Int. J. Advance. Soft Comput. Appl., Vol. 6, No. 1, March 2014
ISSN 2074-8523; Copyright © SCRG Publication, 2014

Computing with Spiking Neuron Networks

A Review

Falah Y.H. Ahmed, Bariah Yusob, and Haza Nuzly Abdull Hamed

Soft Computing Research Group, Faculty of Computing, Universiti Teknologi Malaysia, UTM Skudai, 81310 Skudai, Johor, Malaysia
e-mail: falah_1980@yahoo.com; bariah.yusob@gmail.com; haza@utm.my

Abstract

Spiking Neuron Networks (SNNs) are often referred to as the third generation of neural networks. Highly inspired from natural computing in the brain and recent advances in neurosciences, they derive their strength and interest from an accurate modeling of synaptic interactions between neurons, taking into account the time of spike firing. SNNs overcome the computational power of neural networks made of threshold or sigmoidal units. Based on dynamic event-driven processing, they open up new horizons for developing models with an exponential capacity of memorizing and a strong ability to fast adaptation. Today, the main challenge is to discover efficient learning rules that might take advantage of the specific features of SNNs while keeping the nice properties (general-purpose, easy-to-use, available simulators, etc.) of traditional connectionist models. This paper presents the history of the "spiking neuron", summarizes the most currently-in-use models of neurons and synaptic plasticity, the computational power of SNNs is addressed and the problem of learning in networks of spiking neurons is tackled.

Keywords: *Spiking neuron, artificial neural networks.*

1 Introduction

The human brain consists of an intricate web of billions of interconnected cells called neurons. The study of neural networks in computer science aims to understand how such a large collection of connected elements can produce useful computations, such as vision and speech recognition. A real neuron receives pulses from many other neurons. These pulses are processed in a manner that may result in the generation of pulses in the receiving neuron, which are then transmitted to the other neurons (Fig. 1A). Neurons compute by transforming input pulses into output pulses. Artificial Neural Networks (ANN) try to capture the essence of this computation as depicted in figure 1B. The rate at which a neuron fires pulses is abstracted to a scalar activity-value, or output, assigned to the neuron. Directional connections determine which neurons are input to other neurons. Each connection has a weight, and the output of a particular neuron is a function of the sum of the weighted outputs of the neurons it receives input from. The applied function is called the transfer function, F(∑) which is binary because thresholding force neurons have as output a "1" or a "0", depending on whether or not the summed input exceeds some threshold. Sigmoid neurons apply a sigmoidal transfer-function, and have a real-valued output (inset Fig. 1B, solid response in dotted line). ANNs are sets of connected artificial neurons. Its computational power is derived from clever choices for the values of the connection weights. Learning rules for neural networks prescribe how to adapt the weights to improve performance given some task. An example of a neural network is the Multi-Layer Perceptron (MLP, Fig. 1C). Learning rules like error back propagation [1] allow it to learn and perform many tasks associated with intelligent behavior, like learning, memory, pattern recognition, and classification [2,3].

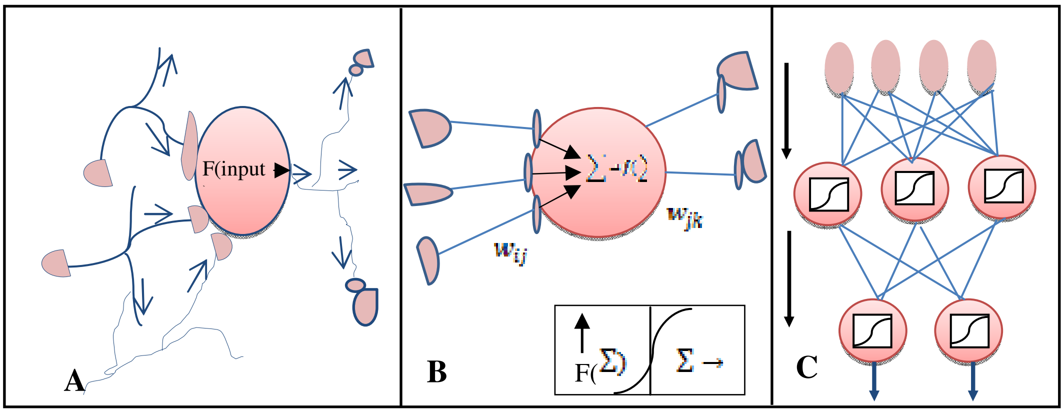

Fig. 1. Artificial neural networks

With the introduction of sigmoidal artificial neurons, and learning rules for training networks consisting of multiple layers of neurons [1,4], some of the deficiencies of the earlier neural networks were overcome. The most prominent example was the ability to learn to compute the XOR function. Since then, multi-layer networks of sigmoidal neurons have been shown to cope with many useful computations, such as pattern classification, pattern recognition, and unsupervised clustering.

In 1949, Hebb hypothesized that, to achieve sufficient flexibility and productivity in a neural network, it would be useful to have the network dynamically linked all the neurons such that detection of different properties of an object can be done as an array or assembly. The purpose of an assembly would be to classify a signal from the output of the constituent neurons, each coding for different properties, without losing sight of the fact that they all add up as part of the same object [5]. Objects composed of different atomic parts could thus be efficiently detected. An example would be to have a neuron that detects the color red, and another neuron that detects the shape of an apple. When linked together in an assembly, these neurons would indicate the presence of a red apple. By having neurons that can each detect a particular atomic, property, a linking mechanism allows the system to be productive, in the sense that by just having a limited set of detectors for atomic properties, any combination of these properties can be enough to express the complete classification. For instance, linking separate red, green, yellow, apple, banana and pear, detectors allow the expression of nine differently colored objects [6].

In the presence of a single object, composed of a number of properties, a simple on-off detector-signal for each property is sufficient to correctly describe the particular composition. However, in the presence of multiple objects this simple compositional signaling scheme is ambiguous [7], and more powerful means of linking atomic elements into composite structures (like .red apple.) in neural networks are needed. Classification in the presence of multiple objects has so far remained elusive or at best sketchy [7]. Even though the usefulness of such schemes has been well recognized e.g. for vision [8], speech recognition [9], and the representation and manipulation of symbolic information [7], several researchers even argued that the representation of compositional information is impossible in neural networks [10,11].

The starting point of many theses on ANN is the notion originally put forward by Sander Bohte [12]. That is, the binding problem can be resolved by a type of neural network (based on the real biological model) where there is complete connection between all the spiking neurons. Malsburg, 1999 [7] proposed that in order for neurons to be sensitive to coding for features that belong to the same object, these neurons would synchronize the times at which they emit spikes (the .synchrony-hypothesis.). Neurons coding for features belonging to different objects would then fire out of phase, allowing for multiple compositional objects to be represented simultaneously. Assembly-dependent correlation between neurons was interpreted as support for this idea [13], and much research into the temporal properties of spiking neurons (in neuroscience as well as in computational modeling) then followed.

2 Computational Process in ANNs

Neurons for first generation ANNs send binary signals (high) only if the summation of the weighted received signals goes up above a threshold value. This implies that the activation function used is solely a step function. When connected as multi-layer perceptron (MLP) with one hidden layer, the signals sent (output of MLP) will be of the Boolean type which may be utilized to help in the computation which can make this first generation ANN as universal approximators [2].

The function as universal approximators can be further improved with the help of second generation ANN. In the second generation ANN, the activation function is continuous (sigmoid function or hyperbolic tangents). The output signals are computed as analog for the two cases, when there is input and when there is output. Back propagation (BP) and recurrent neural networks are used for training and learning purposes. Because BP is flexible and has the ability to approximate any continuous function, it has been used in system identification for real–time chemical process [4].

For the first generation ANNs and for second generation of ANNs, the inputs to each neuron consist of the sum of the incident values multiplied by some weightage. For the third generation ANNs (frequently referred to as Spiking Neural Networks- SNNs) , the inputs to each neuron consist of pulse spikes arriving at random , and the values of the spikes incident onto the neurons are the values of the arriving spikes multiplied by the weights of the preceptors. The incident signals are accepted by the neurons only at the spiking time (time window) of the neurons. The accepted spiking signals can be considered as a type of signals associated with the stabilized frequencies of the neurons. For a signal to be communicated through SNNs, its value has to be converted to a certain time scale. This conversion is known as rate coding or pulse coding. The neurons can only see spiking time or no spiking time. They do not recognize any time in between. Computation is carried out when all the neurons have completely fired. After this firing cycle, the network starts processing on the values of input and comparing them with the attributes to get correct classification .Hence, third generation ANNs (SNNs) are superior to the first generation ANNs or the second generation ANNs. It has been established that SNNs are more biologically realistic than the first generation and second generation ANNs [14,15].

SNNs manipulate spatial-temporal data in their calculation and communication, like what the real neurons actually do [16,4]. The methods of sending and receiving of individual pulses is called rate (pulse) coding. They permit multiplexing of data as amplitude and frequency [17]. Latest findings indicate that neurons can carry out analog calculations in the cortex at an unbelievable rate.

As an example, visual inputs for facial recognition can be analyzed and classified by the brain in less than 100ms [18]. If 10 steps of synaptic excitation are applied to the retina at the temporal lobe, the eye will process the signal within 10ms after the excitation. Therefore, time perception is very short for permitting an averaging method such as rate (pulse) coding [17,18]. When processing speed becomes important, pulse coding technique is preferred [18,19].

3 Spiking Neural Network (SNN)

The third generation neural networks are Spiking Neural Networks (SNN) (Fig. 2) which has become an exciting topic in recent years [20]. SNNs became famous even before the advent of sigmoidal or perceptron neurons [21]. It has been shown that SNNs are suitable for parallel implementation in digital hardware [22], and in analog hardware [23,15].

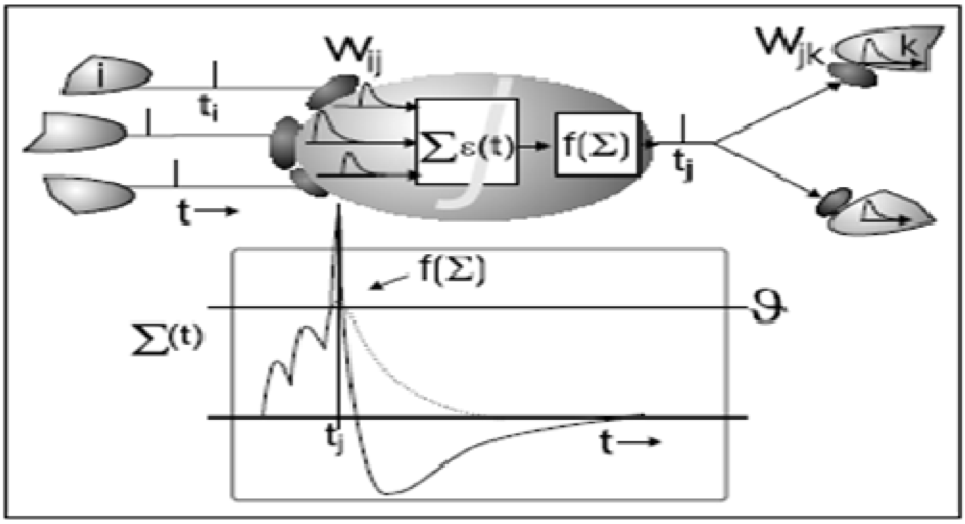

Fig. 2. Spiking neural network [12](Bohte 2003)

Previous generations of neural networks use analog signals to convey information from one neuron to another. Communications between neurons in SNNs use spikes similar to that used by real biological neurons. These spikes are recognized only at the instant they occur. Using weighted sum of the analog input values the previous neuron calculates a value using sum specific non- linear function. This value will determine the delay for the spike output which is targeted for the subsequent neuron. In general, the spiking neuron can be viewed as a leaky integrator. This is because the targeted neurons will integrate the spikes over time and accept the resulting integrated value which is used as membrane potential. Whenever the membrane potential approaches a certain threshold value the neuron sends a spike, after that its membrane potential is reset.

New knowledge in information processing in biological neurons have explained several additional parameters(such as gene and protein expression) which have to be considered for a neuron to spike [15,23,24].These additional parameters may include physical properties of connections [25], the probabilities of spikes being received at the synapses and also the emitted neuro-transmitters or open ion channels [26,27]. Many of these properties have been mathematically modeled and used to study biological neurons [28,29,30,31]. SNN are made up of artificial neurons which communicate with the help of trains which can be considered as pulse coded information [32,33,20,34,23]. SNN are biologically plausible and offer some means for representing time, frequency, phase and other features of the information being processed. SNN has the ability to train the neurons to convert spatial-temporal information into spikes (whose properties include spiking time and spiking rates). When selecting the neuron model for big SNN, there is a tradeoff between the computational efficiency and biological plausibility [35]. If the computational efficiency is more important than biological plausibility, the Leaky Integrate and Fire (LIF) model will be adopted because of its low computational cost.

3.1 Model of spiking neurons and synaptic plasticity

A spiking neuron model accounts for the impact of impinging action potentials spikes on the targeted neuron in terms of the internal state of the neuron, as well as how this state relates to the spikes the neuron fires. There are many models of spiking neurons, and this section only describes some of the models that have so far been most influential in Spiking Neuron Networks.

3.1.1 Threshold-fire models

The threshold-fire models are based on the temporal summation of the contributions from all presynaptic neurons to the membrane potential unit. If this contribution exceeds a threshold ϑ, then the postsynaptic neuron will fire .In this section we will discuss two of these models, the integrate and- fire and the spike response model – SRM [35,32].

3.1.2 Integrate-and-fire models

Integrate-and-fire model was the first model to use spikes over time to convey information [28]. As the input spikes arrive in time, the inner potential of a neuron (postsynaptic potential - PSP) depends on the weights at the input. If the weights are positive, the connections are excitatory and an incoming stimulus acts to increase the PSP. Connections with negative weights are inhibitory as stimulus passing through them act to decrease the PSP. When the postsynaptic potential reaches a threshold, an output spike is released. In the event of an output spike, the PSP is reset to its resting potential, as can be seen in Fig. 3.

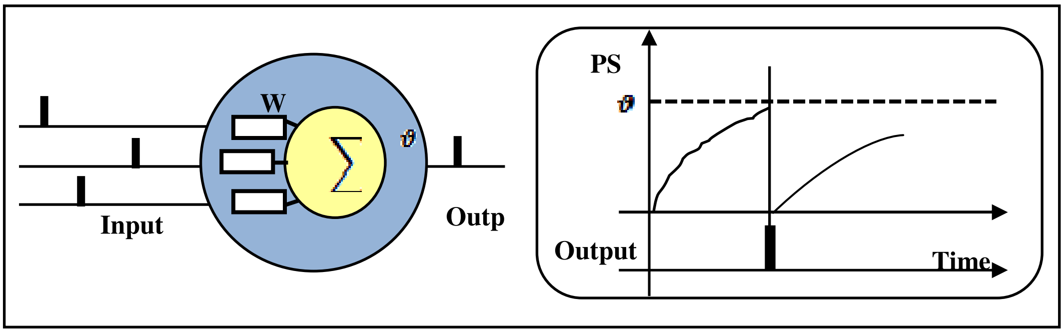

Fig. 3. Integrate-and-fire neuron [35].

3.1.3 Integrate-and-fire with leakage models

Integrate-and-fire neuron has its PSP ruled by a decay term, which decreases the magnitude of PSP over time [28]. When neurons cease to receive input excitation, the PSP gradually decreases and after some time reaches its resting potential. This mechanism, in its simplest form, can be associated with an RC electrical circuit where each neuron is composed of resistors and capacitors. Consequently, neuronal activity can be analyzed using the theory of electrical circuits. The dynamics of a leaky neuron can be expressed by the change in the PSP (excitation or inhibition) upon spike arrival as:

$$PSP_{(t)} = PSP_{(t)} \pm A_{\max}\left(1 - \exp - \left(\frac{t - t_{ini}}{\tau_{rise}}\right)\right) \tag{1}$$

Where $A_{\max}$ is the maximum activation caused by a single spike, t_{ini} is the time of the incoming spike, and τ_{rise} is the excitatory or inhibitory time constant of the neuron. Some simplified models do not consider the exponential term in Equation 1. As a result, upon the arrival of a spike, the PSP is simply added to by the constant $A_{\max}$.

The PSP decay term, on the other hand, is described as:

$$PSP_{(t)} = \pm A_{\max}\left(1 - \exp - \left(\frac{t - t_{ini}}{\tau_{decay}}\right)\right) \tag{2}$$

Where $A_{\max}$ is the maximum activation caused by a single spike, t_{ini} is the time of the incoming spike, and τ_{decay} is the time constant for PSP decay in the neuron. Fig. 4 shows the dynamics of a leaky integrate-and-fire neuron. Experiments have demonstrated that leaky integrate-and-fire neurons can very realistically reproduce the behavior of biological neurons [35].

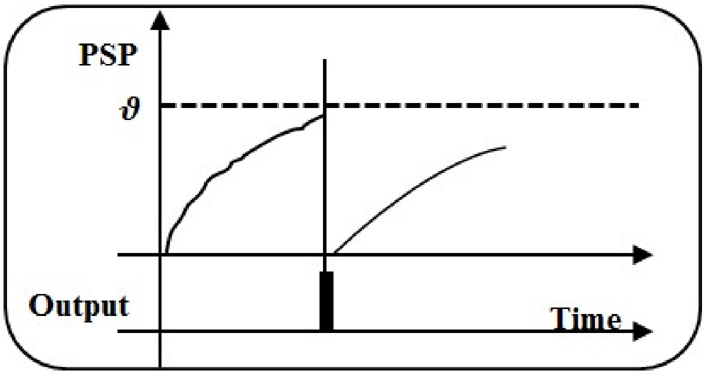

Fig. 4. Typical behaviour of a leaky integrate-and-fire neuron (rise and decay terms).

3.1.4 Different neuron models

Apart from Threshold- Fire Models, there are other models which can be used for SNN. Three different neuron models which are frequently used are as listed below:

i. Izhikevich model

ii. Thorpe's model

iii. Fitzhugh-Nagumo model

3.1.5 Izhikevich model

Izhikevich (IZ) applied bifurcation theory to create this model, which is a 2D ordinary differential equation system [32,30,31].

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + 1$$
$$\frac{du}{dt} = a(bv - u) \tag{3}$$

Resetting of the neuron after every spike is governed by the following formula:

$$\text{If } v \geq 30mV \text{ Then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{4}$$

Where v = the voltage of a neuron, u = adjusting function as a recovery variable, *I (f)* = current input, And a,b,c,d are adjustable parameters.

When the membrane voltage v (t) reaches its top peak (30 mV), a spike is emitted, then the v and u are reset to other values according to the mechanism in rule (3). The resting range in the model is between -70 mV and -60 mV depending on the value of b. i.e., to imitate regular spiking neurons, the scale of time for *a, b, c and d* are set to be 0.02, 0.2, - 65 mV and 2 respectively.

3.1.6 Thorpe's model

Thorpe's model is a simple version of an integrate-and-fire neuron without leakage in which the membrane potential of a post-synaptic neuron i at time t depends on the firing order of all its pre-synaptic neurons *j* [18]:

$$PSP_i = w_{ji} \sum \mathrm{mod}\, order_j \tag{5}$$

Where mod ∈ [0, 1] is the modulation factor, *orderj* is the firing order of a pre-synaptic neuron *j*, *order j* ∈ [0, *m*-1] where *m* is the number of pre-synsptic neurons connected to neuron *i*, *wji* is the synaptic efficacy (weight) of the synapse connecting neuron *i* and neuron *j*. The synaptic modification is proportional to the firing order of the spikes received:

$$\Delta w_{ji} = \mathrm{mod}^{order_j} \tag{6}$$

With the same convention as in (6). Connections to superior order pre-synaptic neurons are given a higher weight. Higher weights results in stronger connections. When a spike is received via one of the neuron's synapse, the neuron's potential *PSPi* builds up. When the *PSPi* reaches a threshold *PSPθi*, neuron *i* fires a spike. After the spike is fired the *PSPi* is set to 0:

$$PSP_i = \begin{cases} PSP_i + P_{ji}, PSP_i < PSP_{\theta i}(\textit{spike received}) \\ 0, PSP_i = PSP_{\theta i}(\textit{spike emitted}) \end{cases} \tag{7}$$

This model has been proven to be an efficient way of modeling the visual system [18] and it has been used to create an audio model [36].

3.1.7 Fitzhugh-Nagumo model

This model is a modified version of a single cell neuron (which is often referred to as HH model). Like the HH model it is possible to get low level steady state for small values of applied current. Intermediate values of current yield stable oscillatory state. Higher values of current will again produce a steady state (of higher values). This model differs from HH in that it uses fewer variables. By imitating the null lines of the HH model with a straight line and a cubic function, a polynomial decreasing model for the following form was obtained [37,38,15].

$$\begin{aligned} \frac{dv}{dt} &= v - \frac{v^3}{3} - w + I(t) \\ \frac{dw}{dt} &= v - \varepsilon(v + a - bw) \end{aligned} \tag{8}$$

In the previous formula, *v* represents the fast ability parameter, and *w* the slow ability parameter; both of which are recovery variables. Parameters a, b and ε are associated with the time scale and dynamic kinetics of the recovery variable. The FitzHugh-Nagumo model deserves special mention [23] because it was discovered after a research work to get a mathematical model of a single cell neuron for the axons of a giant squid (as does the Izhikevich model [38]. The latter is a simple and computationally inexpensive neuron model (suitable for large-scale simulation) that uses two coupled differential equations which are able to reproduce several biologically realistic neuronal behaviors (brain-like activity, bursting, etc.).

3.2 Neural unit models

This section explores in details the spiking neural networks which represent a more realistic neuronal unit models. There are two main applications for a group of spiking neurons connected to each other in a network. SNNs can be used to model brain functions and they are also useful as tools in artificial intelligence. These two applications are discussed in details as follows.

3.2.1 SNNs for modeling brain functions

Traditionally SNNs have been used in computational neuroscience, usually in an attempt to evaluate the dynamics of neurons and how they interact in an ensemble [39]. The Hodgkin-Huxley model of spiking generation [40] can be considered the pioneering work describing the action potentials in terms of ion channels and current flow [41]. Further studies expanded this work and revealed the existence of a wide number of ion channels and that the set of ion channels varies from one neuron to another [42]. Genesis and Neuron [43,44] are examples of widely known simulation tools that use neurons described with ion channels.

New neuron models have been developed using the simulation tools of Genesis and Neuron [43,44,15,35] for which internal and external behaviors of a single neuron are simulated as compartmental electric cable. Because of this , simplified models such as the integrate-and-fire neuron [34], for all intents and purposes have the properties of a single resistor-capacitor (RC) circuit which enables the Izhikevich model [38] be combined with the Hodgkin-Huxley model to produce the integrate-and-fire model which can be described as a two-dimensional system of ordinary differential equations . Spiking neural networks (SNN) use SpikeProp as training algorithms which implement both incremental and batch processing [45].

3.2.2 SNNs in artificial intelligence

Most neural networks are considered as artificial intelligent systems which execute information processing using linear or non-linear processing elements (for instance, a sigmoid function) [46,47,48,49,50,45,15]. Over the years, SNN has been considered too complex and difficult to analyze. Other reasons for leaving SNN aside in artificial intelligence tasks include:

i. Biological cortical neurons have long time constants. Typically fast or slow inhibition can be in the order of dozens of milliseconds and fast or slow excitation can reach hundreds of milliseconds. This dynamics can considerably constrain applications that need fine temporal processing [51].

ii. For biological cortical neurons, there is no prior knowledge of the time coded information. Although it is known that neurons receive and emit spikes, whether neurons encode information using spike rate or precise spike time is still unclear [52]. For those supporting the theory of spike rate coding, it is reasonable to approximate the average number of spikes in a neuron with continuous values and consequently process them with traditional processing units (sigmoid, for instance). Therefore, it is not necessary to perform simulations with spikes, as the computation with continuous values is simpler to implement and evaluate.

However, new discoveries on the information processing capabilities of the brain and the technical advances related to massive parallel processing, are bringing forward the idea of using biologically realistic networks in artificial intelligent systems. Many have questioned the use of rate coding, mainly because they have the assumption that rate coding can be very slow to provide reliable outputs (the average number of spikes needs to be computed over a certain period of time). However, many perceptual experiments have shown to be contrary to this assumption. For instance, a pioneering work has shown that the primate (including human) visual system can classify complex natural scenes in only around 100-150 ms [53]. The same magic numbers of 100-150 ms were obtained by other researchers, when they discovered that unprimed views of common objects can be recognized at a rate of 10 Hz. This rate of information processing is important considering that billions of neurons are involved and the massive volume of information is propagated through several areas of the brain before a decision is made [54].

Such results culminated in a theory suggesting that a single neuron probably exchanges only one or a few spikes before the information processing task is concluded. As a result of Thorpe's work, a simple multi-layer feed-forward network (BP) of integrate-and-fire neurons that can successfully detect and recognize faces in real time was designed [55,56,15]. Other works [12,57,58] also present systems using precise timing of spikes on pattern recognition (clustering, supervised and unsupervised training).

An important landmark in the use of SNNs in artificial information processing is the work of Maass [20], which shows that, theoretically, SNN can be used as universal approximates of continuous functions. Mishra [59] gave examples of spiking neural networks applied to benchmark datasets (internet traffic data, EEG data, XOR problems, 3-bit parity problems, iris dataset) to perform function approximation and supervised pattern recognition. A comparison with a traditional Multi-Layer Perceptron Network (MLP) highlights the differences in performance between the systems in each specific dataset.

3.3 Learning in SNN

This section describes several learning algorithms designed for SNN. Learning in SNN is a complex process since information is represented in time dependent spikes. Most of the SNNs use recurrent network topologies where learning is difficult. Some of the learning algorithms are normally being applied to a specific type of SNN due to its characteristics.

Like traditional neural network, learning in SNN is reinforcement, supervised and unsupervised. Supervised learning is the most commonly used learning algorithm in SNN [15]. Various supervised learning algorithms have been developed for SNN and have been reviewed by Kasabov [23].

3.3.1 Unsupervised learning

As mentioned above, the synaptic efficacy and the strength of the synaptic response may be influenced by the history of activity of the pre- or postsynaptic neurons. This phenomenon is known as synaptic plasticity [60]. There exists strong evidence that this phenomenon is a key factor for the learning processes.

The most common forms of the synaptic plasticity are summarized in Table 1. They differ mainly in their time duration. For instance, some processes (e.g. facilitation) decay at the rate of about 10-100ms; other processes (e.g. long-term potentiating (LTP) or long-term depression (LTD)) persist for hours, days, or longer. The spectrum of time constants is in fact so broad that it covers essentially every time scale, from the fastest (that of synaptic transmission itself), to the slowest (developmental).

Different forms of synaptic plasticity differ according to the conditions required for the induction (cf. Table 1, column 3). Some depends only on the history of presynaptic stimulation, independently of the postsynaptic response. For example, facilitation, augmentation, and postsynaptic potentiation occur after rapid presynaptic stimulation, with stronger stimulation leading to more persistent potentiating. Others depend on some coincidence of pre- and postsynaptic activity or even on the temporal order of pre- and postsynaptic spikes that can determine synaptic potentiation or depression [15,61].

Table 1: Different forms of synaptic plasticity [60,15]. By 'pre' we denote the presynaptic locus of the phenomenon induction, while 'post' stands for the postsynaptic locus.

Phenomenon	Duration	Locus of induction
Short-Term Enhancement		
Paired-Pulse Facilitation (PPF)	100 ms	Pre
Augmentation	10 s	Pre
Post-Tetanic Potentiation (PTP)	1 min	Pre
Long-Term Enhancement		
Short-Term Potentiation (STP)	15 min	Post
Long-Term Potentiation (LTP)	>30 min	Pre and post
Depression		
Paired-Pulse Depression (PPD)	100 ms	Pre
Depletion	10 s	Pre
Long-Term Depression (LTD)	>30 min	Pre and post

3.3.2 Supervised learning

A supervised spike-based processes, such as Spike-Timing Dependent Plasticity (STDP), have already been widely investigated and described in literature [62,63,28,34,54]. However, unsupervised approach is not appropriate for the learning tasks that require an explicit goal definition. In this chapter we focus on the supervised learning methods for precise spike timing in SNN. The goal of the presented survey is to determine what paradigms of neural information coding can be implemented with the recent approaches. We present some representative methods for supervised learning in SNN. For all these methods the common goal of learning can be stated as follows: given a sequence of input spikes trains $S^{in}(t)$ and a sequence of the target output spikes $S^{d}(t)$, find a vector of the synaptic weights w, such that the outputs of the learning neurons $S^{0}(t)$ are close to $S^{d}(t)$.

3.4 Spikeprop network model

The most famous and most used algorithm for supervised learning of feedforward network is the backpropagation algorithm (BP). 80 % of all applications of neural networks use backpropagation algorithm.

The first published back propagation algorithm for SNN is SpikeProp which was proposed by Bohte [11]. Many different versions of SpikeProp have been used [64,65]. SpikeProp algorithm is not similar to classical backpropagation algorithms which have been used frequently [1,66]. The difference between SpikeProp algorithm and classical back propagation algorithm is of course in their adaptation rules. SpikeProp allows the network to gradually adjust towards correct operation, just as traditional networks do [11,54].

The algorithm works in much the same way as back propagation does. It starts with the end neuron, and looks back at the neurons that connect to it. It then adjusts the weights on those connecting edges so that the neuron is more likely to fire at the correct time. This process is continued from the neurons at the output layer of the network back towards the input layer.

This system restricts spiking neural networks to a feed forward model where the neurons are organized into layers [67]. These layers are connected to the previous and next layers by edges, but they are not connected to any neurons within their own layer. In addition, they only affect neurons in the next layer. The adjustment algorithm assumes that all weights rely on the desired firing pattern of the neurons in the next layer. If this assumption does not hold, each weight adjustment must take into account the chance of causing a neuron in a previous layer to fire. Because it is not possible to know how previous layers are related to other layers (including the one in which the current neuron is) the neuron cannot be adjusted (cannot be initialized) without considering the previous neurons in relation to preceding neurons. Initialization of neurons to fire is a problem associated with circular dependence and cannot be solved within this algorithm.

Another flaw in SpikeProp is that, in cases where the potential barely reaches the level of synapse (causing the neuron to fire), the gradient of the potential during the spike is very small which in turn causes the derivative of the error to be very high [67]. This causes neurons that need little adjustment to be dramatically changed. It's an edge case, but it can cause networks to never converge to a correct weighting arrangement. Without a fix for this, 4% of cases never converge, and cases that do converge take 16.7% longer on average.

Finally, networks using SpikeSrop can only fire once in a given time period. This means that, to handle big problems, increasingly large networks must be used. However, none of these issues are endemic to spiking neural networks. It is possible to train a spiking neural network that is both recurrent and has neurons that spike multiple times.

Spiking neural networks (SNNs) are believed to be biologically more plausible [68,69,11,54,15,35] and computationally more powerful than analog neural networks [70].

Computational power of SNNs has yet to be demonstrated, mainly due to the fact that an efficient supervised learning algorithm still unavailable. In contrast to analog neural networks, for which various sophisticated supervised learning algorithms have been developed [71], only a very limited number of supervised learning algorithms are available for training SNNs, which can be attributed to the discontinuous nature of spiking neurons.

SpikeProp adopt error backpropagation procedures which have been used widely in the training of analog neural networks to perform supervised learning [72]. SpikeProp do have weaknesses. The first weakness concerns sensitivity to parameter initialization values. This means that if the neuron is still inactive after initialization, the SpikeProp will not perform training for that weights which will not produce any spike. The second weakness is that SpikeProp is only suitable in cases where there is latency-based coding. The third weakness is that SpikeProp works only for SNNs where neurons spike only once in the simulation time. Finally, SpikeProp algorithm has been designed for training the weights only. To address these weaknesses, several improvements to SpikeProp algorithms have been suggested [73,74,11,54].

4 Conclusion

This paper has given an overview of the current state-of-the Spiking Neuron Networks: its biological inspiration, the models that underlie the networks, some theoretical work on computational complexity and learnability, learning rules, both traditional and novel, and some current application areas. The novelty of the concept of SNNs means that many lines of research are still open and are actively being pursued. Describes in details are the fundamental concepts and methods of ANNs, SNNs and SpikeProp. Many studies in literature have been done to improve the performance of SNNs and SpikeProp algorithm based on ANNs. More work is still required to develop SNN and SpikeProp to improve generalization of error and classification accuracy etc.

ACKNOWLEDGEMENTS

This work is supported by The Ministry of Higher Education (MOHE) under Long Term Research Grant Scheme (LRGS/TD/2011/UTM/ICT/03 – 4L805). The authors would like to thanks Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for the support in R & D, Soft Computing Research Group (SCRG) for the inspiration in making this study a success. The authors would also like to thank the anonymous reviewers who have contributed enormously to this work.

References

[1] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. “Learning representations by back-propagating errors”. *Cognitive Modeling*. Eds: Polk, T.A and Seifert, C.M. (2002). Massachusetts Institute of Technology.

[2] Rebizant, W., Szafran, J., and Wiszniewski, A. “Fundamentals of System Analysis and Synthesis”. *Digital Signal Processing in Power System*

Protection and Control Signals and Communication Technology. (2011). Pp. 29-52. Springer.

[3] Siti M. Shamsuddin, Roselina Sallehuddin and Norfadzila M. Yusof. “Artificial Neural Network Time Series Modeling forRevenue Forecasting”. *Chiang Mai Journal of Science*. (2008). Vol 40, No 6, Special Issue 2. Pp. 411–426.

[4] Jin, Y.-J., Shen, B.-x., Ruo-fanRen, Yang, L., Sui, J. and Zhao, J.-g. “Prediction of the Stryrene Butiend Rubber Performance by Emulsion polymerization using Backpropagation Neural Network”. *Journal of Engineering*. Hindawi Publishing Corproration. (2013). Pp. 515704-515706. http://dx.doiorg/10.1155/2013/515704.

[5] Hebb, D.O. The Organization of Behavior: *A Neuropsychological Theory*. John Wiley & Sons Inc. New York. (1949).

[6] Sander, B., Joost, K. and Han, L. P. “Spike-prop: error- Backpropagation in multi-layer networks of spiking neurons”. *CWI Technical Report SEN-R0036*. (2000).

[7] Malsburg, V.D. “The what and why of binding: The modeler's perspective”. *Neuron, 24*. (1999). Pp.95-104.

[8] Rosenblatt, F. *Principles of neurodynamics: Perception and the theory of brain mechanisms*. Spartan Books. Washington, DC. (1962).

[9] Malsburg, V.D. and Schneider, W. “A neural cocktail-party processor”. *Biol. Cybern*. Vol 54. (1986). Pp. 29-40.

[10] Fodor, J. A. and Pylyshyn, Z. W. (1988). “Connectionism and cognitive architecture: A critical analysis”. *Cognition, 28(1–2).* (1988). Pp. 3-71. doi: http://dx.doi.org/10.1016/0010-0277(88)90031-5

[11] Bohte, S. M., Kok, J. N. and La Poutr´e. “Error-backpropagation in temporally encoded networks of spiking neurons”. *Neurocomputing*, *48*. (2002). pp. 17–37).

[12] Bohte, S.M. *Spiking Neural Networks*. (2003). PhD Thesis. University of Leiden.

[13] Gray, C.M., Konig, P., Engel, A.K. and Singer, W. “Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which re-ects global stimulus properties”. *Nature 338*. (1989). Pp. 334-337.

[14] DasGupta, B. and Schnitger, G. “The power of approximating: a comparison of activation functions”. *Advances in Neural Information Processing Systems, 5*. (1992). Pp. 363-374.

[15] Kasabov, N. "To spike or not to spike: A probabilistic spiking neuron model". *Neural Networks*, 23(1). (2010). Pp. 16-19. doi: http://dx.doi.org/10.1016/j.neunet.2009.08.010

[16] Ferster, D. and Spruston, N. "Cracking the neuronal code". *Science, New Series, vol 270, Issue 5237*. Nov. (2005). Pp. 756-757.

[17] Gerstner, W., Kempter, R. and van Hemmen, J.L. "Hebbian learning of pulse timing in the barn owl auditory system". Eds: Maass, W. and Bishop, C.M. *Pulse Neural Networks*. Chapter 14. MIT Press, Cambridge, M.A. (1998). Pp. 353-377.

[18] Thorpe, S., Delorme, A. and Van Rullen, R. "Spike-based strategies for rapid processing". *Neural Networks*, 14(6–7). (2001). pp. 715-725. doi: http://dx.doi.org/10.1016/S0893-6080(01)00083-1

[19] Belatreche, A. and Paul, R. "Dynamic cluster formation using populations of spiking neurons". *The 2012 International Joint Conference on Neural Network* (*IJCNN*). 10-15 June 2012. (2012). Pp. 1-6.

[20] Maass, W. "Computing with spiking neurons". In: Maass, W., Bishop, C., *Pulsed Neural Networks*. The MIT Press, Cambridge, MA. (2001). Pp. 55-85.

[21] Dayan, P. and Abbott, L.F. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. (2001). MIT Press, Cambridge, MA.

[22] Mehrtash, N., Jung, D., Hellmich, H. H., Schoenauer, T., Lu, V. T. and Klar, H. "Synaptic plasticity in spiking neural networks: a system approach". *IEEE Transactions on Neural Networks,* 14(5), (2003). Pp. 980-992. doi: 10.1109/tnn.2003.816060

[23] Kasabov, N. "Integrative connectionist learning systems inspired by nature: current models, future trends and challenges". Natural Computing. Volume 8, Issue 2. (2009). Pp. 199-218. doi: 10.1007/s11047-008-9066-z

[24] Kojima, H. and Katsumata, S. "An analysis of synaptic transmission and its plasticity by glutamate receptor channel kinetics models and 2-photon laser photolysis". *Proceedings of the 15th international conference on Advances in neuro-information processing - Volume Part I*, (2009). Auckland, New Zealand.

[25] Huguenard, J.R. "Reliability of axonal propagation: The spike doesn't stop here". *Proceedings of the National Academy of Sciences of the United States of America*. Vol 97, no 17. (2000). Pp. 9349-9350.

[26] Ikegaya, Y., Matsumoto, W., Chiou, Y-H., Yuste, R. and Aaron, G. "Statistical significance of precisely repeated intracellular synaptic patterns". *PLoS ONE 3(12): e3983*. (2008). Doi: 10.1371/journal.pone.0003983.

[27] Abbott, L. F. and Nelson, S. B. "Synaptic plasticity: taming the beast". *Nature neuroscience,*. 3 Suppl. (2000). Pp.1178–1183.

[28] Gerstner, W. and Kistler, W.M. *Spiking neuron models: An introduction*. (2002b). Cambridge University Press.

[29] Izhikevich, E.M. "Which model to use for cortical spiking neurons?". *IEEE Transactions on Neural Networks, 15(5)*. (2004). Pp. 1063–1070.

[30] Izhikevich, E.M. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting (Computational Neuroscience)*. (2006). MIT Press.

[31] Izhikevich, E.M. and Edelman, G.M. "Large-scale model of mammalian thalamocortical systems". *Proceedings of the National Academy of Sciences of the United States of America*. Vol 105, no 9. (2008). Pp. 3593-3598.

[32] Gerstner, W. "What´s different with spiking neurons". *Henk Mastebroek and Hans Vos, editors, Plausible Neural Networks for Biological Modelling*. (2001). Kluwer Academic. Pp.23-48.

[33] Maass, W. *The Third Generation of Neural Network Models*. (1997). Technische Universität Graz.

[34] Gerstner, W. and Kistler, W.M. *Spiking Neuron Models*. (2002a). Cambridge University Press.

[35] Kasabov, N. "Evolving, Probabilistic Spiking Neural Networks and Neurogenetic Systems for Spatio- and Spectro-Temporal Data Modelling and Pattern Recognition". *IEEE WCCI 2012, LNCS 7311*. (2012). Pp. 234–260.

[36] Wysoski, S. G., Benuskova, L. and Kasabov, N. Text-independent speaker authentication with spiking neural networks. *Proceedings of the 17th International Conference on Artificial Neural Networks*, Porto, Portugal. (2007).

[37] Fitzhugh R. "Mathematical models of threshold phenomena in the nerve membrane". *Bull. Math. Biophysics*, vo 17. (1955). Pp. 257-278.

[38] Izhikevich, E.M. "Simple model of spiking neurons". *IEEE Transactions on Neural Networks*, 14. (2003). Pp. 1569-1572.

[39] Benuskova, L., Kanich, M. and Krakovska, A. "Piriform cortex model of EEG has random underlying dynamics". *Proc. Of 1st World Congress on Neuroinformatics*, Vienna. (2001). Pp. 287-292.

[40] Hodgkin, A.L. and Huxley, A.F. (1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". *Journal of Physiology*. August 28; 117(4). (1952). Pp. 500-544.

[41] Nelson, M. and Rinzel, J. "The Hodgkin-Huxley model". Eds: Bower, J.M. and Beeman. *The Book of Genesis, Chapter 4*. (1995). Springer, New York. Pp. 27-51.

[42] Marom, S. and Abbott, L.F. (1994). “Modeling state-dependent inactivation of membrane currents”. *Biophysical Journal*. 67 2. (1994). Pp. 515-520.

[43] Genesis. http://www.genesis-sim.org/GENESIS/. Last accessed on 03/12/2007.

[44] Neuron. http://www.neuron.yale.edu/neuron/. Last accessed on 03/12/2007.

[45] ˇS´ıma, J. “Gradient Learning in Networks of Smoothly Spiking Neurons”. *Lecture Notes in Computer Science: Advances in Neuro-Information Perocessing*. Volume 5507. (2009). Pp. 179-186.

[46] Bishop, C. M. *Neural Networks for Pattern Recognition*. (2000). Oxford University Press.

[47] Bohte, S.M. “The evidence for neural information processing with precise spike-timing: A survey”. *Natural Computing* 3(2). (2005). Pp. 195-206.

[48] Gallant, S.I. *Neural Network Learning and Expert Sytstems*. (1995). MIT Press.

[49] Haykin, S. *Neural Networks: A Comprehensive Foundation*. (1998). Prentice Hall PTR.

[50] Negnevitsky, M. *Artificial Intelligence: A Guide to Intelligent Systems*. (2002). Addison Wesley.

[51] Gewaltig, M. “Evolution of Synchronous Spike Volleys in Cortical Networks. Network Simulations and Continuous Probabilistic Models”. *Hamilton, H. ed. Lecture Notes in Artificial Intelligence 1822*. (2000). Springer- Verlag.

[52] Thorpe, S. and Gaustrais, G. “Rank Order Coding”. Boser, J. (eds), *Computational Neuroscience: Trends in Research*. (1998). Plenum Press, New York.

[53] Thorpe, S., Delorme, A. and Van Rullen, R. “Spike-based strategies for rapid processing”. *Neural Networks*, 14(6–7). (2001). Pp. 715-725. doi: http://dx.doi.org/10.1016/S0893-6080(01)00083-1

[54] Gr¨uning5, A. E. and Sporea, I. “Supervised Learning of Logical Operations in Layered Spiking Neural Networks with Spike Train Encoding”. *Logical Operations in Spiking Neural Networks*. (2011). 272.

[55] Delorme, A. and Thorpe, S. “Face identification using one spike per neuron: resistance to image degradations”. *Neural Networks*, 14(6–7). (2001). Pp. 795-803. doi: http://dx.doi.org/10.1016/S0893-6080(01)00049-1

[56] Delorme, A., Gautrais, G., Rullen, R. and Thorpe, S. “A simulator for modeling large networks of integrate and fire neurons”. *Neurocomputing*. Volume 26-27. (1999). Pp. 989-996.

[57] Gueorguieva, N., Valova, I. and Georgiev, G. “Learning and data clustering with an RBF based spiking neuron”. *Journal of Experimental & Theoretical Artificial Intelligence*. 18(1). (2006). Pp. 73-86.

[58] Natschlager, T. and Ruf, B. “Spatial and temporal pattern analysis via spiking neurons”. *Network: Computation in Neural Systems*, 9(3). (1998). Pp. 319-338.

[59] Mishra, D., Yadav, A., Dwivedi, A. and Kalra, P. K. “A Neural Network Using Single Multiplicative Spiking Neuron for Function Approximation and Classification”. *Proceedings of the 2006 International Joint Conference on Neural Networks, 2006. IJCNN '06.* (2006). Pp. 396-403.

[60] Koch, C. *Biophysics of Computation: Information Processing in Single Neurons*. (1999). Oxford University Press: New York.

[61] Long, L.N., and Gupta, A. “Biologically-Inspired Spiking Neural Networks with Hebbian Learning for Vision Processing”. *46th AIAA Aerospace Sciences Meeting*. Jan 7-10. (2008). Reno, NV.

[62] Bonhoeffer, T., Staiger, V. and Aertsen, A. “Synaptic plasticity in rat hip pocampal slice cultures: local ’hebbian’ conjunction of pre- and postsynaptic stimulation leads to distributed synaptic enhancement”. *Proceedings of National Academy of Science USA*. 86(20). (1989). Pp. 8113-8117.

[63] Gerstner, W. Time structure of the activity in neural network models. *Phys. Rev. E*, Volume 51.(1995). pp. 738-758.

[64] Moore, S.C. *Back-propagation in Spiking Neural Networks*. (2002). M.Sc. Thesis, Department of Computer Science, University of Bath, UK.

[65] Jianguo, X. and Embrechts, M. J. “Supervised learning with spiking neural networks”. *Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN’01).* (2001). Pp. 1772-1772.

[66] Roman, N. *Teoretické otázky neuronových sítí*. Matfyzpress, Praha. (1996).

[67] Booij, O. and Nguyen, H. T. “A gradient descent rule for spiking neurons emitting multiple spikes”. *Inf. Process. Lett.*, 95(6), (2005). Pp. 552-558. doi: 10.1016/j.ipl.2005.05.023

[68] Suel, G.M., Garcia, O.J., Liberman, L.M. and Elowitz, M.B. “An excitable gene regulatory circuit induces transient cellular differentiation”. *Nature*, 440(7083). (2006). Pp. 545-550.

[69] Schrauwen, B. and Campenhout., J. V. (2007). “Parallel hardware implementation of a broad class of spiking neurons using serial arithmetic”. *M. Verleysen, editor, Proceedings of the 14th European Symposium on Artificial Neural Networks*. Evere, 624. (2007). Pp. 623–628.

[70] Cabrera, J. L. and Milton, J. G. (2002). On-Off Intermittency in a Human Balancing Task. *Physical Review Letters*, 89(15). (2002). 158702.

[71] Bohte, S.M. “The evidence for neural information processing with precise spike-timing: A survey”. *Natural Computing*, 3(2). (2005). pp. 195–206.

[72] Mergenthaler, K. and Engbert, R. “Modeling the control of fixational eye movements with neurophysiological delays”. *Phys. Rev. Lett*. 98(3): 138104. (2007).

[73] Debanne, D., Gähwiler, B. H. and Thompson, S. M. “Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures”. *The Journal of Physiology*, 507(1). (1998). pp. 237-247. doi: 10.1111/j.1469-7793.1998.237bu.x

[74] Nakazawa, K., Quirk, M.C., Chitwood, R.A., Watanabe, M., Yeckel, M.F., Sun, L.D., Kato, A., Carr, C.A., Johnston, D., Wilson, M.A. and Tonegawa, S. “Requirement for hippocampal CA3 NMDA receptors in associative memory recall”. *Science* 297 (5579). (2002). Pp. 211-218.