Third Generation Neural Networks: Spiking Neural Networks

Samanwoy Ghosh-Dastidar[1] and Hojjat Adeli[2]

[1] Department of Biomedical Engineering, The Ohio State University
[2] Abba G. Lichtenstein Professor, Departments of Biomedical Engineering, Biomedical Informatics, Civil and Environmental Engineering and Geodetic Science, Electrical and Computer Engineering, and Neuroscience, The Ohio State University, 470 Hitchcock Hall, 2070 Neil Avenue, Columbus, Ohio 43210

Abstract. Artificial Neural Networks (ANNs) are based on highly simplified brain dynamics and have been used as powerful computational tools to solve complex pattern recognition, function estimation, and classification problems. Throughout their development, ANNs have been evolving towards more powerful and more biologically realistic models. In the last decade, the *third generation* Spiking Neural Networks (SNNs) have been developed which comprise of *spiking* neurons. Information transfer in these neurons models the information transfer in biological neurons, i.e., via the precise timing of spikes or a sequence of spikes. Addition of the temporal dimension for information encoding in SNNs yields new insight into the dynamics of the human brain and has the potential to result in compact representations of large neural networks. As such, SNNs have great potential for solving complicated time-dependent pattern recognition problems defined by time series because of their inherent dynamic representation. This article presents an overview of the development of spiking neurons and SNNs within the context of feedforward networks, and provides insight into their potential for becoming the next generation neural networks.

1 Introduction

Artificial neural networks (ANNs), inspired by the structure and function of the human brain, have been used as powerful computational tools to solve complex pattern recognition, function estimation, and classification problems not amenable to other analytical tools [1,2,3,4,5,6]. Over time, ANNs have evolved into more powerful and more biologically realistic models [7,8,9,10,11]. Improved understanding of the brain and its modes of information processing has led to the development of networks such as feedforward neural networks [12,13], recurrent networks [14,15], radial basis function neural networks [16,17,18], self-organizing maps, modular neural networks, and dynamic neural networks [19,20,21].

Feedforward ANNs are the most common and utilize various mechanisms for a forward transfer of information across the neural network starting from the input node to the output node. The popularity of feedforward ANNs stems from their

W. Yu and E.N. Sanchez (Eds.): Advances in Computational Intell., AISC 61, pp. 167–178.
springerlink.com

conceptual simplicity and the fact that the primary (but not the only) mode of information transfer in both real and artificial neural networks is feedforward in nature [22,23,24,25]. In fact, other modes of information transfer often involve or are based on feedforward mechanisms to some degree.

Although ANNs have gone through various stages of evolution, until recently, there had not been many attempts to categorize generations of neural networks. This is a particularly difficult task because ANN developments have branched out in many directions and it would not be accurate to label one development as more advanced than another. In addition, such a categorization is subjective and dependent on what is considered advancement. However, in the authors' opinion, if a single clearly identifiable, major conceptual advancement were to be isolated, it would be the development of the mathematically-defined activation or transfer function as the information processing mechanism of the artificial neuron. Due to the importance of the activation function in feedforward ANNs, the discussion on generations of ANN in this article is restricted to the evolution of the artificial neuron from the perspective of feedforward neural networks.

2 Information Encoding and Evolution of Spiking Neurons

Studies of the cortical pyramidal neurons have shown that the timing of individual spikes as a mode of encoding information is very important in biological neural networks [26,27,28]. A presynaptic neuron communicates with a postsynaptic neuron via trains of spikes or action potentials. Biological spikes have a fixed morphology and amplitude [29]. The transmitted information is usually encoded in the frequency of spiking (*rate encoding*) and/or in the timing of the spikes (*pulse encoding*). Pulse encoding is more powerful than rate encoding in terms of the wide range of information that may be encoded by the same number of neurons [30]. In fact, rate encoding can be considered to be a special case of pulse encoding. If the spike timings are known, the average firing rate can be computed.

The early *first generation* neurons developed in the 1940s and 1950s did not involve any encoding of the temporal aspect of information processing. These neurons acted as simple integrate-and-fire units which fired *if* the *internal state* (defined as the weighted sum of inputs to each neuron) reached a threshold. It did not matter when the threshold was exceeded. Translating this assumption to a biological perspective, it implied that all inputs to the neuron were synchronous, i.e. contributed to the internal state at exactly the same time and therefore, could be directly summed. However, unlike biological neurons, the magnitude of the input was allowed to contribute to the internal state. Arguably, this may have represented a primitive form of rate encoding in the sense that a larger input (representing a higher firing rate of the presynaptic neuron) may cause the postsynaptic neuron to reach the threshold. For the sake of simplicity, the mathematical abstraction avoided the modeling of the actual spike train and the input from the presynaptic neuron approximated the average firing rate of the presynaptic neuron. The *fire* state for the postsynaptic neuron was a binary-valued output which returned a value of 1 if the neuron fired and 0 otherwise. This implied that the output from the postsynaptic neuron was not based on rate encoding.

The *second generation* neurons developed from the 1950s to 1990s were also based loosely on rate encoding and defined the internal state in a similar manner. However, they used a mathematically-defined activation function, often a smooth sigmoid or radial basis function, instead of a fixed threshold value, for output determination [27]. In the postsynaptic neuron, the activation function was used to transform the input into a proportionate output which approximated the average firing rate of the postsynaptic neuron. With this development, it became possible for the output to be real-valued. In contrast to the first generation neurons, even the postsynaptic neuron could generate rate encoded information. This model gained widespread acceptance as processing elements in feedforward ANNs because it was compatible with the Rumelhart's widely-used backpropagation (BP) training algorithm [31] which required a continuous and differentiable activation function. The model was significantly more powerful than the one based on first generation neurons and could solve complex pattern recognition problems (the most notable of which in the 1950s was the XOR problem) [32,33,34,35,36,37,38,39,40]. However, the computational power of the neuron still did not reach its full potential because the temporal information about individual spikes was not represented.

In the last decade, *spiking* neurons have been developed and adapted for ANNs to overcome this shortcoming by communicating via the precise timing of spikes or a sequence of spikes. In the literature, spiking neurons have been referred to as *third generation* neurons. Similar to the first generation neurons, a spiking neuron acts as an integrate-and-fire unit and has an *all or none* response. The spiking neuron, however, has an inherent dynamic nature characterized by an internal state which changes with time and each postsynaptic neuron fires an action potential or spike at the time instance its internal state exceeds the neuron threshold. Similar to biological neurons, the magnitude of the spikes (input or output) contains no information. Rather, all information is encoded in the timing of the spikes as discussed in the next section. Even though spiking neurons are discussed within the context of feedforward networks in this article, it must be noted that their application is not limited to only feedforward networks. Spiking neurons have also been used with ANNs similar in concept to Radial Basis Function Neural Networks and Self Organizing Maps with applications in unsupervised clustering and pattern classification.

3 Mechanism of Spike Generation in Spiking Neurons

In general, action potentials or spikes from various presynaptic neurons reach a postsynaptic neuron at various times and induce *postsynaptic potentials* (PSPs). The PSP represents the internal state of the postsynaptic neuron induced in response to the presynaptic spike and is affected by synaptic characteristics such as travel time or delay through the synapse, strength of the synaptic connection, and other biological factors some of which are unknown. Multiple neurons, each with multiple spikes, induce multiple PSPs over time. The postsynaptic neuron acts as a temporal integrator of PSPs induced by all presynaptic neurons and fires when the integrated internal state crosses a threshold.

The effects of various presynaptic spike trains on the postsynaptic potential and the postsynaptic output spike train are illustrated in Fig. 1. In the first two cases, Figs. 1(a) and 1(b), each spike train is considered individually whereas in the third case, Fig. 1(c), the combined effect of the two spike trains shown in Figs. 1(a) and 1(b) is illustrated. Each spike train consists of a sequence of three spikes. The first and the third spike in the presynaptic spike trains occur at the same time instant. The timing of the second spike, however, is different in the two cases. From the perspective of rate encoding, both these spike trains are identical, i.e. the average firing frequency is identical (3 per given time period). This highlights the approximate nature and lower computational power of rate encoding which makes it impossible to differentiate between the two cases in Figs. 1(a) and 1(b).

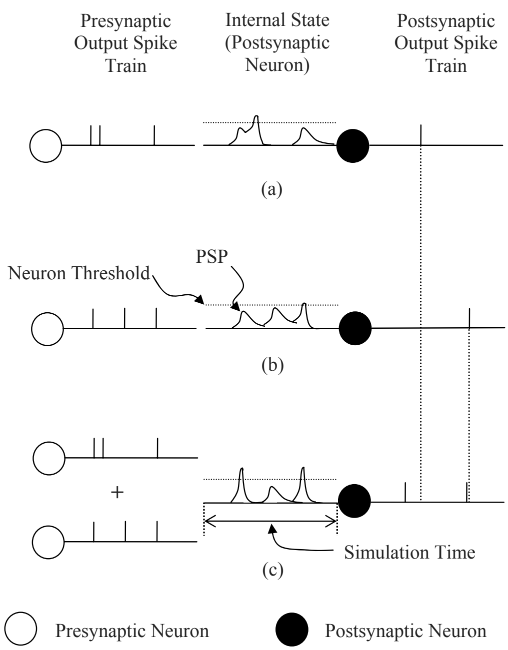

Fig. 1. The effect of various presynaptic spike trains on the postsynaptic potential and the postsynaptic output spike train. (a) and (b) show two spike trains and their individual effects on the postsynaptic neuron, and (c) shows the combined effect of the aforementioned two spike trains on the postsynaptic neuron.

In contrast, the timing of the spikes is considered in pulse encoding. Each spike in the spike train induces a PSP in the postsynaptic neuron at different times. The PSPs are temporally integrated to compute the internal state of the postsynaptic neuron over time as shown in Figs. 1(a) and 1(b). The internal states in the two cases are entirely different and their values exceed the neuronal threshold at different times. This leads to different output spike times from the postsynaptic neuron. An additional source of variation in the PSP is the dependence of the internal state of the postsynaptic neuron on the time of its own output spike. The internal state of a postsynaptic neuron in response to a presynaptic spike is shown in Fig. 2. Had the threshold not been exceeded the internal state of neuron in Fig. 2 would have been represented by the dashed line. The solid line in Fig. 2 shows the internal state of neuron when the threshold is exceeded. Immediately after the firing of an output spike, the internal state of the neuron exhibits a sharp decrease as a result of various biological processes. This phase is known as *repolarization* (Fig. 2) [29,41].

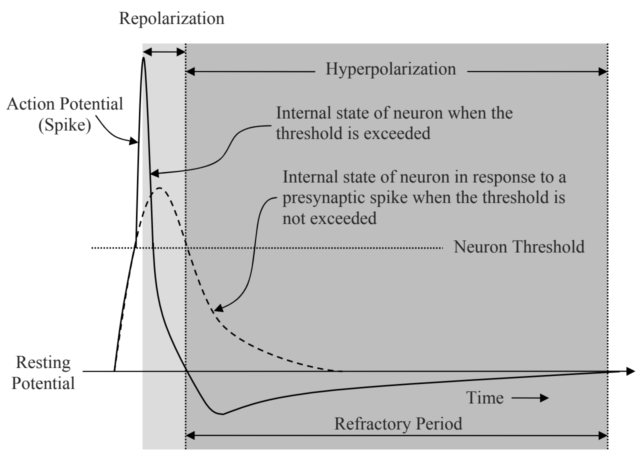

Fig. 2. The internal state of a postsynaptic neuron in response to a presynaptic spike (not shown in the figure) showing the action potential, and repolarization and hyperpolarization phases

In the third case shown in Fig. 1(c), both presynaptic spike trains are input simultaneously to the postsynaptic neuron by two presynaptic neurons. In this case, the internal state of the postsynaptic neuron is not simply the sum of the internal states in the first two cases. An additional factor needs to be considered for the postsynaptic neuron. After the firing of a spike and the resultant sharp decrease in the internal state of the neuron, the internal state is kept at a value lower than the resting potential of the neuron (Fig. 2) by various biological processes that are beyond the scope of this discussion. This phase is known as *hyperpolarization* and shown in Fig. 2 [29,41]. As a result, it becomes difficult for the neuron to reach the threshold and fire again for a certain period of time, known as *refractory period*

(Fig. 2). The internal state of the postsynaptic neuron is obtained by the algebraic summation of the internal states in the first two cases and modified during the repolarization and hyperpolarization phases. The three processes of summation, repolarization, and hyperpolarization lead to the postsynaptic neuron firing output spikes at times different than those for the first two cases. In Fig. 1, the first spike in the third case occurs earlier than the first spike in the first case because the postsynaptic neuron in the third case exceeds the threshold value earlier. The three cases shown in Fig. 1 highlight the importance of the timing of spikes in the presynaptic spike train for encoding information.

4 Models of Spiking Neurons

Spiking neurons can be modeled in many different ways. Many detailed mathematical models have been developed to quantitatively characterize neuronal behavior based on detailed modeling of the neuronal membrane potential and ion channel conductances [42,43,44,45,46]. Networks of such neuronal models have proved to be very valuable in studying the behavior of biological neural networks, neuronal learning mechanisms such as long-term potentiation and depotentiation, and neurotransmitter-based signaling [47]. However, the level of detail, although ideal for reproducing electrophysiological responses accurately, increases the complexity of the model making them difficult to analyze [48,49]. This complexity also imposes a significant computational burden for large neural network based classification or pattern recognition tasks that employ BP as the learning mechanism.

Another obstacle to the use of these detailed models in feedforward ANNs is imposed by the dynamics of the BP algorithm which usually requires a single activation function (representing changes in membrane potential) for backpropagating the error term through the neuron. The detailed models are usually based on multiple differential equations that capture the behavior of different ion channels and currents that affect the membrane potential. It remains to be seen if error backpropagation is even mathematically possible in the face of such complexity.

Spike response models are phenomenological models that are simpler than the detailed models and offer a compromise between computational burden and electrophysiological detail [50,43,51,52,53,54]. As a result, spike response models are preferred for systemic studies of memory, neural coding, and network dynamics. Bohte et al. [55] employed such a spike response model (originally presented by Gerstner [51]) to demonstrate that BP-based learning is possible in such a network. Other spike response models may also be adapted provided that their activation function can be adapted for error backpropagation. In principle, detailed biophysical models and more complicated phenomenological models appear to be better suited to SNNs that are similar in concept to Radial Basis Function Neural Networks and Self Organizing Maps and are not restricted by the requirements and computational burden of BP-based learning.

5 Spiking Neural Networks (SNNs)

SNNs are, simply, networks of spiking neurons. The SNN architecture, as shown in Fig. 3(a) is similar to that of a traditional feedforward ANN. The network is assumed to be fully connected i.e. a neuron in any layer is connected to all neurons in the preceding layer. However, unlike feedforward ANNs where two neurons are connected by one synapse only, the connection between two SNN neurons is modeled by multiple (K) synapses as shown in Fig. 3(b) [56,55]. The number K is constant for any two neurons and each synapse has a weight and a delay associated with it.

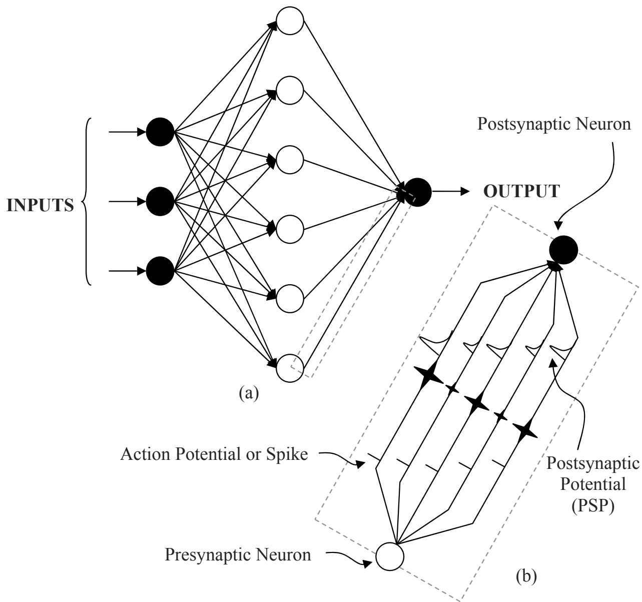

Fig. 3. (a) Spiking neural network architecture; (b) multiple synapses connecting a presynaptic neuron to a postsynaptic neuron

Assuming that presynaptic neuron fires a spike at time t, the k^{th} synapse transmits that spike to the postsynaptic neuron at time $t + d_k$ where d_k is the delay associated with the k^{th} synapse. This architecture enables a presynaptic neuron to affect a postsynaptic neuron by inducing PSPs of varying magnitudes at various time instants. The magnified connection in Fig. 3(b) displays the temporal sequence of spikes (short vertical lines) from the presynaptic neuron, the synaptic weights (proportionate to the size of the star shaped units in the center), and the resulting PSPs (proportionate to the size of the waveform). The modeling of synapses is identical for all neurons, and the k^{th} synapse between any two neurons has the same delay d_k. For the sake of simplicity, neurons in Bohte et al.'s model were restricted to the emission of a single spike. Recently, networks based on spiking neurons that convey information via spike trains (multiple spikes) have also been presented [57,58].

Similar to traditional ANNs, SNN architecture consists of an input layer, a hidden layer, and an output layer (Fig. 3). The number of neurons in the hidden layer is usually selected by trial and error. Since the SNN model is based on spike times, inputs to the SNN have to be preprocessed to convert the continuous real-valued input *features* (or classification variables) into discrete spike times. As a result, the number of original features is converted into a new number of features for input to the SNN. This is known as *input encoding*. Similarly, the number of neurons in the output layer depends on the *output encoding* scheme selected for the classification problem. In SNNs the inputs and outputs can be encoded in a variety of ways. This variety, however, is limited by the assumption of only one spike per neuron.

Until recently, the lack of a continuous and differentiable activation function relating the internal state of the neuron to the output spike times made spiking neurons incompatible with the error backpropagation required for supervised learning. Bohte et al. [55] presented a BP learning algorithm for SNN, dubbed *SpikeProp* similar in concept to the BP algorithm developed for traditional neural networks [31]. Subsequently, SNN was used with various training algorithms such as backpropagation with momentum [59,60], *QuickProp* [59,60], resilient propagation (*RProp*) [60], and Levenberg-Marquardt BP [61] to improve network training performance [11]. QuickProp is a faster converging variant of the original BP learning rule [31] that searches for the global error minimum by approximating the error surface on the basis of local changes in the gradient and weights [62]. RProp is also a fast variant of the BP algorithm where the weights are adjusted based on the direction of the gradient rather than the magnitude. This strategy is specially effective or *resilient* when the error surface is highly uneven and the gradient is not an accurate predictor of the learning rate [63]. Compared with SpikeProp, the aforementioned improved algorithms reportedly provide faster convergence by approximately 600% [11]. Some preliminary research has also been reported regarding the adjustment of other SNN parameters such as neuron threshold, synaptic delays, and the time decay constant defining the shape of the PSP [64]. Recently, new learning algorithms have also been presented for training SNN models that convey information in the form of spike trains [57,58] instead of single spikes.

Computationally, SNN training is usually at least two orders of magnitude more intensive than the traditional ANNs for two reasons [11]. First, multiple weights have to be computed for multiple synapses connecting a presynaptic neuron to a postsynaptic neuron. Second, the internal state of each neuron has to be computed for a continuous duration of time, called *simulation time* (see Fig. 1), to obtain the output spiking times. The time resolution, called *time step*, employed for this computation along with the simulation time and the number of convergence epochs are key factors that affect the actual computation time (real-time) required to train the network. Another difficulty with SNN training is the highly uneven nature of the error surface that can wreak havoc with the gradient descent-based training algorithms. Slight changes in the synaptic weights result in proportionate changes in the postsynaptic potential. But slight changes in the postsynaptic potential may result in disproportionate changes in the output spike times of the postsynaptic neuron. To overcome this training difficulty various heuristic rules are used to limit the changes of the synaptic weights [65,57,11,58].

6 Concluding Remarks

SNNs have been used for complicated time-dependent pattern recognition problems defined by time series because of their inherent dynamic representation. Further, SNNs have been shown theoretically to have the ability to approximate any continuous function [66]. Addition of the temporal dimension for information encoding has the potential to result in compact representations of large neural networks, another advantage for SNNs. However, their widespread acceptance and application is currently limited by the excessive computing times required for training [11]. It may be expected that this will change in the near future for two reasons. First, technology is advancing at a rapid rate and the computational limitations outlined in this manuscript may not remain as limiting. Second, the field of SNNs is of great research interest and developing rapidly as well.

From the perspective of SNN development, in the opinion of the authors, an adaptive adjustment of the number of synapses [64] needs to be investigated with the goal of reducing the number of weights and consequently computational effort, without compromising the classification accuracy. An additional source of computational effort is the input encoding that increases the number of features many times. New methods of input encoding that do not increase the number of features should be explored. Currently, there is great interest in the development of efficient and accurate learning algorithms for feedforward as well as other networks. Novel combinations of these strategies along with improved understanding of biological information processing will contribute significantly to the development of SNNs as the next generation neural networks.

References

1. Adeli, H., Hung, S.L.: Machine Learning - Neural Networks, Genetic Algorithms, and Fuzzy Sets. John Wiley and Sons, New York (1995)
2. Adeli, H., Park, H.S.: Neurocomputing for Design Automation. CRC Press, Boca Raton (1998)
3. Adeli, H., Karim, A.: Fuzzy-wavelet RBFNN model for freeway incident detection. Journal of Transportation Engineering 126(6), 464–471 (2000)
4. Adeli, H.: Neural networks in civil engineering: 1989-2000. Computer-Aided Civil and Infrastructure Engineering 16(2), 126–142 (2001)
5. Ghosh-Dastidar, S., Adeli, H.: Wavelet-clustering-neural network model for freeway incident detection. Computer-Aided Civil and Infrastructure Engineering 18(5), 325–338 (2003)
6. Adeli, H., Karim, A.: Wavelets in Intelligent Transportation Systems. John Wiley and Sons, Hoboken (2005)
7. Adeli, H., Ghosh-Dastidar, S., Dadmehr, N.: Alzheimer's disease and models of computation: Imaging, classification, and neural models. Journal of Alzheimer's Disease 7(3), 187–199 (2005a)

8. Adeli, H., Ghosh-Dastidar, S., Dadmehr, N.: Alzheimer's disease: Models of computation and analysis of EEGs. Clinical EEG and Neuroscience 36(3), 131–140 (2005b)
9. Adeli, H., Jiang, X.: Dynamic fuzzy wavelet neural network model for structural system identification. Journal of Structural Engineering 132(1), 102–111 (2006)
10. Ghosh-Dastidar, S., Adeli, H.: Neural network-wavelet microsimulation model for delay and queue length estimation at freeway work zones. Journal of Transportation Engineering 132(4), 331–341 (2006)
11. Ghosh-Dastidar, S., Adeli, H.: Improved spiking neural networks for EEG classification and epilepsy and seizure detection. Integrated Computer-Aided Engineering 14(3), 187–212 (2007)
12. Adeli, H., Hung, S.L.: A concurrent adaptive conjugate gradient learning algorithm on MIMD machines. Journal of Supercomputer Applications 7(2), 155–166 (1993)
13. Ghosh-Dastidar, S., Adeli, H., Dadmehr, N.: Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection. IEEE Transactions on Biomedical Engineering 54(9), 1545–1551 (2007)
14. Adeli, H., Park, H.S.: Counter propagation neural network in structural engineering. Journal of Structural Engineering 121(8), 1205–1212 (1995a)
15. Panakkat, A., Adeli, H.: Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. Computer-Aided Civil and Infrastructure Engineering 24(4), 280–292 (2009)
16. Karim, A., Adeli, H.: Comparison of the fuzzy-wavelet RBFNN freeway incident detection model with the california algorithm. Journal of Transportation Engineering 128(1), 21–30 (2002)
17. Karim, A., Adeli, H.: Radial basis function neural network for work zone capacity and queue estimation. Journal of Transportation Engineering 129(5), 494–503 (2003)
18. Ghosh-Dastidar, S., Adeli, H., Dadmehr, N.: Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. IEEE Transactions on Biomedical Engineering 55(2), 512–518 (2008)
19. Jiang, X., Adeli, H.: Dynamic wavelet neural network model for traffic flow forecasting. Journal of Transportation Engineering 131(10), 771–779 (2005)
20. Jiang, X., Adeli, H.: Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings. International Journal for Numerical Methods in Engineering 71(5), 606–629 (2007)
21. Jiang, X., Adeli, H.: Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular highrise building structures. International Journal for Numerical Methods in Engineering 74(7), 1045–1066 (2008)
22. Adeli, H., Hung, S.L.: An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks. Applied Mathematics and Computation 62(1), 81–102 (1994)
23. Adeli, H., Park, H.S.: Optimization of space structures by neural dynamics. Neural Networks, Vol 8(5), 769–781 (1995b)
24. Adeli, H., Karim, A.: Neural dynamics model for optimization of cold-formed steel beams. Journal of Structural Engineering 123(11), 1535–1543 (1997)
25. Adeli, H., Jiang, X.: Neuro-fuzzy logic model for freeway work zone capacity estimation. Journal of Transportation Engineering 129(5), 484–493 (2003)
26. Sejnowski, T.J.: Open questions about computation in the cerebral cortex. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing, vol. 2, pp. 372–389. MIT Press, Cambridge (1986)

27. Maass, W.: Lower bounds for the computational power of spiking neural networks. Neural Computation 8(1), 1–40 (1996)
28. Maass, W.: Networks of spiking neurons: The third generation of spiking neural network models. Neural Networks 10(9), 1659–1671 (1997a)
29. Bose, N.K., Liang, P.: Neural Network Fundamentals with Graphs, Algorithms, and Applications. McGraw-Hill, New York (1996)
30. Maass, W.: Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. In: Mozer, M., Jordan, M.I., Petsche, T. (eds.) Advances in Neural Information Processing Systems, vol. 9, pp. 211–217. MIT Press, Cambridge (1997b)
31. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
32. Hung, S.L., Adeli, H.: Parallel backpropagation learning algorithms on Cray Y-MP8/864 supercomputer. Neurocomputing 5(6), 287–302 (1993)
33. Park, H.S., Adeli, H.: Distributed neural dynamics algorithms for optimization of large steel structures. Journal of Structural Engineering 123(7), 880–888 (1997)
34. Adeli, H., Wu, M.: Regularization neural network for construction cost estimation. Journal of Construction Engineering and Management 124(1), 18–24 (1998)
35. Adeli, H., Samant, A.: An adaptive conjugate gradient neural network - wavelet model for traffic incident detection. Computer-Aided Civil and Infrastructure Engineering 15(4), 251–260 (2000)
36. Sirca, G., Adeli, H.: Neural network model for uplift load capacity of metal roof panels. Journal of Structural Engineering 127(11), 1276–1285 (2001)
37. Sirca, G., Adeli, H.: Neural network model for uplift load capacity of metal roof panels - closure. Journal of Structural Engineering 129(4), 562–563 (2003)
38. Dharia, A., Adeli, H.: Neural network model for rapid forecasting of freeway link travel time. Engineering Applications of Artificial Intelligence 16(7-8), 607–613 (2003)
39. Panakkat, A., Adeli, H.: Neural network models for earthquake magnitude prediction using multiple seismicity indicators. International Journal of Neural Systems 17(1), 13–33 (2007)
40. Jiang, X., Adeli, H.: Neuro-genetic algorithm for nonlinear active control of highrise buildings. International Journal for Numerical Methods in Engineering 75(8), 770–786 (2008)
41. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science, 4th edn. McGraw-Hill, New York (2000)
42. Hodgkin, A.L., Huxley, A.F.: A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes. Journal of Physiology 117, 500–544 (1952)
43. Rinzel, J., Ermentrout, G.B.: Analysis of neuronal excitability and oscillations. In: Koch, C., Segev, I. (eds.) Methods in Neuronal Modeling, pp. 135–169. MIT Press, Cambridge (1989)
44. Hille, B.: Ionic channels of excitable membranes, 2nd edn. Sinauer Associates, Sunderland (1992)
45. Ermentrout, G.B.: Type I membranes, phase resetting curves, and synchrony. Neural Computation 8, 979–1001 (1996)
46. Hoppensteadt, F.C., Izhikevich, E.M.: Weakly Connected Neural Networks. Springer, New York (1997)
47. Izhikevich, E.M.: Solving the distal reward problem through linkage of STDP and dopamine signaling. Cerebral Cortex 17, 2443–2452 (2007)

48. Abbott, L.F., Kepler, T.B.: Model neurons: From Hodgkin-Huxley to Hopfield. In: Garrido, L. (ed.) Statistical Mechanics of Neural Networks. Springer, Berlin (1990)
49. Kepler, T.B., Abbott, L.F., Marder, E.: Reduction of conductance-based neuron models. Biological Cybernetics 66, 381–387 (1992)
50. Ermentrout, G.B., Kopell, N.: Parabolic bursting in an excitable system coupled with a slow oscillation. SIAM Journal on Applied Mathematics 46, 233–253 (1986)
51. Gerstner, W.: Time structure of the activity in neural network models. Physical Review E 51, 738–758 (1995)
52. Kistler, W.M., Gerstner, W., van Hemmen, J.L.: Reduction of Hodgkin-Huxley equations to a single-variable threshold model. Neural Computation 9, 1015–1045 (1997)
53. Izhikevich, E.M.: Resonate-and-fire neurons. Neural Networks 14, 883–894 (2001)
54. Gerstner, W., Kistler, W.M.: Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge University Press, New York (2002)
55. Bohte, S.M., Kok, J.N., La Poutré, J.A.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4), 17–37 (2002)
56. Natschläger, T., Ruf, B.: Spatial and temporal pattern analysis via spiking neurons. Network: Computation in Neural Systems 9(3), 319–332 (1998)
57. Booij, O., Nguyen, H.T.: A gradient descent rule for multiple spiking neurons emitting multiple spikes. Information Processing Letters 95(6), 552–558 (2005)
58. Ghosh-Dastidar, S., Adeli, H.: A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. Neural Networks 22 (in press, 2009)
59. Xin, J., Embrechts, M.J.: Supervised learning with spiking neural networks. In: Proceedings of the International Joint Conference on Neural Networks, Washington, DC, vol. 3, pp. 1772–1777 (2001)
60. McKennoch, S., Liu, D., Bushnell, L.G.: Fast modifications of the SpikeProp algorithm. In: Proceedings of the International Joint Conference on Neural Networks, Vancouver, Canada, pp. 3970–3977 (2006)
61. Silva, S.M., Ruano, A.E.: Application of Levenberg-Marquardt method to the training of spiking neural networks. In: Proceedings of the International Conference on Neural Networks and Brain, vol. 3, pp. 1354–1358 (2005)
62. Fahlman, S.E.: Faster-learning variations of back-propagation: An empirical study. In: Proceedings of the 1988 Connectionist Models Summer School, San Mateo, CA, pp. 38–51. Morgan Kaufmann, San Francisco (1988)
63. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In: IEEE International Conference on Neural Networks, San Francisco, CA, vol. 1, pp. 586–591 (1993)
64. Schrauwen, B., van Campenhout, J.: Extending SpikeProp. In: Proceedings of the International Joint Conference on Neural Networks, Budapest, pp. 471–476 (2004)
65. Moore, S.C.: Backpropagation in spiking neural networks. Master's thesis, University of Bath (2002)
66. Maass, W.: Fast sigmoidal networks via spiking neurons. Neural Computation 9(2), 279–304 (1997c)