ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 361-366

A Spiking Machine for Human-Computer Interactions

(Design methodology)

Gilles Vaucher

Supélec – Electronics, Signal Processing and Neural Networks team
Avenue de la Boulaie, BP 81127, 35511 Cesson-Sévigné Cedex, France
Gilles.Vaucher@supelec.fr

Abstract. The STANNs (*Spatio-Temporal Artificial Neural Networks*) are spiking neural networks. Coming from a bio-inspired data coding, they are adapted to process spatio-temporal patterns. Their capabilities have been studied for several years in the field of the natural HCIs (*Human-Computer Interactions*): handwritten character recognition, lipreading and speech recognition. Whereas one can observes a renewed interest for this field with the success of mobile telephony and the development of the personal digital assistants, this paper describes how to build a spiking machine with such models with an aim of integrating them in a human-software interface. The approach is illustrated by an industrial application, carried out within the framework of a collaboration between *Supélec* and *France Telecom R&D*. During this study a detailed attention was given to the evaluation of the ease of integration of this technique in a traditional procedure of software development.

1 Introduction

At the crossroads of spiking neurons and traditional ANN models, the STANNs present several assets. They are models made-up of spiking neurons which implement the property of detection of elementary sequences that have the dendritic trees of the biological neurons, according to W Rall. The formalization which is made of these models is not analytical, as it is usually done [4], but algebraic; it is based on the complex numbers [10, 11, 12]. It thus makes it possible to integrate this coding in the traditional ANNs [2, 7] and draws from this fact profit of all the richness of the work already done on theses models, in particular as regards training. The STANNs thus belong at the same time (1) to the family of the spiking neuron networks with which it is possible to make event-driven processing of vectored flows of asynchronous spikes and (2) to the family of the complex-valued neural networks [5] mostly used in signal and image processing.

As regards HCIs, the STANNs were used in several works in on-line handwritten character recognition [8] as well as in unimodal (visual [3], audio [6]) and bimodal (audio-visual [9]) speech recognition: figure 1. In the core of these studies there is a SM (*Spiking Machine* containing STANNs) the design methodology of which is presented hereafter.

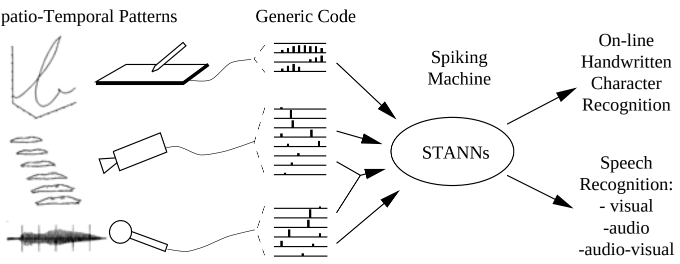

Figure 1: The generic use of the STANNs at the signal level of HCIs

2 SM design

Considering a traditional user-software interface of command-responses type, the introduction of a SM into a uni or multi-modal recognition module is done by splitting up the interface into three blocks. The first ensures the acquisition of the signals (stylus movement, audio or video signal), their pre-processing and their conversion in spikes. The second one is made of the SM whose task is to translate the numerical events (spikes) produced by the first block into symbolic events (letters, words, commands) sent to the third block. Finally the third block is the more traditional part of the interface; its behaviour may be described using a finite-state automata.

2.1 Spike production

The first step of the process after signal acquisition is traditional. It consists of a transformation of the raw signals towards a *good* representation space in which the data present properties of invariance and robustness to noise. For example, for multi or omni-handwritten character recognition, it is important that the resulting coding is invariant to both spatial and temporal translation and scaling. In visual speech processing, more than the stabilities in position and zoom come to be added those concerning brightness and contrast. With regard to the audio speech recognition, the result of the transformation must be robust to the phase and power variations. In practice, depending on the modality processed, traditional transforms from the literature are used.

After this first step the signal is translated to vectored flows of spikes either at the acquisition frequency of the raw signals or when local triggers are detected (more details about this first block in [2, 3, 6, 7, 8, 9]).

2.2 SM components: the STANNs

Whereas the techniques of spike generation used in the first block are specific to the considered modality, the following method proposed to build the SM wants to be generic.

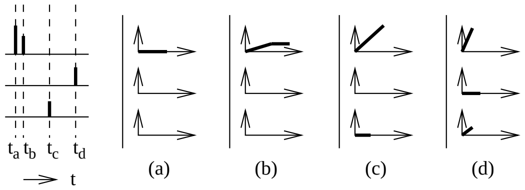

Figure 2: Vectored sequence of spikes, ST coded component by component: (a) The 1st received spike is initially coded by a complex number whose module is equal to the amplitude of the spike and whose phase is null (the date of the spike being referred at the present time). (b) When the 2nd spike arises, it is summed to the previous complex number transformed by $\mathcal{P}$, the $\mathcal{P}$-transform taking account of the time run out between the two spikes. (c-d) Successive coding of the 3rd and 4th spikes received on the other inputs, from which finally the ST representation of the whole vectored sequence of spikes ensues.

It consists in building a feed-forward architecture made of several STANNs. The latter result from the enrichment of traditional ANN architectures, by using a ST (*Spatio-Temporal*) coding of the data and of the parameters. This ST coding which has two degrees of freedom translates a received numerical event (spike), defined by an amplitude and a date, in a complex number while making correspond one to one: (1) the amplitude of the event and the module η of the complex number and (2) the date of the event and the phase φ of the complex number[1]. The complex representation z of the events then evolves in the course of time t by applying the operator $\mathcal{P}$:

$$z_{t+\Delta t} = \mathcal{P}_{\Delta t}(z_t) = \mathcal{P}_{\Delta t}(\eta_t e^{i\varphi_t}) = \eta_t e^{-\mu_S \Delta t} e^{i \arctan(\mu_T \Delta t + \tan \varphi_t)}$$

μ_S and μ_T being two constants (figure 2)[2]. In practice, these constants are both fixed at $1/TW$[3], TW (*Temporal Window*) being the duration of a sort of slipping temporal window, duration which is characteristic of the short-term memorisation process formalized by the $\mathcal{P}$ operator.

This ST coding, which combines both latency and rate coding of the sequences of spikes, is used to build each component of the input vectors X of the neurons. It also intervenes in the coding of the weight vectors W and in the computing of the outputs y. Within the framework of the method presented in this paper, two models of STANNs are used : (1) the ST-SOM which is an ST extension of the Kohonen *Self-Organizing Map* model and (2) the ST-RCE which is drawn from the *Reilly Cooper Elbaum* model while following the same approach. These two types of networks were implemented with the same model of neuron. It is a kernel-unit in which W code a reference sequence, the output of the neuron being provided at any time t by a function f of the hermitian distance d_t between X_t and W. This function is decreasing from 1 to 0 inside a sphere of radius $r > 0$ and is null outside.

[1] Several types of complex numbers were studied. Those used in this paper are the ordinary complexes.

[2] While fixing μ_T at zero, the coding then corresponds to the one used in the traditional leaky integrate & fire model (with a single degree of freedom).

[3] More details about this choice in [2].

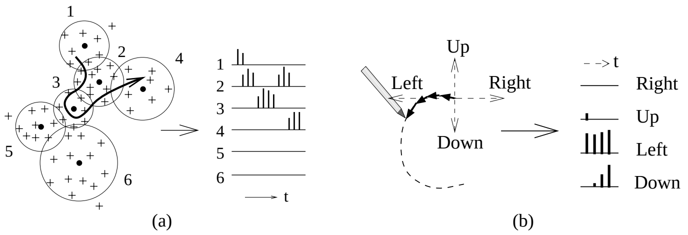

Figure 3: (a) The V → S transform; case of a layer made of six neurons. (b) Conversion of a handwritten letter into a vectored sequence of spikes.

2.3 Synthesis of the SM

The SM, which has a feed-forward architecture, consists of zero to several layers of ST-SOM in cascade followed by a ST-RCE. This last ensures the task of recognition on the basis of the characteristic extracted, layer after layer, by the ST-SOMs which operate on the vectored sequences of spikes within temporal windows of an increasing duration, in some way like the hidden layers of a *Time Delay Neural Network* do.

In fact, the processes carried out by each ST-SOM result from a double transformation: the first one (S → V transform) which translates the vectored sequences of spikes, received by the layer, into an hermitian vector by using the ST coding[4]. During the course of time, this hermitian vector follows under the effect of the $\mathcal{P}$ operator a trajectory which is continuous except at the time of the reception of the spikes.

The second transformation (V → S transform) *a contrario* converts a trajectory in the hermitian space into a vectored sequence of spikes. In fact, during the training, the various trajectories are sampled to produce a set of points in the hermitian space on which the ST-SOM carries out a VQ (*Vector Quantization*). By associating to each neuron a hyper-sphere of influence of radius r[5], one observes at the output of the ST-SOM a vectored sequence of spikes at the frequence of the layer activation, as illustrated on figure 3.a.

Thus, after having *broken up*, thanks to the VQ process, the input sequences into sub-sequences of duration depending on the TW, the ST-SOM produces at its output a sequence of these sub-sequences. Such an architecture then combines both latency and population coding.

The setting in cascade of several ST-SOMs thus makes it possible to analyze the sequences first locally (weak TW) then over durations larger and larger until it is possible to carry out with the ST-RCE the recognition of the global sequences.

[4] To make the explanation of the method simpler, it is supposed here that the connectivity between close layers is complete. In addition, by using a unique TW per ST-SOM, the hermitian vector is thus identical for all the neurons of a layer.

[5] r is computed from the dimensions of the corresponding Voronoï block.

3 Industrial application

After having evaluated by simulation this methodology in handwritten character recognition and speech recognition (visual, audio, audio-visual), an industrial experimentation was carried out in partnership with France Telecom R&D. It was entrusted to a group of students in computer science of Supélec, non-specialists in human-computer interactions, with an aim of validating the potentialities of the method as regards user-software development [1].

The study consisted in developing a pen-oriented interface for acquiring electronic forms with mobile terminals, the capture being done by using the Graffiti$^{(R)}$ alphabet. The letters, digits, punctuation and extended character set having been structured into four group of symbols, four SMs were synthesized.

As in [8], the generation of spikes is made by projecting each stylus movement (differential coding) on the four orientations: right, up, left, down (simplified Freeman decomposition) (figure 3.b). Each SM is made up of: (1) one ST-SOM which *splits* up the character lines into primitives (thus producing an alphabet of characteristic elementary lines)[6] and (2) one ST-RCE which ensures the classification of the characters analysed as sequences of primitives (Global TW).

Six writers produced 150 examples of each symbol. During the simulations, the examples of five writers were used for the training whereas those of the last one were used for the tests. These tests led to 85% of good recognition, 2% of errors and 13% of ambiguities. These last correspond to examples located at the intersection of several coding cells associated to different classes in the ST-RCE. In such cases, it is then possible to produce a *beep* at the user interface level to signal the recognition problem. At last, in exploitation, the system was tested by two users out of the project. One, confirmed, obtained a rate of good recognition of 90% and the other, beginner in the use of Graffiti$^{(R)}$ and in the use of the interface, saw its rate of good recognition falling to 70% at the beginning.

4 Discussion

The proposed methodology has the aim to facilitate the synthesis of spiking neural networks in the field of human-software interfaces development, their role being to translate the spatio-temporal patterns produced by the user into data and commands sent to a software application.

In addition to the potentialities of genericity and multi-modal fusion offered by the event-driven kernel of a SM, the industrial study evoked above showed that the synthesized systems built with such an approach are not much greedy in memory and computing power needs. This technique thus appears promising to develop applications intended for mobile use on low power terminals having only a small power supply. This point was validated by an implementation in a Java$^{(TM)}$ *applet* of a pen-oriented interface in which the handwritten character recognition task is done at each displacement of the stylus (recognition in 10ms

[6] The *length* of the characteristic lines depends on the value of TW.

with a Pentium I, 233MHz). This is made possible thanks (1) to the mechanism of short-term memorizing present in each neuron, mechanism which may be mutualized by several neurons of the same layer, and (2) to the event-driven nature of the implementation in which only a limited number of connections are considered at each slice of time.

The industrial study made it also possible to validate the method, since the non-specialists in handwritten character recognition who implemented it devoted only one and half human×month to the recognition part of the application to obtain altogether acceptable results; the more so as a significant part of this time was devoted to the construction of the database. However, the design methodology of a SM presents still some difficult steps which require a certain know-how: the choice of the number of ST-SOM layers, the choice of the number of neurons in each ST-SOM and the choice of the TW associated to each ST-SOM. We thus work currently on the automation of these choices.

5 Acknowledgements

We make a point of thanking N. Mozayyani and A.R. Baig for their contribution to the progressive working-out of the method presented in this paper.

References

[1] A. Ardouin, N. Brouard, C. Moreau, R. Plouvier, S. Rouchy, and G. Vaucher. Un exemple d'interface orienté stylo à base de stann. In *Journées Neurosciences et Sciences de l'Ingénieur*, sept 2002.

[2] A. R. Baig. *Une approche méthodologique de l'utilisation des STAN appliquée à la reconnaissance visuelle de la parole.* PhD thesis, Univ. Rennes 1 - Supélec, avril 2000.

[3] A. R. Baig, R. Séguier, and G. Vaucher. A spatio-temporal neural network applied to visual speech recognition. In *Int. Conf. on ANN*, pages 797–802, Sept. 1999.

[4] W. Gerstner and W. Kistler. *Spiking Neuron Models.* Cambridge Univ. Press, 2002.

[5] A. Hirose. *Complex-Valued Neural Networks: Theories and Applications.* World Scientific Publishing Co. Pte. Ltd., Singapore, 2003. (to appear).

[6] D. Mercier and R. Séguier. Spiking neurons (stanns) in speech recognition. In *3rd WSEAS Int. Conf. on Neural Networks and Applications*, feb. 2002.

[7] N. Mozayyani. *Introduction d'un codage spatio-temporel dans les architectures classiques de réseaux de neurones artificiels.* PhD thesis, Univ. Rennes 1 - Supélec, Juil. 1998.

[8] N. Mozayyani and G. Vaucher. A spatio-temporal perceptron for on-line handwritten character recognition. In Springer, editor, *Int. Conf. ANN*, pages 325–30, Oct. 1997.

[9] R. Séguier and D. Mercier. Audio-visual speech recognition, one pass learning with spiking neurons. In *Int. Conf. on ANN*, Aug 2002.

[10] G. Vaucher. Neuro-biogical bases for spatio-temporal data coding in ann. In Springer, editor, *Int. Conf on ANN*, pages 703–8, July 1996. Lect. Notes in Computer Sc. 1112.

[11] G. Vaucher. An algebra for recognition of spatio-temporal forms. In D. facto publications, editor, *European Symp. on ANN*, pages 231–6, April 1997.

[12] G. Vaucher. An algebraic interpretation of psp composition. *BioSystems*, 48(1-3):241–6, Sept-Dec 1998.