Computation with Spiking Neurons*

Wolfgang Maass

Institute for Theoretical Computer Science

Technische Universität Graz

Klosterwiesgasse 32/2, A-8010 Graz, Austria

maass@igi.tu-graz.ac.at, http://www.cis.tu-graz.ac.at/igi/maass

Tel.: +43 316 873-5822, Fax.: +43 316 873-5805

1 Introduction

Computation in the brain is primarily carried out by *spiking neurons*. These are computational units that differ strongly from the computational units

*Research for this article was partially supported by the ESPRIT Working Group NeuroCOLT, No. 8556, and the Fonds zur Förderung der wissenschaftlichen Forschung (FWF), Austria, project P12153.

of artificial neural network models. A spiking neuron *fires* at certain points in time, thereby sending through its axon (and its dendrites) a stereotyped electric pulse which is commonly referred to as *action potential* or *spike*. The size and shape of a spike is independent from the input to the neuron, but the *time* when a neuron fires depends on its input.

In virtually every artificial computing machine one is keen on making sure that the timing of individual computation steps adheres to a global schedule, which is independent of the values of the input variables. For example, a gate on layer d of a feedforward neural network is required to produce its output at step d of the computation – regardless of the values of the inputs to the network. In contrast to that, the firing times of neurons in a biological neural system depend on the input to that system. Hence networks of spiking neurons are capable of exploiting time as a resource for coding and computation in a much more sophisticated manner than virtually all other common computational models. Consequently an analysis of computations in networks of spiking neurons requires us to re-examine the computational role of time and space, which are the most fundamental dimensions of any computation. Some of the new insights resulting from this research have triggered the design of novel electronic hardware for computing with pulses, especially in the area of real-time sensory processing with low power analog VLSI circuits.

We will review in this article currently existing theoretical models and paradigms for computing in networks of spiking neurons, and we will compare their computational power with that of common artificial neural network models.

2 A Formal Model for Computation in a Network of Spiking Neurons

A spiking neuron j fires at certain time points $t_j^{(1)}, t_j^{(1)}, \ldots,$ sending each time a stereotyped spike down its axonal tree. At the synapses that connect these axonal branches with other neurons the spike triggers the emission of neurotransmitter, thereby causing positive pulses (excitatory postsynaptic potentials: EPSP's) or negative pulses (inhibitory postsynaptic potentials: IPSP's) on the membrane potential of the postsynaptic neurons i. The time course of these EPSP's and IPSP's resulting from a firing of neuron j at time $t_j^{(f)}$ can be described by terms of the form $w_{ij}\varepsilon(t - t_j^{(f)})$, where the function $\varepsilon(t - t_j^{(f)})$ has value 0 if $t - t_j^{(f)}$is less than the transmission time Δ_{ij}

from the soma of neuron j to the soma of neuron i, then increases almost linearly and finally decays exponentially back to 0. The factor w_{ij} is positive in the case of an EPSP and negative in the case of an IPSP. There exist various mathematical models that predict the firing times $t_i^{(1)}, t_i^{(2)}, \ldots$ of a biological neuron i in dependence of the firing times $t_j^{(f)}$ of "presynaptic" neurons j. Because of its mathematical simplicity, the most useful model for a computational analysis is the *spike response model* (see INTEGRATE-AND-FIRE NETWORKS). Here the membrane potential $u_i(t)$ at time t at the soma of neuron i is written as a sum $\sum_{j,f} w_{ij}\varepsilon(t - t_j^{(f)})$ consisting of EPSP's and IPSP's that were caused by firings of presynaptic neurons j at preceding times $t_j^{(f)}$, and a refractory term $\eta(t - \hat{t}_i)$ that depends on the last time $\hat{t}_i$ when neuron i has fired before time t. If $t - \hat{t}_i$ is positive but very small (less than 2 ms) then $\eta(t - \hat{t}_i)$ is strongly negative, whereas for larger values of $t - \hat{t}_i$ its value returns back to 0. According to the spike response model the neuron i fires whenever the resulting sum $u_i(t) = \eta(t - \hat{t}_i) + \sum\limits_{j,f} w_{ij}\varepsilon(t - t_j^{(f)})$ reaches a certain firing threshold ϑ from below. In comparison with the Hodkin-Huxley model and the standard equations for integrate-and-fire neurons this spike response model is much easier to handle for the purpose of analyzing computations, since its formulation does not involve differential equations.

On the other hand, with suitable choices of the response functions η and ε (which may depend on the neurons i and j) this model is able to capture quite well the dynamics of biological spiking neurons (see INTEGRATE-AND-FIRE NETWORKS).

3 Computing with one Spike per Neuron

Typical models for artificial neural networks (ANALOG NEURAL NETWORKS) assume that the inputs and outputs of a computation consist of analog values (i.e, rational or real numbers), or just of binary values. Hence, in order to compare their computational power with that of networks of spiking neurons, we have to adopt some convention for transforming analog or binary values into spikes, and vice versa – just for the input and output of the network of spiking neurons (internally the network of spiking neurons may use spikes in any way it wants). Apparently biological networks of spiking neurons utilize various different schemes for encoding information via spikes, hence there is no unique "biologically realistic" choice.

If the spiking neurons are not subject to significant amounts of noise then one can carry out computations in network of spiking neurons where every spike matters, comparable to the role of bits in a digital computer. In fact, it can be rigorously proven for this case that some finite network of spiking neurons can simulate a universal Turing machine, and hence can in principle simulate *any* digital computer [Maass, 1996]. However, since time is continuous, a single spike can in principle carry even more information than a single bit: its arrival time t may encode an analog number $t-T$ consisting of many bits (where T is some reference time). One of the most intriguing computational questions regarding networks of spiking neurons is the question which analog functions they can compute efficiently via this type of "temporal coding" of inputs (and outputs), which is often referred to as *delay coding*. The first remarkable observation is that for such code a spiking neuron can carry out computational operations that have no counterpart in the computational units of traditional neural network models: a spiking neuron can act as *coincidence detector* for incoming pulses by firing only if a certain number of EPSP's arrive almost simultaneously at its soma [Abeles, 1982]. Hence if the arrival times of the incoming pulses (EPSP's) encode *analog numbers* x_j, a spiking neuron can detect whether some of these numbers x_j have (almost) equal value. For example, a spiking neuron can compute a function $ED_n : \mathbb{R}^n \to \{0,1\}$ with

$$ED_n(x_1, \ldots, x_n) = \begin{cases} 1 \quad , & \text{if there are } j \neq j' \text{ so that } x_j = x_{j'} \\ \\ 0 \quad , & \text{if } |x_j - x_{j'}| \geq c \text{ for all } j \neq j' \quad , \end{cases}$$

where c is some suitable parameter. In this implementation the binary output of this function ED_n is encoded by firing/non-firing of the spiking neuron. The parameter $c > 0$ allows a single spiking neuron to compute this function without having to make hair-trigger decisions – such as deciding whether $x_1 = x_2$ or $x_1 \neq x_2$. Therefore a spiking neuron i is in principle able to compute this function ED_n (with a suitable choice of its firing threshold ϑ) even if there is a little bit of noise on its membrane potential $u_i(t)$. This is remarkable, since artificial neural networks need a substantial amount of hardware for computing this function ED_n (even if the network is allowed to give any output it wants if neither of the two cases in the definition of ED_n apply).

Theorem 3.1 *[Maass, 1997b]*

a) *Any threshold circuit (i.e., any layered feedforward circuit consisting of McCulloch-Pitts neurons) that computes ED_n needs to have at least $\frac{n}{2}\log n$ neurons on its first hidden layer.*

b) Any feedforward analog neural net (i.e., any multi-layer-perceptron consisting of sigmoidal neurons) that computes ED_n needs to have at least $\frac{n-6}{2}$ hidden units.

One can exploit the coincidence-detection-capability of a spiking neuron i even further and raise the firing threshold ϑ so high that EPSP's caused by firings of *all* presynaptic neurons have to arrive nearly simultaneously at the soma of i to make it fire. In this case the spiking neuron can act in the temporal domain like an RBF-unit (i.e., radial basis function unit) in traditional neural network models. If the transmission delays Δ_{ij} from the trigger zone of neuron j to the same of neuron i all have the same value, then such neuron with a high threshold ϑ fires if and only if the firing times t_j have almost the same value for all presynaptic neurons j. If these transmission delays Δ_{ij} vary for different j, then the same neuron i will fire if and only the firing times $\langle t_j \rangle_j$ of the presynaptic neurons j form a certain translation invariant temporal pattern: i fires whenever $t_j \approx T - \Delta_{ij}$ for all j, where T is some number that is independent of j. Hence one may view the vectors $\langle T - \Delta_{ij} \rangle_j$ for $T \in \mathbb{R}$ as the points in the center of some generalized RBF-like unit for the temporal domain that is implemented by the spiking neuron i. This possibility of using spiking neurons as RBF-like computational units in the temporal domain was first observed by Hopfield [Hopfield, 1995]. In the same article Hopfield demonstrates an advantageous consequence of a 1-dimensional translation invariance for vectors in the center of such RBF-like unit: If $t_j = \log u_j$ for some external sensory input variables u_j , the spiking neuron i can detect a pattern $\langle u_j \rangle_j$ formed by these input variables in a manner that is invariant with regard to multiplication of all u_j by a common factor $\lambda > 0$. Since $\log \lambda u_j = \log \lambda + \log u_j$ the pattern detection by the spiking neuron is not affected by this factor λ. It has been argued that this very useful mechanisms may contribute to the amazing ability of biological organisms to classify patterns over a very large scale of intensities, such as for example visual patterns under drastically different lighting conditions.

In [Natschläger and Ruf, 1998] the previous construction of an RBF-unit for temporal patterns has been extended to a self-organizing RBF-network with the help of lateral inhibition between RBF-units.

A characteristic feature of the previously discussed computation of the function ED_n and the simulation of an RBF-unit is the *asymmetry* between coding schemes used for *input* and *output*. Whereas the input consisted of a vector of analog numbers, encoded through temporal delays, the output of the spiking neuron was just binary, encoded through firing or non-firing of that neuron. Obviously for multilayer or recurrent computations with spiking neurons it is desirable to have computational mechanisms that enable a layer of spiking neurons to *output* a vector of analog numbers that is encoded in the same way as the input. For that purpose one needs a mechanism for *shifting* the firing time of a spiking neuron i in dependence of the firing times t_j of presynaptic neurons. Such mechanism is provided by the fact that an earlier firing time t_j tends to advance the firing time of neuron i, and it does so even more if the intermediate synaptic weight w_{ij} has a large value. This can be used to compute a weighted sum in the temporal domain, and yields in combination with a straightforward implementation of a nonlinear activation function via spiking neurons:

Theorem 3.2 *[Maass, 1997a] Any continuous function $F : [0,1]^n \rightarrow [0,1]^m$ can be approximated arbitrarily closely by a network of spiking neurons with inputs and outputs encoded by temporal delays (relative to some reference time).*

Thorpe and his coworkers have proposed to view the *order* of the firing times t_j of different neurons j as the relevant signal conveyed by these neurons, and have explored possible neuronal mechanisms for computing in terms of such code [Thorpe and Gautrais, 1997]. One may view this code as a digitalized version of the previously discussed delay code.

4 Computing on Spike Trains

In the preceding section we had focused on computations where inputs and outputs consisted of a single spike per channel. In this section we will look at the case where inputs and outputs consist of *spike trains.* This can be used for computations on analog variables that are represented in a highly noise-robust manner, or for computations on time-series. A natural way of using spike trains as code for analog time series – which does occur in nature [Rieke et al., 1997] – results from convolving each spike with a suitable smooth kernel function.

The most traditional – but biologically quite problematic – scheme for converting spike trains into analog numbers and vice versa is the *firing rate code*, where one interprets the number of times when a neuron i fires (during a time interval of some unit length) as the output of neuron i. If the firing rates involved are sufficiently high relative to the time lengths of EPSP's and IPSP's, a spiking neuron can approximate for this code the input/output behavior of a sigmoidal neuron, i.e., of the common computational unit of an artificial neural network. However firing rates of neurons in the cortex are relatively low (10 Hz on average for higher cortical areas) whereas most EPSP's last for just 20 ms. In addition in many biological neural systems the amplitude of EPSP's tends to become smaller for high presynaptic firing rates (due to inherent synaptic dynamics that is not reflected in the simple formal model discussed in this article), making it even harder for the postsynaptic neuron to "read" the firing rate of a presynaptic neuron. Hence it is not clear to what extent a network of biologically realistic models for neurons and synapses can carry out computations in terms of such firing rate code.

Networks of spiking neurons are more suitable for carrying out complex analog computations if the inputs of the computation are presented in terms of a *space-rate code* or *population code* where the fraction of neurons within a certain pool that fire during a specific short time interval (say of length 5 ms) encodes an analog number. It is shown in [Maass and Natschläger, 1998] that feedforward networks of spiking neurons can approximate any given continuous multi-variable function if the network inputs and outputs are assumed to encode analog numbers through such space-rate code. Furthermore, with the help of lateral and recurrent connections one can utilize the tails of EPSP's and IPSP's (i.e., their exponentially decaying segments) to implement linear filters via networks of spiking neurons. In fact, via the same mechanisms, or by using instead of the static synapses (represented by the constant factors w_{ij} in the model discussed in section 2) biologically realistic *dynamic synapses*, a feedforward network of spiking neurons can approximate with regard to space-rate coding of time varying analog inputs and outputs a very large class of nonlinear -filters: any nonlinear filter that is time-invariant and has fading memory [Maass and Sontag, 1998].

Due to their coincidence-detection capabilities spiking neurons are quite sensitive to statistical correlations in the firing times of presynaptic neurons (often referred to as "synchronization"). Since the number of possible relations among presynaptic neurons that can be encoded in this way is exponential in the number of presynaptic neurons, a coding mechanism that involves correlations in firing times may substantially enlarge the expressive capabilities of neural networks. The precise computational operations that can be carried by spiking neurons in the presence of a larger number of competing "relations" encoded through firing correlations of presynaptic neurons has remained elusive (see [Milner, 1974], [von der Malsburg, 1981], [Eckhorn et al., 1988], [Maass, 1998]).

Synfire chains [Abeles, 1991] represent a somewhat related model. Each synfire chain consists of a layered feedforward circuit of spiking neurons, where firing activity is propagated from layer to layer. From the point of view of computational power a single synfire chain just implements a binary register (through its two states active/non-active), but via shared neurons the activity of one or several synfire chains may ignite activity in yet another synfire chain. In this fashion one can implement an AND-gate via a (fairly large) network of spiking neurons. Analog versions of such computations are discussed in [Maass and Natschläger, 1998].

5 Discussion

Networks of spiking neurons represent a novel class of computational models where the timing of individual computation steps carries salient information. It is known that they can carry out all computations that standard neural network models can handle, some of them even with substantially fewer gates. In contrast to most artificial neural network models, networks of spiking neurons are particularly well suited for carrying out computations on analog time series. The exploration of possible advantages of *artificial* networks of spiking neurons is just beginning. Implementations of networks of spiking neurons in novel analog VLSI circuits (see ANALOG VLSI FOR NEURAL NETWORKS) offer new ways for real-time sensory processing at the micro-Watt level. More detailed survey articles on the theory and hardware implementation of computing with spiking neurons can be found in [Maass and Bishop, 1999].

References

[Abeles, 1982] Abeles, M. (1982). Role of the cortical neuron: integrator or coincidence detector? *Israel J. Med. Sci.*,18:83–92.

[Abeles, 1991] Abeles, M. (1991). *Corticonics.* Cambridge University Press, Cambridge.

[Eckhorn et al., 1988] Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., and Reitboeck, H. J. (1988). Coherent oscillations: A mechanism of feature linking in the visual cortex? Multiple electrode and correlation analysis in the cat. *Biological Cybernetics*, 60:121–130.

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

36.

[Maass, 1996] Maass, W. (1996). Lower bounds for the computational power of networks of spiking neurons. *Neural Computation*, 8(1):1–40.

[Maass, 1997a] Maass, W. (1997). Fast sigmoidal networks via spiking neurons. *Neural Computation*, 9:279–304.

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

[Maass, 1998] Maass, W. (1998). A simple model for neural computation with firing rates and firing correlations. *Network: Computation in Neural Systems*, 9:1–17.

[Maass and Bishop, 1999] Maass, W., and Bishop, C., eds. (1999) *Pulsed Neural Networks*, MIT Press, (Cambridge).

[Maass and Natschläger, 1998] Maass, W. and Natschläger, T. (1998). A model for fast analog computation based on unreliable synapses. *Neural Computation*, in press.

[Maass and Sontag, 1998] Maass, W. and Sontag, E. D. (1998). Neural systems as nonlinear filters. *Neural Computation*, in press.

[Milner, 1974] Milner, P. M. (1974). A model for visual shape recognition. *Psychological Review*, 81(6):521–535.

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

[Rieke et al., 1997] Rieke, F., Warland, D., de Ruyter van Steveninck, R. R., and Bialek, W. (1997). *Spikes – Exploring the Neural Code*. MIT Press, (Cambridge), MA.

[Thorpe and Gautrais, 1997] Thorpe, S. J. and Gautrais, J. (1997). Rapid visual processing using spike asynchrony. *Advances in Neural Information Processing Systems, vol. 9*, MIT Press, (Cambridge), MA, 901–907.

[von der Malsburg, 1981] von der Malsburg, C. (1981). The correlation theory of brain function. *Internal Report 81-2 of the Dept. of Neurobiology of the Max Planck Institute for Biophysical Chemistry in Göttingen*, Germany. Reprinted in *Models of Neural Networks II*, Domany et al., eds., Springer, 1994, 95–119.