and **test()** function that can choose a model, optimizer, scheduler, and so on; see the end of the **main.py** for details.

3 Report(30 pts.)

You can use **TensorBoard** in PyTorch to record and visualize the loss and accuracy curves. Here is a tutorial introducing TensorBoard. **Based on FCNN**, analysis section 3.1 and section 3.2.

3.1 Compare AdamW and SGD optimizer (10 pts.)

Train the classifiers you implemented using the AdamW (**torch.optim.AdamW**) and SGD (**torch.optim.SGD**) optimizer and compare the loss and accuracy curves. Put the results in your report.

3.2 Compare StepLR and CosineAnnealingLR scheduler (10 pts.)

Train the classifiers you implemented using two learning rate schedulers, including the StepLR (**torch.optim.lr_scheduler.StepLR**) and CosineAnnealingLR (**torch.optim.lr_scheduler.CosineAnnealingLR**) scheduler and compare the loss and accuracy curves. Put the results in your report.

3.3 Visualization (10 pts.)

You have now completed the entire process of this project. Put all the visualizations and results in your report: the loss and accuracy curves and the final classification accuracy scores. For the result of Linear Classifier, you can report with arbitrary optimizer or learning rate scheduler. For FCNN, you should report section 3.1 and section 3.2.

4 Submit

Be sure to zip your code and final report; Name it as **StudentID_YourName_HW5.zip**. **Any wrong name will cost you 0.5 pts in final score.**