

EE 569: Homework #5

Issued: 03/28/2022

Due: 11:59PM, 04/14/2022

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 5×5 . 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 2×2 , reduce the window size to 1×1 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.

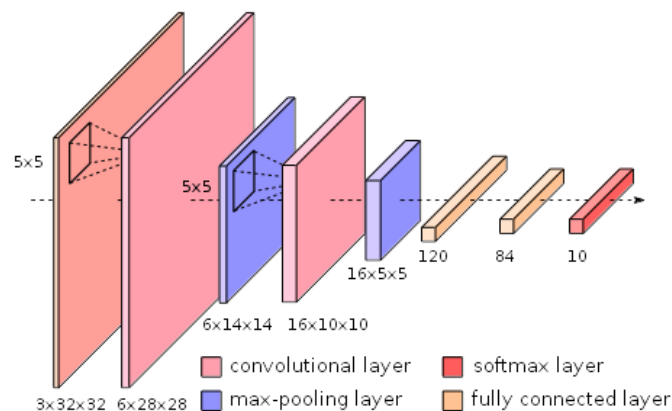


Figure 1: A CNN architecture derived from LeNet-5

This content is protected and may not be shared, uploaded, or distributed.

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 (30%)

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 3 different yet representative hyper-parameter settings (optimizers, 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 2 for Fashion-MNIST. Your best testing accuracy should be no less than 90%.
4. Repeat 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?

This content is protected and may not be shared, uploaded, or distributed.

Note: for each setting, you need 5 runs. Report the [best test accuracy among 5 runs, mean test accuracy of 5 runs, standard deviation of test accuracy among 5 runs] to evaluate the performance.

(c) Analysis on confusion classes and hard samples (30%)

You may achieve good recognition performance on the MNIST dataset in Problem 1(b). Now let's dive deeper into the classification results.

1. Generate the normalized confusion matrix for the 10 classes on the testing set. What are the top three confused pairs of classes? Show one example for each of these three pairs. Describe your observations and explain.
2. Repeat 1 for Fashion-MNIST.
3. Repeat 1 for CIFAR-10.

Note: you may use the best setting you found in Problem 1(b) on each dataset.

(d) Classification with noisy data (20%)

Data in real world application could be noisy with wrong labels. Symmetric Label Noise (SLN) is the type of labeling noise where $\alpha\%$ of the data with true label of class i is labeled as other classes $j \neq i$ with uniform probability. For example, in a 3-class classification problem, the normalized confusion matrix between the true label and the noisy label is close to the following format, where ϵ is the noise level (say, 40%):

$$\begin{bmatrix} 1 - \epsilon & \frac{\epsilon}{2} & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & 1 - \epsilon & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & \frac{\epsilon}{2} & 1 - \epsilon \end{bmatrix}$$

Now you'd like to synthesize the Symmetric Label Noise on the training set of MNIST and investigate the performance of neural networks under different noise levels.

1. Implement the Symmetric Label Noise. Describe your method and show the normalized confusion matrix for $\epsilon = 40\%$.
2. Train LeNet-5 with the noisy training set and measure the testing accuracy. Try $\epsilon = 0\%, 20\%, 40\%, 60\%, 80\%$. Draw the curve of [testing accuracy vs. ϵ]. Note that for each ϵ , 5 runs are needed to calculate the mean and standard deviation of the testing accuracy which then are used to draw your plot.
3. Discuss your observations on result of 2 and analyze.

References

- [1] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324

This content is protected and may not be shared, uploaded, or distributed.

- [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

This content is protected and may not be shared, uploaded, or distributed.