Available online at www.sciencedirect.com

SciVerse ScienceDirect

IERI Procedia 2 (2012) 367 – 371

Procedia IERI

www.elsevier.com/locate/procedia

2012 International Conference on Future Computer Supported Education

Hand Written Digits Recognition Using Digital Learning Networks

Mazin Al Hadidi,Rami Salim Rzouq

Computer Engineering Department, Al Balqa' Applied University,Al-Salt Jordan

Trueamman@yahoo.com

Abstract

A pattern recognition system based on the n-tuple technique is developed and evaluated for use in classifying non-deterministic data with particular reference to unconstrained hand-written numerals.

The system presented in this research fulfils the requirements of simplicity and efficiency making it attractive to practical use in present day industrial environments. This simplicity of operation is afforded by the self evolving nature of the classifier since it is based or a training phase where the recognition logic is developed.

Results are reported on the task of recognizing handwritten digits without any advanced pre-processing. The results are obtained using a RAM-based neural network, making use of small receptive fields. Furthermore, a technique that introduces weights into the RAM net is reported.

Keywords: Pattern recognition; Hand Written; Digital Learning Network (DLN)

* Corresponding author. Tel.: (+962)777131163;.
E-mail address: mhadidi@bau.edu.jo

2212-6678 © 2012 Published by Elsevier B.V. Selection and peer review under responsibility of Information Engineering Research Institute
Open access under CC BY-NC-ND license. doi:10.1016/j.ieri.2012.06.103

1. Introduction

Deferent architectures for off-line recognizing of handwritten digits (and characters) have been proposed in the past [4,3]. This includes traditional pattern recognition schemes as well as schemes based on neural networks. The range of applications is wide and includes postal code reading, automatic data entry, and reading devices for the blind.

The important features when developing the architectures are accuracy, flexibility, and speed. In order to obtain high recognition performance, it is often beneficial to include some intelligent preprocessing before the data are fed to a neural net. On the other hand, it can be difficult to design feature extractors that do not miss important features. It is often so that feature extraction schemes make it easy to recognize 80-90% of the examples, but at the same time, discard information so that it becomes almost impossible to classify the last percentages correctly. However, if the raw data representation is kept together with the extracted features, the original information is of course still available.

In this case the input dimension increases, which might also cause problems. Even though it can be difficult to find suitable pre-processors, there is often no alternative as the number of training examples needed without pre- processing is often too large. In the case of recognizing handwritten digits, it is however easy to get access to training examples. We have therefore tried to investigate what can actually be accomplished with a neural net when no essential pre-processing is being performed. A neural network architecture where the training time scales linearly with the number of training examples is the so-called RAM nets (or n_tuple method). This kind of architecture does not belong to the most widely used types. However, a comparison between RAM nets and other classification architectures (including neural nets) was recently performed for several classification problems [2]. From the obtained results it was concluded that the underlying principle of n-tuple method was a powerful method. Below we report on the results obtained on the task of recognizing handwritten digits using RAM architecture. We have extended the traditional RAM architecture to include inhibition. This new feature increases the recognition performance significantly. The results obtained clearly justifies RAM net as an alternative to the more popular neural network architectures.

2. RAM Neural Nets

RAM-based neural nets belong to the memory-based architectures. These architectures have some appealing features such as one shot learning and fast recall times [5, 4]. A RAM-based neural net can be considered as consisting of a range of Look up Tables (LUTs) that store the weights of the architecture.

Each LUT samples a number of data from the input space. The rows of the tables correspond to different object classes whereas the columns correspond to different patterns of the sampled input data. In the traditional scheme, only the values 0 or 1 are used as weight values. A value of 1 corresponds to a specific feature being present in the training set for a specific class. The output from a specific LUT corresponds to the contents of the column addressed by the given input. The output vectors from all LUTs are added, and subsequently a winner-take-all decision is made to perform the classification of the input example as illustrated in Figure 1.

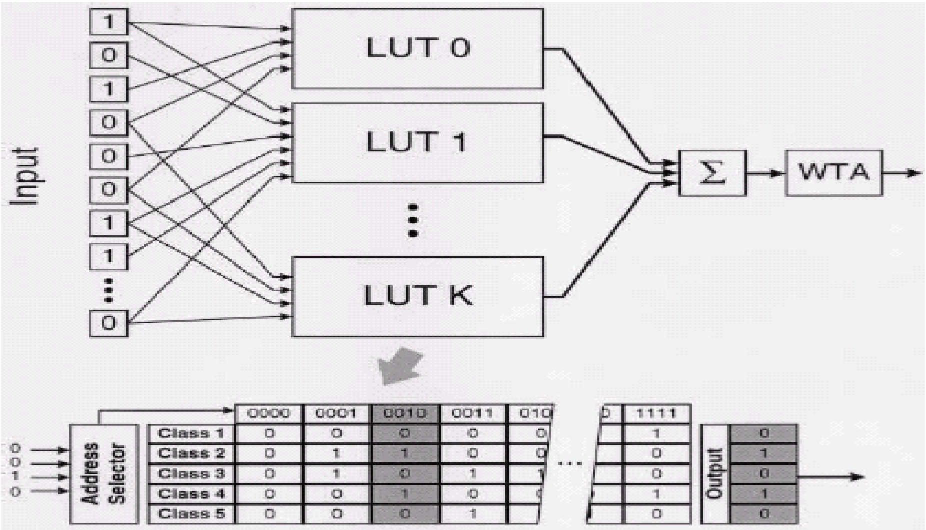

Fig. 1. The RAM Net architecture

The above description can also be put into a mathematical description as in Ref. 2. Let NLUTs be the total number of LUTs in the system, and let Sc describe the set of training examples in class C. Furthermore, let x- denote an example used for training and let y- denote an example outside the training set. A given example will address a specific column of each LUT. Let the column in the ith LUT being addressed by an example y- be denoted ai(y).

When training a RAM-based net it is important to select a proper number of input connections for each LUT. The generalization capabilities of the network are directly related to the number of input bits for each LUT. If a LUT samples all input bits, it will act as a pure memory device and no generalization will be provided. As the number of sampled input bits is reduced, the generalization is increased at an expense of a decreasing number of unambiguous decisions. When the number of training examples increases, it generally becomes more difficult to find features of low dimension that can distinguish between training examples of different classes. Accordingly the number of input connections per LUT will in general have to increase with the number of input examples.

In the traditional scheme the input connections for the different LUTs are chosen at random [2, 1,6]. The performance of the system generally increases with the number of LUTs but for a given size the performance increase obtained by adding extra LUTs becomes very low. As the recall time of the system increases linearly with the number of LUTs and due to memory considerations, it is desirable to minimize the number of LUTs needed to achieve a given performance. Much can be gained if the input connections are selected in a more intelligent way. Such a test is easy to perform on a RAM-based net if one (instead of just writing 0 or 1 in the LUT cells) simply stores the number of examples that visit every cell .Using the information measure it is possible to reduce the number of LUTs needed to achieve a given performance. The information measure is based on entropy and a concept defining a so-called critical example number. The number of critical examples is defined as the number of examples that at least have to be redrawn from the training data to imply a misclassification of the training example in question.

3. The Proposed Technique (Inhibition)

The number of LUT addresses (columns) shared by any two examples belonging to deferent classes should be as small as possible. However, in many situations two deferent classes might only differ in a few of their features. In such a case, an example outside the training set has a high risk of sharing most of its features with an incorrect class. In this situation the RAM net will have an unacceptable high error rate. In order to deal with this problem it becomes necessary to weight deferent features differently for a given class. These weights can then be trained with a perceptron (n-tuple classifier) like learning rule [2].

We present an alternative solution where the contents of the LUTs bear a much closer resemblance to the conventional RAM architecture. Accordingly, it also preserves some essential advantages of the simple RAM-net. In order to illustrate the need for inhibition, a specific example from the task of recognizing hand- written digits is depicted in Figure 2. As illustrated in this figure, the main deference between the considered types of 4's and 9's is the appearance or non-appearance of an upper horizontal bar. If a top bar is met in a test example, it is desirable that the network inhibits class 4. Otherwise there is a risk that too much emphasis is put on the other parts of the test pattern. Figure 2. An example of two digit classes that are very close with exception of the bar or non-bar structure at the top. class 9.

Fig. 2. An example of two digit classes

Due to the limited training set, it is however not necessarily so that columns voting on class 9 but not on class 4 (and vice versa) represent proper distinguishing features between the two classes. Accordingly, a strategy is needed for selecting those columns that are the most likely candidates for inhibition.

The first step in locating column candidates for inhibition is to detect the training examples having low confidence and those being misclassified in a cross-validating test (a simple confidence measure is the vote deference between the winning class and succeeding class). For each of these examples the competing class is registered. All LUT columns that produce votes on the true class but no votes on the competing class are then registered. A small inhibition factor is now stored in the competing class cells of these LUT columns:

Inhibition factor = α inhabit

Weight $\rightarrow$ weight – α inhabit

The inhibition factor is calculated so that the confidence after inhibition corresponds to a desired level. Inevitably this technique will add inhibition to some LUT columns that do not represent important distinguishing features. But the LUT columns representing the distinguishing patterns are likely to be visited by many of the low confidence training examples and accordingly their corresponding cells will obtain larger inhibition factors than the rest. It follows from the above discussion that the inhibit factor should be sufficiently small so that the effect of inhibiting cells once is negligible .The value we use depends on the amount of confidence increase that is needed for a specific example. A typical value of inhibit is 10/NLUTs, where NLUTs is the total number of LUTs.

4. Discussion

The primary objective of the work reported in this research was to investigate a new technique so as to improve the recognition performance of non-deterministic data using RAM-networks which implement n-tuple method of pattern recognition. In this class of data, unconstrained hand –written characters were chosen to optimize the techniques which can be applied to other forms of data.

Inhabitation technique was introduced in order to increase the ability of the RAM net recognizer for feature extraction since the information on the weight of each feature will be available in each class of the recognizer to differentiate between strong features and weak ones. Therefore, weighted RAM net scheme can be utilized to increase the decision confidence between data classes of low separability .

When implementing this technique to the single layer recognizer, it could be found that this technique enhanced the correct recognition to 92.4%, and the rejection was decreased to 7.2%. However, the new technique gave a higher performance and confidence when incorporated with the second layer.

Table 1: Recognition with new technique (inhibition)

	Correct	Reject	error
Modified system with single layer networks	92.4%	7.2%	0.4%
Modified system second layer networks	94.4%	4.8%	0.8%

The inhabitation technique can increase the system performance by emphases on different regions between the patterns (categories) and also emphases on the nature of the feature and it’s affected on pattern. This technique provides a good recognizer for whole range of applications, provide a high confidence in recognition, and it is so easy to implementation in software and hardware.

References

[1] D. Bisset, "Weightless Neural Network ", 1995.
[2] I. Alexander, "An Introduction to Neural Computing ", Chapman and Hall, London,1990.
[3] I. Alexander and T. J. Stonham, "Guide to pattern recognition using random-access memories,” Computers and Digital Techniques, 2, 29-40, 1979.
[4] R. A. Wilkinson, "The first census optical character recognition system conference," Technical Report NISTIR 4912, Nat, Inst. Of Standards and Technol., Gaithersburg, MD, 1992.
[5] R. Rohwer and M. Morciniec, "A theoretical and experimental account of n-Tuple classifier performance,” Neural Comp. 8, 629-642, 1996.
[6] S. Lee, "Multilayer cluster neural network for totally unconstrained handwritten numeral recognition," Neural Networks 8, 783-792, 1995.