World Scientific
www.worldscientific.com

Review Article

SPIKING NEURAL NETWORKS

SAMANWOY GHOSH-DASTIDAR

Department of Biomedical Engineering, The Ohio State University

HOJJAT ADELI

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

Most current Artificial Neural Network (ANN) models are based on highly simplified brain dynamics. They have been used as powerful computational tools to solve complex pattern recognition, function estimation, and classification problems. ANNs have been evolving towards more powerful and more biologically realistic models. In the past decade, Spiking Neural Networks (SNNs) have been developed which comprise of *spiking neurons*. Information transfer in these neurons mimics the information transfer in biological neurons, i.e., via the precise timing of spikes or a sequence of spikes. To facilitate learning in such networks, new learning algorithms based on varying degrees of biological plausibility have also been developed recently. Addition of the temporal dimension for information encoding in SNNs yields new insight into the dynamics of the human brain and could result in compact representations of large neural networks. As such, SNNs have great potential for solving complicated time-dependent pattern recognition problems because of their inherent dynamic representation. This article presents a state-of-the-art review of the development of spiking neurons and SNNs, and provides insight into their evolution as the third generation neural networks.

Keywords: Spiking neuron; Spiking Neural Network; learning algorithm; information encoding; unsupervised learning; supervised learning.

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.[9,22,13,2,46,14,34,148,145,143,147,38,125,42,29] They have also been used for complicated optimization problems with highly nonlinear and discontinuous constraints.[21,117,15,137,144,25,140,146,104] Over time, ANNs have evolved into more powerful and biologically realistic models.[3,4,11,47,48] Improved understanding of the brain and its modes of information processing has led to the development of networks such as feedforward neural networks,[7,50] recurrent networks,[20,133,114] radial basis function neural networks,[82,85,51,95,121] self-organizing maps, probabilistic neural networks,[26,115,19] complex valued neural networks,[79,111,92,36,123,88] and dynamic neural networks.[74,73,75,76,87] Further, neural networks have been integrated with other computational intelligence techniques such as fuzzy logic[37,71,141,112,122,142,35] and genetic algorithm[18,17,93] as well as wavelets, a signal processing technique,[150,84,72,16,90] to solve even more complicated, especially dynamic pattern recognition problems.[128,82,83,127,5,6,151]

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 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.[8,61,62,21,12,10,63] 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 have not been many attempts to categorize generations of neural networks.[98] This is a particularly difficult task because ANN developments have branched out in many directions and it would not be accurate to label one advancement as more significant than another. In addition, such a categorization is subjective and dependent on what is considered advancement. One such clearly identifiable, significant conceptual advancement has been the development of the mathematically-defined activation or transfer function as the information processing mechanism of the artificial neuron.[98] The categorization of *generations* of neural networks in the paper are based on this idea.

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 many biological neural networks.[136,96,98] A presynaptic neuron communicates with a postsynaptic neuron via trains of spikes or action potentials. Biological spikes have a fixed morphology and amplitude.[33] 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.[99] 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 could, therefore, 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 (RBF), instead of a fixed threshold value, for output determination.[96] 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, in this case even the postsynaptic neuron could generate rate encoded information. This model gained widespread acceptance as processing elements in feedforward ANNs. This popularity was further increased due to Rumelhart's backpropagation (BP) training algorithm[126] developed for these ANNs that enabled supervised learning. Since the BP algorithm was constrained by its requirement of a continuous and differentiable activation function, a significant portion of the ensuing research became focused on finding more appropriate continuous and differentiable activation functions. This model was significantly more powerful than the one based on first generation

neurons and could solve complex pattern recognition problems (the most notable early example was the XOR problem).[60,118,24,23,139,39,113,77] 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 past decade or so, to overcome this shortcoming, neurons that can communicate via the precise timing of spikes or a sequence of spikes have been developed and adapted for ANNs. These neurons have been dubbed *spiking neurons*. In the literature, these spiking neurons have been referred to as *third generation* neurons.[98] Similar to the first generation neurons, spiking neurons act as integrate-and-fire units and have an *all or none* response. The spiking neuron, however, has an inherent dynamic nature characterized by an internal state which changes with time. 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.

3. Mechanism of Spike Generation in Biological Neurons

In general, action potentials or spikes from various presynaptic neurons reach a postsynaptic neuron 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 still not completely understood. A spike train consists of a sequence of such spikes. Each spike in the spike train induces a PSP at the time instant it reaches the postsynaptic neuron. Overall, multiple presynaptic neurons, each with multiple spikes, induce multiple PSPs over time. The PSPs are temporally integrated to compute the internal state of the postsynaptic neuron over time. The postsynaptic neuron fires when the integrated internal state crosses a threshold.[48]

The internal state of a postsynaptic neuron is a response to a presynaptic spike as shown in Fig. 1. However, if the postsynaptic neuron fires, i.e.
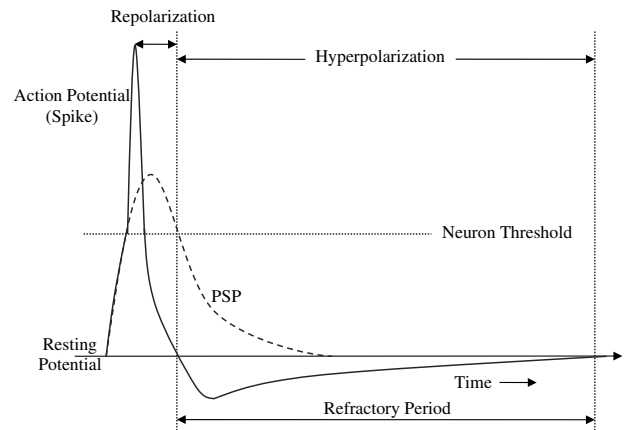

Fig. 1. The internal state of a postsynaptic neuron and the action potential generated in response to a presynaptic spike (not shown in the figure).

generates a spike, the internal state of the postsynaptic neuron does not remain the same as the PSP. In other words, the internal state changes based on the time of the spike from the postsynaptic neuron itself. Immediately after the firing of an output spike, the internal state decreases sharply. This phase is known as *repolarization*.[33,81] After the repolarization phase, the internal state of the neuron is kept at a value lower than the resting potential of the neuron (Fig. 1). This phase is known as *hyperpolarization*.[33,81] 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. 1). The three processes of temporal integration of PSPs, repolarization, and hyperpolarization, together, are responsible for the importance of the timing of spikes in the presynaptic spike train for encoding information. These phases are governed by means of various biological processes including ion channel mechanisms (see Refs. 33 and 81 for details).

4. Models of Spiking Neurons

Spiking neurons can be modeled in many different ways. A number of detailed mathematical or *biophysical* models have been developed to quantitatively characterize neuronal behavior based on detailed modeling of the neuronal membrane potential and ion channel conductances.[57,124,56,40,59,66] 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.[67] Izhikevich *et al.*[68] employed a large network of the spiking neuron models described in Ref. 66 with known firing patterns in the cerebral cortex to study self organization in such a network. Recently, Iglesias and Villa[64] also investigated a large network of spiking neurons described by a different neuron model and various biological processes, and observed a similar pattern of self organization and preferential firing patterns in the neural network.

Such detailed networks are important for studying the effects of various spiking patterns especially in terms of network activation patterns, i.e. which neurons are activated and whether certain neurons are activated more often than others. The level of detail in such models, although ideal for reproducing electrophysiological responses accurately, increases the complexity of the models, makes them difficult to analyze, and imposes a significant computational burden.[1,89,48] As a result, these networks have not been used for real-world classification or pattern recognition tasks that have typically been the domain of traditional ANNs. It may be argued that real-world data can be encoded as spike trains and classified using such biologically plausible networks. However, from the studies of self organization in such networks,[68,64] it appears that there may be a biological preference for certain spike patterns. In the opinion of the authors, appropriate biologically plausible spike train encoding methodologies should be investigated thoroughly for use with such networks.

The approach to designing such biophysical neuronal models is a *bottom-up* approach. Various detailed neuronal characteristics such as properties of the cell membrane, ion channels, mathematical formulations of learning such as long term potentiation and depotentiation, Hebbian learning, and spike time dependent plasticity (STDP) are modeled separately. These separate models are integrated to obtain the overall characterization of neuronal dynamics. On the other hand, phenomenological models represent a *top-down* approach in which the overall behavior of the individual neuron is modeled mathematically. The precise details of neuronal behavior at the level of ion channels or neurotransmitter molecules are not modeled explicitly. Sometimes, the details are

approximately derived only as secondary phenomena. Spike response models are examples of such phenomenological models that are simpler than the detailed biophysical models and offer a compromise between computational burden and electrophysiological details.[41,124,43,91,65,45,48] Therefore, spike response models are preferred for systemic studies of memory, neural coding, and network dynamics.

5. Spiking Neural Networks (SNNs)

It is interesting to note that the phenomenological models can be further simplified to lesser and lesser realistic models until a point is reached where even the spikes are not modeled, which lead to the first and second generation neurons. Evidently, the computational burden reduces significantly and so does the degree of biological realism. In some ways, the level of modeling detail is a function of the available computing power. For instance, consider the ANNs based on second generation neurons. For decades, the modeling of the neurons was limited by the available computing power because the hardware was unable to support large ANNs based on detailed neuronal models. This limitation dictated the design of the learning algorithms. Subsequently, even when advances were made in computing power, proportionate advances were not made in the complexity of the neuronal models because the existing learning algorithms were not compatible with the detailed models.

As a result, two distinct research areas emerged. The field of *Artificial Neural Networks* focused on the behavior of large networks of neuron-like processing units (i.e. the second generation neurons), which were primitive and oversimplified formulations of biological neurons. However, it was demonstrated that even such networks were capable of learning using pseudo-realistic learning algorithms such as backpropagation. ANNs were applied with great success to pattern recognition, classification, and pattern completion tasks in a wide variety of areas.

The other field became known as *Computational Neuroscience*. Within this broad interdisciplinary field, the detailed biophysical and phenomenological models were primarily used in relatively smaller networks to study electrophysiological processes, pattern generation, and the dynamic behavior of small groups of neurons. There have also been studies

involving very large numbers of interconnected biophysical neuron models. However, it has not been possible to use such networks of detailed neurons in a manner similar to ANNs for large real-world pattern recognition and classification tasks as mentioned in the beginning of this article.

Recent advances and the availability of computing power have increased the overlap between the two fields. On the one hand, the processing units, networks, and learning algorithms for ANNs have become biologically more realistic. On the other hand, networks of biophysical neurons have become increasingly larger in size and the biophysical models, more detailed. The available computing power still limits the use of the detailed models in large biophysical neural networks for pattern recognition and classification tasks. As the computing power becomes more readily available, suitable learning algorithms are also being developed for such models. The development of Spiking Neural Networks (SNNs) was the next logical step towards achieving this goal.

Simply stated, SNNs are networks of spiking neurons. The SNN architecture is normally similar to that of a traditional ANN. The processing unit, however, is a spiking neuron, which is typically modeled by a phenomenological model such as a Spike Response Model. As discussed earlier, the use of the biophysical models in certain applications of SNNs is less common due to the computational burden. Therefore, for the purpose of this paper, a distinction is made between SNNs that use phenomenological models and networks that use biophysical models. Only research on the former is reviewed in the rest of this paper.

6. Unsupervised Learning

Unsupervised learning is based solely on the characteristics of the data. The network *learns* the patterns in the data without being guided by any external cues regarding a *desired* outcome. A typical example of unsupervised learning is clustering where data are grouped into clusters (i.e., classified) based on some selected property of the data.[46,52,71] The rationale is that data that are *close* or similar to each other in terms of the selected property should be clustered together. One advantage of unsupervised clustering is the lower computational burden because the process eliminates the need for multiple

iterations through a training data set, which is typically required for supervised learning algorithms such as gradient descent and its variants. As a result, it is not surprising that most initial applications of SNNs were restricted to applications of unsupervised learning.

An early SNN model was presented by Hopfield[58] where the stimuli were represented by the precise timing of spikes and the spike pattern was encoded in the synaptic delays. The neurons acted similar to an RBF neuron, i.e., they fired when the input spike pattern was similar to the pattern encoded as the center of the RBFs. Otherwise, the neuron did not fire. This similarity was modeled by a distance function between the patterns. Similarity or dissimilarity was decided by a fixed threshold. Soon after, an STDP-like learning rule was presented for a similar RBF neuron that used the spike time difference between the presynaptic and postsynaptic spikes as the basis for learning.[44] Using a similar concept, Maass and Natschläger[100−102] modeled the temporal encoding of associative memory using a Hopfield network comprised of spiking neurons. In their work they used a traditional recurrent network architecture. Using these models, it was shown that unsupervised learning and self-organization were possible in networks of spiking neurons.[108,110]

These models were further enhanced to analyze spatial and temporal patterns in the input space and cluster the input data. It was demonstrated that the clustering based on these models converged reliably even when the input data were corrupted by noise.[108,109] For the sake of simplicity, neurons in these models were restricted to the emission of a single spike. Given this limitation, the connection between two SNN neurons was modeled by multiple synapses[108] in order to enable the presynaptic neuron to affect the postsynaptic neuron by inducing PSPs of varying magnitudes at various time instants.

Bohte *et al.*[31] employed a network similar to Natschläger and Ruf[108] and demonstrated that SNNs are capable of clustering real-world data. They used a population encoding scheme to encode the data to improve the accuracy of the SNN. Based on this encoding scheme, they proposed a multi-scale encoding where each dimension of the input space was encoded by multiple neurons with overlapping Gaussian fields. They reported good performance of the network with the Fisher iris dataset and an

arbitrary image segmentation application compared to traditional clustering methods such as *k*-means clustering and self organizing maps. Bohte *et al.*[31] also reported an increased robustness of their model with respect to noise in the data. Finally, they extended their model to a multi-layer RBF SNN that performed hierarchical clustering of the data. More recently, Gueorguieva *et al.*[52] investigated a similar network architecture with an STDP-based learning mechanism discussed earlier and arrive at the same conclusions regarding efficiency and noise. Following the work of Bohte *et al.*,[31] Panuku and Sekhar[116] present a variation of the SpikeProp learning algorithm in which the weights are adjusted in stages, i.e., the weights between the input and hidden layers are adjusted first and the weights between the hidden and the output layers are adjusted next. It is shown that this algorithm can effectively separate linearly separable as well as interlocking clusters.

7. Supervised Learning

7.1. *Gradient descent algorithms for error backpropagation*

Although unsupervised learning was demonstrated in SNNs with a recurrent architecture, until recently, spiking neurons were considered to be incompatible with the error backpropagation required for supervised learning in purely feedforward networks. This incompatibility was due to the lack of a continuous and differentiable activation function that could relate the internal state of the neuron to the output spike times. To demonstrate that BP-based learning is possible in such a network, Bohte *et al.*[30] employed an SNN which comprised of spiking neurons based on Gerstner's[43] spike response model. The architecture of Bohte *et al.*'s model was patterned after the one by Natschläger and Ruf[108] where each connection between a presynaptic and postsynaptic neuron is modeled by multiple synapses and the neurons are restricted to the emission of a single spike.

The learning algorithm presented by Bohte *et al.*,[30] *SpikeProp*, was developed along the lines of the one for traditional neural networks.[126] Subsequently, SNN was used with various training algorithms such as backpropagation with momentum,[149,105] *QuickProp*,[149,105] resilient propagation (*RProp*),[105] and Levenberg-Marquardt

BP[138] to improve network training performance. Feedforward SNNs using these variants of the BP algorithm have been used to solve benchmark classification problems such as the XOR, Fisher iris, Wisconsin breast cancer tumor, and Statlog Landsat satellite spectral image classification problems.[30] 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.[134]

Ghosh-Dastidar and Adeli[48] performed an extensive parametric analysis and studied various combinations of optimal parameters and heuristic rules with the goal of improving the performance of existing SNN models significantly. The improved model was employed for classification of EEGs obtained from three different subject groups: healthy subjects, epileptic patients during a seizure free interval, and epileptic patients during a seizure.[48]

Due to the single-spike restriction for the SNN neuron output, information is primarily encoded with the *time to first spike*. The multiple synapses (with different delays) per connection, however, enable the encoding to retain a temporal aspect. In reality, these multiple synapses perform the function of modeling the spike train rather than modeling the biological aspect of the neuronal connections. In the authors' opinion, a more realistic implementation is multiple synapses per connection where every synapse has the ability to transmit spike trains (rather than single spikes). Ideally, the number of synapses per connection would also be an adaptive and *learnable* parameter. The problem with this strategy of using spike trains is a familiar one. It appeared that it would be impossible to extend the BP-based learning algorithms to such a model. Kaiser and Feldbusch[80] attempted to circumvent this problem entirely by means of a compromise between rate encoding and pulse encoding by using a rate encoding activation function over discrete time periods. Their argument was that individual spikes can be modeled using an infinitesimally small time period that would contain only a single spike.

New learning algorithms have recently been developed that directly adapt the gradient descent based training algorithm for SNNs that convey information in the form of spike trains instead of single spikes.[32,49] In these models multiple synapses

per connection are no longer necessary because the spike trains inherently introduce the temporal component of PSP induction. Multiple synapses are used nevertheless because, together with spike train communication, they represent a general case similar to biological neurons. As mentioned earlier, if the number of synapses is a learnable parameter, each pair of neurons in the network can have a different number of synapses. The biological realism of SNN learning would be further advanced as synapses between neurons are added or removed as required by the learning process. The adaptive adjustment of the number of synapses similar to the model proposed by Schrauwen and van Campenhout[134] holds significant potential and should be investigated further.

Booij and Nguyen[32] argued that the SpikeProp algorithm and its variants were, in principle, applicable to recurrent architectures (in addition to simple feedforward architectures) for SNNs. The only restriction was that the input neurons should not be postsynaptic to any neurons and the output neurons should not be presynaptic to any neurons. They investigated the ability of their learning algorithm to learn two bench mark classification problems. One was the classical XOR problem. The other was the classification of Poisson spike trains. Two spike trains were generated using Poisson processes.[32] Varying degrees of random noise was added to the two spike trains to create ten noisy versions of each spike train. The twenty spike trains were divided into training and testing sets and input to the SNN. Their learning algorithm achieved 89% accuracy in identifying the original noise-free spike trains when presented with the noisy version as input.

Most recently, Ghosh-Dastidar and Adeli[49] presented a *Multi-Spiking Neural Network* (*MuSpiNN*) model where information from one neuron is transmitted to the next in the form of multiple spikes via multiple synapses. They also developed a new supervised learning algorithm, dubbed *Multi-SpikeProp*, for training MuSpiNN. The model and learning algorithm employ the heuristic rules and optimum parameter values discovered in their recent research[48] that improve the efficiency of the original single-spiking SNN model by two orders of magnitude. The classification accuracies of MuSpiNN and Multi-SpikeProp are evaluated using three increasingly more complicated problems: the XOR problem, the Fisher iris classification problem, and the epilepsy and seizure detection (EEG classification) problem. For the iris and EEG classification problems, a modular architecture is employed to reduce each 3-class classification problem to three 2-class classification problems and improve the classification accuracy. For the complicated EEG classification problem, they report a classification accuracy in the range of 90.7-94.8%, significantly higher than the 82% classification accuracy obtained using the single-spiking SNN with SpikeProp.

McKennoch *et al.*[106] present an alternate model for error backpropagation in an SNN that uses *theta neurons* as the processing units. Theta neurons do not model the actual action potential in time domain, which is a discrete response. Instead, the response is transformed to a phase plane formed by plotting the internal state against the recovery variable (mathematical representation of the recovery state of the inactive ion channels during the refractory period). Finally, the phase plane response is expressed in terms of the phase with respect to time, which is a continuous function.[54,53] Although the theta neuron does not precisely model the postsynaptic spikes, the continuous nature of the phase function makes it an attractive alternative in terms of compatibility with the backpropagation algorithm.[106]

A general shortcoming of the SNN models is the computational burden which may be reduced by developing efficient learning algorithms. Additionally, the authors believe that the adaptive adjustment of the number of synapses discussed earlier will be especially effective in reducing the number of synaptic computations, without compromising the classification accuracy. Another source of computational effort is the input encoding that increases the number of features many fold. Novel methods of input encoding that do not increase the number of features and, at the same time, represent the input accurately must be explored. Another problem is the sensitivity of gradient descent-based learning algorithms to the initial state of the SNNs. This has been investigated and heuristic rules have been developed to ensure that the SNN training converges to a global minimum despite the highly uneven error surface.[30,107,149,32,138,105,48] Although significant advances have been made in this aspect, the issue has not been fully resolved.

7.2. *Alternate Models of SNN learning*

To avoid the sensitivity of the learning to the initial SNN state, non-gradient descent algorithms such as genetic algorithms[55,129–132] and evolutionary strategies have been proposed.[28,120] Such algorithms have been used with considerable success for traditional ANNs[77] and are still being investigated for use with SNNs. The learning algorithms presented by Hagras *et al.*[55] and Pavlidis *et al.*[120] adapt only the weights to reach an optimum solution. The optimization objective usually involves minimization of a cost function (typically, a measure of error). Belatreche *et al.*[28,27] present an evolutionary strategy-based learning algorithm that adapts *both* synaptic weights and delays to reach the optimization objective and report classification performance comparable to Bohte *et al.*'s SpikeProp on the XOR and iris classification problems. Jin *et al.*[78] present a Pareto-based multi-objective genetic algorithm to simultaneously adjust the connectivity, weights, and delays of the SNN. The two optimization objectives of this model are to minimize: (1) the number of synaptic connections and (2) either the classification error or the root mean squared error. Despite some promising results, it should be noted that, a known disadvantage of evolutionary algorithms is their prohibitively large computational cost for large problems.

The SNNs discussed thus far have been feed-forward and recurrent networks. These networks have been investigated mostly with regular and fully connected architectures. Such architectures are not common in biological neural networks. A biological network consists of many neurons and their synapses which form the physical network. In reality however, only a small percentage of the neurons in the physical network contributes to the actual information processing. These neurons activate whereas neurons that do not contribute to the output remain silent. The network of activated neurons will be referred to as the *functional* network. Biological learning within such a network is a dynamic process. Silent neurons do not always remain silent and the activated neurons are not required to activate every time. The neurons that comprise the functional network often change based on the input.

This aspect of biological networks is modeled using the so-called *reservoir computing* paradigm. It is possible that biological plausibility was less of a motivation for this development than the search for more efficient supervised learning algorithms for recurrent ANNs.[119] Similar to the traditional recurrent ANNs, these SNNs have the traditional input neurons and output neurons, some with recurrent connection. Between the input and the output neurons, instead of a hidden layer of neurons in the traditional sense, there is a set or *reservoir* of neurons. The architecture of the reservoir is random with sparse connections. Based on the input to the network, the input neurons activate a subset of neurons within the reservoir. This functional network, i.e. the spatial activation pattern, is read by the output neurons, which are called the *readout neurons.* It is important to note that in this model, supervised learning *only* occurs at the synaptic connections between the reservoir and the output layer, usually by means of algorithms such as linear regression. Two examples of reservoir computing, as presented for SNN models, are the Echo-State Network[69,70] and Liquid State Machines.[103] For a comparison of these models, the reader should refer to Refs. 135 and 119.

Supervised Hebbian Learning is another biologically plausible learning strategy that has been proposed for SNNs. Similar to the other learning algorithms, the objective of the output neurons is to learn to fire at the desired spike times. However, in order to supervise the learning of the output neuron, the output neuron is constrained by additional inputs so that it fires only at specific times and not at others.[94] This model was extended by Kasinski and Ponulak[86] to a Liquid State Machine network. In this model, the supervision is not imposed directly on the input neuron. Rather, additional neurons outside the primary network receive the desired signals and influence the synaptic learning in the network. This model is dubbed Remote Supervision Method because the task of supervision is performed by the external neurons. This model was shown to have a high learning ability and accuracy.[86]

8. Concluding Remarks

Researchers have demonstrated that SNNs have significant potential for solving complicated time-dependent pattern recognition problems because of dynamic representation inherent in spiking neurons. Moreover, SNNs have been theoretically shown to have the ability to approximate any continuous function.[97] The addition of the temporal dimension

for information encoding has the potential to result in compact representations of large neural networks, another advantage for SNNs. Despite the interest in supervised learning algorithms for SNNs, their widespread acceptance and development are currently limited by the excessive computing times required for training.[48] This is a problem even with algorithms that are not based on gradient descent, especially when the network size becomes large. This shortcoming may be eliminated in the near future due to advances in computing technology in general as well as the development of more efficient and accurate SNN learning algorithms. Some of these algorithms have been reviewed in this paper. Combinations of such novel 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. L. F. Abbott and T. B. Kepler, Model neurons: From Hodgkin-Huxley to Hopfield, in L. Garrido (ed.), *Statistical Mechanics of Neural Networks* (Springer, Berlin, 1990).

2. H. Adeli. Neural networks in civil engineering: [1989–2000], *Computer-Aided Civil and Infrastructure Engineering* **16**(2) (2001) 126–142.

3. H. Adeli, S. Ghosh-Dastidar and N. Dadmehr, Alzheimer's disease and models of computation: Imaging, classification, and neural models, *Journal of Alzheimer's Disease* **7**(3) (2005) 187–199.

4. H. Adeli, S. Ghosh-Dastidar and N. Dadmehr, Alzheimer's disease: Models of computation and analysis of EEGs, *Clinical EEG and Neuroscience* **36**(3) (2005) 131–140.

5. H. Adeli, S. Ghosh-Dastidar and N. Dadmehr, A wavelet-chaos methodology for analysis of EEGs and EEG sub-bands to detect seizure and epilepsy, *IEEE Transactions on Biomedical Engineering* **54**(2) (2007) 205–211.

6. H. Adeli, S. Ghosh-Dastidar and N. Dadmehr, A spatio-temporal wavelet-chaos methodology for EEG-based diagnosis of Alzheimer's disease, *Neuroscience Letters* **444**(2) (2008) 190–194.

7. H. Adeli and S. L. Hung, A concurrent adaptive conjugate gradient learning algorithm on MIMD machines, *Journal of Supercomputer Applications, MIT Press* **7**(2) (1993) 155–166.

8. H. Adeli and S. L. Hung, An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks, *Applied Mathematics and Computation* **62**(1) (1994) 81–102.

9. H. Adeli and S. L. Hung, *Machine Learning — Neural Networks, Genetic Algorithms, and Fuzzy Sets* (John Wiley and Sons, NY, 1995).

10. H. Adeli and X. Jiang, Neuro-fuzzy logic model for freeway work zone capacity estimation, *Journal of Transportation Engineering* **129**(5) (2003) 484–493.

11. H. Adeli and X. Jiang, Dynamic fuzzy wavelet neural network model for structural system identification, *Journal of Structural Engineering* **132**(1) (2006) 102–111.

12. H. Adeli and A. Karim, Neural dynamics model for optimization of cold-formed steel beams, *Journal of Structural Engineering* **123**(11) (1997) 1535–1543.

13. H. Adeli and A. Karim, Fuzzy-wavelet RBFNN model for freeway incident detection, *Journal of Transportation Engineering* **126**(6) (2000) 464–471.

14. H. Adeli and A. Karim, *Wavelets in Intelligent Transportation Systems* (John Wiley and Sons, Hoboken, NJ, 2005).

15. H. Adeli and H. Kim, Cost optimization of composite floors using the neural dynamics model, *Communications in Numerical Methods in Engineering* **17** (2001) 771–787.

16. H. Adeli and H. Kim, Wavelet-hybrid feedback least mean square algorithm for robust control of structures, *Journal of Structural Engineering* **130**(1) (2004) 128–137.

17. H. Adeli and S. Kumar, Concurrent structural optimization on a massively parallel supercomputer, *Journal of Structural Engineering* **121**(11) (1995) 1588–1597.

18. H. Adeli and S. Kumar, Distributed genetic algorithms for structural optimization, *Journal of Aerospace Engineering* **8**(3) (1995) 156–163.

19. H. Adeli and A. Panakkat, A probabilistic neural network for earthquake magnitude prediction, *Neural Networks* **22**, in press.

20. H. Adeli and H. S. Park, Counter propagation neural network in structural engineering, *Journal of Structural Engineering* **121**(8) (1995) 1205–1212.

21. H. Adeli and H. S. Park, Optimization of structures by neural dynamics, *Neural Networks* **8**(5) (1995) 769–781.

22. H. Adeli and H. S. Park, *Neurocomputing for Design Automation* (CRC Press, Boca Raton, FL, 1998).

23. H. Adeli and A. Samant, An adaptive conjugate gradient neural network — wavelet model for traffic incident detection, *Computer-Aided Civil and Infrastructure Engineering* **15**(4) (2000) 251–260.

24. H. Adeli and M. Wu, Regularization neural network for construction cost estimation, *Journal of Construction Engineering and Management* **124**(1) (1998) 18–24.

25. F. Ahmadkhanlou and H. Adeli, Optimum cost design of reinforced concrete slabs using neural dynamics model, *Engineering Applications of Artificial Intelligence* **18**(1) (2005) 65–72.

26. R. Banchs, H. Klie, A. Rodriguez, S. G. Thomas and M. F. Wheeler, A neural stochastic multiscale optimization framework for sensor-based parameter estimation, *Integrated Computer-Aided Engineering* **14**(3) (2007) 213–223.

27. A. Belatreche, L. P. Maguire and M. McGinnity, Advances in design and application of spiking neural networks, *Soft Computing — A Fusion of Foundations, Methodologies and Applications* **11** (2007) 239–248.

28. A. Belatreche, L. P. Maguire, M. McGinnity and Q. X. Wu, A method for supervised training of spiking neural networks, in *Proceedings of the IEEE Conference Cybernetics Intelligence: Challenges and Advances* (Reading, UK, 2003), pp. 39–44.

29. Y. Benedic, P. Wira and J. Merckle, A new method for the re-implementation of threshold logic functions with cellular neural networks, *International Journal of Neural Systems* **18**(4) (2008) 293–303.

30. S. M. Bohte, J. N. Kok and H. La Poutr, Error-backpropagation in temporally encoded networks of spiking neurons, *Neurocomputing* **48**(1–4) (2002) 17–37.

31. S. M. Bohte, J. N. Kok and H. La Poutr, Unsupervised classification in a layered network of spiking neurons, *IEEE Transactions on Neural Networks* **13**(2) (2002) 426–435.

32. O. Booij and H. T. Nguyen, A gradient descent rule for multiple spiking neurons emitting multiple spikes, *Information Processing Letters* **95**(6) (2005) 552–558.

33. N. K. Bose and P Liang, *Neural Network Fundamentals with Graphs, Algorithms, and Applications* (McGraw-Hill, NY, 1996).

34. N. Bourbakis, P. Kakumanu, S. Makrogiannis, R. Bryll and S. Panchanathan, Neural network approach for image chromatic adaptation for skin color detection, *International Journal of Neural Systems* **17**(1) (2007) 1–12.

35. E. P. Carden and J. M. W. Brownjohn, Fuzzy clustering of stability diagrams for vibration-based structural health monitoring, *Computer-Aided Civil and Infrastructure Engineering* **23**(5) (2008) 360–372.

36. V. S. Chakravarthy, N. Gupte, S. Yogesh and A. Salhotra, Chaotic synchronization using a network of neural oscillators, *International Journal of Neural Systems* **18**(2) (2008) 157–164.

37. L. C. Chao, Fuzzy logic model for determining minimum bid markup, *Computer-Aided Civil and Infrastructure Engineering* **22**(6) (2007) 449–460.

38. M. A. Christodoulou and C. Kontogeorgou, Collision avoidance in commercial aircraft free fight, via neural networks and non-linear programming, *International Journal of Neural Systems* **18**(5) (2008) 371–387.

39. A. Dharia and H. Adeli, Neural network model for rapid forecasting of freeway link travel time, *Engineering Applications of Artificial Intelligence* **16**(7–8) (2003) 607–613.

40. G. B. Ermentrout, Type I membranes, phase resetting curves, and synchrony, *Neural Computation* **8** (1996) 979–1001.

41. G. B. Ermentrout and N. Kopell, Parabolic bursting in an excitable system coupled with a slow oscillation, *SIAM Journal on Applied Mathematics* **46** (1986) 233–253.

42. C. Fyfe, W. Barbakh, W. C. Ooi and H. Ko, Topological mappings of video and audio data, *International Journal of Neural Systems* **18**(6) (2008) 481–489.

43. W. Gerstner, Time structure of the activity in neural network models, *Physical Review E* **51** (1995) 738–758.

44. W. Gerstner, R. Kempter, J. van Hemmen and H. Wagner, A neuronal learning rule for sub-millisecond temporal coding, *Nature* **383** (1996) 76–78.

45. W. Gerstner and W. M. Kistler, *Spiking Neuron Models. Single Neurons, Populations, Plasticity* (Cambridge University Press, New York, 2002).

46. S. Ghosh-Dastidar and H. Adeli, Wavelet-clustering-neural network model for freeway incident detection, *Computer-Aided Civil and Infrastructure Engineering* **18**(5) (2003) 325–338.

47. S. Ghosh-Dastidar and H. Adeli, Neural network-wavelet microsimulation model for delay and queue length estimation at freeway work zones, *Journal of Transportation Engineering* **132**(4) (2006) 331–341.

48. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for EEG classification and epilepsy and seizure detection, *Integrated Computer-Aided Engineering* **14**(3) (2007) 187–212.

49. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Neural Networks* **22**, in press.

50. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection, *IEEE Transactions on Biomedical Engineering* **54**(9) (2007) 1545–1551.

51. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, *IEEE Transactions on Biomedical Engineering* **55**(2) (2008) 512–518.

52. N. Gueorguieva, I. Valova and G. Georgiev, Learning and data clustering with an RBF-based spiking neuron network, *Journal of Experimental and Theoretical Artificial Intelligence* **18**(1) (2006) 73–86.

53. B. Gutkin, D. Pinto and B. Ermentrout, Mathematical neuroscience: From neurons to circuits to

systems, *Journal of Physiology — Paris* **97** (2003) 209–219.

54. B. S. Gutkin and G. B. Ermentrout, Dynamics of membrane excitability determine interspike interval variability: A link between spike generation mechanisms and cortical spike train statistics, *Neural Computation* **10**(5) (1998) 1047–1065.

55. H. Hagras, A. Pounds-Cornish and M. Colley, Evolving spiking neural network controllers for autonomous robots, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 5 (2004), pp. 4620–4626.

56. B. Hille, *Ionic Channels of Excitable Membranes*, 2nd edn. (Sinauer Associates, Sunderland, MA, 1992).

57. A. L. Hodgkin and A. F. Huxley, A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes, *Journal of Physiology* **117** (1952) 500–544.

58. J. Hopfield, Pattern recognition computation using action potential timing for stimulus representation, *Nature* **376** (1995) 33–36.

59. F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Networks* (Springer-Verlag, New York, 1997).

60. S. L Hung and H. Adeli, Parallel backpropagation learning algorithms on Cray Y-MP8/864 supercomputer, *Neurocomputing* **5**(6) (1993) 287–302.

61. S. L. Hung and H. Adeli, Object-oriented back propagation and its application to structural design, *Neurocomputing* **6**(1) (1994) 45–55.

62. S. L. Hung and H. Adeli, A parallel genetic/neural network learning algorithm for MIMD shared memory machines, *IEEE Transactions on Neural Networks* **5**(6) (1994) 900–909.

63. H. T. Huynh, Y. Won and J. J. Kim, An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks, *International Journal of Neural Systems* **18**(5) (2008) 433–441.

64. J. Iglesias and A. E. P. Villa, Emergence of preferred firing sequences in large spiking neural networks during simulated neuronal development, *International Journal of Neural Systems* **18**(4) (2008) 267–277.

65. E. M. Izhikevich, Resonate-and-fire neurons, *Neural Networks* **14** (2001) 883–894.

66. E. M. Izhikevich, Simple model of spiking neurons, *IEEE Transactions on Neural Networks* **14**(6) (2003) 1569–1572.

67. E. M. Izhikevich, Solving the distal reward problem through linkage of STDP and dopamine signaling, *Cerebral Cortex* **17** (2007) 2443–2452.

68. E. M. Izhikevich, J. A. Gally and G. M. Edelman, Spike-timing dynamics of neuronal groups, *Cerebral Cortex* **14**(8) (2004) 933–944.

69. H. Jaeger, The echo state approach to analysis and training recurrent neural networks, Technical Report TR-GMD-148, German National Research Center for Information Technology, (2001).

70. H. Jaeger and M. Lukosevicius, Optimization and applications of echo state networks with leaky-integrator neurons, *Neural Networks* **20**(3) (2007) 335–352.

71. X. Jiang and H. Adeli, Fuzzy clustering approach for accurate embedding dimension identification in chaotic time series, *Integrated Computer-Aided Engineering* **10**(3) (2003) 287–302.

72. X. Jiang and H. Adeli, Wavelet packet-autocorrelation function method for traffic flow pattern analysis, *Computer-Aided Civil and Infrastructure Engineering* **19**(5) (2004) 324–337.

73. X. Jiang and H. Adeli, Dynamic wavelet neural network for nonlinear identification of highrise buildings, *Computer-Aided Civil and Infrastructure Engineering* **20**(5) (2005) 316–330.

74. X. Jiang and H. Adeli, Dynamic wavelet neural network model for traffic flow forecasting, *Journal of Transportation Engineering* **131**(10) (2005) 771–779.

75. X. Jiang and H. Adeli, Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings, *International Journal for Numerical Methods in Engineering*, 71(5) (2007) 606–629.

76. X. Jiang and H. Adeli, Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular highrise building structures, *International Journal for Numerical Methods in Engineering* **74**(7) (2008) 1045–1066.

77. X. Jiang and H. Adeli, Neuro-genetic algorithm for nonlinear active control of highrise buildings, *International Journal for Numerical Methods in Engineering* **75**(8) (2008) 770–786.

78. Y. Jin, R. Wen and B. Sendhoff, Evolutionary multi-objective optimization of spiking neural networks, in J. P. Marques de Sa, L. A. Alexandre, W. Duch and D. Mandic (eds.), *Proceedings of the 17th International Conference on Artificial Neural Networks*, Vol. 4668 of *Lecture Notes in Computer Science* (Springer, Berlin, 2007), pp. 370–379.

79. T. D. Jorgensen, B. P. Haynes and C. C. F. Norlund, Pruning artificial neural networks using neural complexity measures, *International Journal of Neural Systems* **18**(5) (2008) 389–403.

80. F. Kaiser and F. Feldbusch, Building a bridge between spiking and artificial neural networks, in J. P. Marques de Sa, L. A. Alexandre, W. Duch and D. Mandic (eds.), *Proceedings of the 17th International Conference on Artificial Neural Networks*, Vol. 4668 of *Lecture Notes in Computer Science* (Springer, Berlin, 2007), pp. 380–389.

81. E. R. Kandel, J. H. Schwartz and T. M. Jessell, *Principles of Neural Science*, 4th edn. (McGraw-Hill, NY, 2000).

82. A. Karim and H. Adeli, Comparison of the fuzzy-wavelet RBFNN freeway incident detection model with the california algorithm, *Journal of Transportation Engineering* **128**(1) (2002) 21–30.

83. A. Karim and H. Adeli, Incident detection algorithm using wavelet energy representation of traffic patterns, *Journal of Transportation Engineering* **128**(3) (2002) 232–242.

84. A. Karim and H. Adeli, Fast automatic incident detection on urban and rural freeways using the wavelet energy algorithm, *Journal of Transportation Engineering* **129**(1) (2003) 57–68.

85. A. Karim and H. Adeli, Radial basis function neural network for work zone capacity and queue estimation, *Journal of Transportation Engineering* **129**(5) (2003) 494–503.

86. A. Kasinski and F. Ponulak, Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks, in *Proceedings of the 15th International Conference on Artificial Neural Networks: Biological Inspirations — Lecture Notes in Computer Science*, Vol. 3696 (Berlin, Springer, 2005), pp. 145–153.

87. N. Katada and H. Nishimura, Dynamic memorization characteristics in neural networks with different neuronal dynamics, *International Journal of Neural Systems* **17**(3) (2007) 161–170.

88. S. Kawata and A. Hirose, Frequency-multiplexing ability of complex-valued hebbian learning in logic gates, *International Journal of Neural Systems* **18**(2) (2008) 173–184.

89. T. B. Kepler, L. F. Abbott and E. Marder, Reduction of conductance-based neuron models, *Biological Cybernetics* **66** (1992) 381–387.

90. H. Kim and H. Adeli, Hybrid control of smart structures using a novel wavelet-based algorithm, *Computer-Aided Civil and Infrastructure Engineering* **20**(1) (2005) 7–22.

91. W. M. Kistler, W. Gerstner and J. L. van Hemmen, Reduction of Hodgkin-Huxley equations to a single-variable threshold model, *Neural Computation* **9** (1997) 1015–1045.

92. M. Kobayashi, Pseudo-relaxation learning algorithm for complex valued associative memory, *International Journal of Neural Systems* **18**(2) (2008) 147–156.

93. H. Lee, E. Kim and M. Park, A genetic feature weighting scheme for pattern recognition, *Integrated Computer-Aided Engineering* **14**(2) (2007) 161–171.

94. R. Legenstein, C. Naeger and W. Maass, What can a neuron learn with spike-timing-dependent plasticity? *Neural Computation* **17**(11) (2005) 2337–2382.

95. H. Liu, X. Wang and W. Qiang, A fast method for implicit surface reconstruction based on radial basis functions network from 3D scattered points, *International Journal of Neural Systems* **17**(6) (2007) 459–465.

96. W. Maass, Lower bounds for the computational power of spiking neural networks, *Neural Computation* **8**(1) (1996) 1–40.

97. W. Maass, Fast sigmoidal networks via spiking neurons, *Neural Computation* **9**(2) (1997) 279–304.

98. W. Maass, Networks of spiking neurons: The third generation of spiking neural network models, *Neural Networks* **10**(9) (1997) 1659–1671.

99. W. Maass, Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons, in M. Mozer, M. I. Jordan and T Petsche (eds.), *Advances in Neural Information Processing Systems*, Vol. 9 (MIT Press, Cambridge, MA, 1997), pp. 211–217.

100. W. Maass and T. Natschläger, Networks of spiking neurons can emulate arbitrary Hopfield nets in temporal coding, *Network: Computation in Neural Systems* **8**(4) (1997) 355–372.

101. W. Maass and T. Natschläger, Associative memory with networks of spiking neurons in temporal coding, in L. S. Smith and A. Hamilton (eds.), *Neuromorphic Systems: Engineering Silicon from Neurobiology* (World Scientific, 1998), pp. 21–32.

102. W. Maass and T. Natschläger, Emulation of Hopfield networks with spiking neurons in temporal coding, in J. M. Bower (ed.), *Computational Neuroscience: Trends in Research* (Plenum Press, 1998), pp. 221–226.

103. W. Maass, T. Natschläger and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Computation* **14**(11) (2002) 2531–2560.

104. R. V. Mayorga and M. Arriaga, Non-linear global optimization via parameterization and inverse function approximation: An artificial neural networks approach, *International Journal of Neural Systems* **17**(5) (2007) 353–368.

105. S. McKennoch, D. Liu and L. G. Bushnell, Fast modifications of the SpikeProp algorithm, in *Proceedings of the International Joint Conference on Neural Networks* (Vancouver, Canada, 2006), pp. 3970–3977.

106. S. McKennoch, T. Voegtlin and L. Bushnell, Spike-timing error backpropagation in theta neuron networks, *Neural Computation* **21**(1) (2009) 9–45.

107. S. C. Moore, Backpropagation in spiking neural networks, Master's thesis, University of Bath (2002).

108. T. Natschläger and B. Ruf, Spatial and temporal pattern analysis via spiking neurons, *Network: Computation in Neural Systems* **9**(3) (1998) 319–332.

109. T. Natschläger and B. Ruf, Pattern analysis with spiking neurons using delay coding, *Neurocomputing* **26–27**(1–3) (1999) 463–469.

110. T. Natschläger, B. Ruf and M. Schmitt, Unsupervised learning and self-organization in networks of

spiking neurons, in U. Seiffert and L. C. Jain (eds.), *Self-Organizing Neural Networks. Recent Advances and Applications*, Vol. 78 of *Springer Series on Studies in Fuzziness and Soft Computing* (Springer-Verlag, Heidelberg, 2001).

111. T. Nitta, The uniqueness theorem for complex-valued neural networks with threshold parameters and the redundancy of the parameters, *International Journal of Neural Systems* **18**(2) (2008) 123–134.

112. Y. Nomura, H. Furuta and M. Hiorkane, An integrated fuzzy control system for structural vibrations, *Computer-Aided Civil and Infrastructure Engineering* **22**(4) (2007) 306–316.

113. A. Panakkat and H. Adeli, Neural network models for earthquake magnitude prediction using multiple seismicity indicators, *International Journal of Neural Systems* **17**(1) (2007) 13–33.

114. A. Panakkat and H. Adeli, Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators, *Computer-Aided Civil and Infrastructure Engineering* **24**(4) (2009) 280–292.

115. A. Pande and M. Abdel-Aty, A computing approach using probabilistic neural networks for instantaneous appraisal of rear-end crash risk, *Computer-Aided Civil and Infrastructure Engineering* **23**(7) (2008) 549–559.

116. L. N. Panuku and C. C. Sekhar, Clustering of non-linearly separable data using spiking neural networks, in J. P. Marques de Sa, L. A. Alexandre, W. Duch and D. Mandic (eds.), *Proceedings of the 17th International Conference on Artificial Neural Networks*, Vol. 4668 of *Lecture Notes in Computer Science* (Springer, Berlin, 2007), pp. 390–399.

117. H. S. Park and H. Adeli, A neural dynamics model for structural optimization — application to plastic design of structures, *Computers and Structures* **57**(3) (1995) 391–399.

118. H. S. Park and H. Adeli, Distributed neural dynamics algorithms for optimization of large steel structures, *Journal of Structural Engineering* **123**(7) (1997) 880–888.

119. H. Paugam-Moisy and S. Bohte, Computing with spiking neuron networks, in J. Kok and T. Heskes (eds.), *Handbook of Natural Computing* (Springer-Verlag, 2009).

120. N. G. Pavlidis, D. K. Tasoulis, V. P. Plagianakos and M. N. Vrahatis, Spiking neural network training using evolutionary algorithms, in *Proceedings of the International Joint Conference on Neural Networks* (Montreal, QC, 2005), pp. 2190–2194.

121. W. Pedrycz, R. Rai and J. Zurada, Experience-consistent modeling for radial basis function neural networks, *International Journal of Neural Systems* **18**(4) (2008) 279–292.

122. K. Perusich, Using fuzzy cognitive maps to identify multiple causes in troubleshooting systems, *Integrated Computer-Aided Engineering* **15**(2) (2008) 197–206.

123. V. S. H. Rao and G. R. Murthy, Global dynamics of a class of complex valued neural networks, *International Journal of Neural Systems* **18**(2) (2008) 165–171.

124. J. Rinzel and G. B. Ermentrout, Analysis of neuronal excitability and oscillations, in C. Koch and I. Segev (eds.), *Methods in Neuronal Modeling* (MIT Press, Cambridge, MA, 1989), pp. 135–169.

125. J. Ruiz-Pinales, R. Jaime-Rivas, E. Lecolinet and M. J. Castro-Bleda, Cursive word recognition based on interactive activation and early visual processing models, *International Journal of Neural Systems* **18**(5) (2008) 419–431.

126. D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning internal representations by error propagation, in D. E. Rumelhart and J. L McClelland (eds.), *Parallel Distributed Processing*, Vol. 1 (MIT Press, Cambridge, MA, 1986), pp. 318–362.

127. C. Sabourin, K. Madani and O. Bruneau, Autonomous biped gait pattern based on fuzzy-CMAC neural networks, *Integrated Computer-Aided Engineering* **14**(2) (2007) 173–186.

128. A. Samant and H. Adeli, Enhancing neural network incident detection algorithms using wavelets, *Computer-Aided Civil and Infrastructure Engineering* **16**(4) (2001) 239–245.

129. K. Sarma and H. Adeli, Fuzzy discrete multicriteria cost optimization of steel structures, *Journal of Structural Engineering* **126**(11) (2000) 1339–1347.

130. K. Sarma and H. Adeli, Fuzzy genetic algorithm for optimization of steel structures, *Journal of Structural Engineering* **126**(5) (2000) 596–604.

131. K. C. Sarma and H. Adeli, Bi-level parallel genetic algorithms for optimization of large steel structures, *Computer-Aided Civil and Infrastructure Engineering* **16**(5) (2001) 295–304.

132. K. C. Sarma and H. Adeli, Life-cycle cost optimization of steel structures, *International Journal for Numerical Methods in Engineering* **55**(12) (2002) 1451–1462.

133. A. M. Schaefer and H. G. Zimmermann, Recurrent neural networks are universal approximators, *International Journal of Neural Systems* **17**(4) (2007) 253–263.

134. B. Schrauwen and J. van Campenhout, Extending SpikeProp, in *Proceedings of the International Joint Conference on Neural Networks* (Budapest, 2004), pp. 471–476.

135. B. Schrauwen, D. Verstraeten and J. van Campenhout, An overview of reservoir computing: Theory, applications and implementations, in *Proceedings of the 15th European Symposium on Artificial Neural Networks — Advances in Computational Intelligence and Learning* (2007), pp. 471–482.

136. T. J. Sejnowski, Open questions about computation in the cerebral cortex, in D. E. Rumelhart

and J. L. McClelland (eds.), *Parallel Distributed Processing* (MIT Press, Cambridge, MA, 2, 1986), pp. 372–389.

137. A. B. Senouci and H. Adeli, Resource scheduling using neural dynamics model of Adeli and Park, *Journal of Construction Engineering and Management* **127**(1) (2001) 28–34.

138. S. M. Silva and A. E. Ruano, 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 (2005), pp. 1354–1358.

139. G. Sirca and H. Adeli, Neural network model for uplift load capacity of metal roof panels, *Journal of Structural Engineering* **127**(11) (2001) 1276–1285.

140. G. Sirca and H. Adeli, Cost optimization of prestressed concrete bridges, *Journal of Structural Engineering* **131**(3) (2005) 380–388.

141. J. F. Smith, H. ThanhVu and T. H. Nguyen, Autonomous and cooperative robotic behavior based on fuzzy logic and genetic programming, *Integrated Computer-Aided Engineering* **14**(2) (2007) 141–159.

142. A. Stathopoulos, L. Dimitriou and T. Tsekeris, Fuzzy modeling approach for combined forecasting of urban traffic flow, *Computer-Aided Civil and Infrastructure Engineering* **23**(7) (2008) 521–535.

143. S. Suresh, N. Kannan, N. Sundararajan and P. Saratchandran, Neural adaptive control for vibration suppression in composite fin-tip of aircraft, *International Journal of Neural Systems* **18**(3) (2008) 219–231.

144. A. R. Tashakori and H. Adeli, Optimum design of cold-formed steel space structures using neural dynamic model, *Journal of Constructional Steel Research* **58**(12) (2002) 1545–1566.

145. N. Tsapatsoulis, K. Rapantzikos and C. Pattichis, An embedded saliency map estimator scheme: Application to video encoding, *International Journal of Neural Systems* **17**(4) (2007) 289–304.

146. I. Villaverde, M. Grana and A. d'Anjou, Morphological neural networks and vision based simultaneous localization and mapping, *Integrated Computer-Aided Engineering* **14**(4) (2007) 355–363.

147. E. I. Vlahogianni, M. G. Karlaftis and J. C. Golias, Temporal evolution of short-term urban traffic flow: A non-linear dynamics approach, *Computer-Aided Civil and Infrastructure Engineering* **23**(7) (2008) 536–548.

148. H. Wersing, S. Kirstein, M. Gtting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. Steil, H. Ritter and E. Körner, Online learning of objects in a biologically motivated visual architecture, *International Journal of Neural Systems* **17**(4) (2007) 219–230.

149. J. Xin and M. J. Embrechts, Supervised learning with spiking neural networks, in *Proceedings of the International Joint Conference on Neural Networks*, Vol. 3 (Washington, D.C., 2001), pp. 1772–1777.

150. Z. Zhou and H. Adeli, Time-frequency signal analysis of earthquake records using mexican hat wavelets, *Computer-Aided Civil and Infrastructure Engineering* **18**(5) (2003) 379–389.

151. W. Zou, Z. Chi and K. C. Lo, Improvement of image classification using wavelet coefficients with structured-based neural network, *International Journal of Neural Systems* **18**(3) (2008) 195–205.