STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 1, 2010

HANDWRITTEN DIGITS RECOGNITION USING NEURAL COMPUTING

CĂLIN ENĂCHESCU AND CRISTIAN-DUMITRU MIRON

Abstract. In this paper we present a method for the recognition of handwritten digits and a practical implementation of this method for real-time recognition. A theoretical framework for the neural networks used to classify the handwritten digits is also presented.

The classification task is performed using a Convolutional Neural Network (***CNN***). ***CNN*** is a special type of multi-layer neural network, being trained with an optimized version of the back-propagation learning algorithm. ***CNN*** is designed to recognize visual patterns directly from pixel images with minimal preprocessing, being capable to recognize patterns with extreme variability (such as handwritten characters), and with robustness to distortions and simple geometric transformations.

The main contributions of this paper are related to the original methods for increasing the efficiency of the learning algorithm by preprocessing the images before the learning process and a method for increasing the precision and performance for real-time applications, by removing the non useful information from the background.

By combining these strategies we have obtained an accuracy of ***96.76%***, using as training set the ***NIST*** (National Institute of Standards and Technology) database

Received by the editors: December 15, 2010.

2010 *Mathematics Subject Classification.* 68T05, 68T10, 62M45.

1998 *CR Categories and Descriptors.* I.7.5 [**Computing Methodologies**]: Document and text processing – *Optical character recognition (OCR)*; I.5.1 [**Computing Methodologies**]: Pattern Recognition – *Models – Neural Nets*.

Key words and phrases. Convolutional neural networks, supervised learning, handwritten digits recognition.

This paper has been presented at the International Conference Interdisciplinarity in Engineering (INTER-ENG 2009), Târgu-Mureş, Romania, November 12–13, 2009.

1. Introduction

The ***NIST*** database used to train the neural network [4] contains 60,000 images with distorted handwritten digits, 10,000 images being used for testing and considered as a reference benchmark for applications for handwritten digits recognition.

In order to demonstrate the efficiency of our proposed neural network we will compare the learning performances with a classical ***MLP*** (Multy Layer Perceptron) [5] neural network.

In order to measure the performances of our proposed neural network we will use two different types of neural networks architectures: the first one, a classical ***MLP*** neural network and the second an original ***CNN***, proposed in this paper.

2. The Convolutional Neural Network - CNN

CNN is a feed-forward neural network capable to extract topological properties from an image. It extracts features from the input raw image using the first hidden layer and classifies the patterns with the help of the final hidden layer.

The first two layers of the neural network can be considered as a trainable feature extractor [2]. We can add a trainable classifier to the feature extractor, considering 2 fully connected layers (a universal classifier) [2].

The number of hidden neurons contained in the hidden layers is variable and by varying this number we can control the learning capacity and the generalization capacity of the overall classifier [5].

The architecture of the ***CNN*** used to learn the ***NIST*** database is depicted in Figure 1.

The general processing strategy for a ***CNN*** is to extract simple features at a higher resolution, and then to convert them into complex features at a coarser resolution. Each convolutional hidden layer can reduce the resolution of the initial image by a factor of $(n-3)/2$. For this purpose, each neuron contained in a convolutional hidden layer will process a single region from the map of the previous layer. In this way, a neuron is functioning like an independent micro kernel, named ***convolutional kernel*** [2]. The convolutional kernel contains as inputs the corresponding region from the previous map to be processed.

The width of the ***convolutional kernel*** is chosen to be centered on a unit-pixel (odd size), to have sufficient overlap for not losing information (3 units

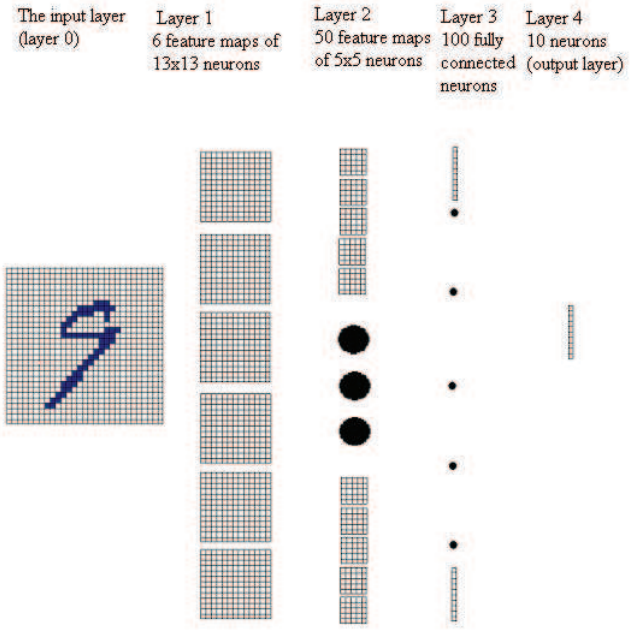

FIGURE 1. The architecture of the ***CNN***

are too small with only one unit overlap, 7 units represents a 70% overlap). A ***convolution kernel*** of size 5 is shown in Figure 2.

With no padding, a sub-sampling [2] of 2, and a kernel size of 5, each ***convolution layer*** reduces the feature size from n to $(n-3)/2$. Since the initial ***NIST*** database contains images of size 28×28, the nearest value which generates an integer size after 2 layers of convolution is 29×29.

The ***CNN*** architecture is composed of 4 layers. The first layer is a convolutional layer composed of 6 feature maps of 13×13 units. Each neuron processes only a region of 5×5 pixels from the input image, ignoring the rest of the information from the picture. The second layer is also a convolution layer that is processing a region of 5×5 outputs from each feature map of the previous layer, the other outputs being ignored. The third layer is composed of 100 neurons fully connected with the neurons from the second layer. The output layer is composed of 10 neurons, one neuron for each pattern that has to be classified.

A trainable weight is assigned to each connection, but all units of one feature map share the same weights. This characteristic is justified by the fact

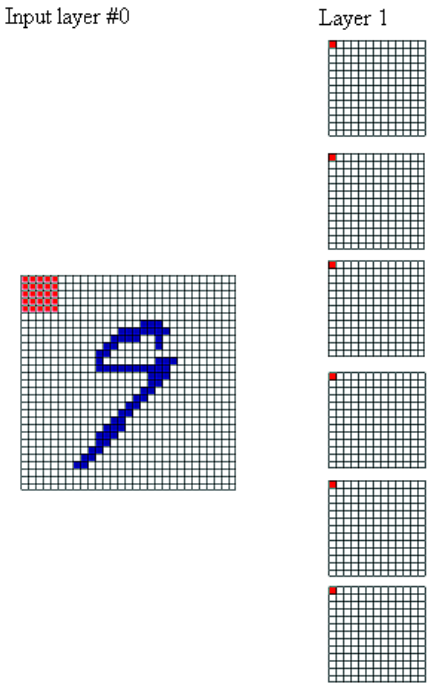

FIGURE 2. The inputs of the first neuron from each feature map in Layer1

that an elementary feature detector useful on one part of the image is unlikely to be useful for the entire image. Moreover, the weight sharing technique allows the reduction of the numbers of trainable parameters. For instance, ***CNN*** has only 132,750 trainable parameters out of 303,450 connections.

2.1. **Cutting out useless information.** Although ***CNN*** are known to be able to extract information from images with minimum preprocessing, the removal of the un-useful pixels from the background improves performances, not only during the learning process, but more important, during the use of the ***CNN*** in practice. One of the reasons is that we have the ***NIST*** training database that contains 60,000 images of handwritten digits but, all of them are centered and have the same width. In practice we will have to extract out digits from bigger images and resize them to 29×29 pixels, so we can present them as inputs to the neural network. Trying to cut out a digit from an image, resizing and centering the image, we can obtain the same results as

with the digits contained in the ***NIST*** database. Another advantage of this strategy is obtained when an image of the digit is zoomed in and the relevant patterns are accentuated. In Figure 3 we have on the left a sample image from the ***NIST*** database and on the right the same image with the background removed.

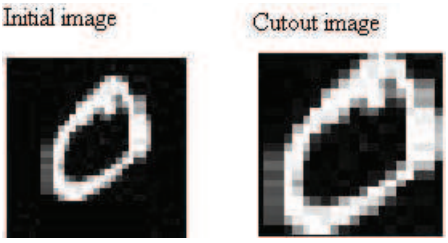

FIGURE 3. Background removal

2.2. **Pre-encoding of the training images.** The encoding of the images used in the learning process is a crucial improvement of the learning process because it dramatically decreases the time needed to pass one learning epoch [5]. The images are codified using the following rule: the useful information, the white pixels, are set to 1 and the background pixels are set to 0.

After this encoding process we will store the encoding images in a special training file. Before starting the learning process we will load all this samples from the training file into the internal memory of the computer. It is very important to run the learning process with the deactivated virtual memory option of the operation system - this will eliminate the need to use the external memory which is slow compared with the internal memory. An example of encoding is presented in Figure 4.

3. The learning process

The learning process was based on the ***NIST*** training database that contains 60,000 images of handwritten digits. Because the training set is huge, the duration of a single learning epoch took almost 73 minutes on a computer with Intel Dual Core processor - 1.86 GHz and with 3 GB of RAM. This long time span of a learning epoch was given by the fact that all of the images had to be loaded from the hard disk, a slow external memory device, compared with the internal memory. To reduce the duration of a learning epoch we

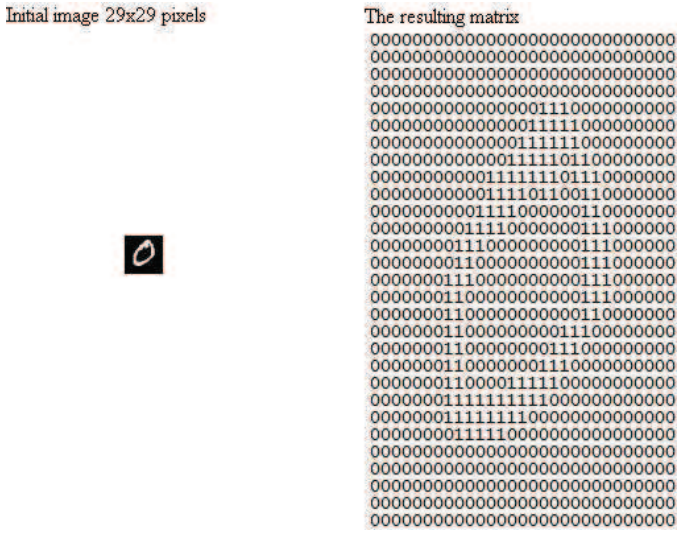

FIGURE 4. Encoding example for digit 0

decided to upload all the images from the ***NIST*** learning database into the internal memory (RAM).

By pre-encoding the learning images and by loading them directly into the RAM at the start of the learning process, we have reduced the time span of a learning epoch to approximately 52 minutes, which means an improvement of **40%** in performance. An important observation is to deactivate the virtual memory option before the start of the learning process, in order to be sure that the training dataset is not stored on the HDD, in the swap file of the operating system.

The initial values of the ***synaptic weights*** have an important effect on the learning capabilities of the ***CNN***. Through experiments, we have found that the best results are obtained with initial random synaptic weights taking values in the interval [-0.005, 0.005].

The ***learning rate*** has also an important effect on the learning process. We have started the learning process with a learning rate of 0.01, afterwards the learning rate being decreased with 10% at every 3 epochs.

We have used as ***activation function*** the ***hyperbolic tangent*** [5], instead of the ***sigmoidal function*** [5], because it has the advantage of being symmetrical related to the origin, and has a greater range of values in interval [-1,1] instead of interval [0,1]. A good practice is to limit the range of values

for the hyperbolic tangent activation function, we have trained our ***CNN*** and calculated the error of the output neurons using the interval [-0.8, 0.8] instead of interval (-1. 1).

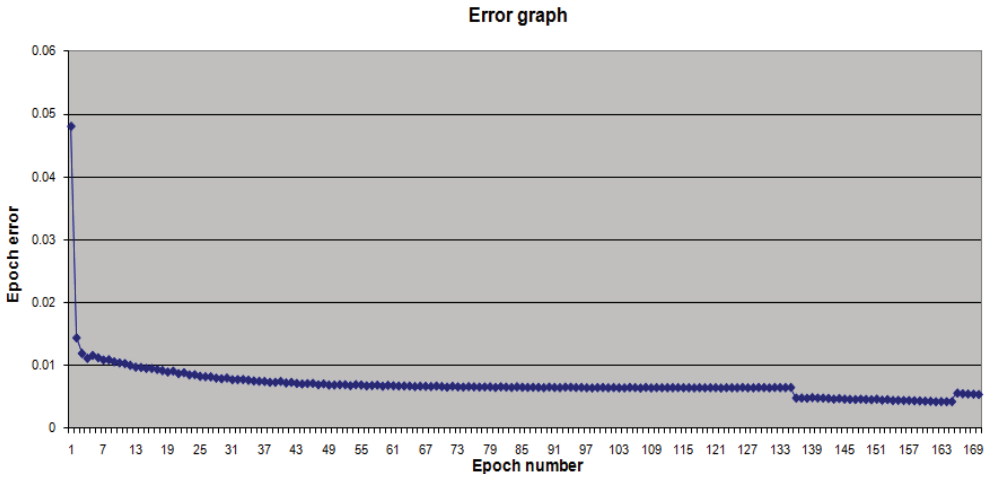

FIGURE 5. ***CNN*** learning error based on the ***NIST*** training images

In practical applications, when we try to extract digits from an image it is very difficult to find the exact algorithm for centering and resizing the digits to 29×29 resolution, similar to the ***NIST*** training set. This is why we trained the ***CNN*** using the initial ***NIST*** training images and a new training set that we have obtained after the removal of the non-useful information from the training images, using the technique presented before.

4. The testing process

After stopping the training process, we have tested the generalization capabilities of the neural network using **10,000** images (the initial ***NIST*** testing database and the testing images obtained from cutting out the useless information from the ***NIST*** testing images). To obtain better performance and to save time we have used in the testing process the pre-encoding method and the loading procedure of the testing images in the internal memory. The ***NIST*** testing database is considered to be a benchmark and a very difficult testing set for handwritten digits recognition applications.

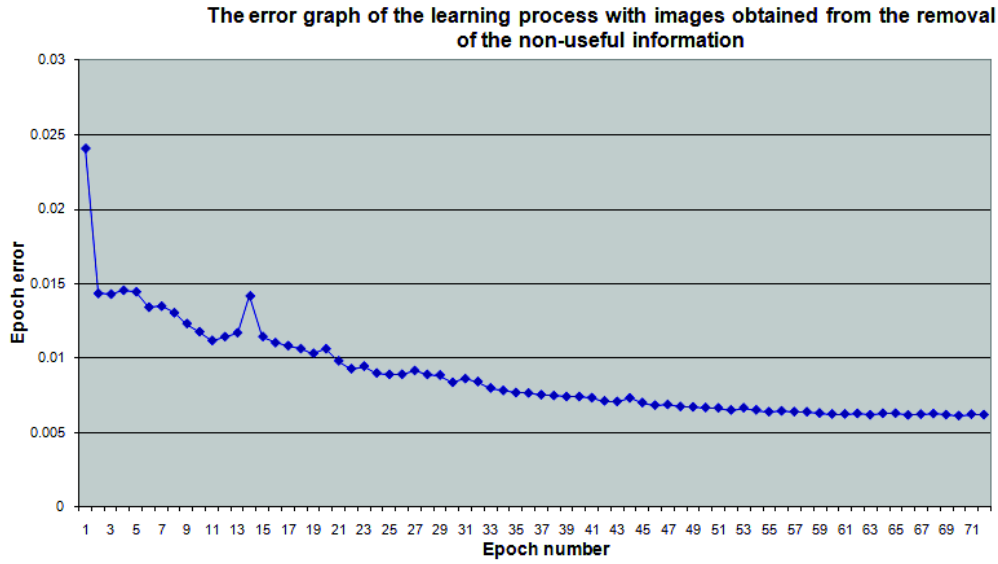

FIGURE 6. ***CNN*** learning error based on the ***NIST*** training images with the useless information removed

Our ***CNN*** obtained the following performances: with the original ***NIST*** testing images **96.74%** accuracy and **96.56%** accuracy with the images without background.

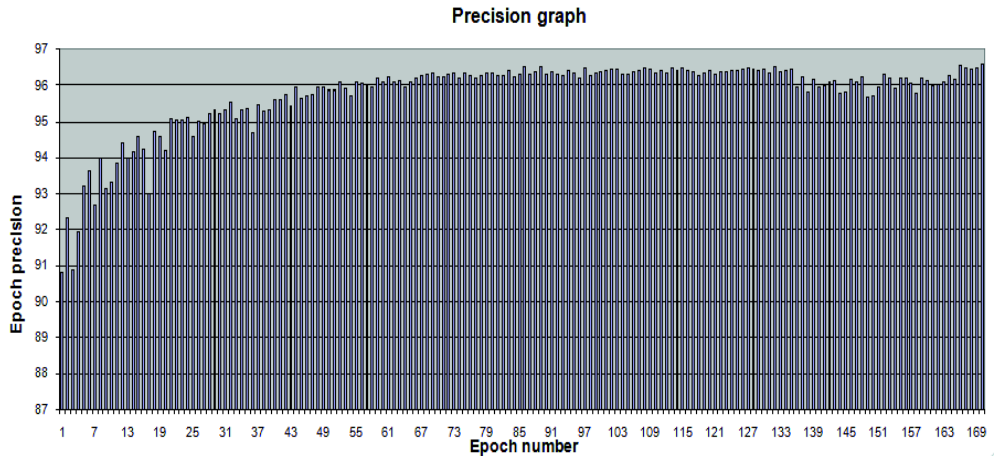

FIGURE 7. ***CNN*** accuracy based on the ***NIST*** testing images

Table 1 contains the best results obtained so far in the world, they range from an error of 12% up to an error of 0.4%, using the ***NIST*** testing database with 10,000 images.

TABLE 1. Results of the learning process for different neural networks

Method	Error	References
Linear classifier	12%	[1]
Linear classifier (nearest neighbor- NN)	8.4%	[1]
Pair wise linear classifier	7.6%	[1]
K-NN, Euclidean	5.0%	[1]
2 layer NN 300 hidden units	4.7%	[1]
2 layer NN 1000 hidden units	4.5%	[1]
2 layer NN 1000 hidden units using distortions	3.6%	[1]
1000 RBF + linear classifier	3.6%	[1]
Our CNN	***3.26%***	***this paper***
LeNet-5	0,8%	[3]
Convolutional NN elastic dis-tortions	0.4%	[2]

5. Conclusions

This article presents a method for analysis of visual documents and can be considered a starting point for the problem of handwritten letters recognition and handwriting recognition. In the future we will try to extend this architecture to support the recognition of handwritten letters based on the ***NIST*** database of handwritten letters containing 800,000 images. New methods for training the neural network must be found in order to speed up the learning process and to manage this high dimensional learning set.

References

[1] E. Kussul, T. Baidyk, "Improved method of handwritten digits recognition. UNAM", Centro de Instrumentos, Cd. Universitaria A.P. 70-186 , CP 04510, Mexico D.F, 2002.

[2] P.Y. Simard, D. Steinkraus, J.C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis", Microsoft Research, One Microsoft Way, Redmond WA 98052, 2003.

[3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, v. 86, pp 2278-2324, 1998.

[4] http://www.cs.toronto.edu/ roweis/data.html

[5] C. Enăchescu, "Calculul neuronal", Casa Cărţii de Ştiinţă, ISBN: 978-973-133-460-8, Cluj Napoca, 2009.

"Petru Maior" University of Târgu-Mureş, N. Iorga 1, 540088, Târgu-Mures, Romania

E-mail address: ecalin@upm.ro

E-mail address: miron.cristian@stud.upm.ro