CSE432/532 Machine Learning Basics

Final Exam Project

General Instructions:

1. This Final Exam Project includes two parts, the program that is used to implement the CNN and the report that discusses the detailed information about the CNN model.
2. For the report, please submit a ".pdf" file to the canvas. For the program, please finish it in the colab and submit the ".ipynb" file.
3. Please follow the instructions of the new CNN architecture to write the code. You can use any tools or libraries (i.e., Pytorch, TenserFlow) to implement this CNN model.
4. Please follow the examples in Chapter 6.1 to write the report. You need to include the following information: the number of parameters (with and without the bias) for each layer and the memory space required (size of the output) for each layer.
5. You need to finish this Final Exam Project individually.
6. Please submit the code and report by May 6, 23:59PM.
7. No starting code, you can leverage HW4.
8. Using the GPU will save you a lot of time. Check HW4 to find out how to use GPU in Colab.

Task:

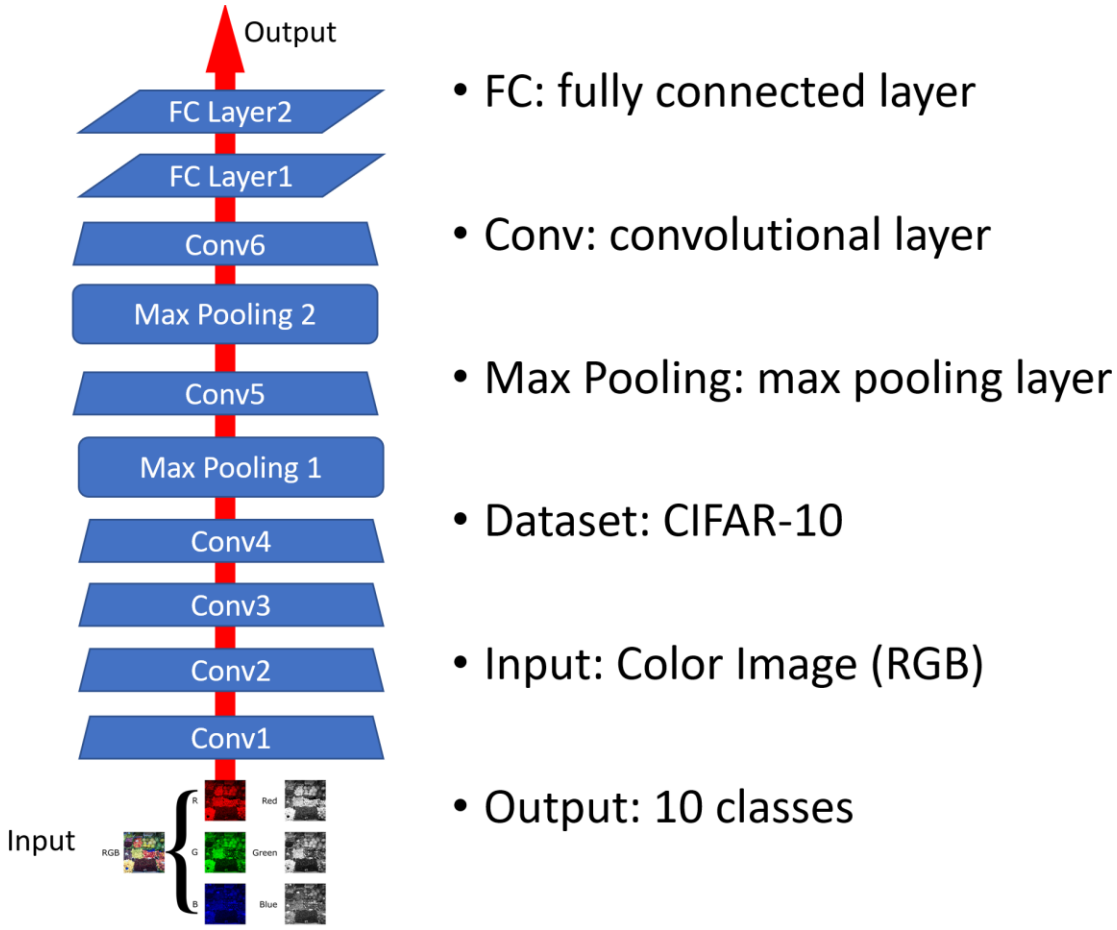
Similar to the homework 4, the task is image recognition. Given an image (colored, 32*32*3), your CNN model should recognize what class this image belongs to.

Dataset:

You will use the same dataset (i.e., the CIFAR-10) as what you used in HW4.

The CNN model

In this Final Exam Project, you are required to implement one specific CNN model (namely FinalExamNet), the structure of which is shown in the following Figure.



The detailed information about each layer is shown in the following table.

Layer	configurations	
Conv1	Number of Filters: 8 Kernel_size:11 Stride:1 Padding:5	
Conv2	Number of Filters: 16 Kernel_size:7 Stride:1 Padding:3	
Conv3	Number of Filters: 16 Kernel_size:5 Stride:1 Padding:2	
Conv4	Number of Filters: 16 Kernel_size:5 Stride:1 Padding:0	
Conv5	Number of Filters: 16 Kernel_size:5 Stride:1 Padding:0	
Conv6	Number of Filters: 160 Kernel_size:5 Stride:1 Padding:0	
Max pooling 1	Kernel_size:2 Stride:2 Padding:0	
Max pooling 2	Kernel_size:2 Stride:2 Padding:0	
FC layer 1	Input Features:160 Output Features:160	Neurons: 160
FC layer 2	Input Features:160 Output Features:10	Neurons:10

If you set the learning rate as 0.01, momentum=0.9, SGD, weight_decay=5e-4 (same as HW4)

Epochs=50

Batch_size=128

Activation function using Relu()

The Final test accuracy should be around 62% (Similar code can be found in HW4 “main” function)

In your report, please include the following information

Layer	Size of output data	Size of parameters
Conv1		
Conv2		
Conv3		
Conv4		
Conv5		
Conv6		
Max pooling 1		
Max pooling 2		
FC layer 1		
FC layer 2		

Assume each memory unit takes 4 bytes.