TDNN with Masked Inputs

Fabrice Alleau; Emilie Poisson; Christian Viard-Gaudin; Patrick Le Callet.

Polytech Nantes, School of Electronic and Computer Engineering - France
IRCCyN, Communications and Cybernetic Research Institute of Nantes (UMR CNRS 6597)

Mailing Adress : Polytech Nantes – Equipe Image et Vidéocommunications
Rue Christian Pauc – La Chantrerie – B.P. 50609 – 44306 Nantes Cedex 3 – France

Phone number : 33 2 40 68 30 40
Fax number : 33 2 40 68 32 32

Electronic Adresses :
fabrice.alleau@polytech.univ-nantes.fr
emilie.poisson@polytech.univ-nantes.fr
christian.viard-gaudin@polytech.univ-nantes.fr
patrick.lecallet@polytech.univ-nantes.fr

Abstract :

A novel architecture based on a Time Delay Neural Network is proposed in this paper. The main idea is the introduction of selection masks which allow to specify neurons to a subset of features included in their receptive field. It turns out that the number of free parameters can be decreased while the performances of recognition remains very high. This architecture has been designed for embedded online handwriting recognition systems where memory capacity has to be reduced as much as possible.

Keywords :
Time Delay Neural Network, Masks, Online handwriting Recognition.

Topic and subject Area :
ICICS – Computer Systems, neural and fuzzy systems
ICICS – Signal Processing, signal detection and reconstruction.

TDNN with Masked Inputs

Fabrice Alleau, Emilie Poisson, Christian Viard Gaudin and Patrick Le Callet

Polytech Nantes, School of Electronic and Computer Engineering - France
IRCCyN, Communications and Cybernetic Research Institute of Nantes (UMR CNRS 6597)

Abstract

A novel architecture based on a Time Delay Neural Network is proposed in this paper. The main idea is the introduction of selection masks which allow to specify neurons to a subset of features included in their receptive field. It turns out that the number of free parameters can be decreased while the performances of recognition remains very high. This architecture has been designed for embedded online handwriting recognition systems where memory capacity has to be reduced as much as possible.

1. Introduction

The ability to transcribe handwritten notes to a computerized text format would be of a great interest for many small mobile devices such as Personal Digital Assistant (PDA), smart-phone, digital pen. The challenge for such a system is to achieve high recognition rate with limited memory and computing resources. Some commercial products are already on the market (Transcriber by Microsoft; MyScript by Vision Objects), they perform quite well but they need a quite powerful system (Tablet PC). The objective of this work is to optimise a small architecture which would be able to compete with a more complex and traditional one.

Handwriting recognition can be divided into two fields which differ in the form in which the data is presented to the system. In *off-line* handwriting recognition, the user writes on a document which is later digitized by a scanner. The data is presented to the system as an image, requiring a segmentation of the writing from the image background before recognition can be done. In contrast, the field of *online* handwriting recognition requires that the user writes on a digitizing area using a special stylus, so that the user's written strokes are captured as they are being formed by sampling the pen's (x, y) coordinates at evenly spaced time intervals. The use of a pressure-sensitive switch on the pen tip indicates pen-up/pen-down status and disambiguates stroke segmentations.

State of the art on-line handwriting recognition systems are based on statistical approaches involving either Artificial Neural Network (ANN) and/or Hidden Markov Models (HMM). The former being applied as a global recognizer while the latter tries to fit the data with a more local point of view.

In this paper, we will focus on ANN systems which allow us to work at the character level. These systems have proved to be very efficient as a character recognizer, but a traditional Multi Layer Perceptron architecture requires quite a great number of parameters to achieve high recognition rate. Our objective is to optimise a Convolutional Neural Network (CNN) architecture. Indeed, as stressed by the recent article [5], it presents remarkable properties to handle directly 2D patterns avoiding the subtle stage of the extraction of relevant features.

A second objective is to improve the network performance. by the incorporation of a priori knowledge. Prior knowledge can either be incorporated into the structure itself or into the pre-processing stages. This second way was tested and discussed in [10]. Here, we focus on the first approach, we test the contribution of masks in TDNN, these masks aim at selecting relevant features [8] and adding relevant information in the different layer of the network.

2. Convolutional Neural Networks

The first important experiments on neural networks for handwriting recognition have been proposed in the late eighties [6]. The architecture of these networks was basically Multi-Layer Perceptron with back-propagation learning. More recently, Convolutional Neural Networks [3] have been derived from MLP, they incorporate important notions such as weight sharing and convolution receptive fields. In that sense, they are capable of a local, shift-invariant feature extraction process.

A perceptron has a fully-connected architecture, one of its main deficiencies is that the topology of the input is ignored : the input variables can be presented in any order without affecting the result of the training. For a CNN, a hidden neuron is connected to a subset of neurons from the preceding layer. It is the local receptive field for this neuron. Thus, each neuron can be seen as a specific local feature detector unit. Furthermore, the weight sharing constraint reduces the number of parameters in the system, facilitating thus the generalization process. This type of network has been applied successfully for digit recognition [3][9].

3. TDNN, a Convolutional Neural Networks

The TDNN, Time Delay Neural Network is a convolutional neural network with temporal shift which was first introduced for speech recognition [7]. It has since been transposed for sequential data (see Penacée [3], LeNet5 [5][7]). It is thus particularly suited to process online handwriting signals. We have carefully defined the topology of the network : size of the receptive fields, number of layers, constraints on the weight sharing and also the learning algorithms (1st/2nd order) [9]. The selected architecture of the TDNN consists of two principal parts (see figure 1). The first part, corresponding to the lower layers, implements the successive convolutions which enable it to gradually transform a sequence of feature vectors into another sequence of higher order feature vectors. The second part corresponds to a traditional MLP, it receives as input all the outputs of the extraction part.
We used online data (X, Y) from the Unipen [2] and IRONOFF [11] databases. They were resampled in order to avoid the influence of the pen speed and to obtain a fixed number of points per sample (50 points). Then, a preprocessing module extracts normalized features from each point: position (2), direction (2), curvature (2), pen status (1), for a total of 7 characteristics per point (see figure 2). Concerning learning, the network is trained with a traditional technique based on a stochastic gradient. It gives, according to our tests, as good results as a second order learning method.

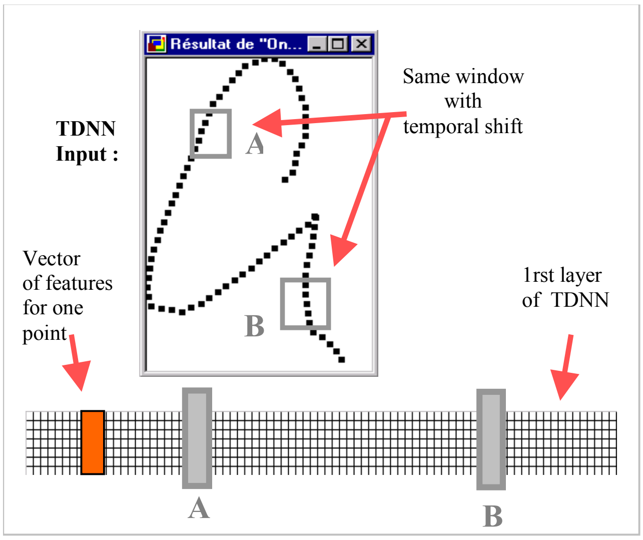
Résultat de "On...
TDNN Input :
A
B
Same window with temporal shift
Vector of features for one point
1rst layer of TDNN
A
B

Figure 2 : TDNN Input

Table 1 presents the comparative performances obtained with the best configurations that we have obtained with a MLP and a TDNN.

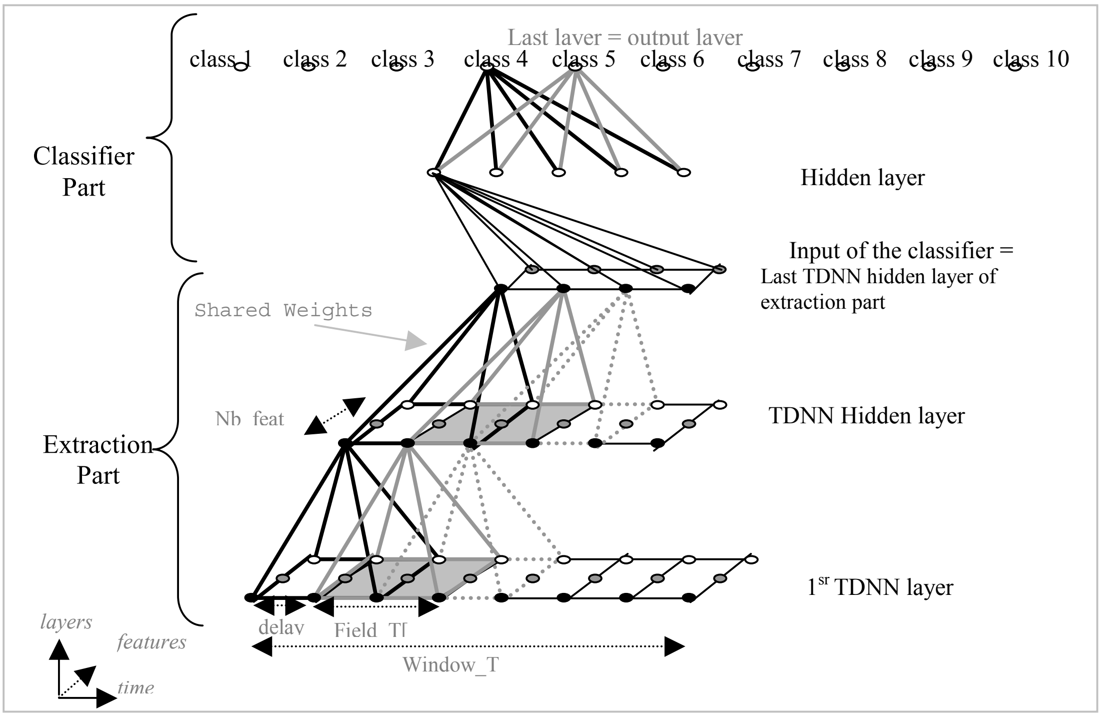

Figure 1 : TDNN architecture

Table 1: TDNN and MLP Recognition Performances.

	Learning set	Test set	**TDNN %**	**MLP %**
UNIPEN + IRONOFF database				
10 Digits	33 141	6 722	**97,9**	**97,5**
26 Lowercase	100 514	21 339	**91,6**	**91,1**
26 Uppercase	79 015	12 795	**94,0**	**92,9**

We can emphasize the significantly higher performances obtained by the TDNN on the three subsets: digits, lowercase and uppercase characters. Indeed, it allows to decrease the error rate up to 16% on the digit set. In addition, the TDNN architecture requires less storage capacity due to its constraint on the weight sharing. For example, the number of coefficients reduces from 36,110 for a MLP (100 neurons on the hidden layer) to 17,930 for the TDNN-digit, (size of receptive field: 20, delay: 5, local features: 20, 100 hidden units for the classifier). This is a factor two reduction rate. Consequently, TDNN architecture presents real advantages for embedded applications. Moreover, it is established that with equal performances (same bias), the simpler a system is, the better its capacities of generalization (lower variance) are [1], known as the famous principle of the Occam's razor "Pluralitas non est ponenda sine neccesitate".

In order to specify the features extracted by a given neuron at the superior level, we propose to introduce a binary programmable selection mask which will allow to select or to inhibit the inputs of the underneath level (figure 3). The definition of different selection masks processing the same receptive field allows to extract information in different subspaces of the original space. A-priori knowledge is used to define the masks and several different configurations are evaluated.

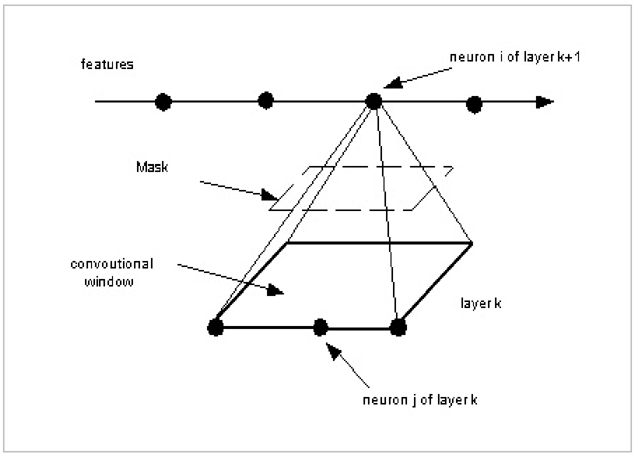

Figure 3 : New structure with a mask

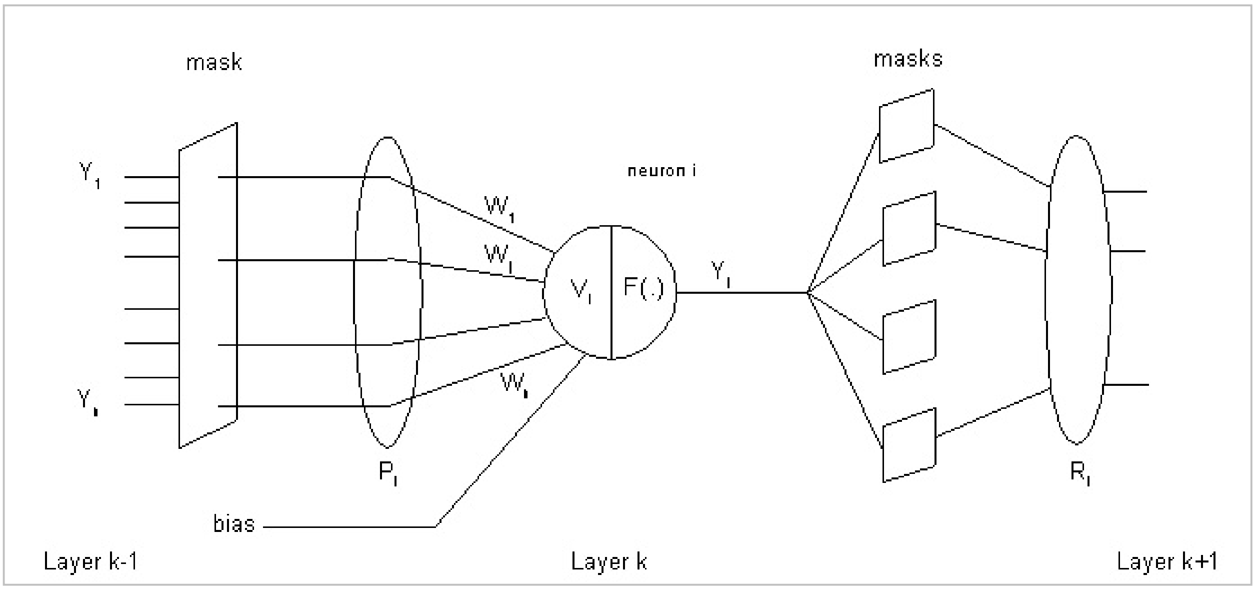

Figure 4 : Zoom on the inputs and outputs of a specific neuron i

4. Interest of masks in a TDNN

With the standard TDNN architecture has tested above, every input which belongs to the receptive field of a neuron participates to the convolution operation. For instance, at the input level, all the seven features derived from a point of the character are involved.

With the notation of the figure 4, a neuron i of the layer k+1 computes the weighted sum function of the state of the neurons (Y1 to Yn) of the lower layer k viewed in its convolutional window. The masks enable to hide some neurons. In that case, the weighted sum corresponds just to a subset of the neurons of the lower layer. The set of theses neurons is defined in the figure 4 by the notation Pi.

4.1 Description and determination of the masks

From the configuration of the TDNN presented in section 3, we have incorporated the mask inside the structure. Many parameters like the size of the mask, the choice of features were determinated by a series of experimental results. Up to now, we have only introduced the masks between the first layer (input layer) and the hidden layer of the TDNN. Each neuron of this hidden layer see a mask applied to the seven input features . The features in the mask are ordered as follow : position-x, position-y, direction-x, direction-y, curvature-x, curvature-y and a flag (P) if the pen is up or down. The value of each bit of the mask is set to 1 if we want to consider the feature or to 0 otherwise. Seven different configurations have been tested. Each configuration has a total number of 20 neurons for a given receptive field. Table 2 describes the masks which are associated to these neurons.

Table 2 : Mask configurations

	X	Y	Dx	Dy	Cx	Cy	P	N	T
TDNN1	1	1	1	1	1	1	1	20	20
TDNN2	1	1	0	0	0	0	1	20	20
TDNN3	0	0	1	1	1	1	1	20	20
TDNN4	1	1	0	0	0	0	1	10	20
	0	0	1	1	1	1	1	10	
TDNN5	0	0	0	0	0	0	1	2	20
	1	1	0	0	0	0	1	6	
	0	0	1	1	0	0	1	6	
	0	0	0	0	1	1	1	6	
TDNN6	1	0	1	0	1	0	1	10	20
	0	1	0	1	0	1	1	10	
TDNN7	1	0	0	0	0	0	0	3	20
	0	1	0	0	0	0	0	3	
	0	0	1	0	0	0	0	3	
	0	0	0	1	0	0	0	3	
	0	0	0	0	1	0	0	3	
	0	0	0	0	0	1	0	3	
	0	0	0	0	0	0	1	2	

In this table, N is the number of neurons which share the same masks for the same receptive field and T is the total number of neurons defined on this field.

The TDNN 1 configuration is the classic TDNN where each input of the receptive field is validated. With TDNN 2, each neuron accesses only to the primary data. In the TDNN 3 architecture, conversely, the derived data are used. TDNN 4 is a compromise between TDNN 2 and 3. TDNN 5 takes into account individually the kind of data : position, direction and curvature. TDNN 6 is dedicated to process two one-dimensional signals. TDNN 7 processes one by one each input features.

4.2 Analysis of the results

These 7 configurations has been tested on the same data as presented in table 1 (UNIPEN + IRONOFF). The first line of table 3 corresponds to the TDNN1 described above in section 3, the given results (first value : number of free parameters and second value : recognition rate on test data) are our baseline references.

Table 3 : Recognition rates of TDNN with different configuration of mask.

TDNN	Digit	Uppercase	Lowercase
1	17 930 **97,9%**	19 546 **94,0%**	19 546 **91,6%**
2	16 330 98%	17 846 93,6%	17 846 91,1%
3	17 130 97,8%	18 746 93,7%	18 746 91,4%
4	16 730 98,0%	18 346 94,1%	18 346 91,7%
5	16 250 98,1%	17 866 94,0%	17 866 91,9%
6	16 730 98,1%	18 346 94,1%	18 346 91,7%
7	**15 530** **98,1%**	**17 146** **94,2%**	**17 146** **92,0%**

We can notice from table 3 that configuration 7 slightly outperforms the others. It concerns either the recognition rate or the number of free parameters of the TDNN. This configuration corresponds to a decoupling stage of the features, they are processed separately with no interaction in the first layer. Basically, we can interpret this step has a preprocessing filter which is applied specifically and independently on each of the 7 features.

5. Conclusion and Perspectives

The architecture which has been developed here has a great potential of flexibility. With the experiments that we have conducted, we have been able to simplify the standard TDNN configuration, the decreasing in the number of parameters being of 13 %, while at the same time the error rate has been decreased of 10 %. With respect to a traditional MLP architecture, it is still better, the decreasing in the error rate is 24 % and for the number of parameters it reaches 57 %. This is very favorable for the implementation in a low cost real time system. Furthermore, there is still room from improvement, such an approach can be extended to the upper layers of the network.

In addition, this type of architecture can be used in several kinds of applications where the selection of many features is difficult and the combination of these features is hard to obtain simply. For example, we have in mind to adapt such a system for video quality assessment. Previously, we have shown how to assess quality on frames [4], our method gives results well correlated with human judgement. We would like to extend it to video. In this context, the main difficult is to express the combination of each frame quality to get a final quality mark for an entire video sequence. For instance, it is well known that the temporal variation of frame quality affects the overall quality but this observation covers several effects remaining misunderstood. We plan to use the purposed system in this paper in order to conduct the combination of quality features coming from frames using several temporal windows.

References

[1] C.M. Bishop, "Neural Networks for Pattern Recognition", *Oxford University Press*. ISBN0-19-853849-9, p 116-161, 1995.

[2] I. Guyon, L. Schomaker, S. Janet, M. Liberman , and R. Plamondon, "First UNIPEN benchmark of on-line handwriting recognizers organized by NIST". *Technical Report* BL0113590-940630-18TM, AT&T Bell Laboratories, 1994.

[3] I. Guyon, J. Bromley, N. Matic, M. Schenkel, H. Weissman, "Penacee: A Neural Net System for Recognizing On-line Handwriting", In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks*, volume 3, pp. 255-279, Springer, 1995.

[4] P. Le Callet, D. Barba "A robust quality metric for color image quality assessment", ICIP, Barcelona, 2003

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Intelligent Signal Processing*, pp. 306-351, 2001.

[6] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Handwritten digit recognition with a backpropagation neural network". In D. Touretzky editor, *Advances in Neural Information Processing Systems 2*, pp. 396-304, 1990.

[7] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," in *The Handbook of Brain Theory and Neural Networks*, (M. A. Arbib, ed.), 1995.

[8] P. Leray, P. Gallinari, "Feature Selection in neural networks", Behaviormetrika (special issue on Analysis of Knowledge Representation in Neural Network Models) Vol.26, No.1, January 1999

[9] E. Poisson, C. Viard-Gaudin, « Réseaux de neurones à convolution : reconnaissance de l'écriture manuscrite non contrainte », *Valgo 2001* (ISSN 1625-9661), N° 01-02, 2001.

[10] E. Poisson, C. Viard-Gaudin, P.M. Lallican, « Multi-Modular architecture based on convolutional neural networks for online handwritten charcter recognition », ICONIP'02, International Conference on Neural Information Processing, Volume 5, pp. 2444-2449, Singapore, Novembre 2002.

[11] C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, "The IRESTE ON-OFF (IRONOFF) Handwritten Image Database", ICDAR'99, pp. 455-458, Bangalore, 1999.