UDK 621.3:(53+54+621+66), ISSN0352-9045 Informacije MIDEM 23(1993)2, Ljubljana

LOGIC NEURAL NETWORKS FOR A CHARACTER RECOGNITION PROBLEM

J. Ficzko, A. Dobnikar

KEYWORDS: logic neural network, pyramidal architecture, optimal architecture, a-algorithm, GHA Generalizeg Hebbian Algorithm, pattern recognition, character recognition, Alexander pyramida, system comparison, comparison of results

ABSTRACT: The paper presents a comparison between the recognition performance of logic neural networks for different pyramidal architectures and different basic elements of pyramids. The capabilities of Aleksander's pyramids /1/ do not reach those of the architecture based on the Generalized Hebbian Algorithm. The later depends on statistical features of patterns .

Logične nevronske mreže za problem razpoznavanja znakov

KLJUČNE BESEDE: mreže nevronske logične, arhitektura piramidna, arhitektura optimalna, a-algoritem, GHA Hebbian algoritem posplošeni, razpoznavanje vzorcev, razpoznavanje znakov, Alexander piramida, primerjava sistemov, primerjava rezultatov

POVZETEK: V članku je podana primerjava rezultatov razpoznavanja znakov z različnimi logičnimi nevronskimi mrežami za različne piramidne arhitekture in različne gradnike piramid. Osnovne Aleksandrove /1/ piramide po svojih zmožnostih ne dosegajo optimalne piramidne arhitekture, ki je določena na osnovi posplošenega Hebbovega algoritma. Ta arhitektura se prilagaja statističnim značilnostim vzorcev, zato so rezultati razpoznavanja po tej poti bistveno boljši.

1. Introduction

In pattern recognition problems the dimensionality of input data is often of order 100 or more. When using neural networks, we face with the problem of having large number of inputs per neuron in the first layer. This practicaly disable the hardware realization of neural network. Therefore we want to find the architecture with a limited connectivity between the input and the first layer of the neural network. We were looking for such architecture in two ways. First we tried to use the pyramidal architecture introduced by Aleksander /1/. It is composed of logic nodes (PLN or G-RAM) with reduced connectivity towards input layer that can be easily realized by RAM elements and some additional logic. Secondly we were looking for the architecture that depends on input data. As the Generalized Hebbian Learning Algorithm performs feature extraction by limiting the weight vectors of the 1st layer neurons toward the eigen vectors of the receiptive input field of the pattern we used this type of a layer as the second alternative.

The paper is organized in three parts. In the first part the data set for a character recognition problem is introduced.

In the second part the comparative results (number of necessary learning cycles, recognition rate) of Aleksander's pyramids - for a different architectures, different basic elements and consequently different learning algorithms are presented.

The pyramidal architecture based on GHA is introduced in the third part together with the testing results for a character recognition problem.

Conclusion finally gives a comment and a suggestion for further work.

2. The data set for a character recognition problem

The data set for a character recognition problem is composed of characters obtained from the scanner. Characters are taken from OCR font (digits from 0 to 9 and delimiters "S" and "h"). The data consists of 15 samples for each class, so the total is 180 characters. For learning purposes 10 samples of each class were used, so 5 samples of each class were available for testing. This data set was used with pyramidal architecture based on GHA.

Observing the recognition rate of Aleksander's pyramids, the ideal representatives for each class were chosen as training set. The testing samples were obtained by adding some amount of noise to the training set. The testing set T1 consists of samples that are in 1 bit different from learning samples, T2 consists of samples that are in 2 bits different from learning samples and T3

of samples that are in 3 bits different from learning samples. Such a learning set was used because of inability, in sense of capacity, of Aleksander's pyramids to learn the same data set as the pyramidal architecture based on GHA.

In both cases, the samples are 64-dimensional and represent a character on 8 x 8 pixel grid. The GHA pyramidal architecture is used also for 1024-dimension cases that represent a character on a 32 x 32 pixel grid.

3. Pyramidal architectures of logic neurons

Concerning the number of inputs to the first layer of Aleksander's pyramids (n), the number of inputs to one element (N), the depth of the pyramid (D) and relation $n = N^D$, where n = 64, with varying N=2,4,8,16,32,64 (different element at different layers) there are 32 different pyramidal architectures. Out of 32 possibilities we decided for 7 architectures that were used for testing:

- P(64;2,2,2,2,2,2)
- P(64;2,2,4,4)
- P(64;4,4,2,2)
- P(64;8,4,2)
- P(64;2,4,8)
- P(64;4,4,4)
- P(64;8,8)

The number c on the k-th place following ";" means the pyramid with c- input elements in the k-th layer. The selection of 7 pyramids includes all pyramids that have equal elements in all layers, as well as pyramids with different number of layers, where we set the elements with the greatest number of inputs in the first or the last layer.

We used the A learning algorithm (introduced by Aleksander /2/) with PLN elements with 3 possible contents in the first case and the A algorithm with additional spreading phase /3/ with G-RAM elements with 3 possible contents in the second case, where different numbers of spreading cycles were performed.

The comparison between different architectures and different learning algorithms was carried out on the basis of learning speed (number of learning cycles that are necessary to respond to the learning set without an error) and capability of generalizing to the testing set (the recognition error on the testing set).

The speed of learning of the particular architecture did not depend on learning algorithms.

Figures 1., 2., and 3. show the speed of learning where the pyramid P(64;8,8) is the fastest and P(64;2,2,2,2,2,2) the slowest. We find out that the speed of learning directly depends on the total number of memory locations in the pyramid. The greater the number, the faster the speed of learning. The lowest pyramid P(64;8,8) has 2304 memory locations whereas the highest - P(64;2,2,2,2,2,2) 252 only.

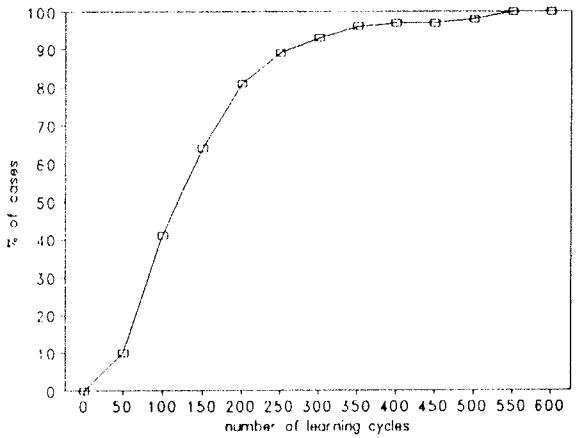

Fig. 1: The speed of learning of P(64;2,2,2,2,2,2)

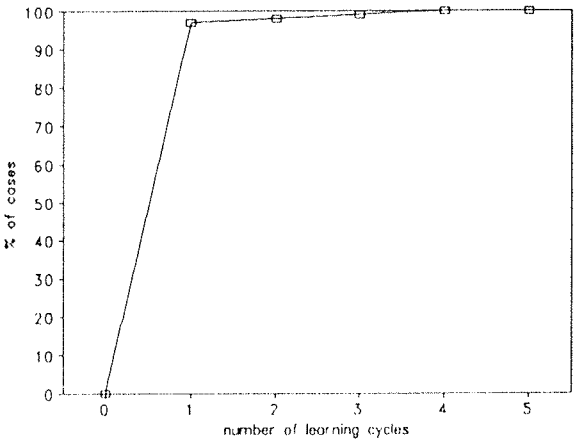

Fig. 2: The speed of learning of P(64;8,8)

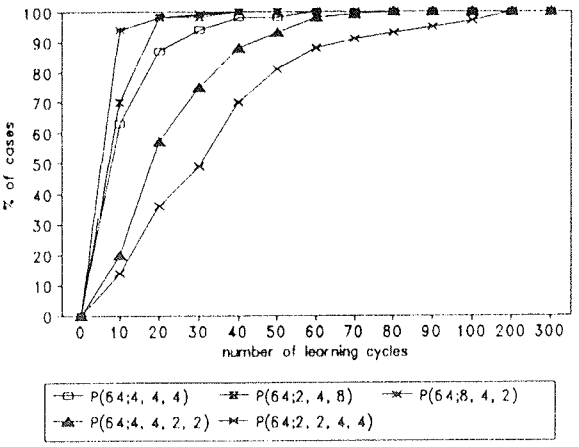

Fig. 3: The speed of learning of P(64;4,4,4), P(64;2,4,8), P(64;8,4,2), P(64;4,4,2,2) P(64;2,2,4,4)

Considering a capability of generalization, it is important to mention when does the recognition error occur. When the input signals address a location with undefined content, the output will be for 50% of time wrong. Therefore, after learning, the number of memory locations that have undefined contents should be as small as possible. The spreading phase carried out after the learning phase reduces the number of memory locations with undefined content. The classification error is therefore in the case of A algorithm with a spreading phase usually smaller than without a spreading phase.

The results shown in Figure 4. for T1 indicate that the highest pyramid P(64;2,2,2,2,2,2) generalizes to the testing set with the smallest error, then follow P(64;2,2,4,4), P(64;4,4,2,2), P(64;4,4,4), P(64;2,4,8), P(64;8,4,2) and the lowest pyramid P(64;8,8) which gives the worst testing results. If the spreading phase is performed, the order of pyramids does not change, but the classification error of the worst pyramid, as well as of the others, is almost equal to the classification error of the best pyramid (see Figure 5.).

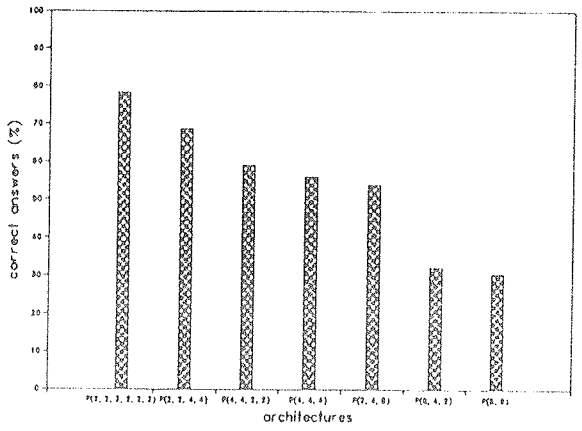

Fig. 4: Correctly recognized characters from testing set T1 with different architectures implemented with 4 pyramids (spreading phase excluded)

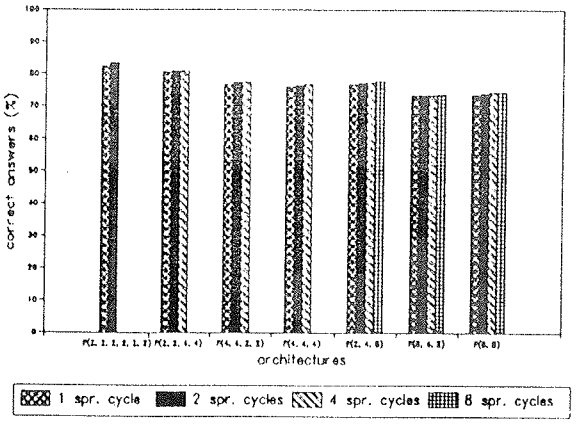

Fig. 5: Correctly recognized characters from testing set T1 with different architectures implemented with 4 pyramids (speading phase included)

Although the classification error is reduced by the spreading phase, the results for Aleksander's pyramids with G-RAMs as well as PLN elements are not acceptable. For a real application of a character recognition problem the achieved classification results are not sufficient.

4. The pyramidal architecture based on GHA for a character recognition problem

The problem of pattern recognition could be divided into two phases:

- feature extraction (or feature selection) and
- classification

One of the best known statistical technic for feature extraction is Karhunen-Loeve Transform /4/, which is the numerical intensive problem. The Karhunen-Loeve Transform could be performed iteratively /5/ with Generalized Hebbian Algorithm that is given by equation:

$$w_{ij}(t+1) = w_{ij}(t) + \gamma(t)\left(\gamma_i(t)\, x_j(t) - y_i(t) \sum_{k \le i} w_{kj}(t)\, y_k(t)\right),$$

where learning rate Y(t) corresponds the conditions:

$$\lim_{t \to \infty} \gamma(t) = 0 \text{ and}$$

$$\sum_{t=0}^{\infty} \gamma(t) = \infty.$$

In the case of a single layer network, an optimally trained layer allows linear reconstruction of the inputs to that layer with a minimal squared error.

A neural network that we used for a character recognition problem based upon a GHA is a single layer locally connected neural network with a linear activation function. The whole input field is divided into smaller input fields (we choose N = 4 bits for each field) and the GHA is then performed on every input field independent of other fields. At the beginning two nodes are mapped onto each input field, but considering the output variances (when they fall bellow a certain threshold) some nodes could be removed.

The results of the GHA are used in logic neurons in that way, that information about the connection strength is stored into RAM locations. To do this, an output of every N-bits binary vector (although it may not be a member of the training set) should be computed from weights and stored into appropriate RAM locations. In this way, the response to the input vector, produced by neural network of conventional neurons, is written to the location that is addressed by the same input vector in logic neural

network. Because of a small input size of basic elements (N=4) this operation does not take a lot of time - computing and storing of outputs for only 16-input combination (of 4 bits each) is necessary.

The decision about class-membership of testing examples is based on two decision rules. The first, proposed by /6/, makes use of reference vectors for each class, while the second /7/ classifies a testing example into the class whose class subspace it has the longest projection in terms of the Euclidian vector norm. Both decision rules lead to the similar results, so only results for first decision rule are presented in this paper.

Figure 6. shows a recognition rate (98.8%) of characters on 8 x 8 pixel grid with 2 nodes mapping to the same input field. Removing nodes with small output variances, without reducing the recognition rate could be done only on one (of 16) input field.

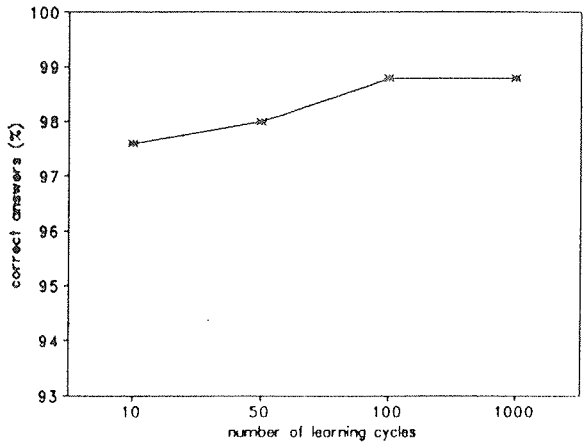

Fig. 6: The recognition rate of characters on a 8 x 8 pixel grid, with 2 nodes per field

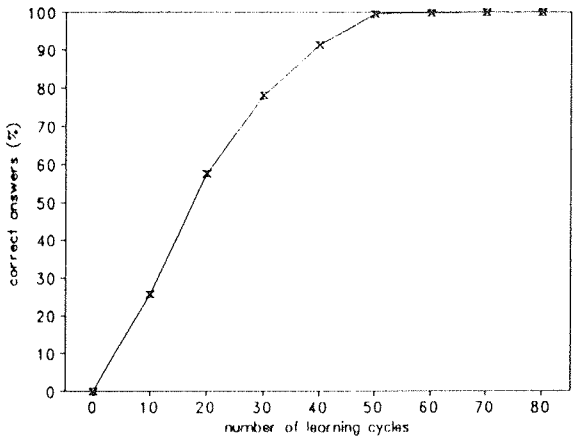

Fig. 7: The recognition rate of characters on a 32 x 32 pixel grid, with 2 nodes per field

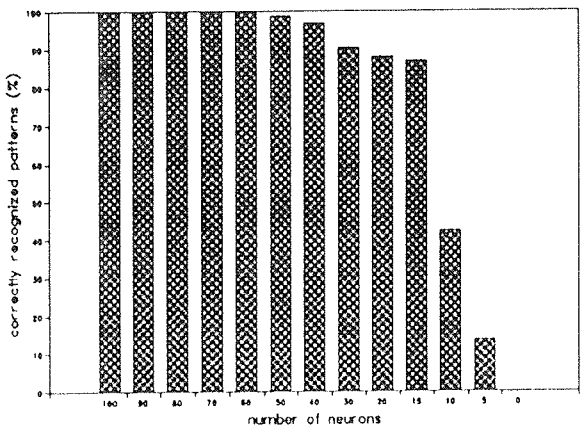

Fig. 8: The recognition rate of characters on a 32 x 32 pixel grid for different number of nodes in the layer

The recognition rate (100%) of characters on 32 x 32 pixel grid with 2 nodes mapping to the same input field is presented in Figure 7. Removing nodes with small output variances is very important for this size of a problem. Some input fields do not provide useful information to distinguish between classes, so they can be removed. Figure 8. shows how the recognition rate depends on the number of nodes covering the whole input field.

5. Conclusion

Neural networks for character recognition problems were studied in the paper. We were looking for the architectures that could be easily implemented in hardware. Two suggestions were outlined in the paper. The first one deals with the so called Aleksander' s pyramids and the second one with the layer based on GHA procedure. We showed that only the second approach succesfully follows the requirements regarding character recognition problem. Its hardware implementation is also obtained easily.

The work presented in the paper opens some new problems also. The one we shall try to deal with first is the problem of classification based on eigen vectors that result in the GHA of learning procedure. For the purpose of this paper we used the standard classification method based on minimal Euclidian distance from reference vectors.

References

/1/ Aleksander I., Canonical neural nets based on logic nodes, 1st IEE Conference on ANNs 1989, London

/2/ Aleksander I., Neural Computing Architectures - The Design of Brain- Like Machines, North Oxford Academic Press, 1989

/3/ Lucy J., Perfect auto-associators using RAM-type nodes, Electronic Let., vol.27, no.10, 1991

/4/ Young Y. T., Calwert W.T., Classification, estimation and pattern recognition, Elsevier, 1974

/5/ Sanger D. Terence, Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network, NNs vol.2,1989

/6/ Penny W., Stonham T., Towards optimal architectures for logic neural nets, ICANN-91

/7/ Oja E., Subspace methods of pattern recognition, Research Studies Press Ltd., Great Britain, 1983

mag. Jelena Ficzko, dipl. ing.
prof. dr. Andrej Dobnikar, dipl. ing.
Faculty of Electrical Engineering and Computer Science
University of Ljubljana
61000 Ljubljana, Slovenia

Prispelo: 21.05.93 ***Sprejeto: 15.06.93***