EE 569: Homework #5: Image Recognition with CNN

Issued: 03/17/2019 Due: 11:59PM, 04/07/2019

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. Do not copy sentences directly from any listed reference or online source. Written reports and source codes are subject to verification for plagiarism. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.
3. You can use Tensorflow or PyTorch.
4. You are allowed to use Keras.

CNN Training and Its Application to the MNIST Dataset (100 %)

You will learn to train one simple convolutional neural network (CNN) derived from the LeNet-5 introduced by LeCun et al. [1]. Furthermore, you need to apply it to the MNIST dataset [2]. The MNIST dataset of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. Figure 1 shows some exemplary images from the MNIST dataset.

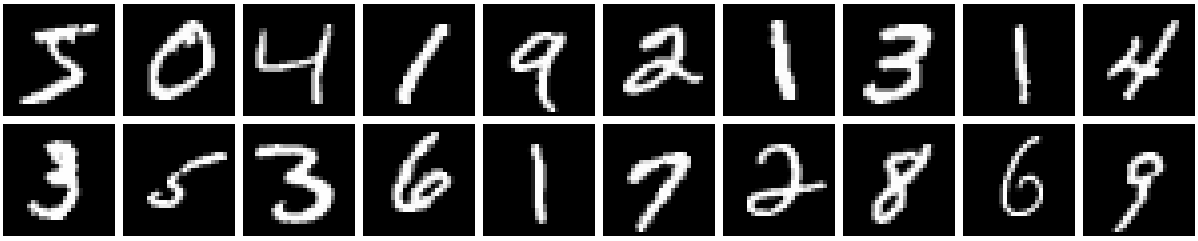

Figure 1: MNIST dataset

The CNN architecture for this assignment is given in Figure 2. 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 *5x5*. 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 *2x2*, reduce the window size to *1x1* by choosing the maximum value of the four responses. The first two *fc* layers have 120 and 80 filters, respectively. The last fc layer, the output layer, has size of 10 to match the number of object classes in the MNIST dataset. Use the popular ReLU activation function [3] for all *conv* and all *fc* layers except for the output layer, which uses softmax [4] to compute the probabilities.

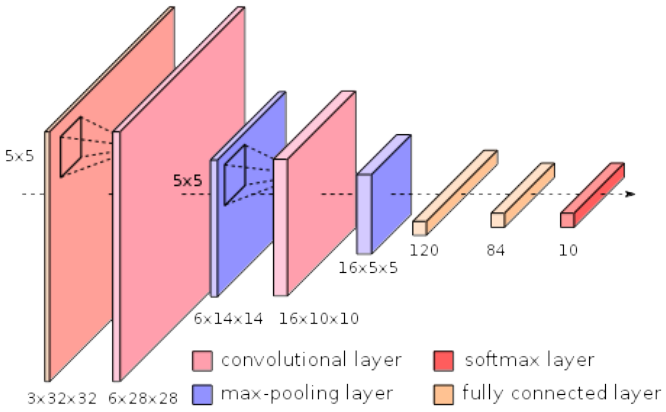

Figure 2: A CNN architecture derived from LeNet-5.

(a) **CNN Architecture and Training (40%)**

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

- 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?
- What is the over-fitting issue in model learning? Explain any technique that has been used in CNN training to avoid the over-fitting.
- Why CNNs work much better than other traditional methods in many computer vision problems? You can use the image classification problem as an example to elaborate your points.
- Explain the loss function and the classical backpropagation (BP) optimization procedure to train such a convolutional neural network.

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

(b) Train LeNet-5 on MNIST Dataset (30%)

Train the CNN given in Fig. 2 using the 60,000 training images from the MNIST dataset.

- Compute the accuracy performance curves using the epoch-accuracy (or iteration-accuracy) plot on training and test datasets separately. Plot the performance curves under 5 different yet representative parameter settings. Discuss your observations and the effect of different settings.
- 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.

(c) Apply trained network to negative images (30%)

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 accuracy on the negative test images using the LeNet-5 trained in part b). 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 accuracy and make discussion.

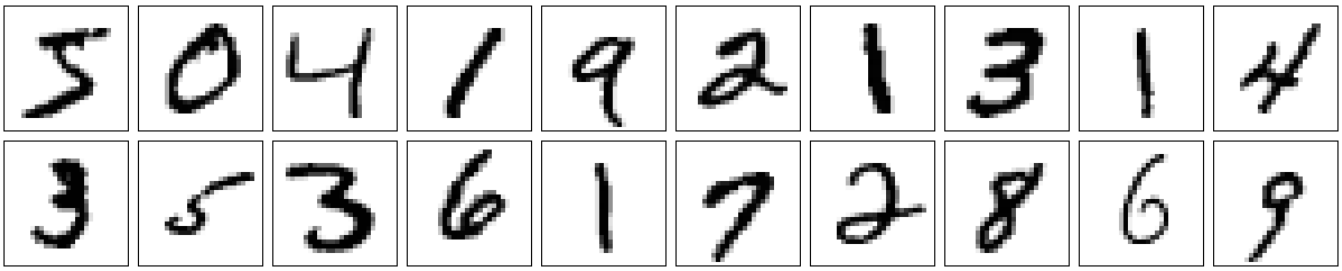

Figure 3: Sample images from the negatives of the MNIST dataset.

References

[1][LeNet-5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.
[2][MNIST] http://yann.lecun.com/exdb/mnist/
[3][ReLU] https://en.wikipedia.org/wiki/Rectifier_(neural_networks).
[4][Softmax] https://en.wikipedia.org/wiki/Softmax_function