MultiScience - XXX. microCAD International Multidisciplinary Scientific Conference
University of Miskolc, Hungary, 21-22 April 2016, ISBN 978-963-358-113-1

Efficient handwritten digit recognition using normalized cross-correlation

Szabolcs Stremler[1], Zsolt Karácsony[2]

University of Miskolc, Department of Applied Mathematics,
H-3515 Miskolc-Egyetemváros, Hungary

Email: [1]stremler@iit.uni-miskolc.hu, [2]matkzs@uni-miskolc.hu

Abstract. The aim of handwriting recognition is to recognize series of handwritten characters by machines. It is a popular field in computer vision and has several commercial applications which requires precise and fast processing algorithms. Our applied research proposes a feasible statistical method and its comparison which showed competitive results in the evaluation phase based on prior works. The benchmark methods are neural networks and support vector machines which are well-known for their convincing performance in many areas of applications. We used the famous MNIST dataset of handwritten digits for testing purposes.

Keywords: machine learning, neural network, support vector machine, image processing

1 Introduction

The main challenge of handwritten digit recognition derives from the diversity of each handwriting style and from the similarity of digits. The performance of the classification algorithm and the feature extraction from the images, while the object turns into data shares the same importance in practice. Several prior work has been made in this field, LeCun et al. [4] proposed gradient-based learning methods, such as multilayer neural networks (NN) and benchmarked them, Ciresan et al. [1] showed that, the error rate of the neural networks corresponds with the hidden layer size and the number of neurons in it, Maji et al. [6] introduced the using of histogram of oriented gradients (HOG) feature extraction method.

Our work shows that a non-learning algorithm can also be competitive in comparison with machine learning techniques, if the database is well represented. The benchmark methods are trained in three different steps:

1. image preprocessing using Otsu's thresholding method (see [8]),
2. HOG feature extraction,
3. NN and SVM training.

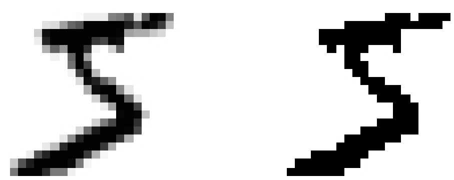
DOI: 10.26649/musci.2016.058

2 Image processing

The original grayscale images in the MNIST dataset are noisy in most cases and requires preprocessing before the feature extraction in order to minimize the error rate in the classification phase.

2.1 Otsu's thresholding method

Otsu's thresholding method [8] converts a grayscale image to a binary image by choosing a threshold to minimize the intraclass variance between the black and white pixels.



(a) Original image (b) Thresholded image

Fig. 1: MNIST digit thresholding

2.2 Histogram of oriented gradients

Histogram of oriented gradients feature extraction method is a powerful technique for describing shapes and which already showed impressive results on several dataset (see[2], [9]). The algorithm works in four different steps:

1. *Gamma and color normalization*
   Many image feature descriptor uses gamma and color normalization to preprocess the image, as Dalal et al. pointed out [2] sometimes it has only modest effect on performance.

2. *Gradient computation*
   The image gradient can be obtained by filtering the image with a 1-D vector horizontally $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$ and vertically $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}^\top$. It describes the directional change in the intensity of an image.

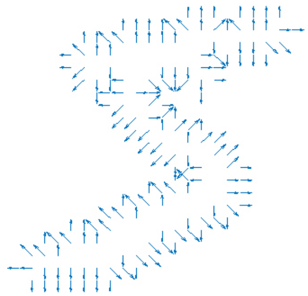


Fig. 2: Image gradient of an MNIST digit

3. *Orientation binning*
   In this step the algorithm splits the image into cells and for each pixel within the cell calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient and the votes accumulates into the bins.

4. *Normalization and descriptor blocks*
   Gradient strengths vary over a wide range, thus normalization turns out to be essential for good results. Descriptor blocks are grouped cells and normalized separately. The final descriptor is the vector of components of the cells of all blocks.

3 Classification algorithms

Classification algorithms plays an important role in the construction of pattern recognition systems and many of them were benchmarked with the MNIST dataset in prior works [4]. The basic approach of classification computes a function $y(n) = f(x(n), w)$, where $x(n)$ is the n-th input pattern, w is the collection of adjustable weights and $y(n)$ can be interpreted as the predicted class label of $x(n)$. Let $\varepsilon(n) = (d(n) - y(n))^2$ be the loss function which measures the discrepancy between the correct label $d(n)$ and predicted label $y(n)$. The average loss function is $\varepsilon_{train}(w)$ and in the simplest setting, the learning problem is to find a weight w which minimizes the function $\varepsilon_{train}(w)$.

3.1 Neural network

In our experiments we used feedforward backpropagation multilayer neural networks with logistic transfer function

$$\varphi(x) = \frac{1}{1 + e^{-ax}}, \quad x \in \mathbb{R},$$

where $a > 0$ and the weights were corrected in each epoch with gradient descent method. The backpropagation method calculates the gradients recursively backward from the output layer, thus the gradient of the j-th neuron in the output layer is

$$\delta_j(n) = (d_j(n) - y_j(n))^2 \varphi'_j(v_j(n)),$$

where $d_j(n)$ is the correct output, $y_j(n)$ is the predicted output, $\varphi'_j(.)$ is the first derivative of the transfer function and $v_j(n)$ is the summing junction. The gradient of the j-th neuron in an arbitrary hidden layer is

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_{k \in \{\text{layer after } j\}} \delta_k(n) w_{kj}(n),$$

where $w_{kj}(n)$ is the weight between the neuron k and j. Therefore the weight of the neuron i which is before the layer of the neuron j is

$$w_{ji}(n+1) - w_{ji}(n) = \Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \tag{1}$$

where $y_i(n)$ is the output of neuron i, $\delta_j(n)$ is the gradient of the neuron j and $\eta > 0$ is the learning rate.

Based on the equation (1) the new weights can be computed in each epoch.

3.2 Support vector machine

Support vector machine is a maximum margin classifier which means it looks for an optimal decision function in a high-dimensional space. The decision function is

$$f(x, \alpha) = \sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + b = 0,$$

where x_i is the input vector, y_i is the class label, $K(\cdot, \cdot)$ is the kernel function, b is the bias and α_i is the Lagrange multiplier. In this equation we have to determine the α_i coefficients and the b bias by the maximization of the function

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to the constraints

$$\sum_{i=1}^{N} y_i \alpha_i = 0,$$

$$0 \le \alpha_i \le C, \qquad i = 1, \ldots, N.$$

We used the following $K(\cdot, \cdot)$ kernel functions in the experiment.

Linear	$K(x, x_i) = \langle x, x_i \rangle$
Polynomial	$K(x, x_i) = (\langle x, x_i \rangle + 1)^p$
RBF (Gaussian)	$K(x, x_i) = \exp(-\frac{1}{2\sigma^2} \left\| x - x_i \right\|^2)$

4 Normalized cross-correlation method

In signal processing the cross-correlation does not include a standardizing factor, but for shape recognition it is advised to use it, if the energy of the image varies with position. Cross-correlation is the similarity measure of two signal at different points in time and it is also applicable for images [5]. For a given region cross-correlation is calculated as

$$\gamma(u,v) = \frac{\sum_{x,y}\Big(f(x,y) - \bar{f}_{u,v}\Big)\Big(t(x-u,y-v) - \bar{t}\Big)}{\sqrt{\sum_{x,y}\Big(f(x,y) - \bar{f}_{u,v}\Big)^2 \sum_{x,y}\Big(t(x-u,y-v) - \bar{t}\Big)^2}},$$

where f is the first image, t is the second image, $\bar{f}_{u,v}$ is the mean of $f(x,y)$ under the region of t, $\bar{t}$ is the mean of t. The cross-correlation matrix can be visualized as a surface as shown below.

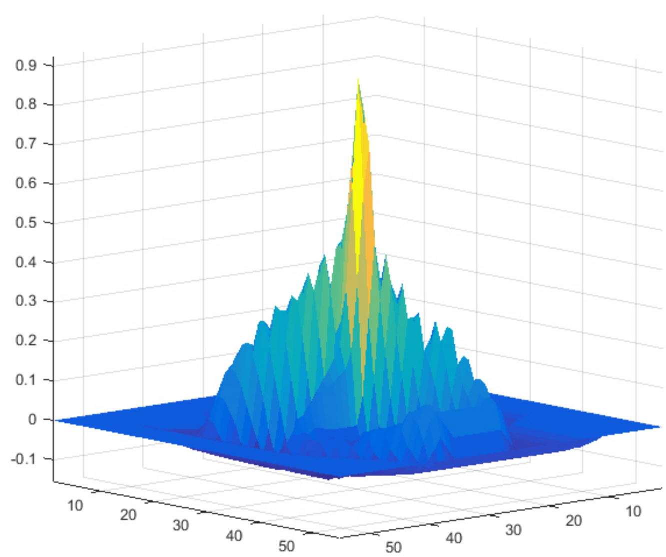


Fig. 3: Image cross-correlation

5 Results

Our first experiment is based on the idea that a dataset in which every class is well represented should show satisfying results for simple similarity measure based classification. Thus, we calculated the cross-correlation for every digit in the MNIST dataset and classified the given digit by the maximum of all global maximums of the cross-correlation matrices. The confusion matrix of the detailed results can be seen in Table 1. The first column consists of the correct labels, the predicted labels are in the first row.

	0	1	2	3	4	5	6	7	8	9	Result
0	976	0	7	3	0	5	4	1	4	5	97.11
1	0	1132	4	0	3	0	4	13	3	9	96.92
2	0	2	994	3	0	0	0	2	3	3	98.71
3	0	0	3	986	0	16	0	0	15	2	96.48
4	0	0	0	0	936	1	1	4	2	8	98.32
5	1	0	0	8	0	849	1	0	5	5	97.70
6	1	1	1	0	3	12	948	0	6	0	97.53
7	1	0	17	4	1	0	0	994	4	7	96.69
8	0	0	4	4	2	7	0	0	926	1	98.09
9	1	0	2	2	37	2	0	14	6	969	93.80
Σ											97.14%

Table 1: Maximum image cross-correlation classification

We used multilayer neural network and SVM as benchmark methods with several different network topology and kernel function, see Table 2. The benchmark results corresponds with the prior works which was made in this area.

Algorithm	Settings	Result	Time
NN	10 neuron	97.8%	42 mp
NN	12 neuron	98.1%	57 mp
NN	36 neuron	98.3%	63 mp
NN	36 12 neuron	98.3%	91 mp
SVM	Linear kernel	98.55%	65 mp
SVM	Gaussian kernel, $\sigma = 1$	98.55%	223 mp
SVM	Polynomial kernel, degree = 3	98.81%	87 mp
SVM	Polynomial kernel, degree = 4	98.87%	80 mp

Table 2: Benchmark results

The results of the first experiment with maximum correlation classification approximates the results of the benchmark methods, but slightly underperform them. The performance of neural networks strongly depends on the number of hidden layers and neurons in it and also in case of SVM higher degree of the kernel function can cause less error rate.

In the second experiment we determine the image cross-correlation coefficients in the same way as mentioned above, but in this case we take the greatest 100 element and calculate the HOG feature vector and its correlation coefficient for each. The predicted class label is the one with the greatest HOG correlation coefficient out of the 100 element. The results can be seen in Table 3.

	0	1	2	3	4	5	6	7	8	9	Result
0	977	0	2	0	1	0	1	0	5	3	98.79
1	0	1131	0	0	1	0	2	4	2	4	98.86
2	0	2	1017	3	0	0	0	5	3	0	98.74
3	1	0	4	988	0	13	0	1	5	5	97.15
4	0	0	0	0	946	0	0	2	1	5	99.16
5	0	0	0	7	0	868	3	0	2	1	98.52
6	1	2	0	1	4	5	951	0	1	0	98.55
7	1	0	7	4	1	1	0	998	3	6	97.75
8	0	0	1	5	2	3	1	0	944	0	98.74
9	0	0	1	2	27	2	0	18	8	985	94.44
Σ											98.07%

Table 3: Maximum HOG feature cross-correlation classification

The overall performance increased with 0.93% and this method outperforms the simplest neural network from the benchmark cases. Most error of the algorithm comes from the misclassification of digit 9.

6 Conclusion

The maximum correlation method approximates the performance of other classifiers if they use relatively simple settings and the dataset is well-represented. The disadvantage of this algorithm is the high computational cost which increases with the number of cases.

7 Acknowledgements

This research was (partially) carried out in the framework of the Center of Excellence of Mechatronics and Logistics at the University of Miskolc.

References

1. D. C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber: Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation, 22(12)* 3207–3220, 2010.
2. N. Dalal, B. Triggs: Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference* 886–893, 2005.
3. I. Fazekas: *Neural networks.* University of Debrecen, 2013.
4. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner: Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE, 86(11)* 2278–2324, 1998.
5. J. P. Lewis: *Fast Normalized Cross-Correlation.* Industrial Light & Magic, 1995.
6. S. Maji, J. Malik: *Fast and Accurate Digit Classification.* EECS Department, University of California, Berkeley, 2009.
7. MNIST: The MNIST Database of Handwritten Digits. `http://yann.lecun.com/exdb/mnist/`.
8. N. Otsu: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics, 9(1)* 62-66, 2005.
9. F. Suard, A. Rakotomamonjy, A. Bensrhair, A. Broggi: Pedestrian Detection using Infrared images and Histograms of Oriented Gradients. *Intelligent Vehicles Symposium, IEEE* 206–2012, 2006.
10. Sz. Stremler: *Neural Networks and Their Applications.* Thesis, University of Miskolc, 2015.
11. P.-N. Tan, M. Steinbach, V. Kumar: *Introduction to Data Mining.* Addison-Wesley Longman Publishing Co., 2005.