Proceedings of 1993 International Joint Conference on Neural Networks

Biologically-Inspired Artificial Neurons : Modeling and Applications

M. Scholles[1], B.J. Hosticka[2], M. Kesper[2], P. Richert[2], and M. Schwarz[2]

[1] Department of Electrical Engineering, University of Duisburg, D-4100 Duisburg, Germany
[2] Fraunhofer Institute of Microelectronic Circuits and Systems, D-4100 Duisburg, Germany

Abstract

Currently used neural networks employ mostly simple neuron models that greatly differ from the "real" biological neurons. To ensure progress in biology-based neural processing, more advanced neuron models must be developed that better reflect the biological functionality. In this communication, we investigate a neuron model which satisfies such requirements to a much higher degree. We also examine some of its learning properties and look at its applications.

1. Introduction

Even though neuron modeling is a science with a considerable long history, models used in today's neural processing are still rather simple when compared with the real-world biological neurons. Computational models have not evolved much past the McCulloch-Pitts original and thus this area needs scientific progress if deeper investigations of biologically-oriented neural processing are to be carried out. "Closer" biological neuron modeling, however, is not the only issue to be tackled but other topics, such as learning and applications of such neurons, must be investigated as well. It is the aim of this contribution to address precisely these issues, although we are aware that there are other problems, too, e.g. question of topologies of networks containing such neurons that must be attended to in the future.

2. Biologically-Inspired Neuron

Essentially, the "biological" neuron model we are searching for must allow modeling of local intraneural computation and synaptic interneural communication. The first feature can be conveniently described by a set of nonlinear differential equations based on the Hodgkin-Huxley model [1], while the second feature can be characterized as nonlinear synaptic "weighting" and integration of interneural action potentials. Such potentials express neural activity changes and appear as pulse trains in neurobiological systems. This kind of pulse processing and transmission is of course not the only characteristic property of these systems: the others are massively parallel computation and communication and learning capabilities.

To start with, we first concentrate solely on the biological neuron. We consider only linear differential equations of the first order so that our model retains some chance for economical realization. We also consider only sampled-data implementations and thus use difference equations only. If $x_i(t)$ denotes the i-th synaptic action potential and $y_j(t)$ the j-th neuron action potential at its output, we can describe the intraneural and interneural activities of a single neuron connected with other neurons using the following expression:

$$\frac{\Delta y_j(t)}{\Delta t} + k_{1j}(t)y_j(t) = k_{2j}(t)\sum_{i=1}^{N} w_{ij}(t)x_i(t) \qquad (1)$$

where w_{ij} is the i-th synaptic weight, and k_{1j} and k_{2j} are gain factors yet to be determined. In addition, biological research has shown [2] that arrival time of pulses and axonal time delays play important roles when evaluating dynamic events and learning, so that they have to be included in our neuron model.

Our neuron model is illustrated in Fig. 1. It corresponds roughly to Eq. 1. Comparing Eq. 1 and Fig. 1, we can deduce that $w_{0j}(t)$ should be equal to $(1 - k_{1j}(t)\Delta t)$, where Δt is a clock period in sampled-data realization. The signal transmission is based on asynchronous pulse processing, as in neurobiological systems. These pulses exhibit constant width (usually a very narrow width) and amplitude, so that only their frequency and phase contain the information. While the original McCulloch-Pitts model relies solely on adjustable synaptic weighting of incoming excitatory and inhibitory signals, postsynaptic summation, and static thresholding, our model also includes adjustable synaptic time delays, variable cell membrane potential with adjustable gain, memory, memory leakage, and dynamic thresholding to obtain variable refractory periods. In our model all these features are variable and can also be learned. As it can be seen, this model is much more complex than the McCulloch-Pitts model, but is also much more powerful.

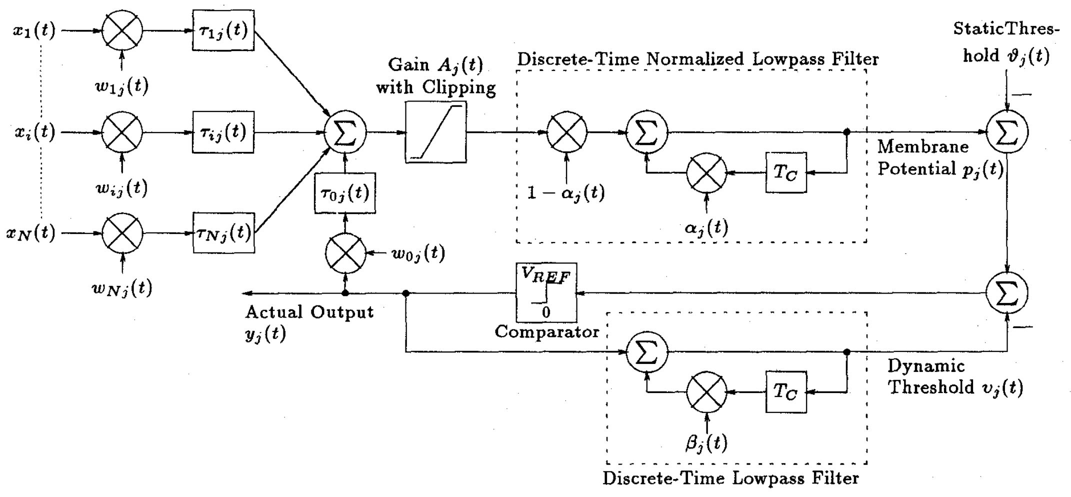

Figure 1: Block diagram of the biologically-inspired neuron model

3. Mathematical Model

Now we will have a closer look at neuron mathematics. The synaptic action potential at the i-th synapse is designated as $x_i(t)$ and modulated onto a pulse rate. The pulses used have a very narrow width τ_p and a uniform amplitude V_{REF}. They arrive at the neuron input, and are weighted and delayed using adjustable synaptic weights $w_{ij}(t)$ and delays $\tau_{ij}(t)$, respectively, so that we obtain a summation signal (see Fig. 1):

$$s_j(t) = V_{REF}\left[\sum_{i=1}^{N} w_{ij}(t-\tau_{ij}(t))\cdot x_i(t-\tau_{ij}(t)) + w_{0j}(t-\tau_{0j}(t))\cdot y_j(t-\tau_{0j}(t))\right] \tag{2}$$

for N synapses. This signal is subsequently multiplied by $A_j(t)$, which represents a linear gain with clipping, and then filtered by a normalized discrete-time lowpass filter. Its coefficient is α_j, which corresponds to $\alpha_j = e^{-(T_C/R_\alpha C_\alpha)}$. $R_\alpha C_\alpha$ is the time constant of an equivalent 1st order continuous-time RC-lowpass, and $f_C = 1/T_C$ is the clock rate. The instantaneous membrane potential is available at the output of the lowpass filter as a signal $p_j(t)$. For a constant pulse rate $1/m$ (i.e. normalized with respect to clock frequency $1/T_C$) at one synapse it can be shown that a potential at the lowpass output is generated which is roughly proportional to the pulse rate at its input. Thus if we vary this pulse rate the lowpass filter acts as a demodulator.

The signal $p_j(t)$ controls a pulse generator which fires output pulses each time the membrane potential $p_j(t)$ has exceeded the sum of static and dynamic thresholds $\vartheta_j(t)$ and $v_j(t)$, respectively (see Fig. 1). We have to determine now the output signal at the generator output $y_i(t)$ from the pause between two subsequent pulses. The elapsed time kT_C between two firings is to be calculated from the following equation: $v_j(kT_C) = p_j(kT_C) - \vartheta_j(kT_C) = \beta_j^k(kT_C)V_{REF}$. If we consider that $p_j(t)$, $\vartheta_j(t)$, and $\beta_j(t)$ vary very slowly we can calculate the pulse rate using the Taylor expansion series of the logarithm function as

$$y_j(t) \simeq [1-\beta_j(t)]\left[1 + (p_j(t) - \vartheta_j(t))/V_{REF}\right] \tag{3}$$

where $p_j(t) = A_j(t)s_j(t)$. The pulse rate (again normalized with respect to the clock frequency $f_C = 1/T_C$) at the neuron output can be now derived as

$$y_j(t) = [1-\beta_j(t)]\left\{1 + [A_j(t)s_j(t) - \vartheta_j(t)]/V_{REF}\right\} \tag{4}$$

This expression is valid only if $p_j(t) - \vartheta_j(t) > 0$, otherwise $y_j(t) = 0$.

4. Network Topology

The intriguing feature of the "biological" neuron is its dynamic behaviour, not present in the McCulloch-Pitts neuron. On the one hand this is due to synaptic delays at the input and on the other hand owing to its temporal memory. Thus, the neuron is capable of processing time-varying input signals. The creation of temporal memory, however, requires recursive feedback topologies and thus raises the question of network stability. It is well known that adaptive recursive filters, similar to our structure, exhibit stability problems. The treatment of this problem is highly involved and eludes any brief discussion. It can be shown, however, that the stability requirement can be handled in a much easier way, if we reduce the complexity of the recursive loop. That is if we use only local recursive loops involving single neurons, the stability issue can be solved rather easily. It still yields a rather powerful dynamic neuron, as we shall see. This is due to the fact that the "biological" neuron uses time as an extra dimension, although it does not necessarily mean that we can do without additional recursive loops involving more neurons for solving more complex problems.

5. Derivation of Learning Algorithms

We now proceed to learning algorithms. For this, we always consider the adjustment of only a single neuron parameter at a given time. The algorithm we are going to employ is that borrowed from transversal adaptive filters which commonly use the least-mean-square (LMS) algorithm [3] for their coefficient adaption. The algorithm tries to minimize the expression $\xi_j = E\left(\delta_j^2\right)$, where E stands for expectation, i.e. statistical average, and δ_j stands for the output error of the j-th neuron, i.e. the difference between the desired output d_j and the actual output y_j of the j-th neuron: $\delta_j(t) = d_j(t) - y_j(t)$.

As the neuron structure of Fig. 1 represents a recursive structure due to the feedback loop containing w_{0j} and τ_{0j}, stability issue is highly involved, as pointed above. In addition, the feedback makes the adaption procedure rather complex. Hence, we have chosen the simplest algorithm for recursive structures first derived by P.L. Feintuch [4]. This algorithm neglects the storage character due to the recursive computation when calculating the coefficient updates. Therefore, the parameters w_{0j} and τ_{0j} can be treated as the other synaptical weights and delays, with $x_0(t) = y_j(t)$. As far as the stability condition is concerned, it is rather simple in our case owing to the 1st order denominator: to guarantee stability we must ensure $|[1-\beta_j(t)]A_j(t)w_{0j}(t)| < 1$ at all times.

Each time we try to minimize $\xi_j = E\left(\delta_j^2\right)$, we have to change a neuron parameter ϕ (ϕ is one of the parameters w_{ij}, τ_{ij}, A_j, ϑ_j, and β_j). Since the LMS algorithm replaces the expectation $E\left(\delta_j^2\right)$ by the square of the error δ_j itself, we can derive the equation

$$\phi(t+\Delta t) = \phi(t) + K\cdot\left(-\nabla_\phi(\xi_j)\right) = \phi(t) + K\cdot\left(-\nabla_\phi(\delta_j^2)\right) = \phi(t) + 2K\delta_j\frac{\partial y_j}{\partial\phi} = \phi(t) + \Delta\phi \qquad (5)$$

to fulfill the minimization criterion, where K is an adaptivity factor. Eq. (5) is the well known delta rule, which leads to the following expressions for the adaptation of the neuron parameters:

$$\begin{aligned}
\Delta w_{ij}(t) &= 2\eta A_j(t)\delta_j(t)\left[1-\beta_j(t)\right]x_i\left(t-\tau_{ij}(t)\right) && (6)\\
\Delta\tau_{ij}(t) &= -2\mu A_j(t)\delta_j(t)\left[1-\beta_j(t)\right]\cdot\left[x_i\left(t-\tau_{ij}(t)\right)\frac{\partial w_{ij}(t-\tau_{ij})}{\partial t} + w_{ij}\left(t-\tau_{ij}(t)\right)\frac{\partial x_i(t-\tau_{ij})}{\partial t}\right] && (7)\\
\Delta A_j(t) &= 2\zeta\delta_j(t)\left[1-\beta_j(t)\right]\cdot s(t)/V_{REF} && (8)\\
\Delta\vartheta_j(t) &= -2\rho\delta_j(t)\left[1-\beta_j(t)\right]/V_{REF} && (9)\\
\Delta\beta_j(t) &= -2\epsilon\delta_j(t)\left\{1+\left[A_j(t)s_j(t)-\vartheta_j(t)\right]/V_{REF}\right\} && (10)
\end{aligned}$$

where η, μ, ζ, ρ, and ϵ are adaptivity factors, which control the adaption rate. A low constant factor yields slow convergence for low-level signals but ensures stability for high-level signals. In case we wish to eliminate this signal dependence, a variable adaptivity factor can be employed. The factor can be controlled using signal properties, e.g. signal power. This principle can be applied to all adaption algorithms. Note that in case we have more inputs, the total input signal power is easy to evaluate.

6. Application

There are certainly many applications where the "biological" neuron can be used. Instead of discussing a wide variety of these we will concentrate on single application and discuss it in more depth. The one we have chosen belongs to the area of time-domain beamforming. This method is commonly used for sonar ranging, steering, and focussing. It involves a uniformly spaced hydrophone array, which receives wavefronts reflected from a target. The arrival delays at the individual transducers are caused by acoustical delays in the water and are affected by the hydrophone spacing, the sound velocity in the water, and the "look" direction angle.

We now perform the following experiment, which is shown in Fig. 2: each transducer is connected to a single "biological" neuron which acts as a lowpass filter to transform the sinusoidal pulses into almost triangular shape. The outputs of all these neurons are then fed to separate synapses of a single "biological" neuron. Adjusting only the synaptic delays of this single neuron by "learning" according to Eq. 7 in such a way that delivers maximum at the neuron output, we achieve a coherent addition of the in-phase input signals. Therefore, the synaptic weights have to be normalized, so that the output of the beamformer is equal to unity if the transducer array is steered in the right direction. The choice of the synaptic weights also affects the selectivity of the beamformer, because the weighting determines windowing of the input sequence. Proper windowing yields small sidelobes and high rejection of off-beam signals. This can be easily computed for our uniformly spaced array, but for irregular arrays it is very difficult to determine the optimal weighting. Of course, neural networks could optimize weighting for such cases by "learning".

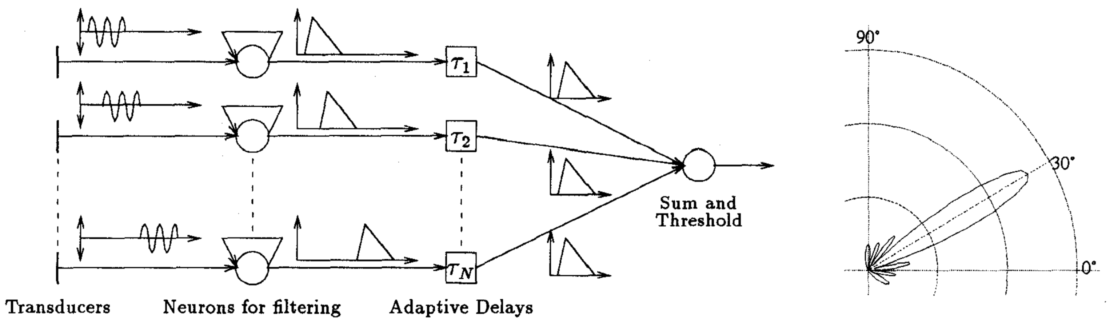

Figure 2: Neural network for one-dimensional beamforming (on the left) and beam pattern for a target at $\vartheta = 30°$ with uniform weighting and transducer spacing (on the right)

The set of synaptic delays after the adaption has been completed now corresponds to a certain "look" angle. We can use another, this time a simple neural network for evaluation that will form the beam after having learned the relationship between chosen reference sets of delays and angles. Fig. 2 shows a beam pattern for a target at $\vartheta = 30°$ which was recorded rotating the transducer array about its perpendicular. An 8-element transducer array was used in the experiment, with corresponding number of "biological" neurons and synapses with variable delays. The delays include a fixed delay potential that allows generation of both, positive and negative delay variations: $\tau_i' = t_0 \pm \tau_i$. The evaluation network was a standard three layer back-propagation neural network. The resolution at the output was 2°/neuron for 90 neurons in the output layer. The beam was formed after 150 iterations with an average error less than 0.01 percent. This linear transducer array can be easily extended to a two-dimensional beamformer using a net described above for each row and column.

There are other numerous applications of "biological" neurons in statistical pattern recognition. Their adaptive time-delays makes them perfect candidates for neural networks performing e.g. spatio-temporal conversions and lateral inhibition, or, in combination with their intrinsic memory capability, for recognition of time-warped signals, like human speech.

7. Summary

We have presented a model of a biologically inspired neuron and developed its mathematical description, including learning rules for all relevant parameters. Especially, the learnable synaptic time delay in conjunction with the neuron memory capability allows applications in statistical pattern analysis. As an example we showed the usefulness of the neuron model in time-domain beamforming of sonar signals.

The authors greatfully acknowledge useful discussions with Prof. R. Eckmiller (University of Bonn).

References

[1] A.L. Hodgkin and A.F. Huxley: *A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve*, J. Physiol. (1952) 117, pp. 500–544.

[2] C.E. Carr and M. Konishi: *Axonal delay lines for time measurement in the owl's brainstem*, Proc. of the Natl. Acad. of Science of the USA, Vol. 85, pp. 8311–8315, Nov. 1988.

[3] B. Widrow and M.E. Hoff, Jr.: *Adaptive Switching Circuits*, 1960 IRE Western Show and Convention Record, Part 4, pp. 96–104, Aug. 23, 1960.

[4] P.L. Feintuch: *An Adaptive Recursive LMS Filter*, Proc. IEEE, Vol. 64, No. 11, pp. 1622–1624, Nov. 1976.