

Spiking Neurons (STANNs) in Speech Recognition

DAVID MERCIER, RENAUD SÉGUIER

Supélec : Équipe Electronique Traitement du Signal et Neuromimétisme
Avenue de la Boulaie - BP28 - 35511 Cesson Sévigné
FRANCE.

Abstract: - Our team has worked for about ten years on a neural network model that use not continuous inputs but pulse inputs, being inspired by the nature of biological neuron's inputs. This spiking neuron model, called STANN (Spatio-Temporal Artificial Neural Network), enables thus to process spatio-temporal data, that is to say data where the spatial information evolves in the variation in time. This family of neural networks has been put into practice with handwritten character recognition and lipreading problems, which has shown its potential. On the other hand, we have also proposed a generic method to generate spikes when raw data are not. In this article, these tools are confronted with audio. The objective is speech recognition of digits on a well known database: Tulips1.

Key-Words: - Spiking Neurons , STANN , Vector Quantization , Speech Recognition

1 Introduction

Speech recognition is a problem studied for many years and of course numerous neural network models have been tested for this operation (among many other kind of tools). But until now, none of the recent spiking neuron models, being inspired by the biological neurons, has been used. According to us, the most likely reason of this is that audio signal is far from a pulse signal and its conversion is not obvious.

In this article, we shows that the simple and generic pretreatment we proposed in [17] for lipreading in order to generate spikes can be extended to audio signal and that its using with STAN (one of these recent spiking neuron models whose efficiency has been shown over handwritten character recognition [15] and lipreading [2]) gives satisfactory results.

Section 2 presents how we chose the representation of the sound as input of our system. Section 3 deals with the generation of pulses in order to use STANNs presented in Section 4. Section 5 shows the first simulations in different conditions. Some concluding remarks and prospects are given in section 6.

2 Audio signal and its processings : which choice?

As it will be presented in 3., vector quantization is used to generate pulses so the representation of the sound is chosen only according to this constraint. Direct vector quantization on the sound is not robust since the speech is too oscillating and too phase sensitive. On the other hand, vector quantization on the

sound to analytic ends has already been used but on the results of the processings resumed in Appendix 1.

In [7], the authors make a vector quantization to recognize words. Four dictionaries are created: one for the vector with 12 cepstrum coefficients, one for the 12 delta cepstrum coefficients, one for the delta log energy and one for the delta delta log energy. In [8], a system based upon neural nets, makes the objective estimation of rebuild sound. It uses 14 MFCC. The authors tested both a multilayer perceptron (MLP) and a radial basis function net (RBF), and found a better robustness with the RBF, that is to say the neural net that uses prototypes. In [10], the vector quantization is compared when applied on wavelet coefficients and on linear prediction coefficients (LPC). In the special case of digit recognition, LPC works better. In [13], a system that generates 8 facial animation parameters (according to MPEG-4) is presented. Two solutions are considered. On the one hand, a vector quantization is made on large vectors containing both the 8 facial animation parameters and 16 linear prediction coefficients. Then the parameters are those in the prototype closest to LPC. On the other hand, a MLP using 16 cepstrum coefficients (computed from LPC) give the 8 facial animation parameters. In [5], 12 MFCC are quantized (through a Kohonen self organized map or a neural gas) and used to make the unsupervised learning of a temporal organization map (TOM).

Eventually, since cepstral coefficients, log energy and their delta values are already available in the database (see 5.), and since vector quantization works, among many others, on these parameters, we decided

to use them. But there are still many possibilities. We can use only the 12 cepstrum coefficients, the 12 cepstrum coefficients and the log energy, or the whole 26 parameters.

3 Pulse generation

In order to generate easily impulses from multidimensional signals evolving continuously in variation in time, we proposed in [17] to make a static vector quantization (VQ) on the signals captured at different moments. The vector quantization enables to associate a shape prototype to the static shape (defined at a moment by the values of every sensor). The generic procedure defines four steps but in this special case of audio, we restricted it to three steps:

Learning:

- 1) Definition of the M static prototypes.

Exploitation :

- 2) Identification at each time t of the prototype P_k that is the closest to the input signal $X(t)$.
- 3) A pulse is emitted. The M outputs of the pretreatment module are equal to zero but the output associated to prototype P_k on which is generated a pulse. The value of the pulse is in this application defined to 1.

Thus, to apply the STAN on audio problems, once the nature of the inputs is chosen, we need to define only the number of prototypes for the vector quantization.

The objective is to have the system described on Fig.1

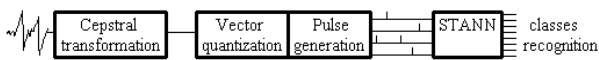


Fig.1: Speech recognition system with STANN

We now present the next and last tool: STAN

4 Classification system: STANN

4.1 STAN

The STAN (Spatio-Temporal Artificial Neuron) is an artificial neuron model created by Vaucher [18], whose underlying coding had been integrated in classical neural architectures [16]. Its principle is to code two-degrees-of-freedom discrete events (amplitude and date) with complex numbers that are also biva-

riant (amplitude and phase).

A STAN is defined by four elements (Fig.2). First, as in classical models, a STAN is characterized by its *weight vector* (W), its *potential function* (V or D) and its *transfer function* (F). The last parameter, called TW , represents the size or the temporal window inside which sequences of impulses should be identified (it can be compared to the maximal delay in a classical dynamical neuron).

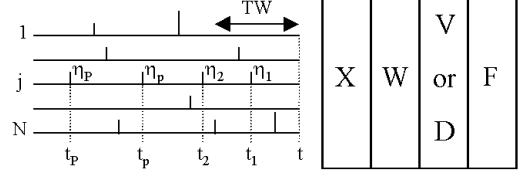


Fig.2: STAN: Spatio-Temporal Artificial Neuron

Calculations are managed according to the following way: a η_1 amplitude impulse emitted at time t_1 on the j^{th} component of the input vector X , is coded at current time t by the complex number:

$$x_j(t) = \eta_1 e^{-\mu_s \tau_1} e^{-i \text{atan}(\mu_T \tau_1)} \quad (1)$$

$$i = \sqrt{-1} \quad \tau_1 = t - t_1 \quad \mu_s = \mu_T = \frac{1}{TW}$$

If a second impulse, with an amplitude equal to η_2 , is emitted at time t_2 on the same component, it is added to current state and this result ages. Thus:

$$x_j(t_2) = \eta_1 e^{-\mu_s(t_2 - t_1)} e^{-i \text{atan}(\mu_T(t_2 - t_1))} + \eta_2 \quad (2)$$

$$= \rho e^{i\phi}$$

and later:

$$x_j(t) = \rho e^{-\mu_s(t - t_2)} e^{-i \text{atan}(\tan \phi + \mu_T(t - t_2))} \quad (3)$$

The operation (2) is realized each time a new impulse is emitted.

Once the input vector X is computed, the potential is equal either to the hermitian product:

$$V(X, W) = \sum_{j=1}^N \overline{w_j} \cdot x_j \quad (4)$$

or to hermitian distance:

$$D(X, W) = \sum_{j=1}^N (x_j - w_j) \cdot \overline{(x_j - w_j)} \quad (5)$$

These potential functions in the complex domain are similar to dot product and euclidian distance in the real numbers.

4.2 Integration in nets : the STANNs

Since an algebra with a dot product and a distance is available, this neuron model has been able to be integrated in classical neural network architectures, adapting easily the learning and relaxation algorithms to complex algebra. Thus [16] and [1] presents spatio-temporal adaptation of the multilayer perceptron (ST-MLP), of the radial basis function neural networks (ST-RBF), of the Reilly, Cooper and Elbaum networks (ST-RCE), of the Kohonen self-organised maps (ST-Kohonen) and its no neighborhood version (ST-Kmeans).

A general utilization procedure for the networks using hermitian distance in order to classify spatio-temporal signals has then be defined in [1].

In our example, as in [17], we use a ST-RCE classifier (see Appendix 2).

5 Tests

To apply STAN on audio, we decided to work with a base that already exists and is well known. We chose a bimodal base, that is to say a base where we can exploit both the images and the sound to make the recognition of digits. From this base, we took only the sound. It is the base called Tulips1 [14]. In this base, twelve persons say twice in english digits from 1 to 4. According to the digit, the sequence and the locutor, from 6 to 16 images are available. Audio information is available under two forms : a raw form and a pre-treated form via 26 audio parameters per image: 12 cepstrum coefficients, the log-energy function, and their derivatives.

In order to have a good idea about the learning and robustness capacities of the STAN in audio, we also decided to test digit recognition with three distinct protocols : the monolocator recognition, the multilocator recognition, and the recognition with an unknown locutor.

5.1 Monolocator recognition

The principle is quite simple : we take the first sequence as the learning base and the second sequence as the test base. We make this operation for each of the 12 persons.

The best results were with 29 prototypes on 26 parameters. The score is 97.6%, i.e. only one mistake: digit 4 with Isaac, recognized as 1. With 10 prototypes on 24 parameters (12 cepstrum coefficients and 12 delta cepstrum coefficients), we already have

89.6% of good recognition. The mistake matrix is :

		Recognized digit			
		1	2	3	4
Digit	1	12	0	0	0
	2	0	12	0	0
	3	1	1	10	0
	4	3	0	0	9

5.2 Multilocator recognition

The principle is to learn by a unic network the pronunciation of many people. Every first sequence is taken as learning base. Every second sequence is added to the test base. Best results were obtained with 45 prototypes quantizing the 26 parameters. Success rate is 93.8% with the following mistake matrix:

		Recognized digit			
		1	2	3	4
Digit	1	10	0	1	1
	2	0	12	0	0
	3	0	0	12	0
	4	1	0	0	11

Nevertheless, with only 25 prototypes, we already obtained a success rate of 91.7%. The matrix of confusion is in this case:

		Recognized digit			
		1	2	3	4
Digit	1	9	1	1	1
	2	0	12	0	0
	3	0	1	11	0
	4	0	0	0	12

5.3 Unknown locutor recognition

The protocol is that of [11]: 22 sequences (2 sequences of 11 people) are used for learning. The tests are made on the 2 sequences of the last person. We made this twelve time, changing the tested person. Best results were with 25 prototypes on 26 parameters. The mean result is 82.3%. The mistake matrix

is:

		Recognized digit			
		1	2	3	4
Digit	1	18	2	1	3
	2	0	22	1	1
	3	2	3	18	1
	4	3	0	0	21

6 Conclusions et perspectives

In this article, we confront the utilization of STANNs with three classical problems in speech recognition: recognition of digits in molocutor conditions, multi-locutor conditions and unknown locutor conditions.

The first results of speech recognition by spiking neural networks are quite promising and let us envisage about two directions of work.

On the one hand, other sound processings than cepstrum transformation should be tried before vector quantization in order to optimize the robustness of the whole system, especially in unknown locutor conditions. Indeed, the state of art shows vector quantization applied on LPC or wavelet coefficients.

On the other hand, since we have exactly the same classification system for lipreading and speech recognition, it will be interesting to try to use both kind of information in a unic system, making thus a efficient bimodal recognition system.

References:

- [1] A.R. Baig, Une approche méthodologique de l'utilisation des STAN appliquée à la reconnaissance visuelle de la parole, *Université Rennes I, PhD Report*, 2000.
- [2] A.R. Baig, R. Séguier and G. Vaucher, A Spatio-temporal Neural Network applied to visual speech recognition», *Ninth International Conference on Artificial Neural Networks (ICANN)*, Vol.2, 1999, pp.797-802.
- [3] R. Boite, and M. Kunt, Traitement de la parole, *Presses polytechniques romandes, complément au traité d'électricité*, 1987.
- [4] I. Daubechies, Ten lectures on wavelets, *SIAM*, 1992.
- [5] S. Durand and F. Alexandre, Learning Speech as Acoustic Sequences with the Unsupervised Model, TOM, *Proceedings 8th International Conference on Neural Networks and their Applications*, 1995
- [6] J.L. Flanagan, Speech analysis, synthesis and perception, *Springer-Verlag, 2nd edition*, 1972.
- [7] V. Fontaine, H. Leich and J. Hennebert, Influence of vector quantization on isolated word recognition, *Proceedings of EUSIPCO*, 1994, pp. 115-118.
- [8] Q. Fu, K. Yi and M. Sun, Speech quality objective assessment using neural network, *ICASSP*, Vol 3, 2000, pp. 1511-1514.
- [9] G.M. Georgiou and C. Koutsougeras, Complex domain backpropagation, *IEEE trans. on circuits and systems - II : Analog and digital signal processing*, May 1992.
- [10] M. Krishnan, C. Neophytou and G. Prescott, Wavelet transform speech recognition using vector quantization, dynamic tyme wrapping and artificial neural networks, *Preprint*, 1994.
- [11] J. Luettin, Visual Speech And Speaker Recognition, *Université de Sheffield, PhD Report*, 1997.
- [12] T. Masters, Signal and image processing with neural networks, *John Wiley & Sons, Inc.*, 1994.
- [13] S. Morishima and H. Harashima, Speech-to-signal media conversion based on VQ and neural network, *ICASSP*, 1991, pp. 2865-2868.
- [14] J.R. Movellan, Visual Speech Recognition with Stochastic Networks, in G. Tesauro, D. Toruetzky, & T. Leen (eds.) *Advances in Neural Information Processing Systems*, Vol 7, MIT Press, Cambridge, 1995.
- [15] N. Mozayyani A.R. Baig and G. Vaucher, A Fully Neural Solution for on-Line Handwritten Character Recognition, *International Joint Conference on Neural Networks (IJCNN)*, 1998.
- [16] N. Mozayyani, Introduction d'un codage spatio-temporel dans les architectures classiques de réseaux de neurones artificiels : Application à la reconnaissance de caractères manuscrits, *Université Rennes I, PhD Report*, 1998.
- [17] R. Séguier and D. Mercier, A generic pretreatment for spiking neuron : application on lipreading with STANN, *International Conference on Artificial Neural Nets and Genetic Algorithms (ICANNGA)*, 2001, pp. 153-156.
- [18] G. Vaucher, A la recherche d'une algèbre neuronale spatio-temporelle, *Université Rennes I, PhD Report*, 1996.
- [19] M. Vetterli and J. Kovacevic, Wavelets and sub-band codind, *Prentice Hall PTR*, 1995.

Appendix 1: Audio processings

This appendix is a very brief quote of the different existing sound processings and their relations.

A1.1 Auto-Regressive model (AR)

The AR model is a mathematical model based upon the simple putting up an equation of the biological model of the speech production system and giving an all pole transmittance for the system. The sound signal at the larynx's output is likened either to a white noise (unvoiced sound) or to the response of a Dirac pulse group through a second order filter. The vocal canal is modelised by a series of acoustic tubes, i.e. a succession of second order resonators. A theoretic justification can be read in [6].

Eventually, the AR model consists in saying that the sound X is the result of the filtering through an all-poles filter H of a source U that is either a gaussian centered white noise or a pulse train whose frequency is the pitch.

In the temporal domain, it amounts to the following recurrence:

$$x[k] + \sum_{i=1}^p a_i \cdot x[k-i] = \sigma \cdot u[k] \quad (6)$$

That is to say a sample is a linear combination of the previous samples and the excitation. It is this recurrence that defines the p -order auto-regressive model.

A1.2 Linear prediction analysis

Historically speaking, the linear prediction analysis is the most important speech analysis techniques. The principle is to use recurrence relation of AR model (6) by considering that the source U is always null and that the autoregressive model is self-sustained (with the proper initialisation). The coefficients a_i are calculated in order to minimise the mean square error of the estimated signal.

A1.3 Spectral analysis

Of course, at the very beginning of speech analysis, the Fourier transform was very used since it was the unique tools that dealt with frequencies. But this transformation is not satisfactory. If it makes it possible to know the present frequencies in the signal, it does not make it possible to know when. Whereas the

vocal signal can be regarded as stationary only over one very short period (from 10 to 30ms), this information is important for speech analysis.

The first solution to this problem was the short-term Fourier Transform. The concept is to modulate the sine of the Fourier transform by multiplying it by a window function. Most of the time, this window function is zero outside some defined range (as rectangular window or a hamming window). But it just needs to have a finite energy so it can be also a gaussian (in this case, it is also called the Gabor transform) ([3] [19]).

A1.4 Formants

The formants are the maximum of the envelope of the spectrum function. The first maxima is the pitch and only the second, third and fourth formants are exploited. Used for a long time to identify phonemes, there are less used today (see [3]).

A1.5 Filter bank analysis

The first application of filter bank analysis was the spectrogram that can plot the short-term power in different frequency bands as a function of time (very close to short-term Fourier transform). But the problem is that the filter bandwidth has to be chosen. A narrow band resolves harmonics but blurs the temporal resolution (of burst for example). A wide band resolves fine temporal details but loses fine frequency details.

So, and with also the idea of approximating the sensitivity of the human ear, non-linear frequency scales were proposed. The best-known non-linear frequency filter bank is probably the Mel scale. The highest the frequency is, the largest the bandwidth is and the best the temporal resolution is.

A1.6 Gabor filters and wavelets

Short-term Fourier analysis suffers the same resolution problem as linear filter bank analysis. So more recent methods, as Gabor filters and wavelets, were created in order to have a good temporal resolution in high frequencies, and poor onset information but a good harmonic resolution at low frequencies.

In Gabor filters, a sine wave modulates a gaussian window. but this window does not have a constant size. It depends on the frequencies that are studied.

Thus the complex expression of a Gabor filter is :

$$g(t, \sigma, w_0) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{t^2}{2 \cdot \sigma^2}} \cdot e^{j \cdot w_0 \cdot t} \quad (7)$$

For wavelets, the temporal atom is more complicated (define as a low pass filter and the complementary high pass filter with some properties [4], [19]) but the rescale principle is quite similar. Once we have the mother function $\psi(t)$, we find the whole discret base thanks functions $\Psi_{s,\tau}(t)$ defined by :

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \cdot \psi\left(\frac{t-\tau}{s}\right) \quad (8)$$

s is the scale that expand or compress the wavelet. τ is the shifts of the function.

A1.7 Cepstral analysis

Inspired by homomorphic treatments, the cepstral analysis consists in transforming the temporal convolution in addition, using the log function in the frequency domain. Indeed, if:

$$x(t) = h(t) \bullet u(t) \Leftrightarrow X(z) = H(z) \cdot U(z) \quad (9)$$

then

$$\log(X(z)) = \log(H(z)) + \log(U(z)) \quad (10)$$

Most of the time the spectral amplitude is enough. In any case, we can see that slowly varying components of $\log(X(z))$ are represented by the low frequencies and the fine details by the high frequencies. Hence another Fourier transform is the natural way to separate the components of H and U .

Extremely fortunately, for the calculation of the cepstrum coefficients, the function logarithm and two Fourier transforms are not necessary. Indeed, only the first cepstrum coefficients are useful. They can be approximate by a non-linear filter bank (Mel filter cepstrum coefficients: MFCC) or from the coefficients of the autoregressive model since the cepstrum coefficients respect the following recurrence relation (see [3]):

$$c_k = a_k + \frac{1}{k} \cdot \sum_{i=1}^{k-1} i \cdot c_i \cdot a_{k-i} \quad 1 \leq k \leq p \quad (11)$$

Appendix 2: The used ST-RCE

A RCE is a two-layer networks. In the first layer, the weights are associated to points in the input space and each neuron compute the distance between the input vector X and its weight vector W (function D as in a RBF). The transfer function is normally a rectangle function which gives 1 if X is in the sphere of influence of the neuron centered on W , and 0 otherwise. In our case, we need at least one neuron to be activated so we prefer a *Winner Takes All* function, that is to say the neuron with the closest weight vector to X is activated at 1 and all the other have 0. The second layer has got as many neurons as there are classes. Links between the first and the second layer are binary links such that an active neuron of the first layer activates at least one and only one neuron of the second layer. During the learning phase, which needs several passes, the number of neurons on the first layer is changeable. At initialization, the layer is empty. For each new example, distances are computed between the input vector and the different weight vectors. Three cases are possible:

- The example is in no sphere of influence. A new neuron is thus created: its weight vector is equal to the example and its radius of influence is equal to k times the lowest distance to other neurons (k is lower but close to 1).
- The example is in the sphere of influence of a neuron associated to another class (by the binary link between the layers). The radius of influence of this neuron is changed to k times the distance between the neuron and X , and a new neuron is created as in first possibility.
- The exemple is in the sphere of influence of a neuron associated to the same class. No modification is made.

The learning is performed with the base until the presentation of every example creates no new neuron.