

This is a Java implementation of the convolutional network used for digit recognition. This is a joint project between me and Yingjie Lin.

The Java project contains ten files. C\_layer and S\_layer are the convolution and subsampling layers. Classifier is the classifier that classifies the input. Network class assembles C\_layer, S\_layer and Classifier. Vector, Matrix and image classes are supporting classes to simplify implementation. Initializer initializes a parameter file to record kernels, bias and scalar multipliers. InfoReader reads information from the parameter file.

To compile the program, type "javac \*.java" at the command line (you need JDK1.6), to run the program, type "java Test\_Network filename", where filename is one of the input test files provided in the zipped file and this file is in the same folder as the project. Input test files are text files composed of pixel values of a 36\*36 image. The first digit in the filename represents the digit this file is representing(ex: b\_0\_0, b\_0\_1 etc means this file represents digit 0). So to test b\_0\_0.txt, type "java Test\_Network b\_0\_0.txt".

The dataset contains 6000 training samples and 1000 test samples for each digit. The network has three convolution layers and three subsampling layers. First convolution layer has 6 output images, second layer has 16 images and the third has 120 images. First I trained the network for about 100,000 samples in total, and then test each digit for 1000 samples. The following is the result:

Correct percentage for digit 0 is 0.978

Correct percentage for digit 1 is 0.985

Correct percentage for digit 2 is 0.952

Correct percentage for digit 3 is 0.903

Correct percentage for digit 4 is 0.955

Correct percentage for digit 5 is 0.974

Correct percentage for digit 6 is 0.928

Correct percentage for digit 7 is 0.919

Correct percentage for digit 8 is 0.888

Correct percentage for digit 9 is 0.859