EE 569: Homework #4: Image Recognition with CNN and Saak

Issued: 03/23/2018 Due: 11:59PM, 04/29/2018

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 interface tool Keras, which is built upon Tensorflow, for deep learning programming in Problem 1.
4. You can use the github codes for the Saak transform implementation in Problem 2.

Problem 1: CNN Training and Its Application to the MNIST Dataset (50 %)

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.

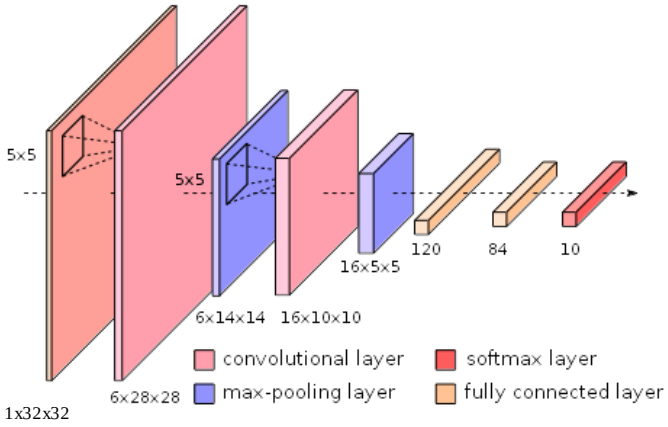

Figure 2: A CNN architecture derived from LeNet-5.

(a) **CNN Architecture and Training (20%)**

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 words in your report.

(b) Train LeNet-5 on MNIST Dataset (15%)

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

- Explain how you initialize the network parameters (filter weights, learning rate, decay and etc.) and discuss the effect of different settings.

- 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.
- 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) Improve the LeNet-5 for MINST dataset (15%)

This is an open part for you to understand the CNN architecture. Please feel free in modifying the baseline CNN in Part (b) to improve the classification accuracy. For example, you can increase the depth of the network by adding more layers, or/and change the number of filters in some layers. You can also try different activation functions or optimization algorithms. Going deeper, you can even refer to any modern techniques in the literature. Report the best accuracy that you can achieve and describe your network architecture and the training parameter setting to reach this result. Discuss the sources of performance improvement. Your grading in this part will be based on your obtained performance in comparison with other students in the same class.

Problem 2: Saak Transform and Its Application to the MNIST Dataset (50%)

In this problem, please read and understand the paper "On Data-Driven Saak Transform" [5] introduced by professor Kuo.

(a) Comparison between the Saak Transform and the CNN architecture (10%)

Use your own language to explain the similarities and differences between the Saak transform and the CNN solution. The length should be at least of 1 page. Do not copy any sentences from [5] or other papers directly, which is plagiarism. The scores will depend on your degree of understanding.

(b) Application of Saak transform to MNIST dataset (30%)

The overall process of this task can be divided into three parts: (1) Saak coefficients generation, (2) Saak coefficients selection and dimension reduction, (3) classification. In the first part, it is a purely unsupervised process based on the statistics from the data. The details are given below.

Step 1: Saak coefficients generation

There are 5 stages in the Saak transform pipeline. Here are the detailed steps to find the Saak coefficients in each stage:

- For each input image (or data cube), select the non-overlapping patch region with spatial size 2x2. Then calculate the variance of each patch and remove the small-variance patches.
- Perform Principle Component Analysis (PCA) to the zero-mean patch data and get the PCA transform matrix. Transform all the input data patches by selecting the important spectral components in the PCA matrix.
- Augment transform kernels through the sign-to-position format conversion.
- Repeat this process for each Saak transform stage. For your convenience, the number of important component in each stage (total 5 stages) are: 3, 4, 7, 6, 8.

Step 2: Saak coefficients selection and dimension reduction

After getting responses from all Saak stages, you will get a feature vector of 1500 dimension. Perform the F-test on it and select features with larger F-test scores (around 1000 dimension). Then, perform another round of PCA to reduce the feature dimension to 32, 64 and 128.

Step 3: Utilize images of the same class label to collect features of training samples and compare the performance of the SVM and the RF classifiers with three feature dimensions as given in Step 2.

(1) Report the best classification accuracy that you can achieve on the train and test sets for all 6 cases.

(2) Compare the performance to the results in Problem 1 and briefly explain your observations.

(c) Error Analysis (10%)

Please compare classification error cases arising from the CNN solution and the Saak transform. What percentages of errors are the same? What percentages are different? Please give explanations to your observations. Also, please propose ideas to improve the CNN solution, the Saak transform solution or both and justify your proposal. There is no need to implement your proposed ideas.

References

[1][LeNet-5]http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf
[2][MNIST] http://yann.lecun.com/exdb/mnist/h
[3][ReLU] https://en.wikipedia.org/wiki/Rectifier_(neural_networks).
[4][Softmax] https://en.wikipedia.org/wiki/Softmax_function
[5] Saak Transform https://arxiv.org/abs/1710.04176