CS 290 Paper Summaries and Commentary

Ren Trista A. de la Cruz

November 17, 2020

Computing with Spikes

Computing with Spikes [3] gives a quick overview of the idea behind computing models based on *spiking neurons* and the (then) current research that the author Wolfgang Maass and his colleagues were conducting. A significant portion of the paper is dedicated to describing the mechanisms of a spiking neuron.

The field that studies spiking neuron models (or spiking neural networks) is in the intersection of the field of neuroscience and the field of theoretical computer science. Unlike abstract models of computation like Turing machines and counter machines in theoretical computer science, spiking neuron models are significantly more complex since they are used to model and study the computational process of the nervous system. Spiking neuron models should be abstract enough for them to be used as models of computation that can solve abstract problems (in theoretical computer science) but they should be detailed enough so that they can be used to explain phenomena in the nervous system (in neuroscience).

Artificial neural networks are computing models that are 'inspired' biological neural network but the activation mechanism (using real functions like sigmoid and trigonometric functions) of the neuron in these models does not represent how activation occurs in a biological neuron. The activation mechanism in a spiking neuron model more closely represents the activation mechanism of a biological neuron. Aside from the difference in activation mechanism, the organization of an artificial neural network is also different to the organization of a spiking neural network.

Maass et al.'s research involves determining the computational function of *neural microcircuits*. In a spiking neuron model, these microcircuits are model using spiking neurons. The research asks how exactly does a spiking neuron work. The research involves studying actual biological neurons in order to observe how they produce and process spikes. Other aspects of their research involve the study of how the spiking neurons are organized (into networks), how memory are stored in the networks, and how learning is done by the networks.

A large part of the paper describes the spiking neuron and gives a general idea behind the spiking mechanism. The neuron has three main parts: the *soma*, the *dendritic tree*, and the *axonal tree*. The soma is the body of the neuron that produces the signal called *spikes*. The dendritic tree is the 'input' region of the neuron where it receives signals from other neurons. The axonal tree is the 'output' region of the neuron where it sends out signals to other neurons. A part of the axonal tree (output region) of one neuron can be 'connected' to a part of the dendritic tree (input region) of another neuron. This 'connection' or interface between a neuron's axonal tree and another neuron's dendritic tree is called a *synapse*.

A neuron has a *membrane potential*, a voltage value based on the difference between the charge inside and the charge outside the membrane of the neuron. A neuron has a resting membrane potential which is about -70 millivolts. The 'spike' in a spiking neuron model refers to an *action potential* in a neuron. An action potential is a sudden increase (around 40 millivolts) and then a sudden decrease of the neuron's membrane potential (happens in less than 3 milliseconds). The term 'spike' refers to that even of the membrane potential spiking.

When a neuron receives a certain combination of inputs (spikes from other neurons received by the dendritic tree), it will produce a spike. There is threshold mechanism in a neuron. If the 'combination' of input spikes passes a certain threshold, the neuron will spike. The amplitude of the spike does not change for a neuron. The input spikes can dictate if the neuron will spike or not but not how 'large' the spike is. A series of input spikes can, however, dictate the timing (i.e. frequency) in which the neuron spikes.

When a neuron spikes, the spike travels along the axon then reaches the axon terminals. The axon terminals can be connected to dendrites of other neurons. The synapse, that connects the axon terminal to dendrite of another neuron, is responsible from 'processing' the spike and then passing a 'processed' spike to the next neuron. The synapse has an internal state/configuration. This state is affected by the spikes it receives. In a sense, this is a form of memory. This state affects how the synapse processed incoming spikes.

In the biological neuron, the spiking event is a result of a combination of electrochemical activities that involve the neuron membrane, voltage-gate ion channels, potassium and sodium ions, etc. The activities in a synapse involve neurotransmitters, neurotransmitter transporters and receptors, etc. In the spiking neuron models, these electrochemical activities are abstracted and simplified in order to have manageable models of computation but they are still somewhat faithful to their biological analogues (at least much more faithful compared to artificial neural network models).

A spiking neuron can only produce a spike, a single type of signal. A neuron can not produce a 'large' spike or 'small' spike. All spikes produced are identical. Information in spiking neuron models are encoded by spiking patterns (in time) produced by a neuron. This pattern in known as a *spike train.* For example, a spike can be represented by '1' and no spike represented by '0'. A spiking neuron's activities (spiking or resting) can be represented by a string (the spike train) over the binary alphabet $\{0, 1\}$. This string is a pattern in time instead of being a pattern in space. One can only observe the spike train by looking at the activities of a neuron for a certain period of time.

In summary, a spiking neuron model is a model of computation based on the spiking mechanism of a biological neuron. It is much more complex than other models of computation because it also has the role of modelling neural phenomena. Spiking neural network s are much more similar to biological neural network unlike artificial neural networks.

Third Generation Neural Networks: Spiking Neural Networks

Third Generation Neural Networks: Spiking Neural Networks [1] gives an overview of a spiking neural network model which a *third generation* neural network.

Neural networks can be categorized in to *generations*: the *first*, the *second*, and the *third* generation. The categorization is both historical and technical. On the historical side, the generation simply refers to the order in which the neural network is invented/introduced. First generation neural networks came first, followed by the second generation, then followed by the third generation. On the technical side, the paper focuses on the activation mechanism of the neurons in order to differentiate the neural networks from different generations.

The first generation networks use neurons that have an *integrate-and-fire with threshold* mechanism. The first generation neuron has an internal state that is the weighted sum of the input signals from other neurons. The integration process refers to the computation of the weighted sum (internal state). If the internal state passes a given threshold value for the neuron, then the neuron fires a signal. This process is the 'firing' process that is conditional on the internal state and the threshold value. The second generation networks use neurons that have an *integrate-and-activate* mechanism. Similar to the first generation, a neuron computes the weighted sum of the input signals. The weighted sum is then feed as input to an *activation function.* The activation function can be a hyperbolic function, a sigmoid (logistic) function, a binary step function (similar to the threshold mechanism in first generation neuron), etc. The output of the activation function will be the output of the neuron. Unlike first generation neurons with a single (discrete) type of signal, the output of second generation neurons is continuous. If the activation function used by a second generation network is continuous and differentiable, then the *back-propagation* algorithm can be used to train the network on labelled data.

The first and second generation networks have a lot in common with each other. In fact, the first generation networks are a special case of the second generation networks. Both generations have the integration process of computing the weighted sum and they both use activation functions. First generation networks specifically use the binary step function while the second generation networks can use any activation functions. Additionally, both first and second generation networks are synchronized. This means at every step in the computation, all neurons in the network perform integration followed by function activation at the same time. All neurons produce output signals at the same time.

Third generation networks use *spiking* neurons. A spiking neuron is similar to a first generation neuron in two ways. It outputs a single type of signal called a *spike* and a spike is produced if the internal state of the spiking neuron reaches a specified threshold. What differentiate third generation networks with the first and the second generation networks are the definition of the internal state of a neuron and the timing of the firing of neurons in the network.

In a spiking neuron, the internal state is more of a continuous value that changes in time and is affected the incoming input spikes. In a spiking neural network (third generation), neurons are not synchronized. A spiking neuron can fire any time as long as its internal state passes the threshold specified for the neuron. For example, neuron A and neuron B are connected to neuron C which means they send the spikes they generate to neuron C. It is possible that in a given time period, both neuron A and neuron B send one spike each to neuron C but at different points in time. The arrival times of input spikes from neuron A and neuron B affect the resulting internal state of neuron C. The resulting internal state of neuron C after receiving both spikes at the same time will be different to the resulting internal state if the spikes arrived at different times. The resulting internal state of neuron C after receiving a spike from neuron A at time t and a spike from neuron B at time $t + d$ will be different to the resulting internal state of neuron C if it received neuron A's spike at time t and neuron B's spike at time $t + d'$ (where $d \neq d'$). The timing of spikes are important in a spiking neural network. In general, in a given time period, we can look at the outputs of neuron A and neuron B as sequences of spikes. These sequences are called *spikes trains*. A spike train is a pattern of spikes in time.

Third generation networks do not all fall under one model. There are different spiking neural network models. Different models may have different mechanisms for the spiking neuron. They can also have different ways of connecting the neurons and building the network. Different spiking neuron models have different ways of defining how sets of spike train inputs affect the internal state of a neuron. It is possible to have very detailed spiking mechanism that models the behavior of a biological spiking neuron. Some spiking neuron models use a system of differential equations to define how the internal state changes with respect to input spike trains. These kind of models are computationally intensive to simulate. i.e. The computer is essentially solving differential equations for each neuron in the networks. Such models are better from a neuroscience perspective but worse for a computational perspective. Spiking neural networks in used in computer science primarily use a much simpler spiking neuron model. Additionally, it is easier to adapt the back propagation algorithm to simpler spiking neuron models.

The paper presents a particular spiking neural network model where, similar to a common feed-forward second generation neural network, neurons are arranged into layers and each neuron in one layer is connected to all neurons in the next layer. The difference is that, in this particular spiking neural network model, the ‘connection’ from one neuron to another neuron in the next layer is actually a set of K connection from one neuron to the next. For example, if the model has $k = 5$, when neuron A is connected to neuron B then there are $k = 5$ connections from neuron A to neuron B. Each connection has its own weight and delay. When neuron A sends a spike, a copy of the spike will go through the five connections.Initially, the output of neuron A will appear as five separate spike trains which are possibly different from each other due to different delays in the connections. One can look at these spike trains as discrete (square) waves. In each of the connection, there is a *synapse* that transforms the discrete wave into a more smooth (continuous) wave like a sine wave. The synapse also ‘amplifies’ this smooth wave by the weight associated with the connection. The discrete wave (spike train) is called the *pre-synaptic potential* while the continuous wave is called the *post-synaptic potential.* One can combine these post-synaptic potentials (continuous waves) from all connections into a single continuous wave. In summary, one can look at the output of a neuron as set of spike trains that are transformed into continuous waves and combined into a single continuous wave.

A neuron will receive multiple continuous waves (inputs) from all neurons in the previous layer. The integration performed by the neuron is the process of combining all these continuous wave inputs. If some of the peaks in the combined wave input passes the threshold specified in the neuron, then the neuron will send a spike out. For a combined continuous wave input, the spiking neuron will produce a spike train output.

Both input (continuous wave, post-synaptic potential) and output (discrete wave, spike train) of a spiking neuron are waves which means they are pattern in time. One can look at spiking neural networks as another form of generalization. i.e the generalization of the second generation networks by allowing input and output as sequences (in the form of waves) instead of single values (in the second generation network, weighted sum is the input and the function result is output). In this case, the second generation networks will appear as spiking neural networks that exclusively deal with sequences with single elements (single value input and output).

Spiking neural networks are particularly effective when it comes to learning time-dependent patterns (time series). They can also be use to create smaller networks (fewer neurons/layers) if one can effectively encoding both input and output as patterns in time. For example, in a second generation neural network for image processing, all features (pixel values) are fed to the network at the same time. If the image is large, the network that processes the image will also be large. By encoding parts of the images (sets of pixels) as sequences, one can create a smaller spiking neural network that can deal with these sequences as input and produced the corresponding output sequences. Finding the time-based encoding for both input and output is not a trivial task.

One disadvantage of using spiking neural network is its computationally intensive training. There are a lot of parameters (different weights and delays of the connections) the training algorithm needs to adjust. The error surface of a spiking neural network is also highly uneven. i.e. A slight change in the delay/weight of one of the connections from one neuron to next neuron can result in a disproportionate change in the output spike train.

Network of Spiking Neurons: Third Generation Neural Network Models

Network of Spiking Neurons: Third Generation Neural Network Models [2] compares the computational power of a particular spiking neural network model with first and second generation neural networks. Specifically, [2] compared spiking neural networks with neural networks that use McCulloch-Pitts/threshold gates (first generation) and neural networks that use sigmoidal gates (second generation).

The spiking neural network model used in the paper was defined formally. The spiking neural network model was introduced by the author in previous works. In [2], a spiking neural network is composed of a set V of spiking neurons. The set $E \in V \times V$ is the set of synapses that connect neurons together. Each synapse has an associated weight. The weight of the synapse that connects neuron u to neuron v is a non-negative real number denoted as $w_{u,v}$. At any given time t, some neuron v will have an associated *potential* denoted as $P_v(t)$. Aside from a weight, a synapse also has an associated *response function* $\varepsilon_{u,v}$. If neuron u fires at time s, then at time t the value $\varepsilon_{u,v}(t-s)$ computed using the response function is a factor that contributes to the value of the neuron v's potential $P_v(t)$. Specifically, the term $w_{u,v} \cdot \varepsilon_{u,v}(t-s)$ contributes to the potential $P_v(t)$. i.e.. $P_v(t) = \sum_{u:(u,v)\in E} \Big(\sum_s w_{u,v} \cdot \varepsilon_{u,v}(t-s) \Big)$. Neuron v also has an associated *thresholding function* Θ_v. At time t, the threshold at neuron v is the value $\Theta_v(t - t')$ where t' is the last time neuron v fired. At time t, neuron v fires if its potential $P_v(t)$ is above the threshold $\Theta_v(t - t')$.

Two types, *Type A* and *Type B*, of spiking neural network were compared to threshold gates neural network and sigmoidal gates neural networks. *Type A* spiking neural networks use piecewise constant ('step') functions for both the response and thresholding functions. *Type B* spiking neural networks use continuous piecewise linear functions for both the response and thresholding functions.

Threshold gates neural networks can be used to compute boolean functions. Since type A spiking neural networks are very similar to threshold gates neural networks then one can see that a threshold gates neural network that computes a particular boolean function can easily be converted to a type A spiking neural network that computes the same function. This means that in terms of boolean function computation, type A spiking neural networks is at least as powerful as threshold gates neural network.

The paper also show a stronger result that states that type A spiking neural networks are more 'computationally powerful' than first and second generation neural networks of the same size when it comes to computing boolean functions. A more specific statement of these result is that there are boolean functions whose spiking neural networks (the networks that compute the functions) are smaller (in terms of the number of neurons) than smallest first and second generation networks that compute the same functions. i

An instance of these boolean function is the *coincidence-detection* function CD_n. CD_n takes $2n$ boolean variables: $x_1, ..., x_n, y_1, ..., y_n$. $CD_n(x_1, ..., x_n, y_1, ..., y_n) = 1$ if there is at least one pair (x_i, y_i) such that $x_i = y_i = 1$. Otherwise, $CD_n(x_1, ..., x_n, y_1, ..., y_n) = 0$. Excluding input neurons, the paper showed that threshold gates neural network need at at least $n/log(n+1)$ neurons while sigmoidal neural networks with piecewise polynomial activation function needs $\Omega(n^{1/2})$ neurons to compute the function CD_n. On the other hand, with proper time encoding, a single spiking neuron can compute CD_n.

There are similar results for functions that take vectors of non-negative real numbers as input and have boolean output. i.e. Functions of the form: $F : (\mathbb{R}^+)^n \rightarrow \{0, 1\}$. There are functions of that form that can be computed by smaller spiking neural networks that are significantly smaller than the smallest first and second generation networks that can compute the same functions.

Spike-based Strategies for Rapid Processing

Spike-based Strategies for Rapid Processing [4] discusses different spiked-based information coding as alternative to the de facto standard rate-based information coding.

Rate-based coding (or rate coding for short) is the idea that information is conveyed by the rate of firing of a nerve cell. For example, in the human visual system, information from the retina is eventually transmitted to the visual cortex. Different visual stimuli will change the firing rates of neurons in the visual cortex. The paper states that (at the time of writing) neurophysiologists often assume that all useful information can be summarized by plots of neuron firing rate as a function of time.

There have been experiments that strongly suggest that there are situations where rate coding seems to be insufficient for tasks. For example, there are visual processing tasks like image recognition that can be done quickly and so they require rapid processing. Some visual tasks (e.g. reacting to visual stimuli) can be done in 100 to 150 milliseconds. If one combines this information with the observation that neurons rarely fire above 100 Hz and the fact that there are around 10 layers of neurons between the eye's photoreceptor and neurons in the visual cortex, then one can see that the neurons in those ten layers can fire at most once. i.e. At the rate of 100 Hz, the average firing time of a neuron is 10 ms. Given the 10 layers of neurons from photoreceptor to a neuron in the visual cortex, a spike that travels that path will need at least 100ms. This means neurons in those 10 layers only has two possible rates: 0 spikes per unit of time or 1 spike per unit of time. Those neurons can only process and communicate 1 bit of information when using rate coding. Other sensory pathways require an even faster processing. Bats, for example, can process auditory stimuli in 8ms. Experiments that demonstrate the need for rapid processing suggest that rate coding does not support the required bandwidth of the input (visual/auditory stimuli).

The paper [4] discusses some spike-based coding that can be used as alternative to rate coding. The paper looks at a group of n neurons and measure the amount of information they can transmit using the different spike-based coding. *Count coding* simply counts the number of spikes sent by the n neurons. If there are n neurons, then one can receive 0 to n spikes. There are $n + 1$ possible values that those n neurons can communicate. This results in the count coding (for n neurons) having a bandwidth of $log_2(n + 1)$ bits. *Binary coding* simply looks at the output (spike or no spike) of the

n neurons. Given some ordering, one can look at the sequence of neuron outputs as a bit string of length n. This means binary coding (for n neurons) has the bandwidth of n bits. *Timing coding* looks at the arrival time of the spikes from n neurons. In timing coding, if one is observing t time intervals, then the arrival time of the spike can be in any of those t time intervals. The arrival time of the spike can specify one of t values which means a single neuron can transmit $log_2(t)$ bits. Given n neurons, the bandwidth of timing coding is $n \cdot log_2(t)$ where t is the number of time intervals being observed. *Rank order coding* looks at the arrival time of spikes from n neurons and rank the neurons in terms of the spike arrival time. There are $n!$ possible ranking which means rank order coding has one of the highest bandwidth which is $log_2(n!)$ bits.

Commentary on Spiking Neural Networks

Spiking neural networks bring the concept of *pattern/information in time* (back) to neural computing. There are patterns that are inherently patterns in time. Time series data, music or audio, and even texts are patterns in time because they contain information ordered in time. Since first and second generation neural networks do not encode information in time, time-based patterns are processed by these networks by looking at them as patterns in space. For example, neural networks that process audio usually represent audio using its spectrogram. A spectrogram is a two-dimensional image that visualizes the spectrum of frequencies of the audio/signal. One can see that one of the first steps taken to process audio (pattern in time) is to transform it from the time domain to frequency domain. One dimension of the spectrogram specifies time while the other dimension specifies the set of frequencies available. Even though a spectrogram still has a time dimension, neural networks that process as 2D images taking the whole spectrogram as input in a single step. In spiking neural networks, patterns in time can be fed to the network as is since the network can ‘stream’ these patterns instead of taking the entire pattern once as a pattern in space. Streaming patterns in time increases the time for the network to take the input but it can eliminate of other steps like preprocessing (i.e. converting audio to spectrogram) and more compact networks can be created (since the pattern is streamed and not taken all at once).

We can see in [4] that the type of coding scheme used to encoded information is important in order to maximize information bandwidth that a group of neurons can transmit. This idea is applicable for both networks that process time-based patterns and networks that process space-based patterns. Count coding and binary coding both encode information in space but binary coding produces better bandwidth. Timing coding and rank order coding both encode information in space and in time but rank order coding produces a significantly better bandwidth.

There is no single spiking neural network model. There are different spiking neural network models. For example, the spiking neural network model described in [2] is different from the model described in [1]. Different spiking neural network models can have different ways of coding information because their spiking neuron mechanisms can be different from each other. The spiking neuron mechanism will dictate what kind of coding can be used in the model. i.e. Some spiking neuron mechanisms can only process specific types of coding. This means that when you are trying to use a spiking neural network to perform some computation you either already have a type of coding (for the input/output of your computation) in mind and you need to select a spiking neural network model that can use that type of coding or you can select a spiking neural network model and work with the types of coding available in that model.

References

[1] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Third generation neural networks: Spiking neural networks. In Wen Yu and Edgar N. Sanchez, editors, *Advances in Computational Intelligence*, pages 167–178, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

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

[3] Wolfgang Maass. Computing with Spikes. *Special Issue on Foundations of Information Processingof TELEMATIK*, 2002.

[4] Simon Thorpe, Arnaud Delorme, and Rufin Van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14(6-7):715–725, July 2001.