Li Deng

The MNIST Database of Handwritten Digit Images for Machine Learning Research

In this issue, "Best of the Web" presents the modified National Institute of Standards and Technology (MNIST) resources, consisting of a collection of handwritten digit images used extensively in optical character recognition and machine learning research.

Handwritten digit recognition is an important problem in optical character recognition, and it has been used as a test case for theories of pattern recognition and machine learning algorithms for many years. Historically, to promote machine learning and pattern recognition research, several standard databases have emerged in which the handwritten digits are preprocessed, including segmentation and normalization, so that researchers can compare recognition results of their techniques on a common basis. The freely available MNIST database of handwritten digits has become a standard for fast-testing machine learning algorithms for this purpose. The simplicity of this task is analogous to the TIDigit (a speech database created by Texas Instruments) task in speech recognition. Just like there is a long list for more complex speech recognition tasks, there are many more difficult and challenging tasks for image recognition and computer vision, which will not be addressed in this column.

DATA
General site for the MNIST database:
http://yann.lecun.com/exdb/mnist

Code to read the MNIST database:
http://www.mathworks.com/matlabcentral/fileexchange/27675-read-digits-and-labels-from-mnist-database

The MNIST database was constructed out of the original NIST database; hence, modified NIST or MNIST. There are 60,000 training images (some of these training images can also be used for cross-validation purposes) and 10,000 test images, both drawn from the same distribution. All these black and white digits are size normalized, and centered in a fixed-size image where the center of gravity of the intensity lies at the center of the image with 28×28 pixels. Thus, the dimensionality of each image sample vector is $28 * 28 = 784$, where each element is binary. This is a relatively simple database for people who want to try machine learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. Using the references provided on the Web site, students and educators of machine learning can also benefit from a rather comprehensive set of machine learning literature with performance comparison readily available.

EVALUATION OF MACHINE LEARNING ALGORITHMS USING MNIST
General evaluation results on MNIST:
http://yann.lecun.com/exdb/mnist/

Details of logistic regression evaluated on MNIST:
http://deeplearning.net/tutorial/logreg. html

Many well-known machine learning algorithms have been run on the MNIST database, so it is easy to assess the relative performance of a novel algorithm. The Web site http://yann.lecun.com/exdb/mnist/ was updated in December of 2011 to list all major classification techniques and

their results that were obtained using the MNIST database. In most experiments, the existing training data from the database were used in learning the classifiers, where "none" is entered in the "Preprocessing" column of the table on the Web site. In some experiments, the training set was augmented with artificially distorted versions of the original training samples. The distortions include random combinations of jittering, shifts, scaling, deskewing, deslanting, blurring, and compression. The type(s) of these and other distortions are specified in the "Preprocessing" column of the table as well.

A total of 68 classifiers are provided in the comparison table on the Web site, where "Test Error Rates (%)" and links to the corresponding reference(s) are provided. These 68 machine learning techniques are organized into six broad categories:

- linear classifiers
- k-nearest neighbors
- boosted stumps
- nonlinear classifiers
- support vector machines (SVMs)
- neural nets (with no convolutional structure)
- convolutional nets.

Each category contains up to 21 entries with very brief description of each in the "Classifier" column of the table. Much of the early techniques published in [1] are listed in the table.

BRIEF ANALYSIS OF THE MACHINE LEARNING ALGORITHMS EVALUATED ON MNIST
Comparing all the 68 classifiers listed on the MNIST Web site, we can make a brief analysis on the effectiveness of various techniques and of the preprocessing methods.

"The first entry to the handwritten image database." Cartoon by Tayfun Akgul (tayfun.akgul@ieee.org).

SUMMARY

The MNIST database gives a relatively simple static classification task for researchers and students to explore machine learning and pattern recognition techniques, saving unnecessary efforts on data preprocessing and formatting. This is analogous to the TIMIT database (a speech database created by Texas Instruments and Massachusetts Institute of Technology) familiar to most speech processing researchers in the signal processing community.

Just like the TIMIT phone classification and recognition tasks that have been productively used as a test bed for developing and testing speech recognition algorithms [7], MNIST has been used in a similar way for image and more general classification tasks. The Web site we introduce in this column provides the most comprehensive collection of resources for MNIST. In addition, this "Best of the Web" column also provides an analysis on a wide range of effective machine learning techniques evaluated on the MNIST task.

AUTHOR

Li Deng (deng@microsoft.com) is a principal researcher at Microsoft Research, Redmond, Washington.

Neural net classifiers tend to perform significantly better than other types of classifiers. Specifically, convolution structure in neural nets accounts for excellent classification performance. In fact, the record performance, about 0.27% error rate or 27 errors in the full 10,000 test set, is achieved by a committee of convolutional nets (with elastic distortion in augmenting the training set) [2]. Without the "committee," one single very large and deep convolutional neural net gives also a very low error rate of 0.35% [3].

The use of distortions, especially elastic distortion [4], to augment the training data is important to achieve very low error rates. Without such distortion, the error rate of a single large convolutional neural net is increased from 0.35% to 0.53% [5].

The depth of neural nets also accounts for low error rates. With both convolution structure and distortions, the deep versus shallow nets give the error rates of 0.35% [3] and 0.40–0.60% [4], respectively. Without convolution structure and distortions or other types of special preprocessing, the lowest error rate in the literature, 0.83%, is achieved using the deep stacking/convex neural net [6]. The error rate is increased to 1.10% [7] when a corresponding shallow net is used.

Behind neural net techniques, k-nearest neighbor methods also produced low error rates, followed by virtual SVMs. Note that preprocessing is needed in both cases for the success.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] D. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Proc. ICDAR*, 2011.

[3] D. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. IJCAI*, 2011.

[4] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. ICDAR*, 2003.

[5] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE Int. Conf. Computer Vision (ICCV 2009).*

[6] L. Deng and D. Yu, "Deep convex network: A scalable architecture for speech pattern classification," in *Proc. Interspeech*, Aug. 2011.

[7] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Proc. ICASSP*, Mar. 2012.

[SP]