Spiking neuron network for image segmentation
Boudjelal Meftah, M. Debakla, M. Zaagane, A. Benyettou, Olivier Lezoray

To cite this version:

HAL Id: hal-00329522
https://hal.archives-ouvertes.fr/hal-00329522
Submitted on 11 Oct 2008
**Abstract**

The process of segmenting images is one of the most critical ones in automatic image analysis whose goal can be regarded as to find what objects are presented in images. Artificial neural networks have been well developed. First two generations of neural networks have a lot of successful applications. Spiking Neuron Networks (SNNs) are often referred to as the 3rd generation of neural networks which have potential to solve problems related to biological stimuli. They derive their strength and interest from an accurate modeling of synaptic interactions between neurons, taking into account the time of spike emission. SNNs overcome the computational power of neural networks made of threshold or sigmoidal units. Based on dynamic event-driven processing, they open up new horizons for developing models with an exponential capacity of memorizing and a strong ability to fast adaptation. Moreover, SNNs add a new dimension, the temporal axis, to the representation capacity and the processing abilities of neural networks. In this paper, we present how SNN can be applied with efficacy in image segmentation.

**Key words:** Classification, Segmentation, Learning, Microscopic cellular images, Segmentation, Spiking Neuron Network.

**1. Introduction**

Image segmentation consists of subdividing an image into its constituent parts and extracting these parts of interest. A large number of segmentation algorithms have been developed since the middle of 1960’s [1], and this number continually increases at a fast rate.

Simple and popular methods are threshold-based and process histogram characteristics of the pixel intensities of the image. Of course, thresholding has many limitations: the transition between objects and background has to be distinct and the result does not guarantee closed object contours, often requiring substantial post-processing.

Region-based methods have also been developed; they exploit similarity in intensity, gradient, or variance of neighboring pixels. Watersheds methods can be included in this category. The problem with these methods is that they do not employ any shape information of the image, which can be useful in the presence of noise.

Meanwhile, artificial neural networks are already becoming a fairly renowned technique within computer science. Since 1997, Maass [2,3] has quoted that computation and learning has to proceed quite differently in SNNs. He proposes to classify neural networks as follows:

- **1st generation:** Networks based on McCulloch and Pitts’ neurons as computational units, i.e. threshold gates, with only digital outputs (e.g. perceptrons, Hopfield network, Boltzmann machine, multilayer perceptrons with threshold units).
- **2nd generation:** Networks based on computational units that apply an activation function with a continuous set of possible output values, such as sigmoid or polynomial or exponential functions (e.g. MLP, RBF networks). The real-valued outputs of such networks can be interpreted as firing rates of natural neurons.
- **3rd generation of neural network models:** Networks which employ spiking neurons as computational units, taking into account the precise firing times of neurons for information coding.

The use of spiking neurons promises high relevance for biological systems and, furthermore, might be more flexible for computer vision applications [4].

Spiking neurons with leaky integrator synapses have been used to model image segmentation and binding by synchronization and desynchronization of neuronal group activity. The model, which is called RFSLISSOM, integrates the spiking leaky integrator model with the RF-LISSOM structure, modeling self-organization and
active PSPs variations. Whenever the potential increases, the PostSynaptic Potential (PSP) which is a function of time, increases significantly. Each action potential, or spike, emitted by a neuron is inhibitory or excitatory. If the synaptic weight is inhibitory, the IPSP is negative: Sharp decreasing of the potential from the resting potential. A smooth return to the resting potential is observed.

The paper is organized as follows: in the first Section, related works present in the literature of spiking neural network (SNNs). The second Section is the central part of the paper and is devoted to the description of the SNN segmentation method and its main features. The results and discussions of the experimental activity are reported in the third Section. Last Section concludes.

2. Spiking Neural Network

2.1 Biological background

Neurons are remarkable among the cells of the body in their ability to propagate signals rapidly over large distances. They do this by generating characteristic electrical pulses called action potentials, or simply spikes that can travel down nerve fibers.

Neurons are highly specialized for generating electrical signals in response to chemical and other inputs, and transmitting them to other cells. The elaborate branching structure of the dendritic tree allows a neuron to receive inputs from many other neurons through synaptic connections [6].

The membrane potential \( U_j(t) \) of a postsynaptic neuron \( N_j \) varies continuously through time (cf. Figure 1). Each action potential, or spike, emitted by a presynaptic neuron \( N_i \) and the axon that carries the neuronal output to other cells. Some important morphological specializations are the dendrites that transmit signals in response to chemical and other inputs, and the axon that carries the neuronal output to other cells. The elaborate branching structure of the dendritic tree allows a neuron to receive inputs from many other neurons through synaptic connections [6].

The state \( U_j(t) \) of neuron \( N_j \) at time \( t \) is given by:

\[
U_j^{(t)}(t) = \sum_{t_j^{(f)} \in F_j} \eta_j(t - t_j^{(f)}) + \sum_{i \in \Gamma_j} \sum_{t_i^{(f)} \in F_i} w_{ij} \epsilon_{ij}(t - t_i^{(f)} - d^k) \tag{1}
\]

where \( \eta_j \) models the potential reset after a spike emission, \( w_{ij} \) describes the response to presynaptic spikes. For the kernel functions, a choice of usual expressions is given by:

\[
\eta_j(s) = -\vartheta e^{-\frac{s}{\tau}} H(s) \tag{2}
\]

where \( H \) is the Heaviside function, \( \vartheta \) is the threshold and \( \tau \) a time constant defining the decay of the PSP. The function \( \epsilon(t) \) is an \( \alpha \)-function as:

\[
\epsilon(t) = \begin{cases} \frac{t}{\tau} e^{\left(1 - \frac{t}{\tau}\right)} & \text{for } t > 0 \\ 0 & \text{else} \end{cases} \tag{3}
\]
3. Segmentation Using Spiking Neural Network

However, before building a SNN, we have to explore three important issues: network architecture, information encoding and learning method. After that we will use the SNN to segment images.

3.1 Network architecture

The network architecture consists of a fully connected feedforward network of spiking neurons with connections implemented as multiple delayed synaptic terminals (cf. Figure 2).

The network consists of an input layer, a hidden layer, and an output layer. The first layer is composed of three inputs neurons (RGB values) of pixel. Each node in the hidden layer has a localized activation \( \Phi = \Phi(||X - C_n||, \sigma_n) \) where \( \Phi(., .) \) is a radial basis function (RBF) localized around \( C_n \) with the degree of localization parameterized by \( \sigma_n \).

Choosing \( \Phi(Z, \sigma) = e^{-\frac{Z^2}{2\sigma^2}} \) gives the Gaussian RBF. This layer transforms real values to temporal values. The activations of all hidden nodes are weighted and sent to the output layer. Instead of a single synapse, with its specific delay and weight, this synapse model consists of many sub-synapses, each one with its own weight and delay \( d^K \).

Choosing \( \Phi(Z, \sigma) = e^{-\frac{Z^2}{2\sigma^2}} \) gives the Gaussian RBF.

3.2 Information encoding

The first question that arises when dealing with spiking neurons is how neurons encode information in their spike trains, since we are especially interested in a method to translate an analog value into spikes. We distinguish essentially three different approaches [9] in a very rough categorization:

- Rate coding: the information is encoded in the firing rate of the neurons.
- Temporal coding: the information is encoded by the timing of the spikes.
- Population coding: information is encoded by the activity of different pools (populations) of neurons, where a neuron may participate of several pools.

We have used the temporal encoding proposed by Bohte et al. in [10]. By this method, the input variables are encoded with graded and overlapping activation functions, modeled as local receptive fields.

Each neuron of entry is modeled by a local receiving field. A receiving field is a Gaussian function. Each receiving field i have a center \( C_i \) given by the equation (4) and a width \( \sigma_i \) given by the equation (5) such as: \( m \) is number of receptive fields in each population and \( \gamma = 1.5 \).

\[
\begin{align*}
    c_i &= \frac{i - 1.5}{m - 2} \quad (4) \\
    \sigma_i &= \frac{1}{\gamma(m-2)} \quad (5)
\end{align*}
\]

3.3 Learning method

The approach presented here implements the Hebbian reinforcement learning method through a winner-takes-all algorithm [11]. For unsupervised learning, a Winner-Take-All learning rule modifies the weights between the input neurons and the neuron first to fire in the output layer using a time-variant of Hebbian learning: if the start of a PSP at a synapse slightly precedes a spike in the output neuron, the weight of this synapse is increased, as it had significant influence on the spike-time via a relatively large contribution to the membrane potential. Earlier and later synapses are decreased in weight, reflecting their lesser impact on the output neuron's spike time. For a weight with delay \( d^K \) from neuron i to neuron j we use:

\[
\Delta w_{ij} = \eta L(Dt_{ij})
\]

And

\[
L(Dt) = (1 + \beta)e^{\frac{(Dt-\alpha)^2}{2(K-1)}} - \beta
\]

with \( \kappa = 1 - \frac{\beta}{2ln(1+\beta)} \)
The learning window is defined by the following parameters:

- \( v \): this parameter, determines the width of the learning window where it crosses the zero line and affects the range of \( \Delta t_{ij} \), inside which the weights are increased.

- Inside the neighborhood the weights are increased, otherwise they are decreased.

- \( \beta \): this parameter determines the amount by which the weights will be reduced and corresponds to the part of the curve laying outside the neighborhood and bellow the zero line.

- \( \alpha \): because of the time constant \( \tau \) of the EPSP, a neuron \( i \) firing exactly with \( j \) does not contribute to the firing of \( j \), so the learning window must be shifted slightly to consider this time interval and to avoid reinforcing synapses that do not stimulate \( j \).

### 4. Experimental Results and Discussion

We have chosen an image from Berkeley Segmentation Dataset and Benchmark [12] defined in pixel grid of 250x250 pixels (Fig 4).

To show the influence of the number of neurons at exit on the number of areas of the segmented image, we had fixed the number of sub-synapses at 14 between two neurons, the step of training to 0.35, the choice of the base of training is random starting from the image source of 5% and numbers of receiving fields with 18 (6 for each value of intensity) and we varied the number of classes at exit. The images obtained are shown in Figure 5:

To show the influence of the number of sub-synapses on the number of areas of the segmented image we had fixed the number of area at exit at 10, the step of training to 0.35, the choice of the base of training is random starting from the image source of 5% and numbers of receiving fields with 18 (6 for each value of intensity) and we varied the number of subsynapses. The images obtained are shown in Figure 6:
to 0.35, the choice of the base of training is random starting from the image source of 5%, the number of sub-synapses at 14 and we varied numbers of receiving fields. The images obtained are shown in Figure 7:

*Fig. 7. Segmented image with 4 and 6 receptive fields for each value of intensity.*

To show the influence of the percentage of simple training on the number of classes of the segmented image we had fixed the number of area at exit at 10, the step of training to 0.35, the number of sub-synapses at 14 and numbers of receiving fields with 18 (6 for each value of intensity) and we varied the number of percentage of simple training. The images obtained are shown in Figure 8:

*Fig. 8. Segmented image with 5 and 20 % of simple training.*

5. Conclusion
In this paper we applied spiking neural networks to image segmentation. At first, the network is build, a subset of the image pixel is taken to be learned by the network and finally the SNN processes the rest of the image to have as a result an important number of classes quantized the image.

References


