Generalizing from a small set of training exemplars for handwritten digit recognition

Wayne E. Simon and Jeffrey R. Carter
Martin Marietta Astronautics Group
P. O. Box 179
Denver, CO 80201

Abstract

The ability of a neural network to generalize from a small set of handwritten digit training exemplars is dramatically improved with two new techniques. First, the number of neural network inputs is drastically reduced by using a log-polar coordinate system to produce a centered, constant size, constant average brightness image of 65 pixels which still retains sufficient information for discrimination. Second, generalized training examples are constructed from the training exemplars with carefully chosen random variations. The results of this work are impressive. The prior state of the art, Le Cun et al.[1], used binary images, 784 inputs, 4635 nodes, 98442 connections, 9840 training exemplars, and required three days to train on a Sun SPARCstation 1. This work used 65 inputs, 75 nodes, 660 connections, 160 training exemplars, and required one hour to train on an AT-class PC, yet its results appear to be similar to those reported by Le Cun et al.

1. Introduction

The practicality of using neural networks to solve real-world recognition problems will always be limited by the number of exemplars in the available training sets, compared to the number of states in the space of the neural network's input. The effect of this mismatch between the training set and the input space of the neural network is usually called "overtraining." That is, the neural network learns to recognize the exemplars of the training set quite accurately, but the performance on a test set is degraded.

The test case chosen for this work is the recognition of gray-scale handwritten digits. It is a reasonably complex problem, interesting in its own right, and has the advantage that data generation is quite straightforward.

The overtraining problem is approached in two ways. First, the number of inputs to the neural network is minimized. Second, a technique is developed to generalize the training set.

2. Problem

The problem addressed in this paper is commonly referred to as "overtraining." The number of instances of each class to be learned by the neural network is very large, but only a small set of exemplars of each class is available for training the network. The neural network can learn the small training set very well, but the resulting network does not perform satisfactorily on test data. As a simple illustration, consider a binary image on a 256-pixel field (16 x 16). The number of states for this input is then 2^{256} or 10^{77}, a very large number. The number of states in this space filled by the training set is, at most, the number of exemplars in the training set, typically a relatively small number. Then, unless each member of the test set matches a member of the training set, errors must occur. The problem is even more severe if, as in most practical problems, continuous input values must be used.

One useful technique for reducing overtraining is a mechanism which eliminates unnecessary connections from the network, such as the technique which uses the second derivative information of Recursive Error Minimization (REM) equations to remove and add network connections during training[2], called REM Thinning. Such a technique should be used whenever possible, but will not necessarily correct overtraining. Such a technique used alone may not affect overtraining unless it is applied so drastically that it corrects overtraining not by improving the network's performance on the test set, but by degrading the network's performance on the training set to unsatisfactory levels.

The problem of overtraining was discussed in a paper on the I-4I problem[3]. Overtraining was avoided for that problem by generating an infinite training set as the network learned, using a random-number generator. That solution was made possible by the analytical nature of the problem. For most real classification problems, such a solution is not possible.

This paper discusses a real classification problem for which overtraining is a definite problem: learning to identify gray-scale handwritten digits with a training set of sixteen exemplars of each digit.

0-8194-0578-7/91/$4.00

3. Approach

If a topological transform could be defined to include all the exemplars, construction of generalized examples would be simple. A random choice of parameters for the transform would produce an infinite set of training examples. Such a transform is rarely available, so an alternate approach must be used. The approach taken in this work is to find an appropriate representation of the input to a neural network that can be used to minimize the number of inputs to the network and to allow typical examples of the input to be generated from the specific exemplars in the training set. The remainder of this section describes the specific application of this approach to the test case of recognizing gray-scale handwritten digits.

The data used in this work were obtained by collecting handwritten digits from 26 individuals. A wick pen was used, and the writers were asked to make the digits a given size. The digits were then photographed with a digital camera, using 128 x 128 pixels. The maximum linear dimension of the digits ranged from 20 to 90 pixels with widely varying positions within the image. Illumination was from fluorescent fixtures to the side, so that the variation in brightness across the image was sometimes greater than the local variation produced by the pen. Figure 1 shows a row of pixels across the lower portion of an "8" (training writer 16), for which this was true. Humans see both sides of the loop in this image, but the right side is not apparent in Figure 1. In order to isolate the local brightness variations associated with the digit, the digit image was smoothed with a characteristic dimension of the order of the digit size (32 pixels) and subtracted from the original image. The result was then squared and normalized to a maximum pixel value of 255. This corresponds roughly to a spatial bandpass, detection, and brightness normalization. The result for the row of pixels in Figure 1 is shown in Figure 2. The right side of the loop is now apparent, though still smaller than the left side. Note that this process brings bright figures on a dark background and dark figures on a bright background to a common representation.

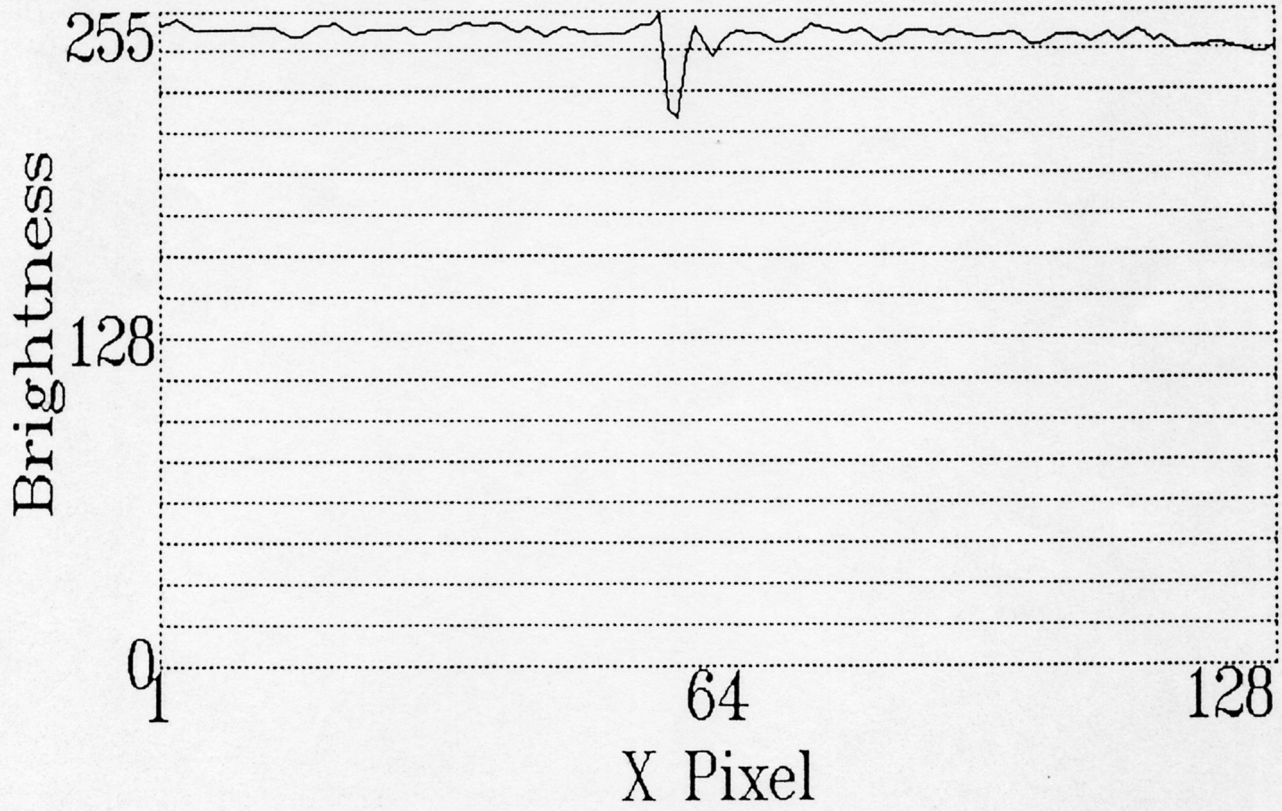

Figure 1. Brightness cross-section of an original gray-scale digit 8

The next step is to choose a method of reducing the number of inputs to the neural network to a much smaller number than the 16384 pixels in the detected images. In many real recognition problems, results must be independent of orientation. Even in the digit problem, the long axis of the digits may vary by 20° or more from writer to writer. This suggests that a polar coordinate system is appropriate. The variation in sensor density in the eye makes plausible a geometric variation in pixel size with distance from the origin. Thus the log-polar (Mellin transform) coordinate system was chosen. Implementation considerations suggest that a simpler process should be substituted for a true transform.

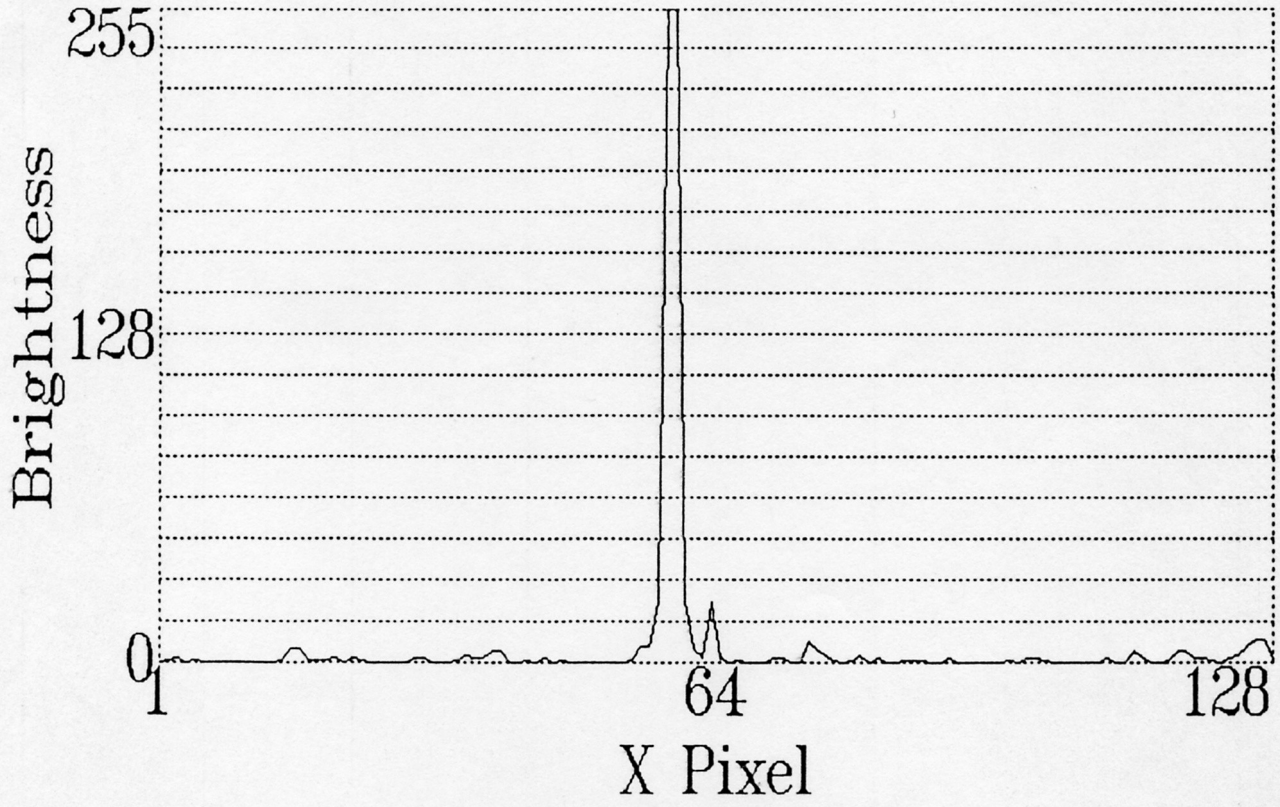

Figure 2. Brightness cross-section of the detected version of the 8 in Figure 1

A figure consisting of circles and rays (Figure 3), is superimposed on the square pixel grid. All the pixels within a patch are summed to obtain the pixel value in the new coordinate system. A simple, hard-wired hardware system could perform this transformation. The following paragraphs describe the specific steps used to perform the transformation.

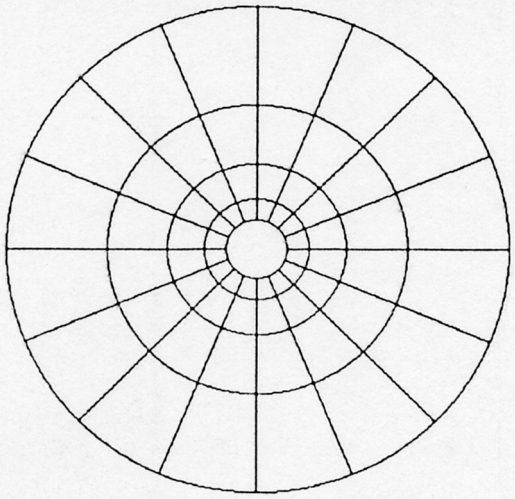

Figure 3. The log-polar coordinate system

A primary requirement for the use of such a system is that the origin must be in the center of the image of the digit. Since the center should be a geometric center, and not a brightness center, a binary image is constructed by thresholding the detected image. These binary images are those shown in Figures 6 and 7 to illustrate the writer-to-writer variation in the digits. All images were binarized with the same threshold, which is why some of the images in Figures 6 and 7 show missing features. The row coordinate of the center is obtained by summing columns and smoothing the sums with a characteristic length of 128 rows. This smoothed function is shifted 64 rows and subtracted from itself. A sign

change in this difference function indicates the position of two rows, 64 pixels apart, with the same value. The midpoint, 32 pixels from the sign change, is then the midpoint of the digit in the row direction. The center column is obtained in the same manner.

This process effectively finds the centroid of the digit. However, if the transformation to log-polar coordinates is done in hardware, the features of the log-polar coordinate system can be used to control the camera position to simply achieve centering without calculating the centroid. This technique is described by Weiman and Juday[4].

With the center of the log-polar coordinate system determined, the outer diameter of the coordinate system must be chosen. A variety of choices were investigated, and the choice described below is believed to give nearly a minimum number of pixels for this task without sacrificing discrimination. The maximum diameter is chosen to be 96 pixels, since this is larger than the largest digit in the data. Twenty rings are constructed in this area, with a ratio of outer to inner diameters of $2^{1/4}$, and overlaid on the digit. The remaining center spot is a unique pixel called the foveal spot. The value of each ring is just the sum of all the original pixels in its area. The value of each ring is then normalized by the sum of all the rings plus the foveal spot.

Now the problem of scaling can be addressed. Starting with the outer ring, the value of each ring is examined. The first ring whose value is greater than 0.02 (2% of the total) defines the maximum diameter of the digit. With this information, the log-polar coordinate system can be chosen and applied to the detected gray-scale image. Starting with the outer diameter of the digit determined from the binary image, four rings are specified, with a ratio of outer to inner diameters of $2^{3/4}$ (1.68). The center is then the foveal spot with a diameter of one-eighth of the digit diameter. 16 equally spaced rays (22.5°) are applied, resulting in a total of 65 pixels. This coordinate system is then applied to the detected gray-scale image. The value of a pixel in the log-polar system is simply the sum of all the square pixels within its area. The log-polar values are then normalized by the sum of all the log-polar pixels. The result is a log-polar coordinate system with 65 pixels, and the digit size varies at most by $2^{1/4}$, or less than ±10% of the maximum diameter.

If the image were mapped to log-polar coordinates in hardware and centered by controlling the camera position as discussed above, scaling could be achieved without these calculations by using the features of the log-polar coordinate system to control the position of a zoom lens on the camera. Just as the log-polar representation allows centering without calculating the centroid, it also allows scaling without performing the calculations described here.

This process produces constant size, constant average brightness images of 65 pixels from original 16384-pixel images of digits in varying location, variable brightness, and with a size range of more than a factor of four. This preprocessing, although it seems complex, is just a set of operations which blur, add, subtract, compare, and normalize, all of which are quite feasible in simple hardware.

Now that all the exemplars of the training set and the test set have been produced, the generalization of the training set is developed. Two factors are used. First, the vertical orientation of the digits is clearly variable. This suggests that a random rotation of the exemplars is appropriate. Second, even after scaling, the size of the digits still has some variation, so small random variation in size is appropriate. The generalized training sets used were obtained by applying a uniformly distributed rotation of ±22.5°, and a uniformly-distributed scale variation of ±3.5%. A specific instance of the generalized training set is obtained by choosing an exemplar, selecting rotation and scale changes with a random-number generator, and interpolating between the pixel values of the exemplar to the positions selected by the random rotation and scale.

A variation in the center of the log-polar coordinate system could also be used; however, experimentation showed that such a variation did not improve the network's results. Apparently, the centering technique described above is very accurate.

4. Results

After many runs with REM Thinning active, it was determined that the problem of classifying gray-scale handwritten digits using the log-polar representation described above could be solved by a network with 65 inputs, no intermediate nodes, and 10 output nodes. Such a network with no intermediate nodes is called a perceptron, even though REM equations were used to train it. The runs described here were made using a perceptron with REM Thinning inactive.

Each of the output nodes represents one of the 10 digits; the output of the network is the digit represented by the output node with the largest value. If this is not the correct digit, the network has made an error; the number of errors divided by the total number of digits on which the network was tested yields a percent error value. A percent error of less than 10% was chosen as an acceptable level of error. For real applications, a more stringent definition of acceptable error may be needed, and more than 16 exemplars of each digit may be used to train the network.

Figure 4 shows the effect of the number of training epochs (one epoch is the presentation of one exemplar of each digit) on the level of error when the network is trained on the training set without generalizing. That is, the same 16 exemplars of each digit are presented to the network repeatedly. As expected, the network can learn the training set very well, but is never able to obtain acceptable results on a test set of 10 examples of each digit. The network is unable to learn the training set exactly because of the combination of the minimization of the number of inputs due to the log-polar transformation and the simplicity of the perceptron network architecture. The minimum error level of 1.3% corresponds to two errors in the 160-exemplar training set. The specific digits are the "9" of training writers 3 and 5 (see Figure 6), which were identified as an "8" and a "5" respectively.

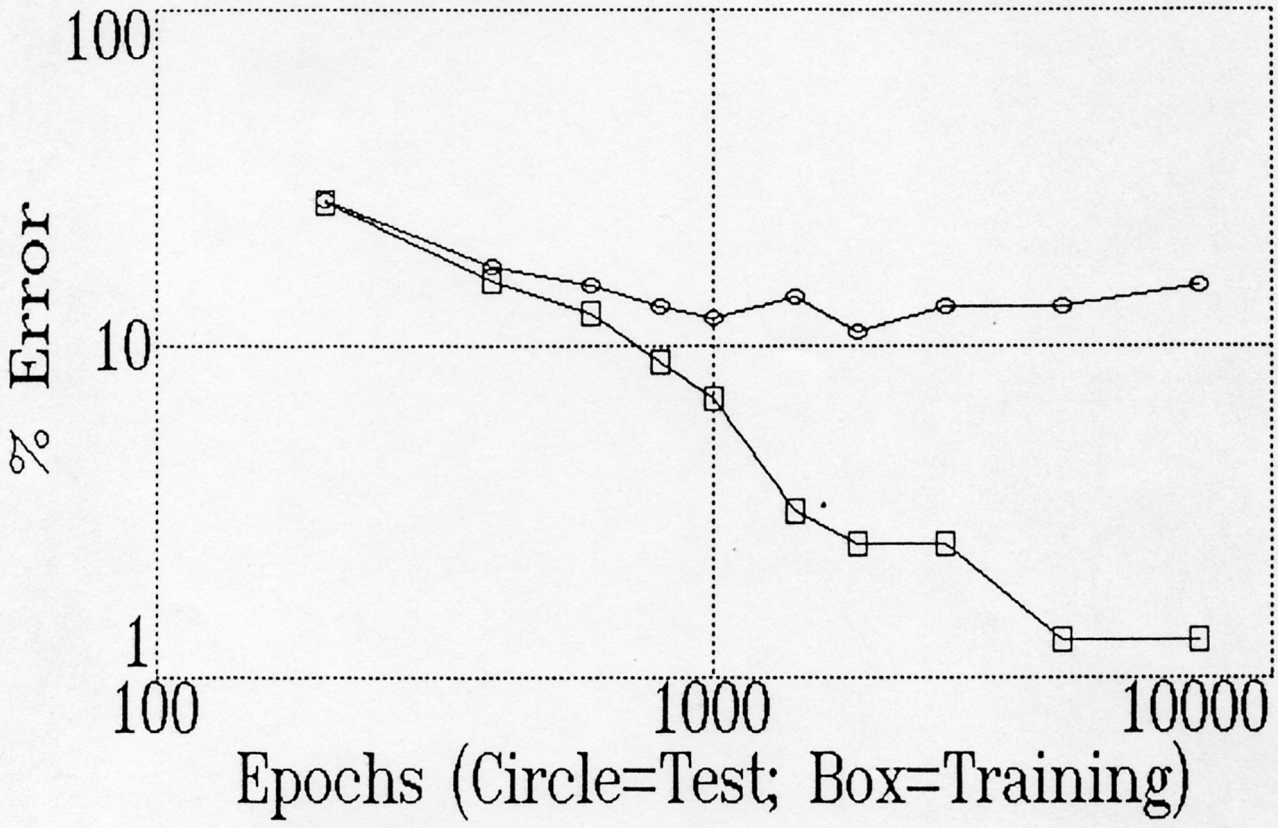

Figure 4. Training on the training set without generalization

When the network is trained on generalized examples generated from the 16 exemplars of each digit as described above, the effect of the number of training epochs on the level of error changes to that shown in Figure 5. Unlike Figure 4, the error level on the test set becomes acceptable and remains acceptable, the error level on the training exemplars is increased, and the difference between the test and training set error levels is decreased.

The error level on the test set is lower because the training examples generated from the training set are more like the test set than are the exemplars of the training set. The error level on the training exemplars is increased because the network has not been trained on the training exemplars, but on the generalized examples generated from them. In effect, the training exemplars constitute a separate test set. With a perfect generalization technique, the error levels for the test and training sets would be identical.

A separate run of 2000 training epochs gave results of 8% error on the test set and 5.6% error on the training exemplars, about what one would expect from Figure 5. This shows that using the techniques described above allows a very simple network (perceptron), trained on a very small training set (16 exemplars of each digit) for a short time (2000 to 3000 epochs), to give acceptable levels of error on both test and training sets. Note that 2000 epochs is just 125 examples generated from each exemplar.

These results are impressive. The prior state of the art, Le Cun et al.[1], used binary images, 784 inputs, 4635 nodes, 98442 connections, 9840 training exemplars, and required three days to train on a Sun SPARCstation 1. This work used 65 inputs, 75 nodes, 660 connections, 160 training exemplars, and required one hour to train on an AT-class PC, yet its results appear to be similar to those reported by Le Cun et al.

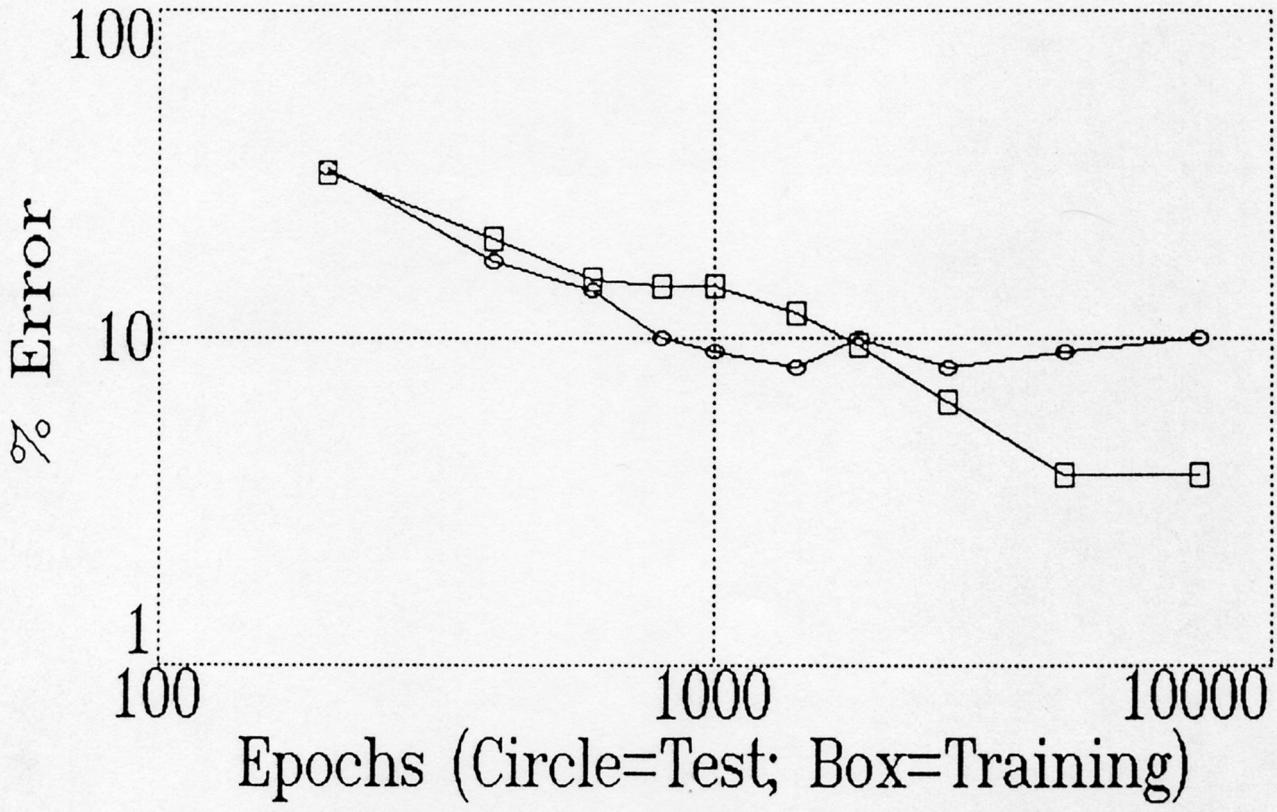

Figure 5. Training on the training set with generalization

Le Cun et al. state that their network achieves an error level of 3.4% on test data, but do not define what they mean by an error. One common measure of error considers each presentation of a test exemplar to be 10 tests, one of each output node. An output node has made an error is it is negative for the digit it represents, or if it is positive for a digit it does not represent. This measure of error is not useful in determining if a system is suitable for a specific application, but is widely used because it gives much lower error rates than the measure defined above. Using this measure of error, the network described here had an error rate of 3% on the test set. If this is the measure of error used by Le Cun et al., this work achieves very similar results with a much simpler network in much less time on a less powerful machine.

5. Conclusion

This work demonstrates that image preprocessing, designed for relatively simple hardware implementation, can drastically simplify the neural network required for handwritten-digit recognition (and, probably, for other recognition and classification problems). Previous work used hundreds of input nodes, hundreds of intermediate nodes, and as many as one-hundred thousand connections for this problem. The present network uses 65 input nodes and 660 connections for the same task.

Most importantly, this work demonstrates that appropriate random perturbation of a small set of exemplars can construct an infinite generalized training set which produces nearly equal recognition on the training and test exemplars and, at the same time, results in better recognition of the test exemplars than is possible with a neural network trained only on the training exemplars.

This work also suggests that relatively simple, real-time hardware with a training set constructed from a larger number of exemplars, and perhaps with a few more log-polar pixels, could certainly attain recognition levels above the 92% demonstrated here. This approach, used with an adequate number of training exemplars and a more complicated network architecture, could probably achieve similar recognition levels on the more difficult problem of handwritten character recognition.

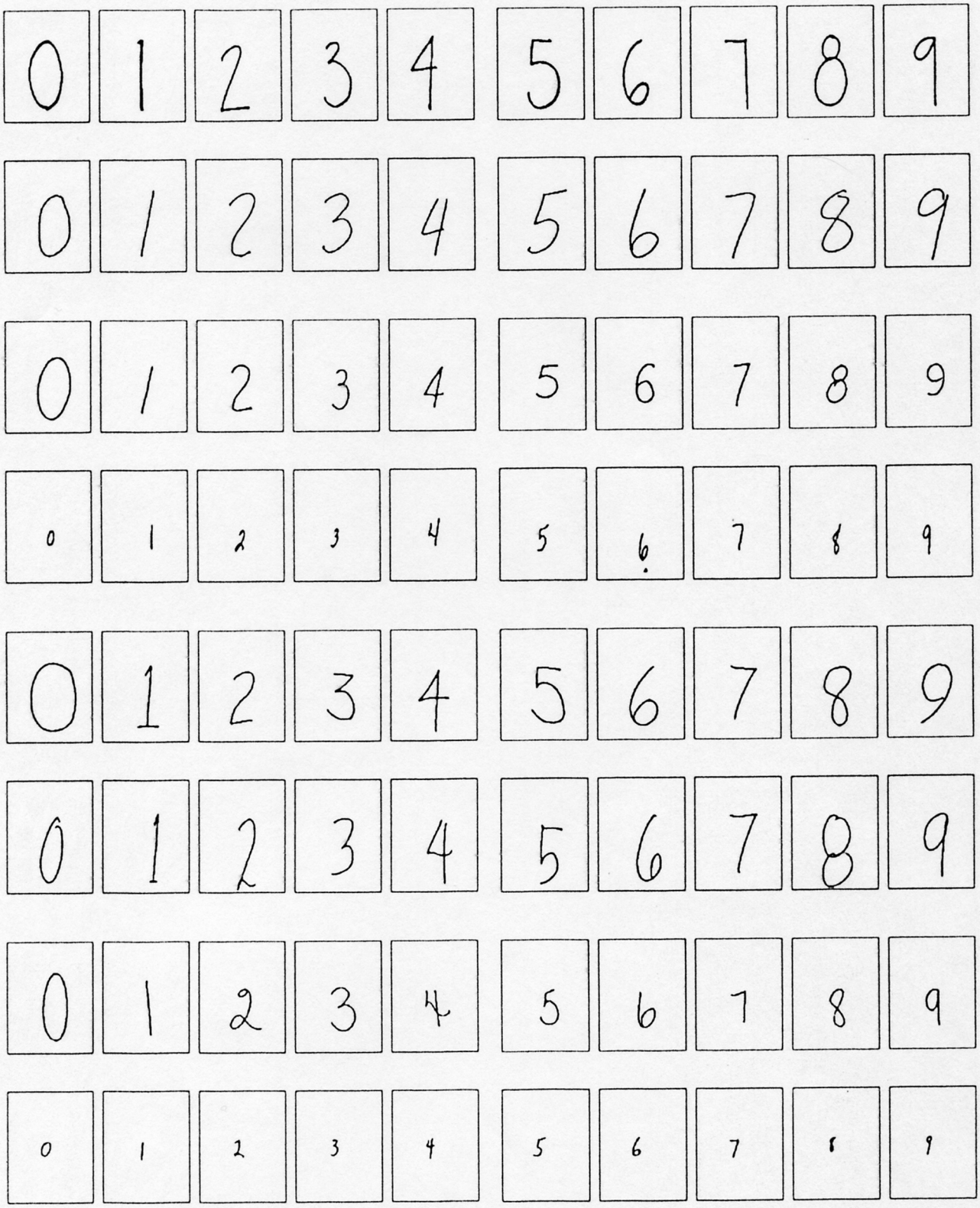

Figure 6a. The first eight writers of the training set.

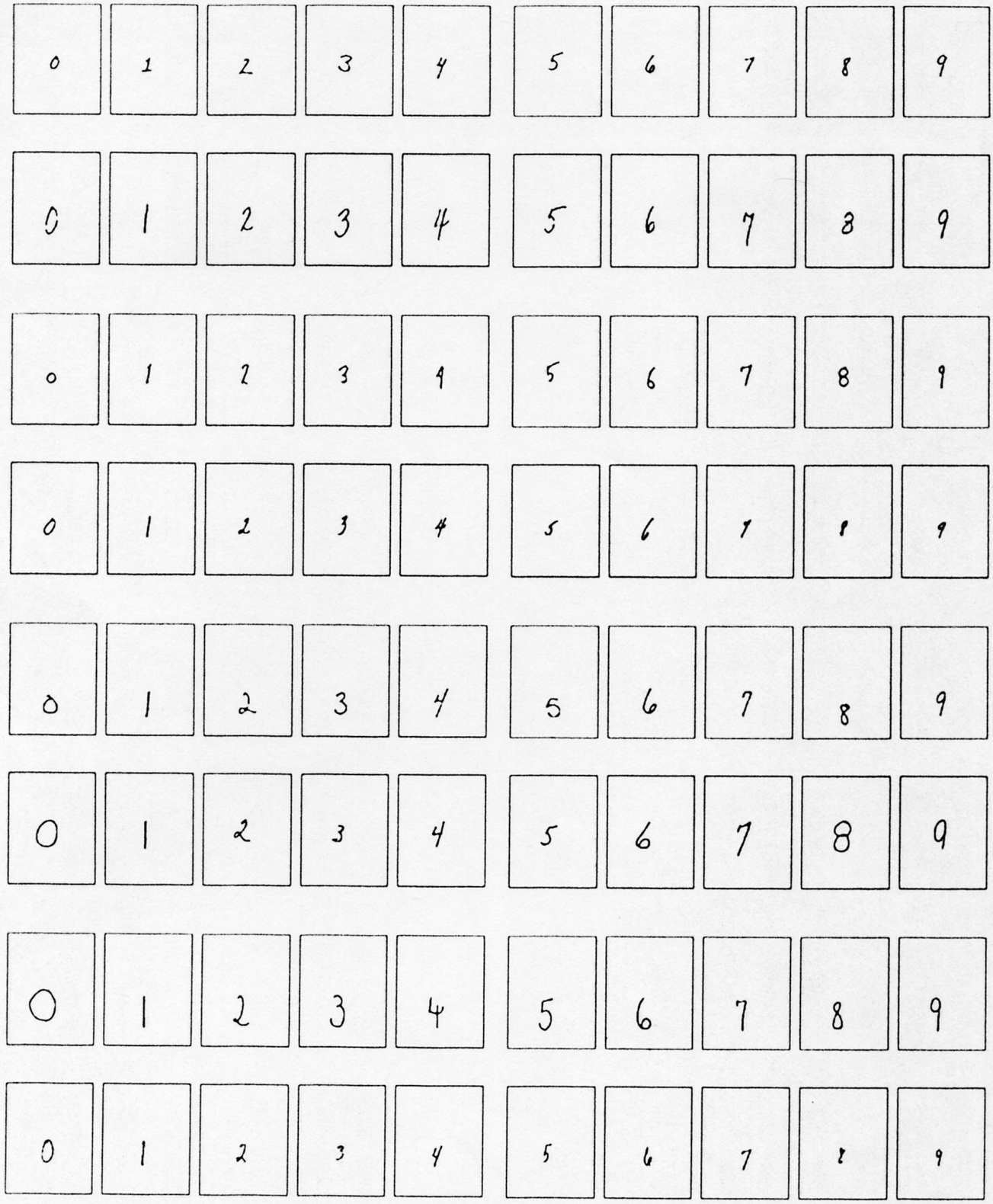

Figure 6b. The last eight writers of the training set.

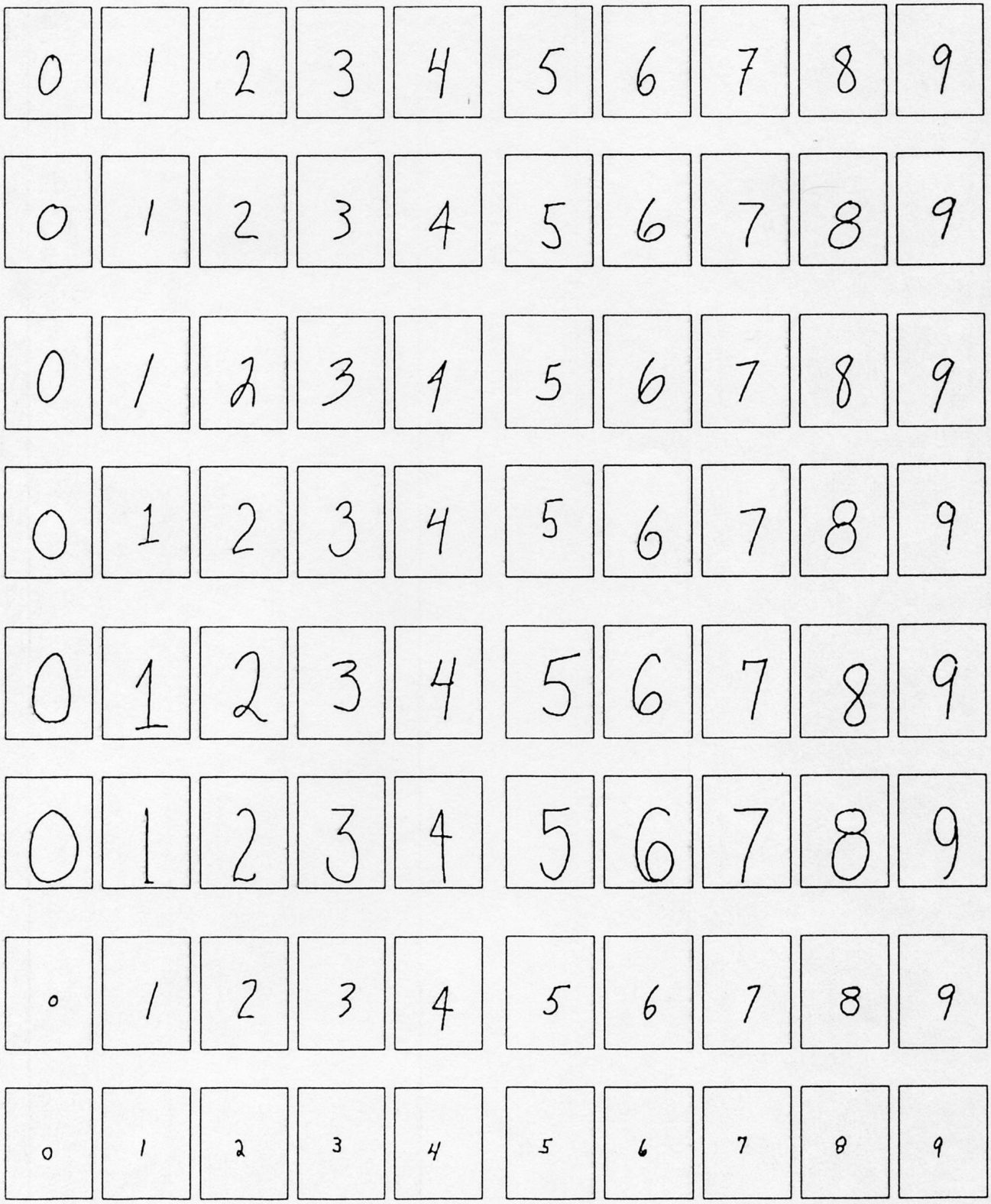

Figure 7a. The first eight writers of the test set.

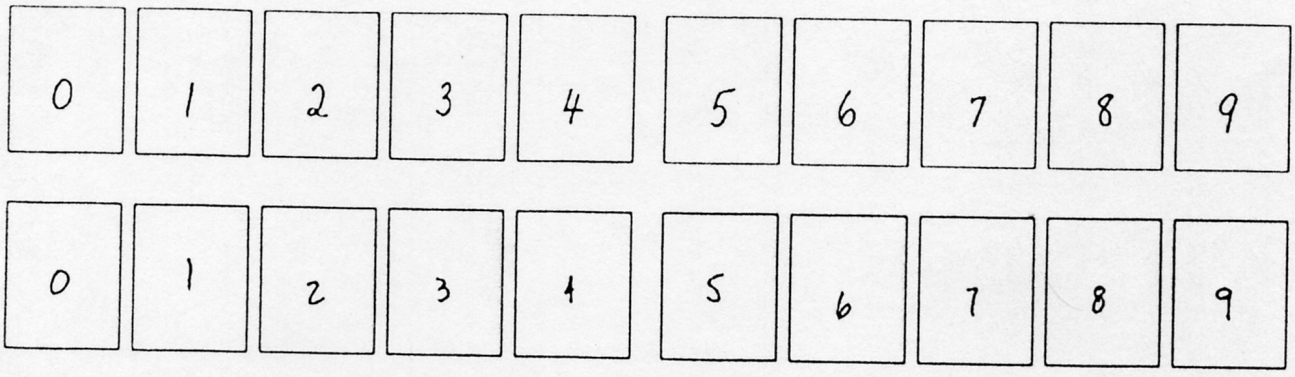

Figure 7b. The last two writers of the test set.

References

1. Le Cun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," in D. Touretzky, editor, *Advances in Neural Information Processing Systems II*, Morgan Kaufmann, 1990

2. Simon, W., and J. Carter, "Removing and Adding Network Connections with Recursive Error Minimization (REM) Equations," *Proceedings of Applications of Artificial Neural Networks*, SPIE, 1990

3. Carter, J., and Simon, W., "Statistical Learning from Nonrecurrent Experience with Discrete Input Variables and Recursive Error Minimization (REM) Equations," *Proceedings of Applications of Artificial Neural Networks*, SPIE, 1990

4. Weiman, C., and R. Juday, "Tracking Algorithms Using Log-Polar Mapped Image Coordinates," *Proceedings of Advances in Intelligent Robotic Systems*, SPIE, 1990