ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
Published By: **Oriental Scientific Publishing Co., India.**
www.computerscijournal.org

ISSN: 0974-6471
September 2013,
Vol. 6, No. (3):
Pgs. 287-294

Effect of the Neuron Coding by Gaussian Receptive Fields on Enhancing the Performance of Spiking Neural Network for An Automatic Lipreading System

ASMAA OURDIGHI and ABDELKADER BENYETTOU

Department of Computer Science, University of Sciences and Technology
Mohamed Boudiaf of Oran, SIMPA Laboratory, Oran, 31000, Algeria.

(Received: September 07, 2013; Accepted: September 13, 2013)

ABSTRACT

The artificial neural networks have been generally based on rate coding in the earliest stage of computational neuroscience development. What if all the idea of computational paradigm involving the propagation of continuous data affected straight the enhancing of neural network performance and the main objective becomes how to encode the data for modeling biological behavior. The spiking neural networks (SNN) were founded around this concept where not only the network topology, neuron model and plasticity rule should be defined, but also used the timing of the spike to encode and compute information. In this paper, we proposed an automatic lipreading system for spoken digits based on spike response model (SRM). We experimentally demonstrated the impact of the coding strategy to improve the results by comparing two strategies: Spike time coding and population coding by using Gaussian receptive fields (GRF); which achieved 75% and 83.33% accuracy, respectively, on Tulips1.0 dataset.

Key words: Spiking Neural Network (SNN), Spike Response Model (SRM), Automatic lipreading, Spike time coding, Population coding, Gaussian receptive fields (GRF).

INTRODUCTION

Nowadays, all the spiking neuron models are dedicated to match with the biological realistic idea. Each model is built to fit with many coding strategy. The spiking neurons origins are founded around the limitations and features of previous existing models. Historically, the formal neuron proposed by McCulloch and Pitts in[1] was only a binary automat modeling the biological behavior with propagation of a simple binary input and output data in synchrony mode[2]. Limited by the lack of research in the neuroscience, this representation was still far from its original construct. In 1963, the publication of Hodgkin and Huxley works bring an exact mathematical description of the action potential. Their contribution invited several subjects of research to describe in details the dynamics of variations of the ionic current. However, the reproduction of the neuron activity specifies to call an excessive number of variables, which often prevent the comprehension of the model and makes it difficult to implement. Although, there works added more realistic part for the artificial neuron, the concept was practically abandoned. Thus, a whole new generation of

neurons was confirmed far from the previous modeling while basing itself only on the automatic aspect with analogical and synchronous inputs; it's called the second generation[2].

The work of[3] reintroduced the return of biological modeling and opens possibility for several research proposals. The model is then called a spiking neuron as third generation of artificial neuron. Also, the model undermines the rate coding concept usually used in the previous generation. The notion of time became an explicit concept of the model where the propagating data was introduced as spike trains.

The spiking neural networks (SNN) are widely used in robotic, speech recognition, image processing, etc. Their performances start to impose itself gradually in varied fields. In the speech recognition, the work of[4] used the liquid state machine (LSM) based on spiking model "integrate and fire" (LIF) by integrating a simulation of ear for the encoding of information. In the works of[5-6] introduced various applications on classification, auto-association and the clustering. More recently in[7], an approach based of spiking neurons able to achieve microscopic cells segmentation for image processing.

In our work, we proposed to realize automatic speech recognition by using the spike response model (SRM) on one-layered feedforward construction. In the first section, we exposed two coding strategy used in our work which is the aim of our study. In the second section, we explain the features and the topology of the chosen SNN for our approach. Then, we describe a gradient descent algorithm that agreeing with the used network. Mainly, with one layer, the SNN received input data determined by spike time coding strategy or population coding using multiple overlapping Gaussian receptive fields.

Finally, in the experimental phase, we used Tulips 1.0 database to compare the results with work of[6]. With its approach based on the evolution of the pixels gray level over the sequences, first, we decided to delimit the number of pixels. Then, we chose to reduce input neurons by using six geometric features of opening mouth. Finally, we expose all the obtained results with our observations and a comparison with similar work.

Coding strategies

In spiking neural network, each neuron encodes information. The neural coding is supposed to identify the link between stimulus and spike response by modeling spike trains. Several methods are proposed in literature to solve this crucial phase[16-17]. In our work, we used two methods to test the data encoding impact on the enhancing of SNN performance.

Spike time coding

In the spike time coding, each input neuron describe an assignation of a threshold. When the action potential reached the threshold, the neuron will fire at a specific timing otherwise the firing is marked in different timing. This coding is confronted by the problem of threshold initialization but several works demonstrated its performance[13].

Gaussian receptive fields (GRF)

The population coding method that encodes an input variable using multiple overlapping Gaussian receptive fields was proposed in[14]. The Gaussian receptive fields are used to generate ûring times from real values. The stimulus features are represented by a set of neurons resulting from projecting input data on a GRF (see figure.1, below). A range of the data is calculated from all values of each feature, then input data is encoded with a population of neurons that cover the whole data range[10,7].

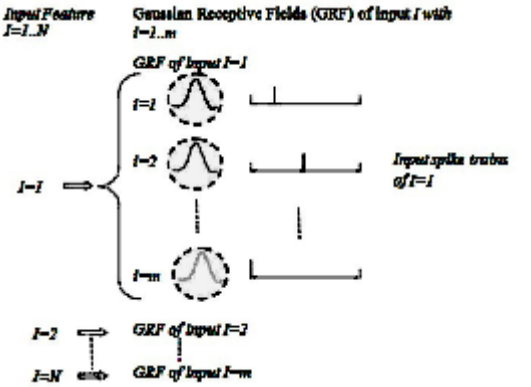

Fig. 1: SNN Input layer with Gaussian Receptive Fields

Formally, the GRF generate spike trains from real values. The aim is to encode with a population of neurons the value of the input feature i following its range . This population of neuron acts as a set of *m* GRF for each input data. Each GRF neuron is described by a center C_i in (1) and a width σ_i in (2).

$$C_i = I_{\min} + \left(\frac{2i-3}{2}\right)\left(\frac{I_{\max} - I_{\min}}{m-2}\right) \quad ...(1)$$

$$\sigma_i = \frac{1}{\gamma}\left(\frac{I_{\max} - I_{\min}}{m-2}\right) \quad ...(2)$$

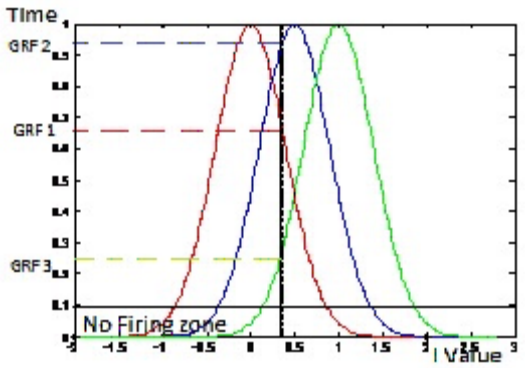

Fig. 2: The firing time values of GRF1, GRF2 and GRF3 is illustrated in time axe for the feature value I=0,4

The spike train is generated following the intersection between the input value and Gaussian graph of GRF (see figure.2, above).

Overview of Spike Response Model

As In this section, we describe the spiking neuron model used in our work, called the Spike Response Model (SRM). Initially, the main features of SRM were presented and described in the work[8]. The model is regarded as generalization of the leaky integrate-and-fire model (LIF). The SRM treats the input spike train to produce a spike response using a simple thresholding concept. The feature of this model lies in the use of a reset kernel function which allowed to create a short refractory phase after a spike emission. Thus, the rest function makes possible to prevent temporarily a neuron to response, even if the accumulation of incoming presynaptic spike train is important.

In our work, the SRM used is a model described in the works of[6,9-10]. In this version, the connectivity linked each presynaptic and postsynaptic neurons is delayed as several neural network model of the second generation[11]. More specifically, this unit uses a block of delayed synapse w_{ji}^{k} to connect presynaptic neuron *i* with a postsynaptic neuron j at a delay for each subsynapse libeled by k. Thus, potential of the neuron j , noted and describing its state, is calculated from the formula (3):

$$u_j(t) = \sum_{t_j^{(f)} \in F_j} \eta(t - t_j^{(f)}) - \sum_{i \in \Gamma_j} \sum_{t_i^{(n)} \in F_i} \sum_{k=1}^{K} w_{ji}^{k} \varepsilon(t - t_i^{(n)} - d^{k}) \quad ...(3)$$

The spike is emitted when the potential reaches the threshold; the time of its emission is noted $t_j^{(f)}$. The firing time is crucial information used for calculation of the potential in posterior moments. In SRM case, we need to note all firing time of presynaptic neurons, also, archive firing time from postsynaptic each neuron. These values take an important part in the refractory phase through the function ·.

Therefore, the output of the neuron j is characterized by the set where firing time is stored every moment of spike per chronological order:

$$F_j = \left\{ t_j^{(f)} : 1 \le f \le n \right\} \quad ...(4)$$

Where *n* is the total number of spikes emitted by the neuron j .

In contrast to the refractory effect, usually, the spike response function μ increase the value of the potential by describing the effect of the incoming spikes from the set of presynaptic of the neuron on the potential of post-synaptic neuron. Many mathematical formulations are possible for the functions μ although the function is generally seen as a short part of increase followed by a long shift. We will use equation (5) where the postsynaptic potential (PSP) is modeled by the difference between two exponential decays:

$$\varepsilon(s) = \left[\exp\left(-\frac{s}{\tau_m}\right) - \exp\left(-\frac{s}{\tau_s}\right)\right] H(s) \quad ...(5)$$

Where denotes the heavy-side step function: $H(s) = 0$ for $s \leq 0$ and $H(s) = 1$ for $s > 0$. Both time-constants τ_m and τ_s (with $0 < \tau_s < \tau_m$) control the increasing and the shifting of the function but more specifically its higher edge.

$$\eta(s) = -\vartheta \exp\left(-\frac{s}{\tau_r}\right) H(s) \quad ...(6)$$

Where ϑ the threshold of the neuron, $H(s)$ represents the function with threshold, and τ_r another time-constant.

For the refractory function · in (6), a simple exponential is used. It is necessary that $\eta(s) = 0$ for. When $s > 0$, its value is negative to ensure the decreasing of the potential. Therefore, it is higher shortly after the time firing.

SNN approach for an automatic Lipreading

An automatic lipreading method is generally integrated as visual feature to support a speech recognition system and enhance their performance. In our work, we exclusively used this technique for the isolated word speech recognition. In this context, the works of booij in [6] used the spiking neuron networks for an automatic lipreading system and achieved a recognition rate of 57% in tulips1.0 test dataset. However, the coding strategy in this work was based on a thresholding of gray-level value of pixels. Therefore, referring to their evolution in the images sequences, the thresholding method transformed the analogical entry into binary vector for each pixel; which is perfectly simplified the input data coding to a spikes train.

Even if this coding seems logical, the image size and length of the sequence directly affects the network parameter. In fact, the number of input neurons will be important because each pixel represents an input neuron (i.e: for their application the image size was 100x75 pixels by sequence that's built a network of 7500 input neurons)

In our approach, we try to optimize the network topology by using new features based on mouth shape instead of using image pixel intensity. Then, we apply two coding strategy: spike time coding and population coding. Thus, in experiments, we use the audio-visual tulips 1.0 dataset. The dataset contains a sequence of images and audio of 12 people pronouncing twice the first four English numbers: "one", "two", "three" and "four".

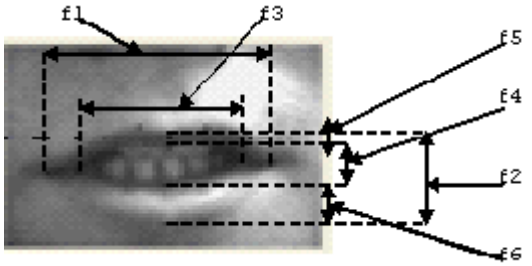

Fig. 3: Six features of the mouth in the pronunciation

For the visual data, each sequence is composed of 6 to 16 images (100x75pixels). More importantly, the dataset contains the evolving of six features for each sequence of images, noted by f_i where $i = 1..6$.
Each feature represents a geometrical distance illustrated above in figure.3. The feature values are in this order:
(f1): The width of outer corners of the mouth.
(f2): The height of outer corners of the mouth.
(f3): The width of inner corners of the opening of the mouth.
(f4): The height of inner corners of the opening of the mouth.
(f5): The height of upper lip.
(f6): The height of lower lip.

Consequently, the input neurons will be reduced to six neurons for spike time coding instead of number of pixels in an image (i.e. 7500 neurons). For population coding, the six features is coded in a group of neurons, so the input layer depends on the number of receptive fields. This reduction of parameters can even optimize the SNN training in terms of time and performance.

Network Topology and coding strategy

In our work, the SNN architecture consists in feedforward network with one layer based on the spike response model. The input data coding is operated by two methods: spike time coding and population coding by Gaussian receptive field. This choice seems ideal considered that the extracted distances of images are real values varying in an interval determined by the opening and size of the mouth.

In the first information encoding, the input layer is composed of six neurons with fixed threshold, the input neuron fire only if its assignation at the sequence is higher than a predefined threshold. However, in the second one, the number of the six input neurons is multiplied by the number of receptive fields. Since, all the spoken digits features evolved over 6 to 16 sequences; the input values is the summation of the same feature according its evolution before applying receptive fields.

The output layer is composed of 4 output neurons depicting the class of the pronounced word ("one", "two", "three" and "four"). The spike train of output neuron contains 20 delays. The attribution of an example to a class is done by comparing the first spike-time. The winning neuron is the one which emitted before the others.

SNN Training

The spiking neural networks are able, by definition, to receive and treat an input spike train to emit another output spike train. By analogy with the conventional versions, the plasticity rule defined the enhancing of the training. The unsupervised training is usually shown more performance in SNN case. For our application, the classes are already defined by labeled sequences for each spoken isolated words wherefore we use the supervised training. We are taking as a starting point the work of[6,10], the learning rule used is a gradient descent algorithm. Formally, the objective is to determine the synaptic weights of the spiking neuron network without hidden layer. Therefore, in order to minimize the error function, all the delayed synapses are modified. This minimization is achieved by gradient descent rule while decreasing proportionally the error function by its derivative. The error value is a very important stage for the learning algorithms. So, in our case, error measurement is determined by the difference between the desired and calculated spike-time of output neurons. That's why it is so important to choose the appropriated coding when we build the SNN because the firing times are the only information returned from an output neuron.

So, we used a learning rule which is inspired by SpikeProp method except that the coding takes into account the first spike $t_j^{(1)}$ and is unaware of all the spikes which come afterwards. The error value can be given by the sum of the differences between desired and obtained time-to-first-spike output:

$$E_{net} = \frac{1}{2}\sum_{j\in J}\left(t_j^{(1)} - \hat{t}_j^{(1)}\right)^2 \qquad ...(7)$$

Where $t_j^{(1)}$ is the desired time-to-first-spike of neuron j and represents the output neuron. In order to minimize the error of the network, we proportionally change each weight with derived from the error with respecting this weight. The modification of the synapse weight goes from the neuron i to the neuron is then noted by:

$$\Delta w_{ji}^k = -\eta \frac{\partial E_{net}}{\partial w_{ji}^k} \qquad ...(8)$$

Where · is a small value constancy determining the learning rate and is the synaptic weight which goes from the input neuron to i the output neuron with a delay .

The synaptic weight influences only spikes delays of the output neuron, we can extend the second factor of the equation (7) to:

$$\frac{\partial E_{net}}{\partial w_{ji}^k} = \frac{\partial E_{net}}{\partial t_j^{(1)}} \frac{\partial t_j^{(1)}}{\partial w_{ji}^k} \qquad ...(9)$$

The first factor, which expresses the error variation of the network according to the time-to-first-spike of the output neuron j, is given by:

$$\frac{\partial E_{net}}{\partial t_j^{(1)}} = t_j^{(1)} - \hat{t}^{(1)}{}_j \quad ...(10)$$

Obtaining the second factor of the formula (9), which expresses the change of spike delay by report with the modified weight, is more difficult, because it does not have their formula to express that. Several successive formulas by replacement of the factors allow this type of calculation, and the following formula is proposed as:

$$\frac{\partial t_j^{(1)}}{\partial w_{ij}^k} = \frac{-\sum_{t_i^{(g)} \in F_i} \varepsilon'(t_j^{(1)} - t_i^{(g)} - d_k)}{\sum_{i,k} \sum_{t_i^{(g)} \in F_i} w_{ij}^k \varepsilon'(t_j^{(1)} - t_i^{(g)} - d_k)} \quad ...(11)$$

Finally, the formula which expresses the change of the weight concretely as follows:

$$\Delta w_{ij}^k = -\eta \frac{-\sum_{t_i^{(g)} \in F_i} \varepsilon'\left(t_j^{(1)} - t_i^{(g)} - d^k\right)}{\sum_{i,k} \sum_{t_i^{(g)} \in F_i} w_{ij}^k \varepsilon'\left(t_j^{(1)} - t_i^{(g)} - d^k\right)} \left(t_j^{(1)} - \hat{t}_j^{(1)}\right) \quad ...(12)$$

Hence, with the definition of the factors: topology, coding strategy and learning algorithm, we can pass to the experimental phase.

EXPERIMENTAL

In this section, we experiment our SNN which performs the spoken isolated words recognition based on visual data with six geometrical distances of mouth, by tracking their evolutions in each sequence. For achieving our experiments, we divided the database into two sets: a training set and a generalization set. The training set consist 6 individuals each one pronounce the four numbers with two recoveries, therefore, we reserved 48 example of training set. The test set contains all 48 vectors remain pronounced by different individuals.

However, we were interested in the information coding proposed in booij's work. We tried to take again his experiment by reducing the input of 7500 neurons related to the size of the image (100x75 pixels) to 2100 neurons determined by a framework size of 60x35 pixels. We specifically localized and extracted the mouth framework from the sequences. Our aims were the elimination of impertinent data. With this approach, after several experiments, we noticed no improvement of the results compared with those obtained by booij. The rate of test always varies between 41% and 46%. Naturally, the training time and generalization time decreases slightly but we detected no utility to reduce the number of neurons in entry with their coding strategy.

So, we launch several times the SNN training with an architecture of six input neurons with the two coding strategy as describe previously. The advantage of this network was the limited size of network parameters which allows a fast propagation of the input data. Consequently, comparing with the previous experience, the learning phase was widely reduced. In All experiences, after trying several values between 10^{-2} and 10^{-7}, the selected learning rate · was equal of 10^{-5}.

For spike time coding, the generalization demonstrates a recognition rate between 64,58% and 75%. In the table.1, we present the results of generalization given by SNN selected previously. The input-neurons thresholds were randomly chosen by observing the data range. In fact, this thresholds initialization was the main inconvenient of the coding stage.

The results obtained by population coding denoted in figure.4 demonstrated the performance of this method where the rates are between 68,75% and 83.33% when the receptive fields varied between 1 and 9.

Table 1. The recognition rates of SNN with spike time coding

	Number of occurrence	Number of occurrence classified	Recognition rate (%) correctly
"one"	12	12	100
"two"	12	8	66,66
"three"	12	9	75
"four"	12	7	58,33
Total	48	36	75

We observed that the accuracy rate increased starting from seven Gaussian receptive fields before stabilizing with increasing the GFR value.

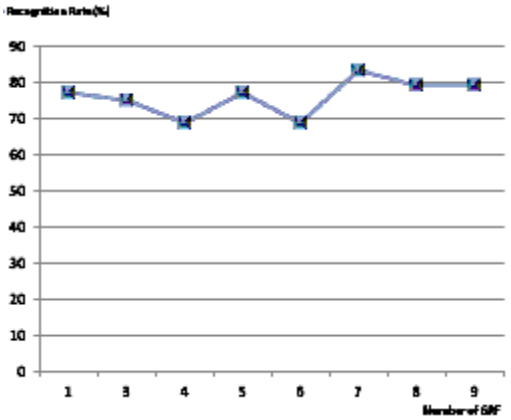

Fig. 4: Evolution of recognition rate with varying number of Gaussian receptive fields

The Table.2 illustrates the pattern in classification accuracy rates. The results showed that the pattern “one” and “two” were correctly classified.

Table 2: The recognition rate of SNN with population coding

	Number of occurrence	Number of occurrence classified	Recognition rate (%) correctly
“one”	12	12	100
“two”	12	12	100
“three”	12	10	83,33
“four”	12	6	50
Total	48	36	83,33

The experimental phase revealed few gaps in the gradient descent Algorithm for each method. We noticed in almost all the launched trainings a divergence and a considerable instability during the minimization of the error. For SNN based on spike response model, we can say that it is enough to change coding strategy to improve the rate of generalization between 20% and 30%.

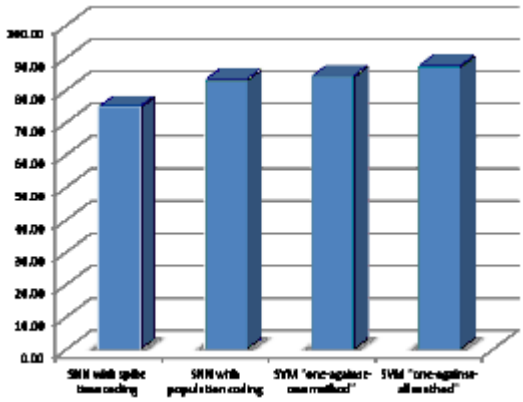

Fig. 5: Comparing recognition rate by applying the SNN approach and SVM with polynomial kernel

For comparing our SNN approach with other methods, we decided to apply support vector machine classifier (SVM). In our experimentation, we tried several kernel functions of the SVM model with two strategies: one-against-one and one-against-all method. The accuracy rates was between 68,75% and 87,5%. The best SVM performance was scored with polynomial kernel which was the only model that enhanced the results comparing our SNN approach. The figure.5 illustrates the recognition rates of one-against-one and one-against-all method which achieved, respectively, 84,38% and 87,5% accuracy rate.

In literature, other works, the researches focused on the definition of the viseme classes to obtain the phonetic transcriptions. The work[15] achieved 92,7% accuracy using Multimodal Sensor Fusion Architecture in Tulips 1.0 dataset. In the work of[6], by using this network with the same database, the results had not exceeded the 57% in the generalization phase. For example, the hidden Markov model (HMM) tested on AVICAR database give rate recognition equal of 50% using the pattern extracts from the image[12]. More recently, a variant of the HMM, called Ergodic HMM[13], provided 66.83% of rate recognition. This application was tested on the Mugshot database and combines with the image pattern. Lastly, SVMs tested on the images of AV database letters. Their result was equal to 62.80%[14].

CONCLUSION

Finally, the obtained accuracy rate of our SNN approach was promising results in à small-scale study. Furthermore, we demonstrated that spike time coding is found to give the least reliable results compared to the neural coding using Gaussian receptive fields. We conclude that coding strategy is an important factor to design the spiking neural network. Consequently, even if the topology and learning algorithm are the main problem to plan in neural network, the coding strategy remains always crucial phase to enhance the network performance. Unfortunately, there is no way to define the appropriated coding. Even if the coding method seems to be the most logical process, only the generalization will determine its efficiency. In fact, the coding strategy adds to the neuron network a new Constraint of initialization.

REFERENCES

1. McCulloch, W.S., Pitts, W. H.A. Logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology.*, **5**(4): 115-133 (1943).
2. 2. Alexandre, F. Les spiking neurons: une leçon de la biologie pour le codage et le traitement des données ., *Traitement et Analyse de l'Information: Méthodes et Applications*(TAIMA)., Tunisia (2005).
3. Maass, W. Networks of spiking neurons: The Third Generation of Neural Network Models. Transactions of the Society for Computer *Simulation International. ,* **14**(4): 1659-1671 (1997).
4. Verstraeten D., Schrauwen B., Stroobandt B. Isolated word recognition using a Liquid StateMachine. Ghent University. ELIS (2005).
5. Schrauwen B. Pulstreincodering en training met meerdere datasets op de cbm. *Master's thesis*; Ghent University (2002).
6. Booij O. Temporal pattern classification using spiking neural networks. *Master's thesis, Intelligent Sensory Information Systems.*, University of Amsterdam (2004).
7. Meftah B., Lezoray O., Benyettou A. Segmentation and Edge Detection Based on Spiking Neuron Networks., *Neural Processing Letters.*, **32**(2): 131-146 (2010).
8. Gerstner W. Spiking neurons. *In W. Maass; C. M. Bisop, editors, Pulsed Neural Networks, MIT Press Cambridge.*, **3**: 54 (1998).
9. Gerstner W., Kistler W.M. Spiking Neuron Models: Single Neurons, Populations, Plasticity. *Cambridge University Press*, (2002).
10. Bohte S. M., Kok J. N., La Poutre H. Error back propagation on temporally encoded networks of spiking neurons. *Neurocomputing.*, **48**: 17-37 (2002).
11. Weibel A. Consonant Recognition by Modular Construction of Large Phonetic Time Delay Neural Network. ATR Interpreting Telephony Research Laboratories., Japan (1988).
12. Yun Fu, XiZhou, Ming Liu, Mark Hasegawa-Johnson, Huang Thomas S. Lipreading by Locality Discriminant Graph. in Proceedings of IEEE International Conference on Image Processing (ICIP) (2007).
13. Kasinski A., Ponulak F. Comparison of supervised learning methods for spike time coding in spiking neural networks. *International Journal of Applied Mathematics and Computer Science.*, **16**(1): 101-113 (2006).
14. Bohte S. M., La Poutre H., Kok J. N. Unsupervised clustering with spiking neurons by sparse temporal coding and multi-layer RBF networks. *IEEE Trans Neural Netw.*, **13**(2): 426-435 (2002).
15. Makkook, M., A Multimodal Sensor Fusion Architecture for Audio-visual Speech Recognition. *Master's thesis.,* University of Waterloo (2007).
16. Cessac B., Paugam-Moisy H., Viéville T. Overview of facts and issues about neural coding by spikes. *J Physiol.*, **104**(1-2):5-18 (2010).
17. Thorpe, S., Delorme, A. Spike-based strategies for rapid processing. *NEURAL NETW.,* **14**(6): 715-725 (2001).