Processing Bank Checks with Genetic Programming and Histograms

Oana Muntean, Mihai Oltean
Faculty of Mathematics and Computer Science
Babeş-Bolyai University, Kogălniceanu 1
Cluj-Napoca, 400084, Romania
oana.muntean85@gmail.com
moltean@cs.ubbcluj.ro

Abstract

In spite of evolution of electronic techniques, a large number of applications continue to rely on the use of paper as the dominant medium. Bank checks are a widely known example. When filled by hand, the processing of the written information requires either a human or a special software which has intelligent abilities. This paper examines the issue of reading the amount of money written on the checks. Genetic Programming (GP) technique is used for dealing with this problem. A new type of input representation is proposed: histograms. Several numerical experiments with GP are performed by using large datasets taken from the MNIST benchmarking set. Preliminary results show a good behavior of the method.

1 Introduction

Nowadays bank checks are documents used all over the world. However a part of them are still processed manually by humans. The most common and important operation is reading of the amount of money written on the check. Automation of check processing is an important application of document analysis techniques [2]. Bank check is a complex document with many information fields. Usually check amount is indicated twice: as a numeral expressed in digits (courtesy amount), and as a phrase expressed in words (legal amount). Check processing system should be able to correctly read the numerical value representing the sum to be paid to the owner of the check.

Handwriting recognition concerns the conversion of the analog signal of handwriting into a digital symbolic representation. The analog signal that we are interested in is in the form of a two-dimensional scanned image of paper.

It is widely accepted that a perfect recognition of digits requires intelligence. Specifically, Artificial Intelligence techniques have been intensively used for solving this problem. Among them, Artificial Neural Networks are the most popular.

In this paper we use GP for handwritten digit recognition. The novelty consists in the representation of the input. Because the matrices, encoding the images involved in the numerical experiments, are too big to be used as direct input for GP we have to extract and use only some partial information. This is why we have constructed the horizontal and vertical histograms [1] and this information was sent as input for GP. According to our knowledge this type of representation was not used before in conjunction with GP.

We have performed several numerical experiments by using a well-known benchmarking data set: MNIST [4]. Results have shown a more than 91% classification accuracy for all the cases. For some tests the accuracy was more than 97%.

The paper is organized as follows: the handwritten digit recognition problem is defined in Section 2. The proposed approach is deeply presented in Section 3. Test problems are given in Section 4. The results of numerical experiments are given in Section 5. Strengths and limitations of the proposed technique are discussed in Section 6. Conclusions and future work directions are given in Section 7.

2 Handwritten recognition problem

The recognition of handwritten characters by a machine is a very difficult task. This work involves identifying a correspondence between the pixels of the image representing the sample to be analyzed (recognized) and the abstract definition of that character. In this paper we focus our attention on the problem of digit recognition.

The main drawback of this problem is the extreme variability of handwritten that produces a large number of features for different writing style. That is why there is no single method for solving handwritten recognition.

Artificial Neural Networks are very popular techniques for solving this problem. Although, GP was suggested [6] as an alternative method for attacking this problem. However, the results presented so far are not very encouraging: huge populations and extensive running times have been involved.

3 Proposed approach

The proposed approach uses standard GP [3] as the main search mechanism.

What is different in our paper is the way in which the input is represented. GP ability to solve problems is highly related to the number of terminals [3]. If we have too many terminals it is more difficult for GP to select the good ones.

The original data sets consist in images encoding digits represented as 20x20 matrices. If all pixels of this matrix are sent as input to GP we would have 400 terminals. This is a huge number for GP.

This is why we have tried to reduce the number of inputs. We can do that by extracting some information from the original 20x20 matrices. In this paper we have built the horizontal and vertical histograms and this information was actually sent as input to GP. This information is further processed by the GP classifier.

Histograms representation is very simple [1]. For each row and each column of the image we count the pixels containing ink (see Figure 1). The obtained number can give us some rough information about what digit we have there. Histograms have been used in the past for handwritten recognition [1]. However, this is the first time when they are used in conjunction with Genetic Programming.

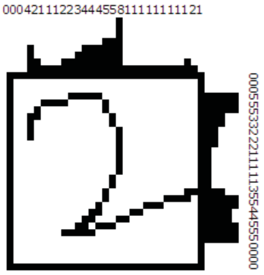

Figure 1. Histogram for digit 2. For each row and column we have counted the ink pixels.

We have used our method as a binary classifier. We want to find out if the method can distinguish between one digit and the rest of the digits. For instance we are interested to find a computer program (mathematical formula) which can distinguish between 0 and the digits from 1 to 9.

In this case we deal with a classification problem with 2 classes. The fitness is computed as the number of incorrectly classified examples over the total number of cases. Thus, the fitness has to be minimized.

For increasing the generalization ability of the method we have divided the training set in 2 parts: a subset for the training purposes only and a validation set. The search process is guided by the training set. When a new individual is generated it is tested against the validation set. The individual with the best error for validation is applied (at the end of the search process) to the test set. Using the validation set is an effective method for avoiding overfitting.

4 Test data

The MNIST is a database [4] of handwritten digits having a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized (to a 20x20 matrix) and centered in a 28x28 fixed-size image. In our case we have extracted only the information from the central 20x20 matrix. By building the histograms we have obtained a dataset with 60000 examples, each of them having 40 attributes.

5 Numerical experiments

We have extracted from MNIST 10 different files containing a dataset for each digit to be classified. In each dataset 30% from the instances represents a digit and the other 70% represents the other digits. Each dataset contains 10000 cases. The validation set has 2000 cases and the digits are represented uniformly (about 10% each). The distribution is the same for test set.

Secondly we have built the histograms and then we have applied GP for all of them.

In the numerical experiments performed in this paper the following parameters for GP are used: a population of 500 individuals evolved for 100 generations, binary tournament selection, point-mutation with 0.1 probability and subtree-swap crossover with 0.9 probability. The function set is $F = \{+, -, *, /\}$. The set of terminal consist of input data for each digit classification (in this case 40 values).

5.1 Results

The classification accuracy obtained by running GP method for all test problems are given. In all runs we have obtained a very good classification error. 30 runs have been performed and the best errors are provided in Table 1.

Table 1. Results obtained by applying GP to the considered test problems

Digit	Error(%)	Digit	Error(%)
0	3.99	5	8.72
1	4.27	6	2.44
2	5.22	7	3.57
3	7.00	8	6.51
4	2.29	9	4.91

Taking into account the values presented we can observe that the best error is obtained for classifying digit 4. This means that is very easy to make distinction between this digit and the rest of digits. The worst result is obtained for classification of 5 and, then, for classification of 3. This means that is very difficult to make distinction between 5 and the rest of the digits.

Note that the results can be further improved by using a larger population or running the search process for more generations.

5.2 Comparison with other GP techniques

In [6] GP with a special representation was used for digit classification. The reported error was between 2% and 5%. In [5] the error was between 3% and 9%. Note that a perfect comparison cannot be made because the representations and the parameter settings are too different.

6 Strengths and weaknesses

Genetic Programming strengths and weaknesses are already known to the research community. This is why we focus our attention to the advantages and disadvantages introduced by the representation with histograms.

The greatest benefit is the fact that we have been able to successfully apply GP for solving this problem. This is due to the reduced number of inputs which are sent to the GP individuals. This was not possible if we sent the entire matrix to GP because the method cannot successfully handle such large amount of inputs.

The limitations of the proposed approach are mainly related to the limitations introduced by the histogram representation. There are several digits which are difficult to be distinguished when using this kind of representation. For instance, in the case of 5 and 3 it is quite difficult to find a very good classification. Luckily, our test data contains digits written by hand. In this case there is a more clear distinction between the digits previously enumerated since the handwritten characters have a huge number of possible representations. This fact has been shown by the results from Table 1 where the classification errors are quite good.

Another weakness is that by using histograms we have reduced the amount of information which was sent to GP classifier. This could lead to some poor results in some cases.

7 Conclusions and further work

In this paper a new way of representing the input for solving handwritten recognition problems using GP has been suggested. The proposed representation was tested on a well-known benchmarking data set. The results of the numerical experiments have shown very good classification accuracy. Further efforts will be focused on introducing more functions (such as sin, exp, lg) in GP algorithm and on testing the method for other datasets (including letters).

References

[1] H. C. Fu. User adaptive handwriting recognition by self-growing probabilistic decision-based neural networks. *IEEE-NN*, 11(6):1373–1384, 2000.

[2] N. Gorski, V. Anisimov, E. Augustin, O. Baret, D. Price, and J. Simon. A2iA check reader: A family of bank check recognition systems. In *ICDAR*, pages 523–526, 1999.

[3] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[5] A. Lemieux, C. Gagne, and M. Parizeau. Genetical engineering of handwriting representations. In *Frontiers in Handwriting Recognition*, pages 145–150, 2002.

[6] A. Teredesai, J. Park, and V. Govindaraju. Active handwritten character recognition using genetic programming. In J. F. M. (et al), editor, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 371–379. Springer-Verlag, 2001.