

**EE 569: Homework #5**

Issued: 03/29/2021

Due: 11:59PM, 04/14/2021

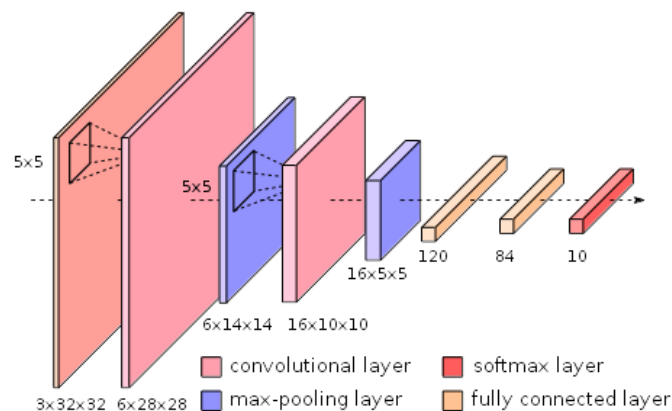
**General Instructions:**

1. Read *Homework Guidelines* for the information about homework programming, write-up and submission. If you make any assumptions about a problem, please clearly state them in your report.
2. You are required to use PYTHON in this assignment. It is recommended to use interface tool PYTORCH. KERAS is an alternative choice if you feel more comfortable with it, which is built upon TENSORFLOW. We only provide sample tutorial using PYTORCH.
3. DO NOT copy codes from online sources e.g. Github.
4. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.

**Problem 1: CNN Training on LeNet-5 (100%)**

In this problem, you will learn to train a simple convolutional neural network (CNN) called the LeNet-5, introduced by LeCun et al. [1], and apply it to three datasets **MNIST** [2], **Fashion-MNIST** [3] and **CIFAR-10** [4].

LeNet-5 is designed for handwritten and machine-printed character recognition. Its architecture is shown in Fig. 1. This network has two *conv* layers, and three *fc* layers. Each *conv* layer is followed by a *max pooling* layer. Both *conv* layers accept an input receptive field of spatial size  $5 \times 5$ . The filter numbers of the first and the second *conv* layers are 6 and 16 respectively. The stride parameter is 1 and no padding is used. The two *max pooling* layers take an input window size of  $2 \times 2$ , reduce the window size to  $1 \times 1$  by choosing the maximum value of the four responses. The first two *fc* layers have 120 and 84 filters, respectively. The last *fc* layer, the output layer, has size of 10 to match the number of object classes in the dataset. Use the popular ReLU activation function [5] for all *conv* and all *fc* layers except for the output layer, which uses softmax [6] to compute the probabilities.



**Figure 1:** A CNN architecture derived from LeNet-5

The following table shows statistics for different datasets:

	Image type	Image size	# Class	# training images	# testing images
MNIST	Gray	28*28	10	60,000	10,000
Fashion-MNIST	Gray	28*28	10	60,000	10,000
CIFAR-10	Color	32*32	10	50,000	10,000

**(a) CNN Architecture (Basic: 20%)**

Explain the architecture and operational mechanism of convolutional neural networks by performing the following tasks.

1. Describe CNN components in your own words: 1) the fully connected layer, 2) the convolutional layer, 3) the max pooling layer, 4) the activation function, and 5) the softmax function. What are the functions of these components?
2. What is the over-fitting issue in model learning? Explain any technique that has been used in CNN training to avoid the over-fitting.
3. Explain the difference among different activation functions including ReLU, LeakyReLU and ELU.
4. Read official documents of different loss functions including L1Loss, MSELoss and BCELoss. List applications where those losses are used, and state why do you think they are used in those specific cases?

Show your understanding as much as possible in your own words in your report.

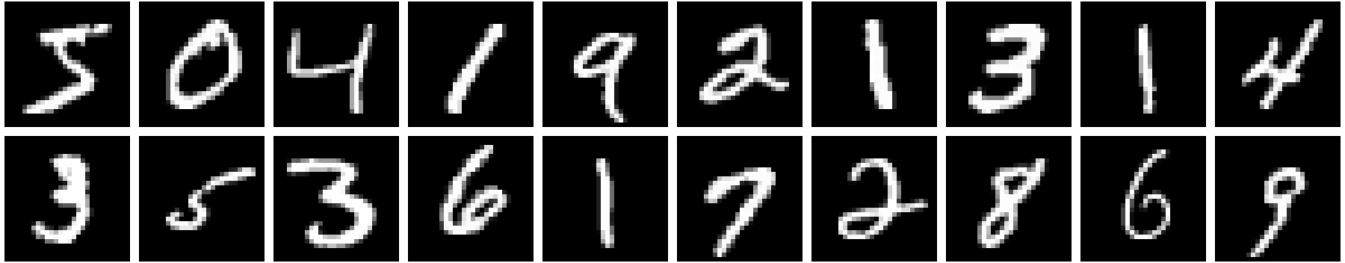
**(b) Compare classification performance on different datasets (Basic: 50%)**

Train the CNN given in Fig. 1 using the training images of MNIST, then test the trained network on the testing images of MNIST. Compute and draw the accuracy performance curves (epoch-accuracy plot) on training and test datasets on the same figure. You can adopt proper preprocessing techniques and the random network initialization to make your training work easy.

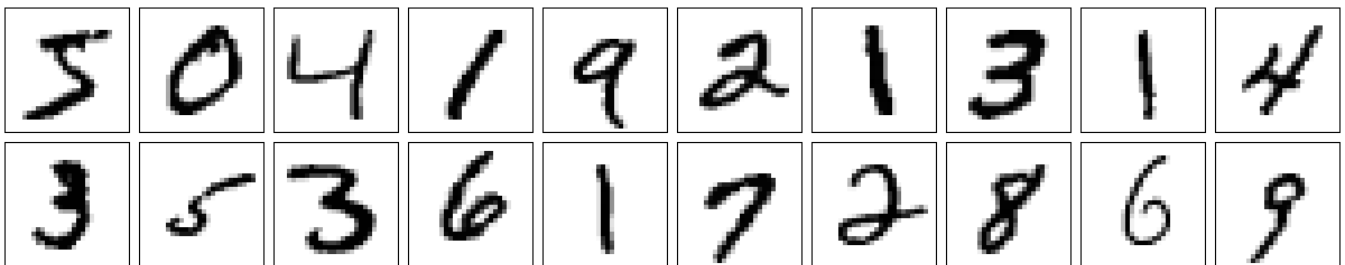
1. Plot the performance curves under 5 different yet representative initial parameter settings (initialization of filter weights, learning rate, decay and etc.). Discuss your observations and the effect of different settings.
2. Find the best parameter setting to achieve the highest accuracy on the test set. Then, plot the performance curves for the test set and the training set under this setting. Your testing accuracy should be no less than 99%.
3. Repeat 1 and 2 for Fashion-MNIST. Your best testing accuracy should be no less than 90%.
4. Repeat 1 and 2 for CIFAR-10. Your best testing accuracy should be no less than 65%.
5. Compare your best performances on three datasets. How do they differ and why do you think there is such difference?

**(c) Apply trained network to negative images (Advanced: 30%)**

You may achieve good recognition performance on the MNIST dataset in Problem 1(b). Do you think the LeNet-5 understands the handwritten digits as well as human beings? One test is to provide a negative of each test image as shown in Fig. 3, where the value of the negative image at pixel  $(x,y)$ , denoted by  $r(x,y)$ , is computed via  $r(x,y)=255-p(x,y)$ , where  $p(x,y)$  is the value of the original image at the same location. Humans have no difficulty in recognizing digits of both types. How about the LeNet-5?



**Figure 2:** Sample images from original MNIST dataset



**Figure 3:** Sample images from the negatives of the MNIST dataset

1. Describe how you can get negatives of the testing set. Implement your idea, then use statistics and sample images to show that you correctly reverse the intensity.
2. Report the accuracy on the negative test images using the LeNet-5 trained in part b). Discuss your result.
3. Design and train a new network that can recognize both original and negative images from the MNIST test dataset. Test your proposed network, report the accuracy and make discussion.

## References

- [1] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324
- [2] <http://yann.lecun.com/exdb/mnist/>
- [3] <https://github.com/zalandoresearch/fashion-mnist>
- [4] <https://www.cs.toronto.edu/~kriz/cifar.html>
- [5] ReLU [https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)).
- [6] Softmax [https://en.wikipedia.org/wiki/Softmax\\_function](https://en.wikipedia.org/wiki/Softmax_function)