Comparative Study of Single Biological Neuron model with a proposed Artificial Neuron model

Sashmita Panda (ysashmita786@gmail.com)
Indian Institute of Technology Kharagpur

Research Article

Keywords: Artificial neuron, Biological Neuron, Artificial neural network (ANN), Back Propation (BP) Algorithm, Spike response model.

Posted Date: May 20th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1668670/v1

Noname manuscript No.
(will be inserted by the editor)

Comparative Study of Single Biological Neuron model with a proposed Artificial Neuron model

Sashmita Panda

Received: date / Accepted: date

Abstract In an attempt to propose a closer model of a single biological neuron, various artificial neural models have been reported in the literature. Many articles are available on artificial neurons which tried to mimic a single biological neuron but those models are unable to produce the spiking patterns of a real biological neuron. Hence there is further scope to develop and investigate alternative improved spiking neural models which will better represent the activities of a biological neuron. With this motivation, much modification is done in an artificial neuron model in order to generate the spike patterns that a real biological neuron produces. The proposed modified single artificial neuron model exhibits the activities of a biological neuron. Modeling of a spiking bio-neuron is still an important exercise in view of possible applications of the underlying features in the areas of neuromorphic engineering, cognitive radio, and spiking neural networks.

Keywords Artificial neuron · Biological Neuron · Artificial neural network (ANN) · Back Propation (BP) Algorithm · Spike response model.

1 Introduction

The simplest dynamic unit of the human central nervous system is a biological neuron. A neuron generates a tiny electrical pulse or'spike' when it is activated beyond a certain threshold value [1]. Transfer of a spike from one neuron to another takes place at a fluidic region called synapse [2]. Neurons are highly specialised for generating and sending electrical signals in response to chemical and other inputs [3]. A neuron model which reproduces the spiking and bursting behavior of known types of cortical neurons is presented in [4] where the model efficiently captures the biological features of the Hodgkin-Huxley model and the computational efficiency of the integrate-and-fire model. Electrochemical impulses and neurotransmitters facilitate neuro-spike communication. Synaptic vesicles, filled with neurotransmitters, reside at an axonal terminal of a neuron. A small fluidic gap between an axon terminal of a pre-synaptic neuron and a spine of another post-synaptic neuron is known as the synaptic cleft. When a spike or an action potential arrives at the axon terminal of a presynaptic neuron, voltage-gated Ca^{2+} channels are activated, allowing an influx of Ca^{2+} ions into an axonal terminal. The vesicles are fused to the neuronal membrane with the help of Ca^{2+} ions, and eventually, neurotransmitters are released into the synaptic cleft. Representation of a stimulus by the neurons in our brain is a topic of much ongoing research and it is expressed that the timing of action potentials or spikes fired by these neurons carry information [5]. Understanding the fundamental concepts of communication among neurons, known as neuro-spike communication leads to reaching bio-inspired nanoscale paradigms. The author in the paper [6] proposed a realistic model to describe the neuro-spike communication. In a systematic way, Hodgkin and Huxley described that the ionic currents in the squid giant axon [7], can be realized by the variation in conductance of Na^+ and K^+, in the axon membrane. By performing many voltage-clamp experiments, a mathematical model consisting of the voltage and time-dependent properties of the Na^+ and K^+ conductances was developed

Sashmita Panda
G.S.Sanayal School of Telecommunications
Indian Institute of Technology Kharagpur
721302, India.
E-mail: ysashmita786@gmail.com

by them [8]. Through this work, a set of differential equations was developed, that described the action potential's ionic basis, termed as Hodgkin-Huxley model [9] later on.

The model [4] describes the spiking and bursting behavior of some cortical neurons. It has the dynamics of Hodgkin–Huxley model along with the integrate-and-fire neuron's computational efficiency. With this model, tens of thousands of spiking cortical neurons can be simulated in real-time. In paper [10], two translinear multipliers and first-order log domain low pass filters are used to create this model. By adjusting the input current as well as the biassed voltages and currents, the spiking patterns of this neuron model created by the chip could be seen. The generalised integrate-and-fire neuron model is another name for the Mihalas-Niebur neuron model [11]. This model shows many spiking and bursting behaviors, that can be seen in real biological neurons [10], [12]. Unlike other simplified Hodgkin-Huxley neuron models, it uses simple first-order differential equations to explain each of the state variables [13]. This neuron model has several advantages like it can be biophysically interpreted i.e. one can learn what might happen in biological neurons. Also, it provides an organized way of adding many other mechanisms and state variables. The spike response model [14], a version of the LIF model also explains how neurons generate action potentials. The spike response model is mostly dependent on filters, whereas integrate-and-fire versions are based on differential equations for the membrane potential. In this study, a neuron model with variable leaking resistor and bias current was discussed, which could precisely simulate the membrane voltage dynamics of a biological neuron cell. [15]. In [16], a simple model that can accurately reproduce the spiking behavior of neurons is investigated.

The spike train enables the creation of models that are far less complex than biophysical models while still capturing a significant portion of their complexity. A two-dimensional neuron model that provides a mathematical explanation of membrane potential evolution as well as an adaptation current [14], [17]. It's an expansion of the exponential LIF neuron that mimics the downswing of an action potential with a reset condition and the upswing with an exponential function. When the membrane potential approaches the threshold voltage, the exponential term causes the voltage to rapidly grow [18]. Subthreshold adaptation and spike-triggered adaptation are also determined by some of the parameters employed in this model. Many artificial models have been produced as a result of biological models. The author in the paper described the first neural network model [19], where the connections of the neurons are by directed weighted paths [20]. The McCullouch-Pitts neuron is a generalized model for providing the threshold in the axon with a nonlinear dynamics that is dependent on time [21]. The value of this model's binary unit is determined by the linear sum of weighted inputs from the other neurons in the network [22]. Many network models have been constructed by researchers to depict the network of artificial neurons. Artificial neural networks (ANN) are presently used in a variety of social, industrial, financial, and scientific settings. A functional approximation, filtering, direct modelling or system identification, inverse modelling or channel equalisation, control, classification, forecasting, pattern recognition, and optimization are some of the common applications in these domains. Rumelhart invented the conventional backpropagation (BP) technique of multilayer artificial neural network (MANN) as a supervised learning strategy [23], [24] which is a gradient descent local optimization technique.

2 Description of single biological neuron model

The block diagram of a single biological neuron model presented in Fig.1 comprises some inputs at the dendritic end. Each of j^{th} $(1 \leqslant j \leqslant J)$ input of i^{th} sequence, x_{ij} is multiplied with its associated gamma-distributed time-varying synaptic weights, $w_{ij}(t)$ to produce the required output $u_i(t)$. In this case, each x_{ij} represents either a '0' or '1,' and each input sequence consists of a J number of bits. The strength of the connection between two neurons varies in practice and is represented as synaptic weight. Either the height of the postsynaptic potential or slope of the postsynaptic current denotes the amplitude response and is determined by weights. In most of the single neuron models, long-term synaptic plasticity is employed, which assumes constant synaptic weights.

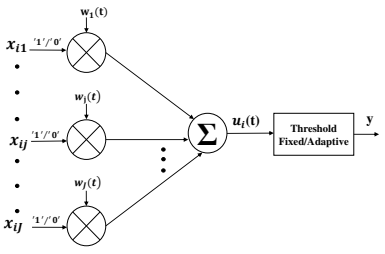

Fig. 1: Block diagram of a single biological neuron model

In Fig.1, $u_i(t)$, which denotes the membrane potential. If it crosses some threshold value, then action potentials in terms of spikes are generated. Here $u_i(t=0)$ is the initial membrane potential at the beginning of the i^{th} observation interval.

The membrane potential , $u_i(t)$ for i^{th} input sequence of biological neuron model is computed as

$$u_i(t) = \sum_{j=1}^{J} x_{ij} w_{ij}(t) + u_i(t=0) \tag{1}$$

where $1 \leqslant i \leqslant I$, I is total number of observation intervals, $1 \leqslant j \leqslant J$, J is total number of bits in a sequence, $x_{ij} \in x_1$, x_j, $....x_J$, The time-varying weight function can be mathematically expressed as

$$w_{ij}(t) = \frac{t}{t_p} h_p f_j e^{\left(1-\frac{t}{t_p}\right)} \tag{2}$$

f_j is the gamma distributed random variable with mean and variance chosen as 0.5 and 0.3 respectively. The symbol 't_p' represents the time when $w_{ij}(t)$ has attained maximum amplitude h_p . In the proposed model, AMPA receptor is cosidered which has typical value of $t_p = 1msec$ and $h_p = 1mV$ respectively. The output 'y' of the model can be expressed as;

$$y = \phi(u_i(t)) \tag{3}$$

$\phi(u_i(t)) = 1$, when $u_i(t) >= TH$ otherwise $\phi(u_i(t)) = 0$, when $u_i(t) < TH$, where 'TH' represents certain output threshold voltage.

3 Development of a Single artificial neuron model equivalent to biological neuron

A mathematical model of a single artificial neuron which is equivalent to a single biological neuron or a spiking neuron is shown in Fig.2. The

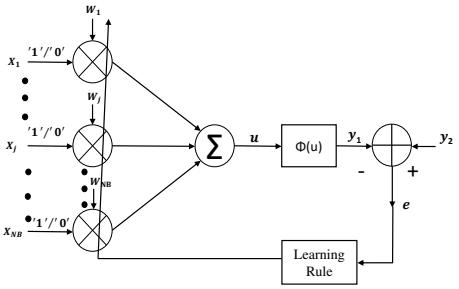

Fig. 2: Block diagram of a single artificial neuron model

model consists of the inputs X_1.....X_J.....X_{NB}. The input sequence is taken in terms of '0's and '1's. Each input is connected with their synaptic weights W_1....W_J.....W_{NB}. Now, the inputs multiplied with synaptic weights are given to summer or an adder which sums all the inputs as a linear combiner. Then, the output of an adder is applied to an activation function or a squashing function which limits the permission of the amplitude range of the output signal to some finite value. The output of the activation function is represented as $\phi(u)$.

3.1 Steps followed by the single Artificial neuron model

In this subsection, a sequence of steps is followed to explain the working principle of a single artificial neuron model which is equivalent to a biological neuron.

Step 1: Take the input sequence in terms of '0's and '1's. Here we provide the input sequence as X_1.....X_J.....X_{NB} where NB=20.
Step 2: Load the outputs obtained from the biological neuron model of vector size (1000x1) and stored it in an excel sheet.
Step 3: Assign the random value of weights in the artificial neuron model, then the input sequence will be multiplied with the random weights and then it will get summed by a summer using the following equation.

$$u = \sum_{j=1}^{NB} x_j w_j \tag{4}$$

Where 'u' is the output of the summing junction and $1 \leqslant u \leqslant NB$.
Step 4: The summing junction output is applied to suitable threshold function where a sigmoid function used as a threshold .

$$y = \phi(u) = \frac{A_H}{1 + e^{-B(u-TH)}} \tag{5}$$

Where y is the output of the single artificial neuron model. 'A_H' is the maximum amplitude of y, B is a scalar quantity and TH is the threshold value after which the output y decreases.
Step 5: Now we can give an input of vector size 1000x20 to the artificial neural model, then in the similar ways explained in step 4 using Eq.5 we will find the output of the artificial neural model.
Step 6: To train the artificial neuron model, we will give the output of the single spiking neuron having vector size 1000x1 which is explained in step 2.

3.2 Training of Atificial neuron model

An artificial neuron model can be trained by the following steps;
Step 1: After providing the ouput of a biological neuron, the model will compute the error function as;

$$e = f(y_2, y_1) = \frac{1}{2}(y_2 - y_1)^2 \tag{6}$$

Where y_2 represents the output of the target or desired output, here y_2 denotes the output of the single biological neuron and y_1 is the actual output of the single artificial neuron model.
Step 2: Find the error between the output of a biological and artificial neuron model. Now by using a learning rule, this error will be feedback to the artificial neuron model in order to update the weights in accordance to get the actual output which will be nearly equal to the desired biological neural model output.
Step 3: The weights can be updated by using the following learning rule;

$$\Delta W = -\eta \frac{\delta e}{\delta w} \tag{7}$$

Where η is the learning rate parameter and e is the error function of the model and its value lies between 0 to 1.

3.3 Development of updated weights for the single artificial neuron model

In this subsection the learning rule of updated weights of the proposed AN model is derived. The output of the summing junction is expressed in Eq.(4) ;

$u = \sum_{j=1}^{NB} x_J w_j$

Now, the output of the single artificial neuron model is represented in Eq.(5) ;
$y = \phi(u) = \frac{A_H}{1+e^{-B(u-TH)}}$

Error function of the model is given in Eq.(6);
$e = f(y_2, y_1) = \frac{1}{2}(y_2 - y_1)^2$

Now updating the weights using the learning rule defined in Eq.(7)

$\Delta W = -\eta \frac{\delta e}{\delta w}$

By using the chain rule Eq.(7) can be written as follows;
$\Delta W = -\eta \frac{\delta e}{\delta w}$
$= \eta(\frac{\delta e}{\delta y_1})(\frac{\delta y_1}{\delta u})(\frac{\delta u}{\delta w_j})$

We know;

$$\frac{\delta e}{\delta y_1} = \frac{1}{2}(y_2 - y_1)^2 = (y_2 - y_1) \tag{8}$$

$$\begin{aligned}
\frac{\delta y1}{\delta u} &= \frac{\delta}{\delta u}(\frac{A_H}{1+e^{-B(u-TH)}}) \\
&= \frac{A_H B(u-TH)e^{-B(u-TH)}}{(1+e^{-B(u-TH)})^2} \\
&= A_H B(u-TH)\frac{1+e^{B(u-TH)}-1}{(1+e^{-B(u-TH)})^2} \\
&= A_H B(u-TH)(\frac{1+e^{-B(u-TH)}}{(1+e^{-B(u-TH)})^2} - \frac{1}{(1+e^{-B(u-TH)})^2}) \\
&= A_H B(u-TH)(\frac{1}{(1+e^{-B(u-TH)})} - \frac{1}{(1+e^{-B(u-TH)})^2}) \\
&= A_H B(u-TH)(y_1 - y_2^2) \\
&= A_H B(u-TH)y_1(1-y_1)
\end{aligned} \tag{9}$$

Where $y = \phi(u) = \frac{A_H}{1+e^{-B(u-TH)}}$

From the model,

$$\frac{\delta u}{\delta w_j} = x_j \tag{10}$$

Hence,

$$\begin{aligned}
\Delta w_j &= -\eta \frac{\delta e}{\delta w_j} \\
&= \eta(\frac{\delta e}{\delta y_1})(\frac{\delta y_1}{\delta u})(\frac{\delta u}{\delta w_j}) \\
&= \eta(y_1 - y_2)A_H B(u-TH)y_1(1-y_1)x_j
\end{aligned} \tag{11}$$

If $e = y_1 - y_2$
and $\delta = ey_1(1-y_1)$
Finally the change of weights in the $j_t h$ branch is derived as;

$$\Delta w_j = \eta \delta A_H B(u-TH)x_j \tag{12}$$

Now, using the learning rule,

$$W_{New} = W_{old} + \Delta W \tag{13}$$

Where W_{New} is the new weight, W_{old} is the older weight and ΔW is the update weight for the model. Substituting the value of Δw_j in the $j_t h$ branch the updated weight can be given as;

$$\begin{aligned}
W_j &= w_j + \Delta w_j \\
W_j &= w_j + \delta A_H B(u-TH)x_j
\end{aligned} \tag{14}$$

Where W_j is the new weight and w_j is the old weight for the artificial neuron model.

Step 4: Now weight of the model which is determined in step 3 is applied for finding new weights.

Step 5: The training procedure will continue through updating the weights till the error function e will be minimum i.e. the output of both single artificial neuron and biological neuron model will be approximately equal to ϵ ($e \leqslant \epsilon$ assume $\epsilon = 0.0001$).

4 Results and Discussions

In this section, the results of output between a single biological neuron (BN) and an artificial neuron (AN) model is analysed and contributions of the study has been highlighted. For simulating the proposed models a total of 50 input sequences or patterns have been applied to both BN and AN models. By applying a threshold voltage of 15mV to the BN model, specific spike patterns at the output have been observed. The typical numerical values used for the simulation study of the proposed model are listed in Table 1.

Table 1: Numerical values for important parameters used in simulation study

Symbols	Description	Typical values
t_p	The time instant when $w_{ij}(t)$ has maximum amplitude	1msec
h_p	The maximum amplitude of $w_{ij}(t)$ which occurs at $t = t_p$	1mv
μ	Mean of Gamma distributed random variable	0.5
σ	Variance of Gamma distributed random variable	0.3
T_s	Sampling Time	0.5msec

For experimental based simulation study, input random binary sequence each of length J (J=20) is considered for a time interval of 5 msec. Each input bit is then multiplied with the time-varying synaptic weights obtained at an interval of T_S msec. After each successive T_S msec time interval, all the twenty partial products are added at the summer node to produce an output. After a time interval of 5 msec, the cumulative sum of outputs due to all ten weights are added to produce a membrane potential. When the magnitude of this potential crosses some predefined threshold value, then the neuron gets fired, and a spike is produced in the final output.

It is observed from Fig.3 that after applying 100 input patterns to a BN, it generates 95 spikes where as in AN model it gives 92 spikes shown in Fig.4. The mean square error reduces to 0.063

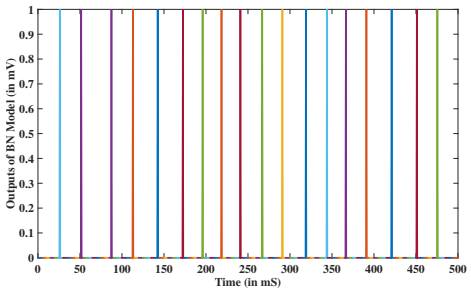

Fig. 3: Plot of output for a single biological neuron model for 100 input patterns

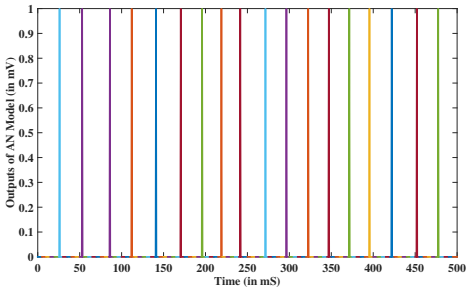

Fig. 4: Plot of output for a single artificial neuron model for 100 input patterns

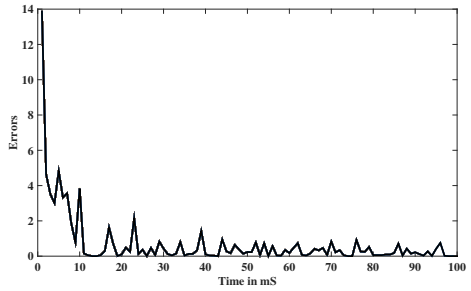

Fig. 5: Plot of error for artificial neuron model for 100 Input Patterns

for an artificial neuron model presented in Fig.5 which represents an artificial neuron can mimics as a biological neuron. Similarly when the number

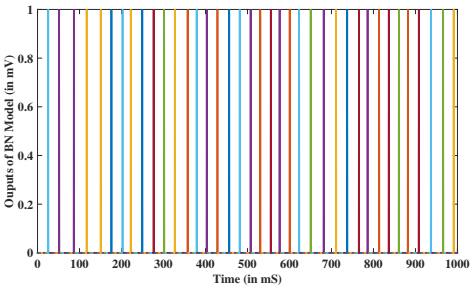

Fig. 6: Plot of output for a single biological neuron model for 200 input patterns

of input patterns increases to 200, then the spikes

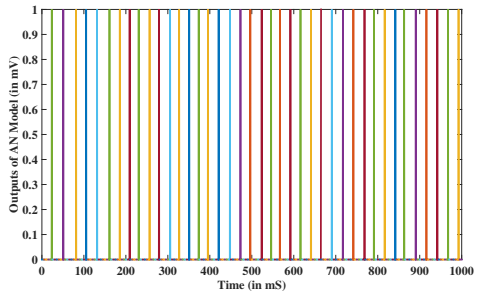

Fig. 7: Plot of output for a single artificial neuron model for 200 input patterns

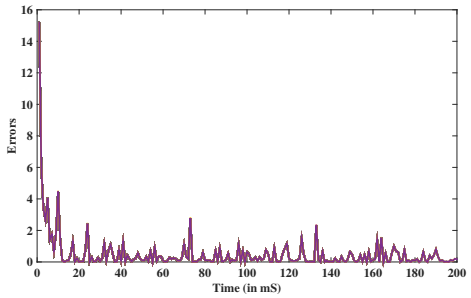

Fig. 8: Plot of error for an artificial neuron model for 200 Input Patterns

generated by the BN model is 38 shown in Fig.6 where as in AN model it produces 40 number of spikes represented in Fig.7. From Fig.8, it is seen that the error in AN model reduces to 0.066.

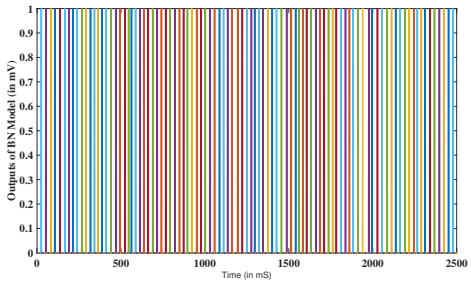

Fig. 9: Plot of output for a single biological neuron model for 500 input patterns

Now after increasing the number of input patterns to 500, both BN model and AN model gives 18 spikes which is shown in Fig.9 and 10 respectively.

Table 2: Comparison of number of spikes generated in BN and AN Model with different input patterns

Number of Input Patterns	Number of Spikes in BN Model	Number of Spikes in AN Model	MSE
100	95	92	0.063
200	38	40	0.066
500	18	18	0.0047

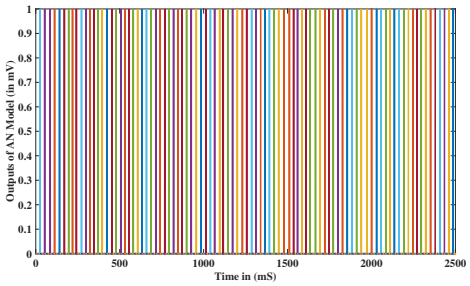

Fig. 10: Plot of output for a single artificial neuron model for 500 input patterns

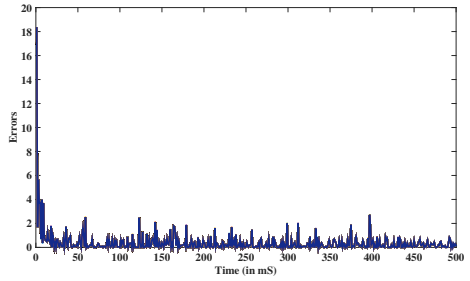

Fig. 11: Plot of error for an artificial neuron model for 500 Input Patterns

It is observed that the mean square error decreases to 0.0047 for an AN model shown in Fig.11 and hence an artificial neuron functions same as an artificial neuron. It can be illustrated that with increase in number of input patterns to both BN and AN model the error decreases and the proposed AN model shows the realistic characteristics of a biological neuron. Table 2 illustrates that when the number of input patterns increases, the artificial neuron exhibits a spiking behaviour that is quite comparable to that of a true biological neuron.

5 Conclusion

An improved biological neuron model is proposed and its output spike patterns are presented in this paper. By applying the same input patterns the response of the biological neuron model is compared with the proposed AN model. The comparison reveals that the mean square error during training phase of an artificial neural model decreases as the number of input patterns or sequences increases, implying that an artificial neuron model acts more like a biological neuron. The simulation findings also show that the output of the artificial neuron model is almost similar to that of a biological neuron.

References

1. S. Schliebs and N. Kasabov, "Evolving spiking neural network—a survey," *Evolving Systems*, vol. 4, no. 2, pp. 87–98, 2013.
2. E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, and A. Hudspeth, *Principles of neural science*, 5th ed. McGraw-hill New York, 2013.
3. P. Dayan and L. F. Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.
4. E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.
5. R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing–based decisions," *Nature neuroscience*, vol. 9, no. 3, pp. 420–428, 2006.
6. H. Ramezani and O. B. Akan, "A communication theoretical modeling of axonal propagation in hippocampal pyramidal neurons," *IEEE transactions on nanobioscience*, vol. 16, no. 4, pp. 248–256, 2017.
7. L. Abbott and T. B. Kepler, "Model neurons: from hodgkin-huxley to hopfield," in *Statistical mechanics of neural networks*. Springer, 1990, pp. 5–18.
8. S. Koslow and S. Subramaniam, "Hodgkin-huxley models," in *Electrophysiological Models In: Databasing the Brain: From Data to Knowledge*. Wiley, 2004.
9. A. L. Hodgkin and A. F. Huxley, "Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo," *The Journal of physiology*, vol. 116, no. 4, pp. 449–472, 1952.
10. A. Van Schaik, C. Jin, A. McEwan, T. J. Hamilton, S. Mihalas, and E. Niebur, "A log-domain implementation of the mihalas-niebur neuron model," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 4249–4252.
11. V. Varghese, J. L. Molin, C. Brandli, S. Chen, and R. E. Cummings, "Dynamically reconfigurable silicon array of generalized integrate-and-fire neurons," in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2015, pp. 1–4.
12. R. Jolivet, T. J. Lewis, and W. Gerstner, "Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy," *Journal of neurophysiology*, vol. 92, no. 2, pp. 959–976, 2004.
13. G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, p. 73, 2011.
14. W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
15. Z. Wang, L. Guo, and M. Adjouadi, "A generalized leaky integrate-and-fire neuron model with fast implementation method," *International journal of neural systems*, vol. 24, no. 05, p. 1440004, 2014.
16. C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz *et al.*, "Generalized leaky integrate-and-fire models classify multiple neuron types," *Nature communications*, vol. 9, no. 1, pp. 1–15, 2018.
17. L. Hertäg, J. Hass, T. Golovko, and D. Durstewitz, "An approximation to the adaptive exponential integrate-and-fire neuron model allows fast and predictive fitting to physiological data," *Frontiers in computational neuroscience*, vol. 6, p. 62, 2012.
18. R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 2005.
19. S. Chakraverty, D. M. Sahoo, and N. R. Mahato, "Mcculloch–pitts neural network model," in *Concepts of Soft Computing*. Springer, 2019, pp. 167–173.
20. B. Mehlig, "Artificial neural networks," *arXiv preprint arXiv:1901.05639*, 2019.
21. H. Szu and G. Rogers, "Generalized mcculloch-pitts neuron model with threshold dynamics," in *Int'l Joint Conf. Neural Networks*, 1992.
22. Y. Takefuji and K. Lee, "An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics," *Biological Cybernetics*, vol. 64, no. 5, pp. 353–356, 1991.
23. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.
24. B. M. Wilamowski, Y. Chen, and A. Malinowski, "Efficient algorithm for training neural networks with one hidden layer," in *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, vol. 3. IEEE, 1999, pp. 1725–1728.