Evaluating the Conventional and Class-Modular Architectures Feedforward Neural Network for Handwritten Word Recognition

MARCELO N. KAPP[1], CINTHIA O. DE A. FREITAS [1], JÚLIO C. NIEVOLA[1], ROBERT SABOURIN[2]

[1]Pontifícia Universidade Católica do Paraná (PUCPR) – Rua: Imaculada Conceição, 1155 – Prado Velho – 80215-901 – Curitiba – PR – Brazil
{mnk,cinthia, nievola}@ppgia.pucpr.br
[2] École de Technologie Supérieure (ETS) – 1100, Rue Notre Dame – Ouest – H3C 1K3 – Montreal (QC) – Canada
robert.sabourin@.etsmtl.ca

Abstract. This paper evaluates the use of the conventional architecture feedforward MLP (multiple layer perceptron) and class-modular for the handwriting recognition and it also compares the results obtained with previous works in terms of recognition rate. This work presents a feature set in full detail to work with handwriting recognition. The experiments showed that the class-modular architecture is better than conventional architecture. The obtained average recognition rates were 77.08% using the conventional architecture and 81.75% using the class-modular.

1. Introduction

The main objective of this work is to evaluate the performance of a Conventional architecture feedforward MLP (multiple layer perceptron) in relation to Class-Modular architecture for the recognition of the handwritten names for the months of the year in Brazilian Portuguese language. This is an important task since it constitutes a sub-problem of bank check date recognition. Although this study deals with a limited lexicon of 12 classes, there are classes that share a common sub-string, which can affect the overall system performance: Jan**eiro**, Fever**eiro**, **Ma**rço, Abril, **Ma**io, **Ju**nho, **Ju**lho, Agosto, Set**embro**, Outu**bro**, Nov**embro** and Dez**embro** [2].

In general, handwriting recognition generates high-dimensional problems [1]. This work also suggests a simple feature set that makes possible the recognition in relatively reduced dimensions.

The power of Artificial Neural Networks (ANNs) resides in its capacity to generate an area of decision of any form. Other works [2, 4, 8] also used ANNs for the recognition of words in lexicons of small dimension.

However, different performances can be obtained with the conventional and modular architectures. Modularity is an essential concept, which should be used appropriately in the design of systems for diverse application areas. Since K classes are involved in the classification module, we can naturally think of the classes as a target of modularity. It leads us directly to the *class modularity* concept [1].

In the class-modular concept, each class should be managed independently of the other classes, at least conceptually [1].

In this work the conventional and class-modular feedforward neural network architectures are evaluated based on a feature set and applying global techniques for the extraction of patterns.

This paper is organized as follows. The Section 2 describes the feature set extracted from the word images. Section 3 and Section 4 introduce respectively the Conventional and Class-Modular architectures. In Section 5 the experimental results are provided with some analyses and discussions. Section 6 presents the concluding remarks and future work.

2. Features Extraction

All the studies of pattern recognition and more specifically of the handwritten one have as one of its relevant points the feature set selection which must represent and discriminate the different found shapes.

In this work, perceptual features [5] and characteristics based on concavities / convexities and another, were explored for the recognition of handwritten names of the months of the year in Brazilian Portuguese language. Basically, they are numbers of occurrences of such features. However, only these discrete primitives do not make the recognition system more robust [9]. Therefore, it was added to the features set a zoning mechanism during the extraction of the primitives.

The zoning is used only in two areas separated by the center of gravity of the word: *left-area* and *right-area*, as shown in Figure 1. It was chosen because in each midfield the occurrence of some features gives more useful information for the pattern classifier.

The ascending and descending zones are computed taking into account the Upper (UL) and Lower (LL) lines. UL and LL are based on maximum horizontal projection histogram of black-white transitions, establishing the central line (CL) [3], as presented in Figure 2.

Figure 1: Zoning mechanism

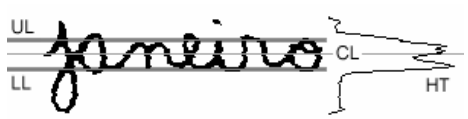

Figure 2: Example of areas detection

The feature set can be described as following:

- Number of loops on the *left/right-areas* (NLL=2 and NLR=3), Figure 1;
- Number of concave semicircles on the *left/right-areas* (NSCVL=3 and NSCVR=5), Figure 3-a;
- Number of convex semicircles on *left/right-areas* (NSCXL=3 and NSCXR=3), Figure 3-b. The concavities and convexities are only extracted in the tuned words. The concave and convex points are obtained by mathematical morphology;
- Number of crossing-points on the *left/right-areas* (NCPL=1 and NCPR=1), Figure 3-c;
- Number of branch-points on the *left/right-areas* (NBPL=3 and NBPR=6), Figure 3-d;
- Number of end-points on the *left/right-areas* (NEPL=3 and NEPR=1), Figure 3-e;
- Number of crossings between the stroke and the horizontal axis (NCH), Figure 3-f;
- Number of ascenders on the *left/right-areas* (NAL=0 and NAR=0);
- Number of descenders on the *left/right-areas* (NDL=1 and NDR=0);
- Proportion of black pixels in relation to the white one (NPP=0.955324), Figure 3-g. The pixels proportion is part of the surface in relation to the context of the word (NPP). A bounded box is used and the proportion can be obtained by the Equation (1) computed inside the bounded box, as follows:

$$prop = (tp - tpp) / tp \qquad (1)$$

where *tp* is the total of pixels inside the bounded box and *tpp* is the total of black pixels of the word stroke,

- Number of vertical lines (NVL=7), Figure 3-h,
- Number of horizontal lines (NHL=0),
- Number of ascenders with loop on the *left/right-areas* (NALL=0 and NALR=0),
- Number of descenders with loop on the *left/right-areas* (NDLL=1 and NDLR=0).

These 14 features are extracted from each word in order to generate a feature vector of 24 dimensions. When a feature is not found in the word, a small value is assumed, for our case, 0.001.

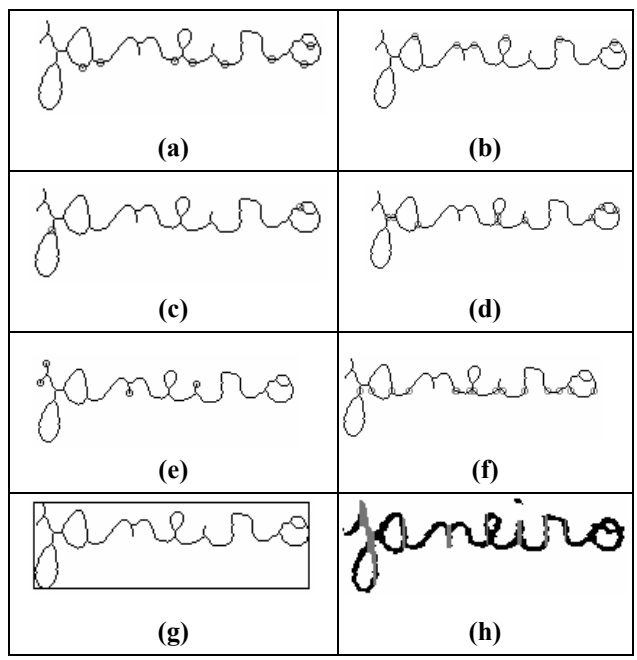

Figure 3: Feature extraction: a) concave semicircles, b)convex semicircles, c) crossing-points, d) branch-points, e)end-points, f) NCH, g)NPP, h)vertical lines

3. Conventional Architecture

The MLP has been used extensively in implementing the *K*-classification module for the word recognition. One of distinct properties of the conventional MLP architecture is that all the *K* classes share one large network [1], as shown in Figure 4. The essential task in designing a character recognition system is to choose a feature type with a good discriminative power and the network should divide the *K* class regions well in the chosen feature space.

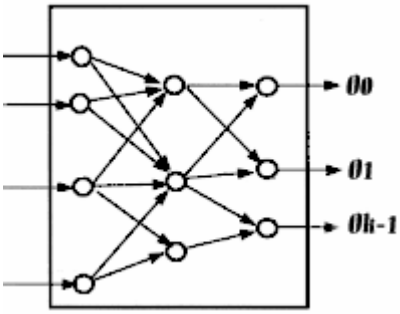

Figure 4: Conventional architecture where *K* classes are intermingled [1]

However, determining the optimal decision boundaries for the *K*-classification module for word recognition in a high-dimensional feature space is very complex, and can seriously limit the recognition performance of the character recognition system using MLP [1,6]. Particularly, when the training set is not large enough compared with the classifier size (i.e., the number of free parameters in the classifier), a problem occurs in convergence [6].

4. Class-modular MLP

A single task is decomposed into multiple subtasks and each subtask is allocated to an expert network. In this paper, as well as in [1], in the class-modular classification, the *K*-classification problem is decomposed into *K* 2-classication subproblems, each for one of the *K* classes. A 2-classification subproblem is solved by the 2-classifier specifically designed for the corresponding class.

The 2-classifier is only responsible for one specific class and discriminates that class from the other *K-1* classes. In the class-modular framework, *K* 2-classifiers solve the original *K*-classification problem cooperatively and the class decision module integrates the outputs from the *K* 2-classifiers.

In Figure 5, we can see the MLP architecture for 2-classifier. The modular MLP classifier consists of *K* sub-networks, M_i for $0 \leq i \leq K-1$, each responsible for one of the *K* classes. The architecture for the entire network constructed by *K* sub-networks is shown in Figure 6.

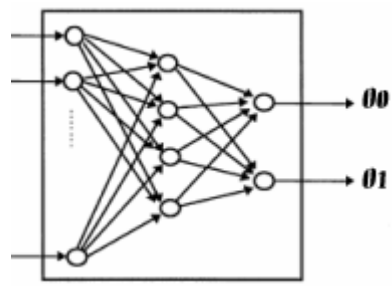

Figure 5: Class-modular architecture: sub-network [1]

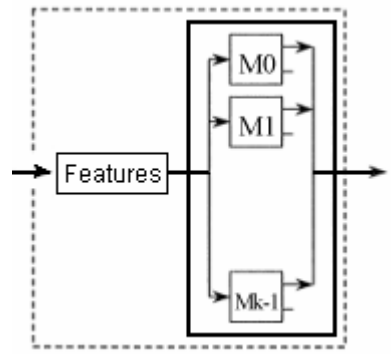

Figure 6: Architecture for the class-modular: whole network with M modules [1]

5. Experimental Results

This section describes the database and presents the results obtained with the conventional and class-modular MLP architectures. To test both architectures, the artificial neural networks used were implemented via the SNNS simulator program [10].

5.1 Database

The database used is composed by names of the months of the year and was collected by UFPB (Federal University of Campina Grande-Paraíba-Brazil), for more details see [2]. In total there are 6000 word images, with 500 of each class. All the images are already preprocessed, i.e., the baseline skew and slant were corrected, reducing the writing variability (different writing styles and particular writing characteristics). For the experiments, the database was randomly divided in three data sets: Training set (60%), Validation set (20%) and Test set (20%). Figure 7 shows sample images from the database.

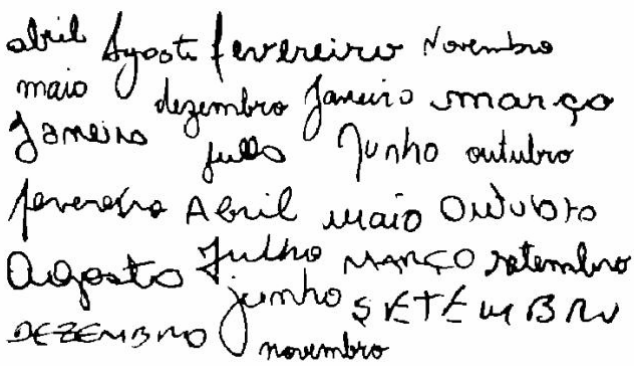

Figure 7: Sample images from the database

5.2 Conventional Architecture Results

Conventional MLP is composed by 24 nodes in the input layer, 45 nodes in the hidden layer and an output layer with 12 nodes. Validation sets were employed in order to avoid over-training. The stop criterion is the increase of the error read in the validation set.

All the classes are trained together. The class that presents the maximum output value is the class considered as recognized. The recognition rate obtained for the conventional architecture is 77,08%. The confusion matrix for the test set is shown in Table 1.

5.3 Class-modular MLP Results

In class-modular MLP, each of K 2-classifier is trained independently of the other classes using the training and validation set. The backpropagation algorithm was used in each of 2-classifiers in the same way as in conventional MLP.

To train 2-classifier for each word class (n = 12), we re-organize the samples in the original training and validation set into n-two groups, *Z0* and *Z1* such that *Z0* has the samples from current class and *Z1* all the other ones, taking account the *a priori* probability for each class.

To recognize the input word patterns, the class decision module takes only the values of O_0 and uses the simple winner-take-all scheme to determine the final class.

A conventional network sees each of the training instances once per epoch. However, in the case of modular network, each subnetwork sees each training instance once per epoch, so the whole network sees each sample K times per epoch [1]. The recognition rate obtained for the class-modular architecture was 81,75%. The confusion matrix for this experiment is presented in Table 2.

Table 1: Confusion Matrix for conventional architecture

Month	J	F	M	A	M	J	J	A	S	O	N	D
Jan	**77**	5	4	0	1	1	1	2	1	2	5	1
Fev	6	**87**	0	0	0	0	0	0	2	4	0	1
Mar	5	1	**74**	2	12	2	2	1	0	1	0	0
Abr	0	1	5	**84**	4	3	2	1	0	0	0	0
Mai	1	2	9	5	**77**	0	2	1	1	2	1	0
Jun	2	2	1	0	4	**77**	10	0	1	3	0	0
Jul	1	1	2	3	4	10	**73**	3	0	3	0	0
Ago	6	2	5	4	0	1	2	**73**	0	0	2	5
Set	2	8	0	1	0	1	0	0	**71**	7	6	4
Out	3	3	3	0	0	2	1	0	9	**76**	3	0
Nov	0	2	6	0	0	1	0	0	3	1	**82**	5
Dez	4	6	0	0	1	1	1	2	4	1	6	**74**

Table 2: Confusion Matrix for class-modular architecture

Month	J	F	M	A	M	J	J	A	S	O	N	D
Jan	**83**	8	2	0	0	2	0	0	0	2	1	2
Fev	5	**83**	1	1	1	0	0	1	3	1	2	2
Mar	3	3	**75**	5	10	0	0	0	0	2	2	0
Abr	1	1	1	**93**	2	0	2	0	0	0	0	0
Mai	1	0	10	5	**80**	2	1	0	0	0	1	0
Jun	1	3	0	0	5	**84**	4	0	1	2	0	0
Jul	1	0	0	6	4	9	**76**	0	0	3	1	0
Ago	2	4	2	3	0	3	0	**78**	0	3	3	2
Set	1	10	0	0	0	0	0	0	**73**	6	8	2
Out	4	4	2	0	0	0	0	0	4	**85**	1	0
Nov	3	2	0	0	0	0	0	0	4	1	**89**	1
Dez	3	5	0	0	0	2	1	1	0	0	6	**82**

5. 4 Discussions

Table 3 summarizes the result obtained in this work and in some other studies [2]. Observe that ANNs in general obtained best results than HMM (Hidden Markov Models), when a similar feature set is applied.

The Conventional network obtained better recognition rate than Directional Features (DF)/ANN. The class-modular network obtained recognition rate similar to the use of the Perceptual Features (PF)/ANN. More than that, each module could yet be optimized, aiming to better rates. Two concluding remarks can be made based on experimental results, as following:

- The class-modular network was superior in terms of the convergence over conventional network (according to the monitoring of the MSE – mean-square error); and
- The class-modular network was also superior in terms of recognition capability regarding the conventional network.

Table 3: Comparison of word recognition results

Set	Recognition
HMM [2]	75.90 %
Directional Features (DF)/ANN [2]	76.60 %
Perceptual Features (PF) /ANN [2]	81.80 %
Conventional Architecture	**77.08 %**
Class-modular MLP	**81.75 %**

6. Conclusions

The results indicate that this research is quite promising and prove to be worthy of further investigations of the class modularity paradigm. A consideration must be made about large-set classification in order to test the effect of the number of classes on the recognition capability (for example: legal amounts [3]).

We proposed and implemented a new feature set with smaller dimension than presented in [2]. Then it generates less parameter to be estimated in the ANN, decreasing the complexity computation without loss of recognition performance. We also observed that the class-modular network was superior in terms of convergence and recognition capability over the conventional network, such as [1].

The conventional architeture has a *rigid* structure composed of an unstructured black box in which all the K classes are altogether intermingled. The modules cannot be modified or optimized locally for each class.

However, the disadvantages of the class-modular architecture are firstly the reorganizations of training, validation and test set to assist each class as described in the Section 5.3, and the training of K networks for the classes of the problem.

The obtained results motivate the continuity of the system development considering a rejection mechanism. Other future work is study a feature set for each one (global and modular architecture) based on dependent-class feature subsets.

7. References

[1] I-S. Oh, C. Y. Suen, "A class-modular feedforward neural network for handwriting recognition", *Pattern Recognition* 35 (2002), 229--244.

[2] J. J. Oliveira Jr., J. M. de C. Carvalho, C. O. de A. Freitas, R. Sabourin, "Evaluating NN and HMM classifiers for handwritten word recognition", *15 th Brazilian Symposium on Computer Graphics and Image Processing* (2002), 210--217.

[3] C. O. de A. Freitas, *Uso de Modelos Escondidos de Markov para o Reconhecimento de Palavras Manuscritas*, Tese de Doutorado, Pontifícia Universidade Católica do Paraná, Brazil, 2001.

[4] M. Cotê, E. Lecolinet, M. Cherriet, C.Y. Suen, "Automatic reading of cursive scripts using a reading model and perceptual concepts", *IJDAR International Journal on Document Analysis and Recognition* (1998) 1: 3--17.

[5] S. Madhvanath, V. Govindaraju, "The Role of Holistic Paradigms in Handwritten Word Recognition", *IEEE Trans. on PAMI* 23 (2001), 149--164.

[6] A. K. Jain, R. P. W. Duin, J. Mao, "Statistical Pattern Recognition: A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), 4--37.

[7] G. Kim, *Recognition of offline handwritten words and its extension to phrase recognition*, PhD Thesis, University of New York at Buffalo, USA, 1996.

[8] D. Ollivier, *Une approche économisant les traitements pour reconnâitre l'écriture manuscrite: application à la reconnaissance des montants littéraux de chéques bancaries*, Thèse de doctorat, Université de Paris XI Orsay, France, 1999.

[9] O. D. Trier, A. K. Jain, T. Taxt, "Feature extraction methods for character recognition-a survey", *Pattern Recognition* 29 (1996), 641-662.

[10] A. Zell et al., *SNNS – Stuttgart Neural Network Simulator, User Manual, Version 4.2*, University of Stuttgart, 1994.