Complexity of Boolean Computations for a Spiking Neuron*

Michael Schmitt†
Lehrstuhl Mathematik und Informatik
Fakultät für Mathematik
Ruhr-Universität Bochum
D-44780 Bochum, Germany
mschmitt@lmi.ruhr-uni-bochum.de

Abstract

We investigate the computational power of a model for a spiking neuron in the Boolean domain by comparing it with traditional neuron models such as threshold gates (or McCulloch-Pitts neurons) and sigma-pi units (or polynomial threshold gates). In particular, we estimate the number of gates required to simulate a spiking neuron by a disjunction of threshold gates and we establish tight bounds for this threshold number. Furthermore, we analyze the degree of the polynomials that a sigma-pi unit must use for the simulation of a spiking neuron. We show that this degree cannot be bounded by any fixed value. Our results give evidence that the use of continuous time as a computational resource endows single-cell models with substantially larger computational capabilities.

1 Introduction

Models of single neurons like the threshold gate or the sigmoidal gate are theoretically well investigated and widely used in applications. These and related types of models, however, do not capture the phenomenon of timing which is believed to be essential for computations in biological neural systems (see e.g. Rieke et al. [1]). Spiking neurons, or leaky-integrate-and-fire neurons as they are called in biophysics and theoretical neurobiology, are an attempt to explore the computational significance of this phenomenon.

We investigate in this paper a simple version of a spiking neuron, the so-called "spiking neuron of type A" according to the terminology of Maass [2]. Despite its simplicity, the model has sufficient qualities so that one can observe the impact that the use of time as a resource has on the computational capabilites of a single neuron.

We focus on the computation of Boolean functions. In Section 2 the definition of the model and references to related work are given. We compare the computational capabilities of a spiking neuron with those of traditional neuron

*Work supported in part by the ESPRIT Working Group NeuroCOLT No. 8556

†This research was done while the author was with the Institute for Theoretical Computer Science at the Technische Universität Graz, A-8010 Graz, Austria.

models: threshold gates and sigma-pi units. Disjunctions of threshold gates are universal computational devices for the class of Boolean functions provided that the number of gates is not restricted. Therefore, the question arises how many threshold gates are required to simulate a spiking neuron by a disjunction of threshold gates. We analyze this threshold number of a spiking neuron in Section 3. A single sigma-pi unit is also capable to compute any Boolean function provided that there is no bound on the degree of the polynomial that the unit may use. In Section 4 we study the degree that a sigma-pi unit must have to be able to simulate a spiking neuron.

Due to space limitations, all proofs are omitted in this extended abstract. They can be found in the complete version [3].

2 Basic concepts and related work

A *Boolean function* on n variables is a function $f : \{0,1\}^n \to \{0,1\}$. A *threshold gate* (or *McCulloch-Pitts neuron*) with n inputs is characterized by $n+1$ real numbers: the *weights* $w_1, \ldots, w_n$ and the *threshold* θ. It computes a Boolean function f where $f(x_1, \ldots, x_n) = 1$ if and only if $\sum_{i=1}^{n} w_i x_i \geq \theta$. A Boolean function computable by a threshold gate is also called *threshold function*.

In this paper we focus on a simple model for a spiking neuron, the so-called "spiking neuron of type A" [2]. For the sake of brevity we will refer to it as *a spiking neuron* henceforth throughout this paper.

A *spiking neuron* v receives inputs in the form of short pulses, also known as *spikes*, from n input neurons $a_1, \ldots, a_n$. For $i = 1, \ldots, n$ there is a connection from a_i to v characterized by two numbers: the weight $w_i \in \mathbb{R}$ and the delay $d_i \in \mathbb{R}^+$ (where $\mathbb{R}^+ = \{x \in \mathbb{R} : x \geq 0\}$). Time is treated as a continuous variable denoted by t. We assume that if input neuron a_i *fires*, that is, emits a spike, at time t_i, this generates a rectangular pulse in v of the form $h_i(t - t_i)$ with

$$h_i(t) = \begin{cases} 0 & \text{for } t < d_i \text{ or } t \geq d_i + 1 \ , \\ w_i & \text{for } d_i \leq t < d_i + 1. \end{cases}$$

Neuron v is assumed to fire as soon as the sum

$$P_v(t) = \sum_{i=1}^{n} h_i(t - t_i)$$

of these pulses reaches a certain threshold θ_v. The functions h_i, known as *postsynaptic potentials*, model the effect that a firing of neuron a_i has on the *membrane potential* P_v at the so-called trigger zone of v, which is the location where the decision is made whether a spike is to be emitted by v.

The firing threshold θ of a biological neuron is a function depending on the time which has passed since its last firing. For Boolean computations we assume that the neuron has not fired for a sufficiently large period of time, so that its firing threshold has returned to its so-called resting value. Therefore we treat the threshold θ as a constant throughout this paper.

The neuron model that we consider here is a simple version of a *leaky integrate-and-fire neuron*. Further discussions of this and other neuron models can be found in the brief article by Softky and Koch [4] or in the surveys by Gerstner [5] and Maass [2]. The latter also contain comprehensive lists of references to relevant literature from biophysics and neurobiology. The main simplification of our model is the embodiment of the postsynaptic potential as a rectangular pulse rather than a continuous function of a similar shape. Rectangular pulses are widely used in silicon implementations of networks of spiking neurons such as described by Murray and Tarassenko [6].

A spiking neuron may be viewed as a digital or analog computational element, depending on the type of temporal coding that is used. In the following we restrict our analysis to the coding of binary values. For this *binary coding* we assume that input neuron a_i fires at time 0 if it encodes a 1, and that it does not fire at all if it encodes a 0. Correspondingly, we assume that v outputs a 1 if it fires as a result of this input from $a_1, \ldots, a_n$, and that v outputs a 0 if it does not fire.

We view the delays d_i as "programmable parameters" of the spiking neuron, in addition to the weights w_i of its synapses. Hence a learning algorithm may change the function computed by the spiking neuron not only by modifying its synaptic efficacies but also by tuning the transmission delays between neurons. For a discussion of the biological relevance of such mechanisms see Maass and Schmitt [7] and the references in there.

When considering the computation of Boolean functions, it is not hard to see (and was already stated in [2]) that the type of a spiking neuron studied here is at least as powerful as a threshold gate. The latter can be simulated by using the same weights and assigning equal values to all delays. Moreover, in [2] a concrete Boolean function has been exhibited that can be computed by a single spiking neuron, but requires at least $n/(2\log((n/2)+1))$ threshold gates when computed by a circuit. Further results in [7] indicate that the expressive power of a spiking neuron with n variable delays is much higher than that of a threshold gate: When the weights are fixed and the delays are programmable, the VC-dimension of a spiking neuron is $\Theta(n \log n)$, whereas for a threshold gate with variable weights it is $\Theta(n)$. (The VC-dimension can be considered a measure for the variety of a class of functions; see [7] for details.) The significant increase of the VC-dimension from a threshold gate to a spiking neuron demonstrates the considerably large impact of a delay as a programmable parameter on the computational power of a single-cell model.

In the following sections we provide further evidence for the large computational capabilities of this neuron model. First we establish bounds for its threshold number. Then we study its relation to sigma-pi units.

3 On the threshold number

The *threshold number* of a Boolean function f is the smallest number k such that f can be computed by a disjunction of k threshold gates. The threshold

number of a computational unit, such as a threshold gate or a spiking neuron, is defined as the maximum threshold number of any Boolean function that can be computed by the unit. Obviously, a threshold gate has threshold number 1. Calculating the threshold number of a computational unit is a natural way to measure how much more powerful the unit is than a threshold gate.

Investigations of the threshold number of Boolean functions can be traced back to Jeroslow [8]. He has shown that every Boolean function on n variables has threshold number at most 2^{n-1}, and further that for each k, where $1 \leq k \leq 2^{n-1}$, there is a Boolean function having threshold number k, with the parity function attaining the maximum value.

Maass and Schmitt [7] have shown that every Boolean function computable by a spiking neuron can be computed by a disjunction of at most $2n-1$ threshold gates. This implies a threshold number of at most $2n-1$ for the spiking neuron. The following result shows that even less than n gates are sufficient.

Theorem 1 *The threshold number of a spiking neuron with $n \geq 2$ inputs is at most $n-1$.*

By an explicit construction Maass [2] has shown that there is a Boolean function that can be computed by a spiking neuron, whereas any threshold circuit that computes this function has at least $n/(2\log(n/2+1))$ gates. This immediately implies that $n/(2\log(n/2+1))-1$ is a lower bound for the threshold number of a spiking neuron. We will now considerably improve this bound.

Theorem 2 *The threshold number of a spiking neuron with n inputs is at least $\lfloor n/2 \rfloor$.*

Theorems 1 and 2 together provide asymptotically tight bounds for the threshold number of a spiking neuron.

Corollary 3 *The threshold number of a spiking neuron with n inputs is $\Theta(n)$.*

The linear upper bound from Theorem 1 raises the question if having a small threshold number is sufficient for a Boolean function to be computable by a spiking neuron. This conjecture is refuted by the following result.

Proposition 4 *For each $n \geq 2$ there is a Boolean function on n variables that has threshold number 2 and cannot be computed by a spiking neuron.*

We remark that the threshold number 2 in Proposition 4 is minimal with this property since every Boolean function with threshold number 1 is a threshold function and hence computable by a spiking neuron.

4 On the threshold order

A threshold gate can be considered as a linear separator, that is, a hyperplane in the n-dimensional space which classifies the set of Boolean vectors depending

on which side they lie with respect to the hyperplane. A natural generalization of this notion is to allow for higher order terms in the weighted sum such that one arrives at polynomial surfaces as classifiers. Neuron models employing polynomials as activation functions are also known as sigma-pi units (see, e.g., Softky and Koch [4]). Such models are computationally more powerful than threshold gates even in the Boolean domain and even when the degree of the polynomial is bounded by some small constant as in the case of, for instance, quadratic surfaces.

In the following we compare the computational capabilities of the spiking neuron with those of polynomial neurons. We present a lower bound for the spiking neuron in terms of the so-called threshold order. The *threshold order* of a Boolean function f is the smallest number m such that f has a separator of degree m, but no separator of degree $m-1$. A *separator of degree* m for a Boolean function f on n variables is given by a weight vector $w_1, \ldots, w_d$ of dimension $d = \sum_{i=1}^{m} \binom{n}{i}$ and a threshold θ, such that

$$\sum_{S \subseteq \{1,\ldots,n\}, 0<|S| \leq m} w_S \prod_{i \in S} x_i \geq \theta \Longleftrightarrow f(x) = 1.$$

We define the threshold order of a computational unit as the largest threshold order of any Boolean function the unit can compute.

The notion of a threshold order has been introduced by Minsky and Papert [9]. (They simply called it order.) In particular they showed that each Boolean function on n variables has threshold order at most n with the parity function being of order exactly n. Obviously, a threshold gate has threshold order 1. We show that the threshold order of a spiking neuron is significantly larger.

Theorem 5 *The threshold order of a spiking neuron with n inputs is $\Omega(n^{1/3})$.*

This result shows that the threshold order of a spiking neuron grows arbitrarily large with the number of inputs. Thus, a polynomial threshold gate, or sigma-pi unit, with a degree that is bounded by a constant is computationally less powerful than a spiking neuron.

Corollary 6 *There is no sigma-pi unit with fixed degree that can simulate a spiking neuron.*

The currently best upper bound is established in the following result.

Proposition 7 *The threshold order of a spiking neuron with $n \geq 2$ inputs is at most $n-1$.*

Finally, by analogy with the proof of Proposition 4 we obtain the following statement as a counterpart to the result on the threshold number.

Corollary 8 *For each $n \geq 2$ there is a Boolean function on n variables that has threshold order 2 and cannot be computed by a spiking neuron.*

5 Conclusions

Time is undoubtedly an essential parameter in neural computation. We have analyzed a neuron model that uses the timing of single firing events to represent input and output values and has transmission delays as programmable parameters. We have shown that the computational capabilities of this model are significantly larger than those of a threshold gate or a sigma-pi unit with fixed degree. In particular, we have established a tight lower bound of $\Omega(n)$ for the threshold number of a spiking neuron, which is the number of gates required to simulate a spiking neuron by a disjunction of threshold gates. Moreover, we have shown that the threshold order, which is the degree of the polynomial used by a sigma-pi unit for simulating a spiking neuron, must grow as $\Omega(n^{1/3})$.

The lower bounds presented here bear far-reaching consequences for the following reason: It is not hard to see that these lower bounds remain valid for models of spiking neurons that employ more realistic functions as postsynaptic potentials. In fact, there is a very mild condition that such a model must satisfy in order for these bounds to hold: It is sufficient that the postsynaptic potential is non-zero for a definite period of time. If this requirement is met the ideas provided here show that the model becomes more powerful by using time as a computational resource.

References

[1] F. Rieke, D. Warland, W. Bialek and R. de Ruyter van Steveninck, *SPIKES: Exploring the Neural Code*, The MIT Press, Cambridge, Mass., 1996.

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

[3] M. Schmitt, On computing Boolean functions by a spiking neuron, TU Graz, 1998, available via http://www.lmi.ruhr-uni-bochum.de/mschmitt/.

[4] W. Softky and C. Koch, Single-cell models, in: *The Handbook of Brain Theory and Neural Networks*, ed. M. A. Arbib, The MIT Press, Cambridge, Mass., 1995, 879–884.

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

[6] A. Murray and L. Tarassenko, *Analogue Neural VLSI: A Pulse Stream Approach*, Chapman & Hall, London, 1994.

[7] W. Maass and M. Schmitt, On the complexity of learning for a spiking neuron, in: *Proceedings of the Tenth Annual Conference on Computational Learning Theory COLT'97*, ACM Press, New York, 1997, pp. 54–61.

[8] R. G. Jeroslow, On defining sets of vertices of the hypercube by linear inequalities, *Discrete Mathematics* **11** (1975) 119–124.

[9] M. L. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*, expanded edition, The MIT Press, Cambridge, Mass., 1988.