INTRODUCTION

The transforming of incoming signals into action potentials by neurons is believed to be the basis for information processing in nervous systems [1,2,3]. In many cases, the accurate representation of involved timings variability is necessary for a correct computation in neural network simulations [4,5,6,7]. A lot of nervous system simulations reported in scientific literature are computed with time-step based methods. This technique is valid to describe many aspects of biological circuits, but some computational aspects (inefficiency, unreliability, etc.) have been highlighted when used in certain scenarios, especially on very large nets [8]. In this work, a very simple and effective analog spiking neural network simulator, based on LIF (Leaky Integrate and Fire) with latency neurons, is presented. It is simulated with an event-driven method, necessary to guarantee the preservation of the original process behavior. In this way, the simulation proceeds without any forcing in order to obtain a compromise between high precision and computational cost. Networks with up to 10⁶ neurons for more than 10¹⁰ spikes, can be simulated in a few minutes (using a standard PC) with a simple MATLAB tool [9]. Plasticity algorithms are also applied to develop bio-inspired applications and emulate interesting global effects as the Neuronal Group Selection [10].

NEURON MODEL, NETWORK ARCHITECTURES AND SIMULATION PROCEDURE

In this work, we have used a LIF (Leaky Integrate-and-Fire) with latency model. The following biophysical features are considered: subthreshold linear decay (i.e., \( \ell = \ell_d \)), spiking threshold (i.e., \( S_R \)), spike latency (i.e., \( t_{sp} \)), absolute refractory period (i.e., \( t_{ref} \)), excitatory and inhibitory effects (i.e., presynaptic inputs, \( P_e \), synaptic strength (i.e., postsynaptic weight, \( P_i \)). The latter quantity can be affected by synaptic plasticity [11]. The concepts just introduced are described by means of the following equations:

1. Passive Mode (\( S < S_R \)):
\[
S = S + S_P (P_e - T_e)
\]
2. Active Mode (\( S = S_R \)):
\[
S = S + S_P (P_e + T_e) + \frac{T_e}{1/T_e - 1} \]

In which \( S \) is the Inner Neuron State (i.e., the membrane potential of the biological counterpart), whereas \( S_R \) represents the previous state. The leakage term is defined as \( T_e = \ell_d \), whereas the rise term \( T_r = (S_R - 1)/dt \) (for \( S = S_R \), and \( T_r \) is a constant). The neuronal behavior of this model is shown in Fig. 1.

Figure 1. Example of the qualitatively inner state behavior of a neuron in passive and active modes. An incoming excitatory input at \( t_{in} \) causes an instantaneous increase of the state from \( S_{in} \) to \( S_{out} =P_e T_e \). At \( t_{in} \), a second excitatory input is applied, then the state increase its value from \( S_{in} \), to the final value \( S_{out} =P_e T_e \). Note that, \( S_{in} = (S - P_e T_e) \) (i.e., under the spiking threshold (\( S < S_R \)), the neuron is affected by a linear decay (passive mode). Moreover, due to the latency effect, the firing is not instantaneous but occurs after a quantity called time-to-fire \( t_{sp} \), active mode. Finally, after the firing, the neuron is reset to its resting potential (i.e., \( S = S_R \), for an absolute refractory period). In the case of inhibitory inputs \( P_i \) is negative (not shown in this figure).

In Fig. 2 are reported some typical architectures that we have implemented.

Figure 2. (a) Cellular Neural Network-like (CNN-like) topology [12] each firing neuron on (both excitatory and inhibitory neurons) transmits its spikes to a certain number of target neurons (i.e., its neighbors). (b) Feedforward Network (FFN) topology: each neuron of a given layer can only transmit its spikes to the neurons belonging to the adjacent layer. (c) Deliberated Network topology: simple structure in which a delayed neuron \( D \) is used in order to realize an effective coincidence detector. For the purpose of emulating a continuous-time behavior an event-driven approach is required [13,14,15]. Therefore, a simple algorithm has been implemented: the simulation proceeds searching for the neurons with the minimum time-to-fire \( t_{in} \), in order to determine the next firing event to be scheduled in a spike timing array list. Then, the evaluation of firing events effects on all the directly targeted neurons is made.

GLOBAL EFFECTS AND APPLICATIONS

By means of the above mentioned architecture some interesting global effects can be studied, such as formation and maintaining of neuronal groups (Fig. 3) as in the Neuronal Group Selection Theory [10], or the jitter reduction [16] (Fig. 4).

Finally, it is possible to implement SNH classifiers (Fig. 5).

REFERENCES


AWARDS


Best Paper Category : “Spiking neural networks as analog dynamical systems: basic paradigm and simple applications” received at the International Conference “Advances in Computer Engineering (ACE 2012)”, Amsterdam, June 2013.