EE 569: Homework #4: Handwritten Digits Recognition with LeNet-5

Issued: 11/4/2016 Due: 11:59PM, 12/4/2016

General Instructions:

1. Read Homework Guidelines and MATLAB Function 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. Do not copy sentences directly from any listed reference or online source. Written reports and source codes are subject to verification for any plagiarism. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.
3. In this homework, you are asked to write a deep learning program to train and test convolutional neural networks. You will develop your program in a framework called *Torch*, which uses *Lua* (a script language similar to python) as the interface. See *HW4_Torch_Instruction* to know how to get started with this framework.

Common Background of Homework #4

In Homework #4, you will learn to train a simple convolutional neural network (CNN) called the LeNet-5 and apply it to the MNIST dataset. The MNIST dataset [1] is formed by images of handwritten digits (0, 1, ..., 9). All digits are size-normalized and centered in an image of size 32 by 32. The dataset has a training set of 60,000 samples and a test set of 10,000 samples. The LeNet-5 is the latest CNN designed by LeCun et al. [2] for handwritten and machine-printed character recognition. Its architecture is shown in Fig. 2. The LeNet-5 has two pairs of convolutional/pooling layers, denoted by C1/S2 and C3/S4 in the figure, respectively. Layer C1 has 6 filters of size 5 by 5. Layer C3 has 16 filters of size 5 by 5. Each of them is followed by a nonlinear activation function. Furthermore, there are two fully connected layers, denoted by C5 and F6, after the two pairs of cascaded convolutional/pooling operations and before the output layer.

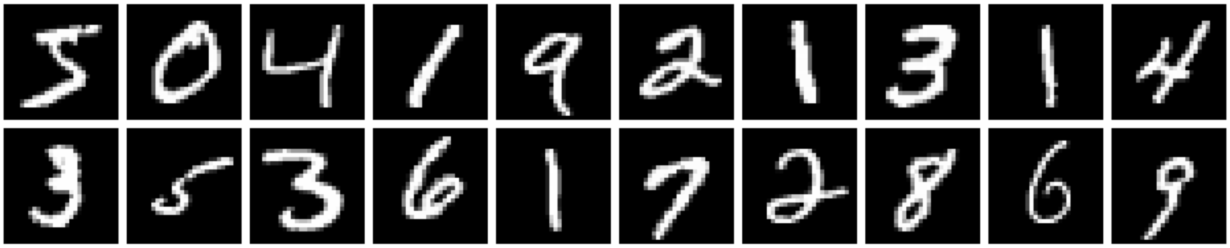

Figure 1: Sample images from the MNIST dataset.

The network to be trained is slightly different from the original architecture proposed in [2]. First, it uses the ReLU function [3] (instead of the hyperbolic tangent function) for nonlinear activation. Second, it uses the softmax function [4] to produce the probability distribution of the output (instead of the ad hoc solution in [2]). These two modifications are commonly used in modern CNNs. We still call the resulting network the LeNet-5 since the modifications are minor.

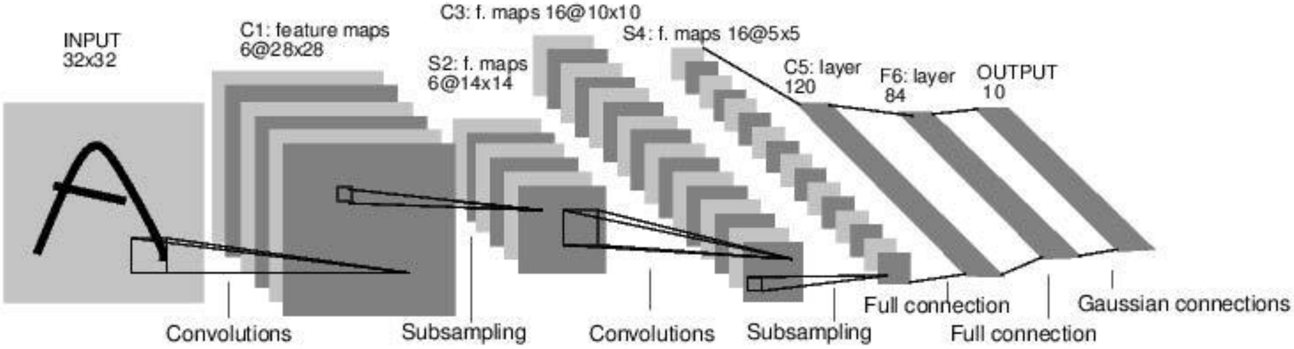

Figure 2: LeNet-5

Problem 1: LeNet-5 Training and Its Application to the MNIST Dataset (55%)

(a) CNN Architecture and Training (15%)

Understand the LeNet-5 architecture fully. To show your understanding, answer the following questions clearly in your report.

1) Describe the architecture and building components of the LeNet-5 in your own words.
2) What is the main difference between the LeNet-5 and the classic three-layer artificial neural network (ANN)?
3) Explain why LeNet-5 works better for the handwritten digit classification problem as compared with ANN?

(b) Application of LeNet-5 to MNIST Dataset (40%)

Build the LeNet-5 based on the description in the last paragraph of Page 1. Train it with the MNIST training dataset.

1) Describe your procedure in the training of the LeNet-5 (e.g., initialization, cost function selection, training parameters selection, etc.).
2) Plot the epoch-accuracy (or iteration-accuracy) curves for the training dataset and the testing dataset in one figure. Discuss your obtained results.
3) Report the mean Accuracy Precision (mAP) and discuss how the mAP value is related to your initialization scheme and training parameter choices. You should do at least two different settings in (2) so as to compare the results of two correct recognition rates.

Note: The training and testing datasets are provided in *mnist-p1b-train.t7* and *mnist-p1b-test.t7*.

Problem 2: Capability and Limitation of Convolutional Neural Networks (45%)

(a) Application of LeNet-5 to Negative MNIST Images (15%)

You may achieve good recognition performance on the MNIST dataset in Problem 1. 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?

1) Report the mAP on the negative test images using the LeNet-5 trained in Problem 1. Discuss your result.
2) 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 mAP result and make discussion.

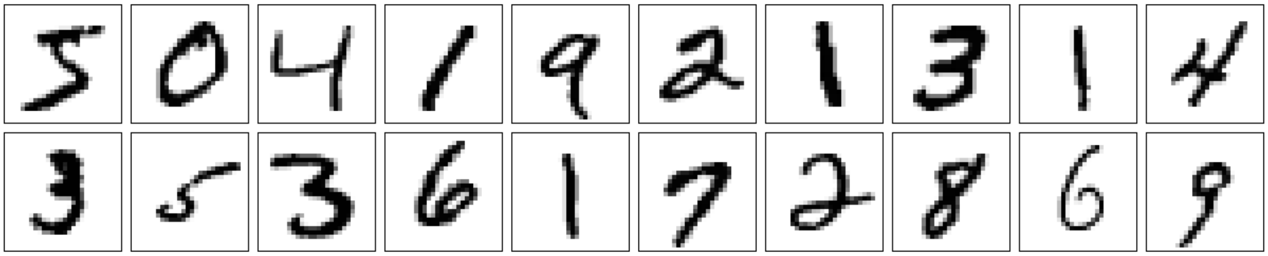

Figure 3: Sample images from the negatives of the MNIST dataset.

(b) Application of LeNet-5 to MNIST Images with Background (15%)

We add background scenes to the MNIST training and testing datasets. They are shown in *mnist-p2b-train.t7* and *mnist-p2b-test.t7*. Each image in this dataset has color background but may not contain a digit. Train the LeNet-5 to classify each input image into one of 11 classes (i.e., 10 digit classes plus one additional class that contains no digit). Report your mAP result and make discussion.

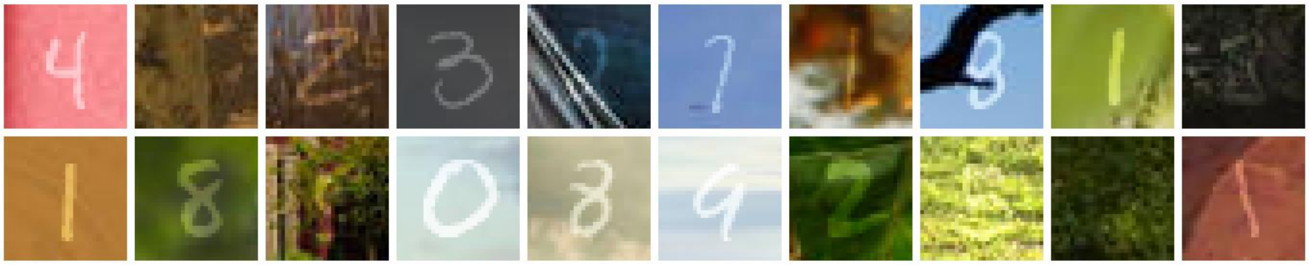

Figure 4: The MNIST images overlaid with background scenes.

(c) Is LeNet-5 Translationally Invariant? (15%)

A classifier is translationally invariant if it can recognize objects regardless of their positions in the image.

1) Is the LeNet-5 a translationally invariant classifier? Design an experiment to verify your conjuncture. Explain your experimental design in detail. Report your experimental results.
2) Please find a reason to explain the observed phenomenon in the first part.
3) If the LeNet-5 is not translationally invariant, find a solution to make it translationally invariant? Run some experiments to test your idea.

Hint: you may pad the input image with extra background to enlarge the input image size (e.g. 36x36 or larger). Then, you can translate the digit arbitrarily yet it is within the original image of size 32x32.

References

[1] [Online] http://yann.lecun.com/exdb/mnist/

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

[3][Online]https://en.wikipedia.org/wiki/Rectifier_(neural_networks).

[4][Online]https://en.wikipedia.org/wiki/Softmax_function

[5] C.-C. Jay Kuo, "Understanding CNN with a mathematical model" 2016, arXiv 1609.04112.