Signal processing and computational model for neural networks

Conference Paper · September 2002

DOI: 10.1109/ICOSP.2002.1180087 · Source: IEEE Xplore

Douglas Baxter
University of Texas Health Science Center at Houston

Signal Processing and Computational Model for Neural Networks

Chen, Houjin Yuan, Baozong
Institute of Information Science, Northern Jiaotong University, Beijing, CHINA
hjchen@public.bta.net.cn

Baxter, D. A. Byrne, J. H.
University of Texas Medical School at Houston, Texas, U.S.A.

Abstract: Neural networks can process information and generate a specific pattern of electrical activity. Through analyzing its biophysical and biochemical structure and integrating information science into medical science, equivalent circuit and computational model were made, and its interior structure and exterior properties can be studied by use of computer simulation. Therefore, one computer simulation system was designed and implemented to cognize the mechanism of generating and transmitting its information. In this paper, only its computational model of ionic currents was described and a novel algorithm was designed to adaptively optimize simulation output data so as to greatly improve simulation efficiency.
Key Words: Neural network; Signal processing, Computational model, Computer Simulation;

1. Introduction

Neural networks have the ability to process information and generate a specific pattern of electrical activity. And the biophysical & biochemical structure of individual neurons and their pattern of synaptic connectivity determine its external electrical properties. The specific role that any one process plays in the overall behavior of the network can be difficult to access due to such factors as interacting nonlinear feedback loops and inaccessibility of the process for experimental manipulation. Therefore, to analyze and cognize the process of information generation and transmission, and its exterior response properties, informatics was integrated into biology, and equivalent circuit and computational model were presented to simulate the electrical properties and one computer simulation system for neural networks was designed and implemented. In order to match the complexity of individual neural networks, the simulation system was designed to be open, controllable and measurable. To achieve those properties, it was implemented in seven modules: neural network, equivalent circuit, computational model, data structure, comprehensive debug, simulation computation, and output display. They were orderly established according to their dependency and formed a pyramid architecture. The neural network is the basis. Equivalent circuit and computational model are used to simulate its electrical properties. Data structure, comprehensive debug, and simulation computation are for making simulation correct and efficient. The output display is for data output. In this paper, only signal processing and computational model were described.

2. Equivalent Circuit

According to the biophysical structure of a neuron, it is mainly composed of three parts: the dendrites for receiving electrical signals from other cells; the soma or cell body for summing electrical potentials from many dendrites and also containing the nucleus; and the axon for conducting electrical signals and transmitting them to other cells. From the biochemical structure of a neuron, there are mainly three single-element ions: potassium (K^+), sodium (Na^+), and chloride (Cl^-), along with some compound ions. The significant variable for information transmission in a neuron is the electrical potential across the membrane of the axon. It is determined by the intracellular and extracellular concentrations of ions: The ionic currents are generated due to that potential. Neurons connected with synapses (electrical synapses and chemical synapses) form a neuron network. An equivalent circuit (Fig.1) for a neuron is modeled according to its biophysical & biochemical structure and its external electrical properties. Here only ion channels are illustrated, and the synapses and external injections are omitted. This electrical circuit primarily describes the electrical properties of membrane potential for a neuron. The neuron has a membrane potential V_m and a membrane capacitance C_m. E is the equilibrium potential of corresponding ion.

The currents arise mainly from three sources: (1) m voltage-dependent ionic conductances (G_{ion}), (2) n conductances (G_{es}) due to electrical synapses, (3) n×p voltage- and time-dependent conductances (G_{cs}) due to chemical synapses (each electrical synapse is with p chemical synapse) .

Among those conductances, g_L is the leakage conductance, and it corresponds to a linear conductance. g_{Na} and g_K are nonlinear conductances and they are voltage-dependent, and with nonlinear properties. Moreover, there are some external current injections to the neuron.

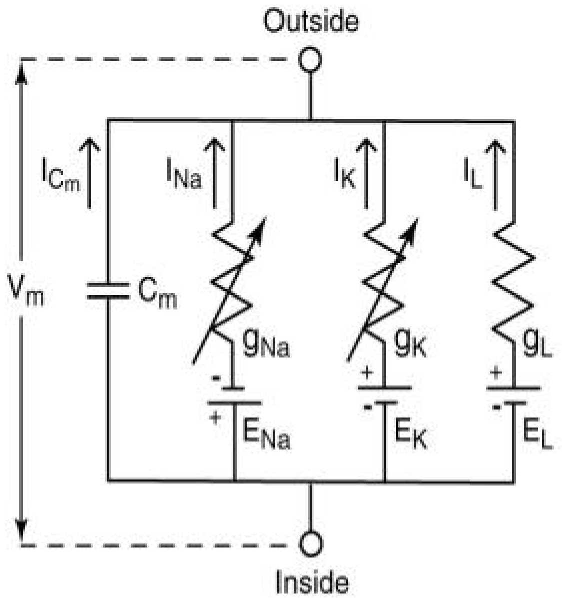

Figure 1. Equivalent circuit of a neuron

3. Computational Modeling

From the equivalent circuit of a neuron, its computational model is established according to circuit theory. The electrical property of a neuron is described by its membrane potential. It is defined by the following differential equation (3-1):

$$C_m \frac{dV}{dt} = I_{stim} - \sum_{j=1}^{m} I_{ion_j} - \sum_{k=1}^{n} I_{es_k} - \sum_{k=1}^{n}\sum_{l=1}^{p} I_{cs_{kl}} \quad (3\text{-}1)$$

Where j, k, l are the indices for the ionic currents, the electrical synaptic currents, and the chemical synaptic currents, respectively; C_m is the membrane capacitance; n is the total number of neurons in the neural network.. I_{stim} is the external current injection.

One of the versatile features of this simulator is to provide multiple models to match the differences of neurons. Here only the computational model for ionic currents is presented here.

3.1 Ionic Current I_{ion}

Based on circuit equivalent and the kinetics of ion channels, the calculation for all ionic currents is expressed in the equation 3-2.

$$I_{ion} = G_{ion} \cdot (V_m - E_{ion}) \quad (3\text{-}2)$$

E_{ion} is the equilibrium potential associated with the conductance. For the calculation of voltage-dependent ionic conductance G_{ion}, five computational models are made in the following equations to account for their specialties of individual neurons.

$$G_{ion} = \bar{g}_{ion} \cdot A^p(v) \cdot B(v) \cdot \prod_{q=1}^{nr} f\left[REG_q\right] \quad (3\text{-}3)$$

$$G_{ion} = \bar{g}_{ion} \cdot m^p(v) \cdot h(v) \cdot \prod_{q=1}^{nr} f\left[REG_q\right] \quad (3\text{-}4)$$

$$G_{ion} = \bar{g}_{ion} \cdot A^p(v) \cdot \prod_{q=1}^{nr} f\left[REG_q\right] \quad (3\text{-}5)$$

$$G_{ion} = \bar{g}_{ion} \cdot m^p(v) \cdot \prod_{q=1}^{nr} f\left[REG_q\right] \quad (3\text{-}6)$$

$$G_{ion} = \bar{g}_{ion} \cdot \prod_{q=1}^{nr} f\left[REG_q\right] \quad (3\text{-}7)$$

Where $\bar{g}_{ion}$ is the maximal conductance, f[REG] represents the conductance modulation by a regulator. *A(v)* and *B(v)* are the voltage-dependent activation and inactivation terms using a time constant method. *m(v)* and *h(v)* are the voltage-dependent activation and inactivation terms using a rate constant method.

3.2 Activation and Inactivation Terms *A* and *B*

The activation and inactivation terms, when using the time constant method, are described by following equations (3-8) and (3-9).

$$\frac{dA(v)}{dt} = \frac{A_\infty(v) - A(v)}{t_A(v)} \quad (3\text{-}8)$$

$$\frac{dB(v)}{dt} = \frac{B_\infty(v) - B(v)}{\tau_B(v)} \quad (3\text{-}9)$$

Where $\tau_A(v)$ and $\tau_B(v)$ are the corresponding time constants; $A_\infty(v)$ and $B_\infty(v)$ are steady-state terms for activation and inactivation.

3.3 Steady-State terms A_∞ and B_∞

Steady-state terms $A_\infty(v)$ and $B_\infty(v)$ for activation and inactivation are voltage-dependent. They are calculated by the equations (3-10) (3-11) respectively.

$$A_\infty(v) = \frac{1}{\left(1 + e^{(V-h)/s}\right)^p} \quad (3\text{-}10)$$

$$B_\infty(v) = \frac{1 - B_{\min}}{\left(1 + e^{(V-h)/s}\right)^p} + B_{\min} \quad (3\text{-}11)$$

Where $h, s, p,$ and $B_{\min}$ are constants.

3.4 Time Constant terms τ_A and τ_B

Time constant terms $\tau_A(v)$ and $\tau_B(v)$ are voltage-dependent. The computational models for them are defined in the equations (3-12) (3-13) respectively.

$$\tau_A(v) = \frac{\tau_{A(\max)} - \tau_{A(\min)}}{\left[1 + e^{(V-h_1)/s_1}\right]^{p_1}\left[1 + e^{(V-h_2)/s_2}\right]^{p_2}} + \tau_{A(\min)} \quad (3\text{-}12)$$

$$\tau_B(v) = \frac{\tau_{B(\max)} - \tau_{B(\min)}}{\left[1 + e^{(V-h_1)/s_1}\right]^{p_1}\left[1 + e^{(V-h_2)/s_2}\right]^{p_2}} + \tau_{B(\min)} \quad (3\text{-}13)$$

Where $\tau_{(\min)}, \tau_{(\max)}, h_1, h_2, s_1, s_2$ are all constants.

3.5 Activation and Inactivation Terms *m* and *h*

The terms *m(v)* and *h(v)* in the rate constant method are described in equation (3-14) (3-15).

$$x = \frac{\alpha_\infty(v)}{\alpha_\infty(v) + \beta_\infty(v)} \quad (3\text{-}14)$$

$$\frac{dx}{dt} = l_x \cdot \left[\alpha_\infty(v)\cdot(1-x) - \beta_\infty(v)\cdot x\right] \quad (3\text{-}15)$$

Where *x is m(v)* or *h(v);*

l_x is a constant;

$\alpha_\infty(v)$ and $\beta_\infty(v)$ are rate parameters.

3.6 Rate Parameters α_∞ and β_∞

The rate parameters $\alpha_\infty(v)$ and $\beta_\infty(v)$ are both voltage-dependent. They are respectively represented in the equation (3-16), (3-17) (3-18).

$$w = \frac{\alpha\cdot(V+b)}{1 \pm e^{(c-V)/d}} \quad (3\text{-}16)$$

$$w = 1 + e^{(c-V)/d} \quad (3\text{-}17)$$

$$w = \alpha \cdot e^{(c-V)/d} \quad (3\text{-}18)$$

Where w may be $\alpha_\infty(v)$ or $\beta_\infty(v)$;

$\alpha, b,$ c and d are constants.

4. Signal Processing

In a network with n neurons, there are m ionic currents, n electrical synaptic currents and n×p chemical synaptic currents. As its total number of conductances may be up to n×(m+n+n×p), so each simulation will deals with a great amount of data. When all simulation data are ready, they are calculated according to their dependency. During simulation, results are graphically displayed in real time. As the step size for numerical computation is very small (10^{-4}~10^{-6}), so each simulation produces up to millions of data. If all data are graphically displayed, the efficiency of simulation will be extremely low. Sometimes it is unbearable. In fact, most neighbor data are identical at the display resolution for some medium. It is unnecessary to display them at the same display position. In addition, graphical display takes much more computer resources than other operations. Therefore, an adaptively optimizing algorithm was designed. It takes display resolution and data characteristics as its optimizing criteria to adaptively filter extra data. Moreover, the simulation results are outputted to four media: screen, file, printer and postscript. They are with different resolutions, and postscript is with highest quality. As the algorithm is run in the core loop, its computation complexity must be low. Its principle is described as follows:

(1) Getting the display width and height for the output medium (screen, file, printer, or postscript);

(2) Setting up optimizing factors (rx, ry) according to the medium type. The greater the factor, the higher the output resolution. Empirical ranges for four media are listed as follows:

screen：	rx=1.0~2.0，	ry=1.0~2.0；
printer：	rx=1.0~5.0，	ry=1.0~5.0；
file：	rx=0.01~ 1.0，	ry=0.01~1.0;
postscript：	rx=1.0~100.0，	ry=1.0~100.0.

(3) Establishing optimizing rules (4-1)~`(4-4) according to the parameters: width, height, rx, ry.

Rule 1	[curX*rx] < width	(4-1)
Rule 2:	[curY*ry] < height)	(4-2)
Rule 3:	[(preX-curX)*rx]≠ 0	(4-3)
Rule 4:	[(preY-curY)*ry] ≠0	(4-4)

Where [ ] is to truncate number into its integer part.

(4) Applying rules to optimize output data.

```
WHILE not the last point, DO
      some current point (curX, curY) generated
      IF Rule1and Rule2 are satisfied, THEN
            IF Rule3 and Rule4 are satisfied, THEN
                  Output(curX, curY);
            ENDIF
        Replace (curX; curY) with (preX, preY)
      END IF
END WHILE
```

where (preX， preY) is previous point;

Here Taking (preX=15.38, preY=246.57), (curX=15.32, curY=246.51) for example,

if rx=1.0, ry=1.0, then current point (curX, curY) is ignored according to the rules (4-3)and (4-4)；

if rx=10.0, ry=10.0，then current point (curX, curY) is outputted；

From its principle, the algorithm is with very low computation complexity and can adaptively optimize the output data. If the output data change rapidly, more data are outputted; else few data are outputted. As its optimizing factors are adjustable, it can achieve excellent tradeoff between computation accuracy and computation efficiency for corresponding output medium. In practical applications, it can make simulation time from days to hours. Sometime it makes some impossible simulation to be possible. Through a great amount of practical applications, it has been shown to be very efficient and critical to the simulation computation.

5. Conclusion

In this paper, signal processing and computational model of a computer simulation system for neural networks is described. And one adaptively optimizing algorithm for signal processing is presented and it is verified to be efficient. And computational model is shown to match their electrical properties of neural networks. As the simulation system was implemented in platform independent JAVA language, it can virtually be executed in almost all computers with plentiful graphical interfaces. Therefore, it has been widely applied in the world. It is useful for theoreticians and experimentalists to explore the properties of complex neural systems.

References:

[1] Robert E. Keen, James D. Spain, Computer Simulation in Biology, Wiley-Liss, Inc. 1992.

[2] Christof Koch, Biophysics of Computation-Information Processing in Single Neurons, Oxford University Press, New York, 1999.

[3] Christof Koch, Idan Segev, The role of single neurons in information processing, Nature neuroscience supplement, Vol.3, November 2000.

[4] Koch, C. Biophysics of Computation, Oxford University Press, NY, 1999.