On Computing Boolean Functions by a Spiking Neuron*

Michael Schmitt

Institute for Theoretical Computer Science
Technische Universität Graz
Klosterwiesgasse 32/2
A-8010 Graz, Austria
mschmitt@igi.tu-graz.ac.at

Abstract

Computations by spiking neurons are performed using the timing of action potentials. We investigate the computational power of a simple model for such 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

Biological neurons communicate by sending spikes among themselves. A spike is a discrete event in continuous time. There is widespread agreement that the computations performed in biological neural networks are not based solely on the rates of these spikes. Neurophysiological experiments have rather lead to the conclusion that neural information processing also takes place in terms of temporal patterns of neural activity (see e.g. Rieke et al. [14]).

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

Models of single neurons like the threshold or the sigmoidal gate are theoretically well investigated and widely used in applications. However, these and related types of models do not capture the phenomenon of timing which is believed to be essential for computations in biological neural systems. 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 article a simple version of a spiking neuron, the so-called "spiking neuron of type A" according to the terminology of Maass [7]. 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 by a spiking neuron. 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 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. We study the degree that a sigma-pi unit must have to be able to simulate a spiking neuron. Results are presented in Section 4. Finally, Section 5 discusses the results and concludes with some remarks.

2 Basic concepts and related work

A *Boolean function* on n variables is a function $f : \{0,1\}^n \rightarrow \{0,1\}$. 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 article we focus on a simple model for a spiking neuron. This particular variant, the so-called "spiking neuron of type A", has been introduced by Maass [7]. 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

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

(see Fig. 1). 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. In a biological context the functions h_i are called *postsynaptic potentials.* They 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.

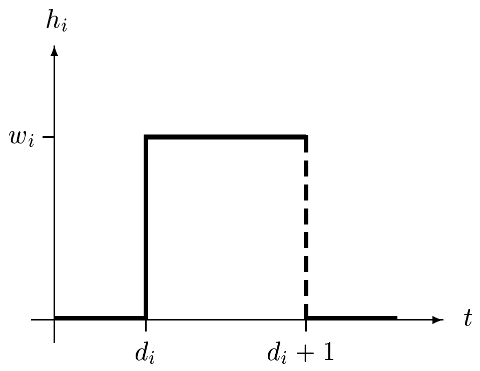

Figure 1: Time course of the postsynaptic potential h_i with delay d_i and weight w_i as evoked by a spike from input neuron a_i. The values at the "jumps" are defined as $h_i(d_i) = w_i$ and $h_i(d_i + 1) = 0$.

The firing threshold θ of a biological neuron is a function depending on the time which has passed since its last firing. Since we are concerned with computations on binary values that are encoded in terms of single firing events, it is convenient to 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 article.

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 of Softky and Koch [15] or in the surveys by Gerstner [2] and Maass [7, 8]. 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 [13]. Theoretical investigations concerning the computational power of networks of a neuron type that employs more complex functions as postsynaptic potentials can be found in Maass [5, 6, 7].

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 [9, 10] and the references in there.

When considering the computation of Boolean functions, it is not hard to see (and was already stated in [7]) 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 [7] 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 [9] 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 [9] for details.) The significant increase of the VC-dimension from a threshold gate to a spiking neuron is somewhat remarkable since from a cardinality point of view, a spiking neuron seems only slightly more powerful than a threshold gate: It has been shown that a spiking neuron can compute at most $2^{n^2+O(n\log n)}$ Boolean functions [9]. On the other hand it is known that a threshold gate can compute at least $2^{n^2(1-10/\ln n)}$ Boolean functions [17]. (This bound almost matches the upper bound 2^{n^2} [12].) It is therefore only the term $O(n \log n)$ in the upper bound of the cardinality that accounts for the higher VC-dimension of the spiking neuron. This demonstrates that a delay as a programmable parameter has a considerably larger impact on the computational power than a weight.

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. We define the threshold number of a set of Boolean functions as the maximum of the threshold numbers of its members. The threshold number of a computational unit, such as a threshold gate or a spiking neuron, is defined as the threshold number of the set of Boolean functions computed by the unit choosing all possible settings of its parameters. Since threshold functions can be characterized as those Boolean functions having threshold number 1, 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 [4] and Chvátal and Hammer [1]. Jeroslow [4] 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. Chvátal and Hammer [1] proved that computing the threshold number is NP-hard. The first explicit definition of the threshold number appears in later work by Hammer et al. [3]. They established bounds for the threshold number of positive Boolean functions. A Boolean function is *positive* (sometimes called monotone) if it satisfies $f(x) \leq f(y)$ for any pair of binary vectors x, y with $x \leq y$. Here $x \leq y$ means that $x_i \leq y_i$ for all $i = 1, \ldots, n$. A binary vector p is said to be a *prime implicant* of a Boolean function f if $f(p) = 1$ holds and p is minimal with respect to $\leq$ having this property. Every positive Boolean function is known to have a unique collection of prime implicants. Hammer et al. [3] show in particular that the threshold number of such a function is less than $\binom{n}{\lfloor n/2 \rfloor}$. Zuev and Lipkin [18] proved that almost all Boolean functions have a threshold number that satisfies the bounds $\Omega(2^n/n)$ and $O((\ln n) \cdot 2^n/n)$.

In [9] (see also [10]) Maass and Schmitt 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. We now prove the following combinatorial result which will be used to establish an improved estimate for the upper bound.

Lemma 1 *Let $S_1, \ldots, S_k$ be a sequence of non-empty subsets of a set $S = \{s_1, \ldots, s_n\}$ such that the following conditions hold:*

(1) $|S_j \cup S_{j+1}| \geq |S_j \cap S_{j+1}| + 2$ *for* $j = 1, \ldots, k-1$.

(2) For each $s_i \in S$ there are unique numbers $l_i, r_i \in \{1, \ldots, k\}$ such that all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, k\}$ satisfy the following:

$$s_i \in S_j \Longleftrightarrow l_i \leq j \leq r_i.$$

Then $k \leq n$, where equality holds if and only if each S_j is a singleton.

Proof. Assume that $l_1, \ldots, l_n, r_1, \ldots, r_n$ are given according to condition (2) and consider the transitions $(j, j+1)$ from S_j to S_{j+1} where $j \in \{1, \ldots, k-1\}$. We assign elements from the sequence $l_1, \ldots, l_n, r_1, \ldots, r_n$ to these transitions as follows:

$$\begin{array}{llll} l_i & \text{is assigned to} & (l_i - 1, l_i) & \text{if } l_i > 1, \\ r_i & \text{is assigned to} & (r_i, r_i + 1) & \text{if } r_i < k. \end{array}$$

Then each element is assigned to at most one transition, but there are also elements that have not been assigned. In particular, since S_1 and S_k are non-empty there is at least one l_i and at least one r_i not being assigned to any transition. Hence at most $2n - 2$ elements remain for assignments. Due to condition (1) there are at least two elements for each transition. Thus there can be at most $n - 1$ transitions, from which $k \leq n$ follows.

Assume now that $k = n$. Then exactly two elements must be assigned to each transition, and there are exactly two elements not being assigned to any transition. This implies that each S_j is a singleton, and thus the "only if" part of the last statementof the lemma is shown. The "if" part is obvious. □

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

Proof. For a spiking neuron v with n inputs there are finitely many points in time where the membrane potential P_v can possibly change. These points are determined by the starting or ending of a postsynaptic pulse h_i which in its turn depends on the value of the corresponding delay d_i. The points give rise to disjoint intervals $I_1, \ldots, I_q$ (ordered according to time) during which P_v is constant. Within each of these intervals I_j the potential P_v is governed by a subset S_j of the set of weights $S = \{w_1, \ldots, w_n\}$. Thus for a given input the neuron fires within interval I_j if and only if the sum of those weights $w_i \in S_j$ that received an input spike is greater or equal to θ_v.

Consider an arbitrary pair of contiguous intervals I_j, I_{j+1}. We show that one of them is not relevant for the firing of v if the corresponding weight sets S_j, S_{j+1} satisfy $|S_j \cup S_{j+1}| = |S_j \cap S_{j+1}| + 1$. Assume therefore without loss of generality that $S_j \cup \{u\} = S_{j+1}$ for some $u \in S$. If $u \geq 0$ then v fires within I_{j+1} whenever it fires within I_j. On the other hand, if $u < 0$ then v fires within I_j whenever it fires within I_{j+1}. Thus for each pair I_j, I_{j+1} with $|S_j \cup S_{j+1}| = |S_j \cap S_{j+1}| + 1$, at least one interval is not relevant for the firing of v. Therefore, the relevant intervals satisfy the conditions of Lemma 1 implying that there are at most n such intervals. Further, if there are exactly n such intervals the corresponding S_i are singletons. The function computed by this spiking neuron is a simple disjunction and can be computed by a single threshold gate. If there are less than n relevant intervals we define for each of them a threshold gate with weights and inputs determined by the corresponding S_j. The disjunction of these at most $n-1$ gates is then equivalent to the spiking neuron. □

We show now that the upper bound is almost optimal up to a factor of 2. For the proof of this statement we use the following result by Hammer et al. [3].

Proposition 3 (Hammer et al. [3]) *Let f be a positive Boolean function and P its collection of prime implicants. If for any $p, q \in P$ $(p \neq q)$ there are four distinct indices i, j, k, l such that $p_i = p_j = q_k = q_l = 1$ and $p_k = p_l = q_i = q_j = 0$, then f has threshold number $|P|$.*

By an explicit construction Maass [7] 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 use now basically the same function to establish a better lower bound that is even linear in n. (Note, however, that this result has no obvious implications for the lower bound of the size of a threshold circuit computing this function.)

Theorem 4 *The threshold number of a spiking neuron with n inputs is at least $\lfloor n/2 \rfloor$.*

Proof. Assume first that n is even. The statement is proven if we show that there is a function that has threshold number $n/2$ and can be computed by a spiking neuron. Consider the function

$$x_1x_2 \vee x_3x_4 \vee \cdots \vee x_{n-1}x_n$$

which can be computed by a spiking neuron as follows [7]: Choose weights and threshold equal to 1 and adjust the delays such that $d_{2i-1} = d_{2i}$ and $d_{2i}+1 < d_{2i+1}$. The function is positive and has the $n/2$ prime implicants $(1100\ldots00)$, $(0011\ldots00)$, $\ldots$, $(0000\ldots11)$. They all satisfy the condition of Proposition 3. Hence, the function has threshold number $n/2$. The statement now follows immediately also for n odd. □

We summarize the results of Theorems 2 and 4 providing asymptotically tight bounds for the threshold number of a spiking neuron.

Corollary 5 *The threshold number of a spiking neuron with n inputs is $\Theta(n)$.*

The linear upper bound from Theorem 2 raises the question if having a small threshold number is sufficient for a Boolean function to be computable by a spiking neuron. The following result blatantly demonstrates that this is not the case.

Proposition 6 *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.*

Proof. Consider the function

$$x_1\overline{x}_2 \vee \overline{x}_1 x_2 \vee x_3 \vee \cdots \vee x_n.$$

It can be computed by a disjunction of two threshold gates: For the first gate choose threshold $\theta^{(1)} = 1$ and weights $w_1^{(1)} = 1, w_2^{(1)} = -1$ and $w_i^{(1)} = 0$ for $i \geq 3$. For the second gate choose threshold $\theta^{(2)} = 1$ and weights $w_1^{(2)} = -1$ and $w_i^{(2)} = 1$ for $i \geq 2$. Hence the function has threshold number 2. On the other hand, restricting the function to inputs x_1, x_2 by fixing 0 to the remaining inputs we obtain the parity function. It is not hard to see (and was already stated in [9]) that the parity of two inputs cannot be computed by a spiking neuron. □

We remark that the threshold number 2 in Proposition 6 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 [15]). 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 [11]. (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. A proof that parity is the only function of threshold order n is due to Wang and Williams [16]. Obviously, a threshold gate has threshold order 1. We show that the threshold order of a spiking neuron is significantly larger.

Theorem 7 *The threshold order of a spiking neuron with n inputs is $\Omega(n^{1/3})$.*

Proof. Assume that $n = 4k^3$ for some $k \geq 1$. Consider the function f defined by the formula

$$x_1 \cdots x_{4k^2} \vee \cdots \vee x_{4k^2(k-1)+1} \cdots x_{4k^3}$$

which is a disjunction of k terms each of them being a monomial in $4k^2$ variables. Each variable occurs exactly once. Obviously, the function can be computed by a spiking neuron: Assign equal values to delays corresponding to inputs in the same monomial such that postsynaptic pulses of inputs from different monomials do not overlap. The weights can then be chosen as 1 and the threshold as $4k^2$.

We show that the function has threshold order at least k using a result by Minsky and Papert [11] on the dual of f. (The dual f^d of a Boolean function f is defined as $f^d(x_1, \ldots, x_n) = \overline{f}(\overline{x}_1, \ldots, \overline{x}_n)$.) Due to Theorem 3.2 of Minsky and Papert [11], f^d has threshold order at least k. By Corollary 2.3 of Wang and Williams [16], the threshold order of a function is equal to the threshold order of its dual. Since $k = (n/4)^{1/3}$, the statement of the theorem follows. □

We have shown 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 8 *There is no sigma-pi unit with fixed degree that can simulate a spiking neuron.*

Currently there is no better upper bound than the trivial bound $n-1$ which is obtained as follows: If the spiking neuron does not compute a threshold function then this function can be computed by a disjunction of threshold gates. As was shown in the proof of Theorem 2 the weight sets of these gates satisfy the conditions of Lemma 1. In particular, each gate has at most $n-1$ inputs. Hence, the function of each gate can be written in disjunctive normal form with monomials of at most $n-1$ literals. This normal form can then easily be turned into a polynomial threshold gate with degree at most $n-1$.

Proposition 9 *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 6 we obtain the following statement as a counterpart to the result on the threshold number.

Corollary 10 *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. Furthermore, the postsynaptic change of the neural membrane potential responding to a presynaptic spike is a time dependent function modelled by a rectangular shaped pulse. In addition to the parameters that characterize traditional neuron models, the spiking neuron has transmission delays as programmable parameters.

We have shown that this model of a spiking neuron has computational capabilities in the Boolean domain that are significantly larger than that of a threshold gate or a sigma-pi unit with fixed degree. In particular, we have established a 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 can be considered as the degree of the polynomial used by a sigma-pi unit for simulating a spiking neuron, must grow as $\Omega(n^{1/3})$.

We have also shown that the lower bound for the threshold number is tight whereas such a result for the threshold order remains an interesting open problem. Upper bounds are important for exploring the limitations of a model. However, the lower bounds presented in this article bear even 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 can be used in the same way to show that the model becomes more powerful due to the use of time as a computational resource.

References

[1] V. Chvátal and P. L. Hammer, Aggregation of inequalities in integer programming, in: *Studies in Integer Programming*, eds. P. L. Hammer, E. L. Johnson, B. H. Korte and G. L. Nemhauser, Annals of Discrete Mathematics 1, North-Holland, Amsterdam, 1977, pp. 145–162.

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

[3] P. L. Hammer, T. Ibaraki and U. N. Peled, Threshold numbers and threshold completions, in: *Studies on Graphs and Discrete Programming*, ed. P. Hansen, Annals of Discrete Mathematics 11, (Mathematics Studies 59), North-Holland, Amsterdam, 1981, pp. 125–145.

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

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

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

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

[8] W. Maass, On the relevance of time in neural computation and learning, in: *Proceedings of the 8th International Workshop on Algorithmic Learning Theory ALT'97*, eds. M. Li and A. Maruoka, Lecture Notes in Artificial Intelligence, vol. 1316, Springer, Berlin, 1997, pp. 364–384.

[9] 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.

[10] W. Maass and M. Schmitt, On the complexity of learning for spiking neurons with temporal coding, Electronic Colloquium on Computational Compexity, ECCC TR97-049, 1997, available via WWW at URL http://www.eccc.uni-trier.de/eccc/. Also available at http://www.cis.tu-graz.ac.at/igi/mschmitt/.

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

[12] S. Muroga, *Threshold Logic and Its Applications*, John Wiley & Sons, New York, 1971.

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

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

[15] 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.

[16] C. Wang and A. C. Williams, The threshold order of a Boolean function, Discrete Applied Mathematics 31 (1991) 51–69.

[17] Yu. A. Zuev, Asymptotics of the logarithm of the number of threshold functions of the algebra of logic, Soviet Mathematics Doklady 39 (1989) 512–513.

[18] Yu. A. Zuev and L. I. Lipkin, Estimating the efficiency of threshold representations of Boolean functions, Kibernetika (Kiev) no. 6 (1988) 29–37. English translation in: Cybernetics 24 (1988) 713–723.