IERG4160 Final Project

Introduction

This project aims at solving Cifar-10 image classification task using neural networks. You need to implement several models, compare the performance among them, and adjust hyper-parameters to improve the performance. You have to finish the project on your own.

Set up the environment

1. Install anaconda: download and install anaconda here: https://www.anaconda.com/
2. Create an environment and activate it:
   ```
   conda create -n ierg4160 python=3.7 -y
   conda activate ierg4160
   ```
3. Install PyTorch: Launch your command line and execute:
   ```
   conda install pytorch torchvision -c pytorch
   ```

Tasks

1. Train the default linear classification model and report the test result via:
   ```
   python main.py
   ```
2. Implement an MLP model with 3 hidden layers whose channels are all 128. Each hidden layer is followed by ReLU activation. Use an additional fully-connected layer to map the dimension into 10 classes. Please report the test result.
3. Implement LeNet, one of the earliest CNN models. The structure is as follows. Please report the test result.

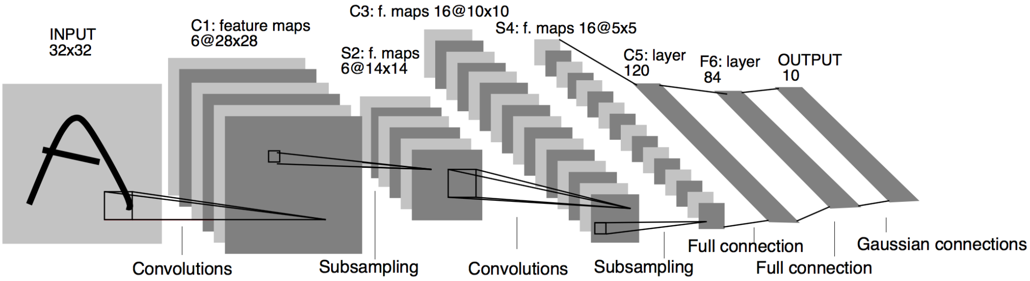

Fig. 1. LeNet has 2 conv layers and an MLP with 2 hidden layers. There is a max pooling layer following the 1st and 2nd conv layer. Don't forget ReLU.

4. Implement AlexNet, the famous CNN model that marked the beginning of the deep learning era. The structure is as follows. Please report the test result.

Layer	Params
Convolution	out_channels=64, kernel_size=3, stride=2, padding=1
ReLU	
Max Pooling	kernel_size=2
convolution	out_channels=192, kernel_size=3, padding=1
ReLU	
Max Pooling	kernel_size=2
Convolution	out_channels= 384, kernel_size=3, padding=1
ReLU	
Convolution	out_channels= 256, kernel_size=3, padding=1
ReLU	
Convolution	out_channels= 256, kernel_size=3, padding=1
ReLU	
Max Pooling	kernel_size=2

Dropout	Dropout ratio = 0.5
Fully connected	out_channels=4096 (please deduce the in_channels)
ReLU	
Dropout	Dropout ratio = 0.5
Fully connected	out_channels=4096
ReLU	
Fully connected	out_channels=10

Fig. 2. AlexNet.

5. Based on **LeNet**, study the influence of hyper-parameters including the batch size, learning rate, momentum, weight decay, type of data augmentations.
6. Adjust the hyper-parameters mentioned above to achieve as high performance as possible. Note that number of epochs, the CNN model is not allowed to change.
7. Write an experiment report based on your experiments. You may use tables and curves to show your observations. You are also encouraged to perform more studies.