On the Complexity of Learning for Spiking Neurons with Temporal Coding*

Wolfgang Maass   Michael Schmitt

Institute for Theoretical Computer Science
Technische Universität Graz
Klosterwiesgasse 32/2
A–8010 Graz
Austria
Fax: ++43 316 873 5805
Phone: ++43 316 873 5811
E-mail: {maass, mschmitt}@igi.tu-graz.ac.at

Abstract

Spiking neurons are models for the computational units in biological neural systems where information is considered to be encoded mainly in the temporal pattern of their activity. In a network of spiking neurons a new set of parameters becomes relevant which has no counterpart in traditional neural network models: the time that a pulse needs to travel through a connection between two neurons (also known as delay of a connection). It is known that these delays are tuned in biological neural systems through a variety of mechanisms. We investigate the VC-dimension of networks of spiking neurons where the delays are viewed as programmable parameters and we prove tight bounds for this VC-dimension. Thus we get quantitative estimates for the diversity of functions that a network with fixed architecture can compute with different settings of its delays. In particular, it turns out that a network of spiking neurons with k adjustable delays is able to compute a much richer class of functions than a threshold circuit with k adjustable weights. The results also yield bounds for the number of training examples that an algorithm needs for tuning the delays of a network of spiking neurons. Results about the computational complexity of such algorithms are also given.

1 Introduction and Definitions

During the last few years the paradigms for computation in biological neural systems have undergone drastic changes. With the help of refined experimental techniques

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

one has learnt that information is not only encoded in the *firing rates* of biological neurons, but often also in the *temporal pattern* of their firing. Whereas threshold circuits and sigmoidal neural nets provide a suitable model for neural computation based on *rate coding*, i.e. in terms of firing rates, they cannot be used for modelling neural computation based on *temporal coding*, i.e. in terms of temporal patterns of neuronal activity. In order to model temporal patterns of activity, one has to consider networks consisting of a different type of computational unit: spiking neurons, or leaky integrate-and-fire neurons, as they are commonly called in biophysics and theoretical neurobiology.

We will focus in this article on a simple version of the spiking neuron model ("spiking neurons of type A" in the terminology of (Maass, 1997b)). This model allows us to study some fundamental new learning problems that arise in the context of computation with temporal coding. Since this model is sufficiently simple, the basic aspects of this new mode of computation are not obscured by the myriad of additional subtleties and complications that occur in a more detailed neuron model. In addition, this simple model for a spiking neuron has the advantage that it provides a link to silicon implementations of spiking neurons in analog VLSI.

1.1 The Model for a Spiking Neuron

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

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

We assume that the neuron v fires 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 these pulses are called *postsynaptic potentials*. The function h_i models the effect of a firing of neuron a_i on the *membrane potential* $P_v(t)$ at the trigger zone of v. The firing threshold θ of a biological neuron depends on the time which has passed since its last firing. For simplicity we assume here that the neuron has not fired for a while (say at least 20 ms), so that its firing threshold has returned to its "resting value" θ_v. There is some disagreement among neurobiologists whether the sign of a synaptic efficacy w_i can change in the course of a learning process. This issue will not be relevant for the results of this article.

The model is a simple version of a leaky integrate-and-fire neuron. In contrast to more complex models (see e.g., (Tuckwell, 1988; Gerstner, 1995; Maass, 1997a)) it models a pulse as a step function, rather than a continuous function of a similar shape. Pulses of this shape are actually very common in silicon implementations of networks of spiking neurons (Murray and Tarassenko, 1994).

A spiking neuron of this type was called a "spiking neuron of type A" in (Maass, 1997b). In this article we will refer to it simply as a *spiking neuron.*

1.2 Temporal Coding

A spiking neuron may be viewed as a digital or analog computational element, depending on the type of temporal coding that is used. For *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.

For *analog coding* we assume that a_i encodes a real number $t_i \in [0,1]$ by firing at time t_i. The output value of v is the time t_v when it fires (or $t_v - T$ for a suitable constant T if one wants to scale the real-valued output of v into a specific range such as $[0,1]$). In case that v does not fire, we assume that this encodes some fixed analog output t_0 (e.g. $t_0 = 0$).

We will consider both types of coding in this article. Moreover, the type of coding for the inputs may differ from that for the output, e.g., analog coding for the inputs and binary coding for the output may occur. We prove each result for that type of coding for which it is more difficult. *Lower* bounds for sample or learning complexity tend to be more difficult for binary coding, *upper* bounds tend to be more difficult for analog coding.

We view in the following the delays d_i as "programmable parameters" of a neuron, in addition to the weights w_i of its synapses. This is reasonable since in biology many mechanisms are known that can change the effective delay between two neurons. One well-known mechanism is the selection of a few synapses out of an initially very large set of synapses between two neurons. Some other biological mechanisms for changing the effective delay between two neurons are discussed in (Agmon-Snir and Segev, 1993; Gerstner *et al.*, 1996).

Our results about the VC-dimension of a spiking neuron are complementary to those achieved in (Zador and Pearlmutter, 1996). In that article the integration time constant and the threshold were viewed as the only variable parameters of a spiking neuron, whereas the effect of variable delays has not been addressed.

1.3 Complexity of Learning

In this article we investigate the complexity of learning for a spiking neuron within the framework of *probably approximately correct learning*, or *PAC-learning* for short. For detailed definitions of this paradigm we refer the reader to (Anthony and Biggs, 1992; Blumer *et al.*, 1989; Valiant, 1984). In Section 2 we estimate the computational power and the sample complexity of a single spiking neuron. We give upper and lower bounds for its computational power when using binary coding in terms of several classes of Boolean functions. The sample complexity is analyzed in terms of the *Vapnik-Chervonenkis dimension*, or *VC-dimension* for short. As the main result of this section we show that for binary and analog coding the VC-dimension of the corresponding function class is $\Theta(n \log n)$. It is well known that the VC-dimension of a function class gives fairly tight bounds on the sample complexity, i.e. the number of training examples needed, for PAC-learning this class. According to (Haussler, 1992), these estimates of the sample complexity in terms of the VC-dimension hold even for agnostic PAC-learning, i.e. in the case when the training examples are generated by some arbitrary probability distribution. In particular, these bounds remain valid when the training examples are not generated by a spiking neuron.

In Section 3 we consider feedforward networks of spiking neurons. We show that such networks can have a VC-dimension that is quadratic in the number of delays that are programmable. Interestingly, this bound matches the quadratic lower bound for sigmoidal nets in terms of the number of weights due to (Koiran and Sontag, 1997). We further show that this bound is asymptotically tight by proving that any feedforward network of spiking neurons has a VC-dimension that is at most quadratic in the number of its edges. Moreover, this upper bound holds even if all delays, weights, and thresholds are programmable and even for analog coding of the inputs. The proof of this bound relies on a well-known and far-reaching result by (Goldberg and Jerrum, 1995).

In Section 4 we investigate the computational complexity of PAC-learning using a particular spiking neuron as hypothesis class. We show that for a bounded set of at least two delay values the consistency problem for the corresponding hypothesis class is NP-complete. This implies that there are no efficient PAC-learning algorithms for these hypothesis classes unless the complexity classes RP and NP are equal. The intractability results presented in this section have also consequences for the case of agnostic PAC-learning. According to known results (Kearns *et al.*, 1992; Höffgen *et al.*, 1995), polynomial-time agnostic PAC-learning with some hypothesis class $\mathcal{H}$ is possible only if the minimizing disagreement problem for $\mathcal{H}$ is in RP. Now, for each hypothesis class $\mathcal{H}$ the consistency problem for $\mathcal{H}$ can be solved in polynomial time if the minimizing disagreement problem for $\mathcal{H}$ can be solved in polynomial time. (More precisely, there is an easily definable polynomial-time reduction from the consistency problem to the minimizing disagreement problem.) Therefore, polynomial-time agnostic PAC-learning is not possible for the hypothesis classes considered in this section, provided that RP $\neq$ NP.

The final Section 5 contains some concluding remarks and discussion.

2 Computational Power and VC-Dimension of a Single Spiking Neuron

We first introduce some notation. The class of Boolean functions that can be computed by a spiking neuron with n binary coded inputs and a binary coded output is denoted by $\mathcal{S}_n^{\mathrm{bb}}$ (where "bb" stands for "binary input and binary output"). Correspondingly, $\mathcal{S}_n^{\mathrm{aa}}$ is the class of functions from $\mathbb{R}^n$ to $\mathbb{R}$ that can be computed by a spiking neuron with analog coding of the inputs and the output. The subclass of $\mathcal{S}_n^{\mathrm{aa}}$ restricted to Boolean output values encoded in binary is denoted by $\mathcal{S}_n^{\mathrm{ab}}$.

We use a similar notation for the threshold gate and for the sigmoidal gate: A threshold gate, also known as Perceptron or McCulloch-Pitts neuron, with inputs $x_1, \ldots, x_n$ has weights $w_1, \ldots, w_n$, where w_i is associated with x_i for $i = 1, \ldots, n$, and a threshold θ. It outputs 1 if $w_1 x_1 + \cdots + w_n x_n \geq \theta$, otherwise 0. By $\mathcal{T}_n^{\mathrm{bb}}$ we denote the class of Boolean functions that can be computed by a threshold gate. A threshold gate with real-valued inputs and binary-valued output corresponds to a half-space over $\mathbb{R}^n$. We denote the corresponding function class by $\mathcal{T}_n^{\mathrm{ab}}$. The sigmoidal gate is a neuron model that computes functions from $\mathbb{R}^n$ to $\mathbb{R}$. We assume that it calculates its output value by applying the standard sigmoidal function $1/(1+e^{-y})$ to the sum $w_1 x_1 + \cdots + w_n x_n - \theta$. We denote the corresponding function class by $\mathcal{T}_n^{\mathrm{aa}}$.

For assessing the computational power of a spiking neuron in the Boolean domain it turns out that it is useful to consider two further classes of Boolean functions: the first is a special type of disjunctive normal form (DNF), the second is a disjunction of linearly many threshold gates. The class $\mu{-}\mathrm{DNF}_n$ is the class of Boolean functions each of which can be written as a DNF formula over n variables where each variable occurs at most once. By $\mathrm{OR}{-}\mathrm{of}{-}O(n){-}\mathcal{T}_n^{\mathrm{bb}}$ we denote the class of Boolean functions that can be computed by a disjunction of $O(n)$ threshold gates.

2.1 Computational Power

It is obvious that in the case of binary coding a spiking neuron has at least the computational power of a threshold gate: just assume that all delays d_i are equal. However, it is easy to see that its computational power is strictly larger. In order to characterize its power more precisely we compare it with the Boolean function classes defined above. The following theorem clarifies the relationships among these classes. It shows that a spiking neuron with binary coding can also compute any function in $\mu{-}\mathrm{DNF}$. On the other hand, it can be simulated by a disjunction of linearly many threshold gates. The results are graphically depicted in Fig. 1.

Theorem 2.1

a) $\mathcal{T}_n^{\mathrm{bb}} \not\subseteq \mu{-}\mathrm{DNF}_n$ *for all* $n \geq 3$.

b) $\mu{-}\mathrm{DNF}_n \not\subseteq \mathcal{T}_n^{\mathrm{bb}}$ *for all* $n \geq 4$.

c) $\mathcal{T}_n^{\mathrm{bb}} \subseteq \mathcal{S}_n^{\mathrm{bb}}$ *for all* n, *and* $\mathcal{T}_n^{\mathrm{bb}} \neq \mathcal{S}_n^{\mathrm{bb}}$ *for all* $n \geq 4$.

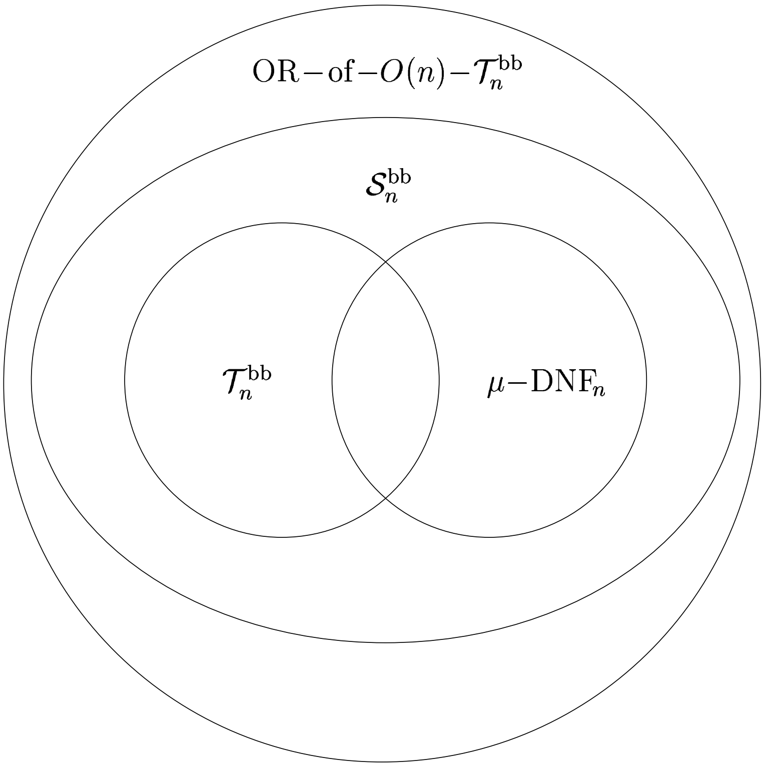
$\text{OR}-\text{of}-O(n)-\mathcal{T}_n^{\text{bb}}$
$\mathcal{S}_n^{\text{bb}}$
$\mathcal{T}_n^{\text{bb}}$
$\mu-\text{DNF}_n$

Figure 1: Upper and lower bounds for the computational power of a spiking neuron in the Boolean domain as established in Theorem 2.1.

d) $\mu-\text{DNF}_n \subseteq \mathcal{S}_n^{\text{bb}}$ *for all* n*, and* $\mu-\text{DNF}_n \neq \mathcal{S}_n^{\text{bb}}$ *for all* $n \geq 3$.

e) $\mathcal{S}_n^{\text{bb}} \subseteq \text{OR}-\text{of}-O(n)-\mathcal{T}_n^{\text{bb}}$*. More precisely, each function in* $\mathcal{S}_n^{\text{bb}}$ *can be computed by an OR of* $2n-1$ *threshold gates. For* $n \geq 2$*, however, there exist functions computable by an OR of two threshold gates that are not in* $\mathcal{S}_n^{\text{bb}}$*.*

Proof. All proofs are straightforward. We give evidence of the inequality claims by presenting for each of them a function that separates the two classes involved for the smallest n. It is then easy to obtain a separating function also for higher values of n.

a) The function $(x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$, which is 1 if and only if the input vector contains at least two 1's, can obviously be computed by a threshold gate. Assume that it can be written as a $\mu-\text{DNF}_3$ formula. Then this formula either contains an AND with only one variable or it consists of at most one AND, both of which contradicts the definition of the function.

b) Consider the $\mu-\text{DNF}_4$ formula $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$. Assuming that it can be computed by a threshold gate implies that $w_1 + w_2 \geq \theta$ and $w_3 + w_4 \geq \theta$, but also that $w_1 + w_3 < \theta$ and $w_2 + w_4 < \theta$. All four inequalities together form a contradiction.

c) The inclusion is obvious. For the inequality consider the function in b) which can be computed by a spiking neuron as follows: Choose equal values for delays which belong to the same AND and take care that pulses from different ANDs do not overlap. This is also the general way of computing a $\mu-$DNF formula by a spiking neuron, which is the first claim of d).

d) For the inclusion see c). For the inequality consider the function $(x_1 \wedge x_2) \vee (x_2 \wedge x_3)$ which is in $\mathcal{S}_3^{\mathrm{bb}}$: Choose delays such that the pulses for x_1 and x_3 do not overlap but that those for each pair x_1, x_2 and x_2, x_3 do overlap. It is easy to see that this function cannot be written as a $\mu-\mathrm{DNF}_3$ formula.

e) A spiking neuron v with binary coding has at most $2n$ points in time at which the potential P_v changes. Hence, there are at most $2n-1$ intervals during which P_v can be non-zero. Define for each of them a threshold gate that has as inputs those x_i having pulses intersecting this interval. For the inequality, it is easy to see that the Exclusive-OR of two bits, which outputs 1 if and only if $x_1 + x_2 = 1$, can be computed by an OR of two threshold gates and is not in $\mathcal{S}_2^{\mathrm{bb}}$. □

2.2 Lower Bound for the VC-Dimension

A *dichotomy* of a set $S \subseteq \mathbb{R}^n$ is a partition of S into two disjoint subsets S_0, S_1 such that $S_0 \cup S_1 = S$. Given a set $\mathcal{F}$ of functions from $\mathbb{R}^n$ to $\{0,1\}$ and a dichotomy S_0, S_1 of S, we say that $\mathcal{F}$ *induces* S_0, S_1 *on* S if there is a function $f \in \mathcal{F}$ such that $f(S_0) \subseteq \{0\}$ and $f(S_1) \subseteq \{1\}$. We say further that $\mathcal{F}$ *shatters* S if $\mathcal{F}$ induces all dichotomies on S. The *VC-dimension* of $\mathcal{F}$ is defined as the largest number m such that there is a set of m elements that is shattered by $\mathcal{F}$. (If arbitrarily large finite sets are shattered by $\mathcal{F}$ then its VC-dimension is defined to be infinite.)

It is well-known that the VC-dimension of the classes $\mathcal{T}_n^{\mathrm{bb}}$ and $\mathcal{T}_n^{\mathrm{ab}}$ is $n+1$ and hence equal to the number of programmable parameters of these neuron models. The following result shows that the VC-dimension of the classes $\mathcal{S}_n^{\mathrm{bb}}$ and $\mathcal{S}_n^{\mathrm{ab}}$ is significantly larger even if only the delays are adjustable and the weights and threshold remain fixed.

Theorem 2.2 *The VC-dimension of a spiking neuron with n variable delays as programmable parameters is $\Omega(n \log n)$. This holds even if the inputs are restricted to binary values and all weights are kept fixed.*

The statement follows from the following more general result choosing $k = (\log(n/2))/2$ and $m = n/2$, and observing that $k \cdot 2^k + m \leq n$ and $k \cdot m = \Omega(n \log n)$. We give a proof for binary coding of the inputs and indicate afterwards how to derive the result for the case of analog coding of the inputs, i.e. the class $\mathcal{S}_n^{\mathrm{ab}}$.

Theorem 2.3 *For each $m, k \geq 1$ there exists a set $S \subseteq \{0,1\}^{m+k\cdot 2^k}$ of cardinality $|S| = m \cdot k$ that can be shattered by a spiking neuron with fixed weights.*

Proof. We first describe the construction of S, then we fix the weights and a part of the delays, and finally we show that for each subset $S' \subseteq S$ there exists a delay vector such that the neuron fires on elements of S' but does not fire on elements of $S \backslash S'$.

The set S consists of $m \cdot k$ elements $\underline{s}^{i,j}$ for $1 \leq i \leq m$, $1 \leq j \leq k$ where the first m bits of $\underline{s}^{i,j}$ are formed by the unit vector $\underline{e}_i \in \{0,1\}^m$. The remaining $k \cdot 2^k$ bits of the elements are defined as follows: Assume a fixed enumeration of all 2^k subsets of the set $\{1,\ldots,k\}$. Reserve for each $A \subseteq \{1,\ldots,k\}$ a block b_A of k bits. The block b_A of element $\underline{s}^{i,j}$ is then defined as

$$\begin{array}{ll} \text{the unit vector } \underline{e}_j \in \{0,1\}^k, & \text{if } j \in A, \\ \text{the zero vector } \underline{0} \in \{0,1\}^k, & \text{otherwise.} \end{array}$$

The weights are defined as $w_i = 1$ for $1 \leq i \leq n$, and the threshold is $3/2$. The delays for the last $k \cdot 2^k$ inputs are fixed in such a way that inputs from the same block b_A have identical delays, but the pulses for inputs from different blocks $b_A, b_{A'}, A \neq A'$ do not overlap. (For instance, integer values $\{0,\ldots,2^k-1\}$ would do this.)

It remains to show that S can be shattered. Let $S' \subseteq S$. The delays for the first m inputs are specified as follows: For each $i \in \{1,\ldots,m\}$ define the set

$$A'_i = \{j \in \{1,\ldots,k\} : \underline{s}^{i,j} \in S'\}$$

and choose the delay for the i-th input equal to the delay of the inputs of block $b_{A'_i}$. Obviously then the neuron fires only for elements of S'. □

Theorem 2.3 can be shown to hold also for analog coding of the input values at the cost of adding an extra input with value 0. Its weight is chosen such that all pulses from inputs that encode 0 are cancelled. This weight can also be kept fixed because all elements of S constructed in the proof have the same number of 0's.

The proof of Theorem 2.3 gives also rise to a lower bound when the number of different values for the delays is bounded. One obtains the bound $\Omega(n \log l)$ where l is the number of different delay values that are allowed.

2.3 Upper Bound for the VC-Dimension

The lower bound of Theorem 2.2 holds for a very restricted spiking neuron with fixed weights and integer delays. The following suprising result shows that this bound is asymptotically tight even if the delays and weights range over arbitrary real numbers.

Theorem 2.4 *The VC-dimension of a spiking neuron with n analog coded inputs and binary coded output is $O(n \log n)$.*

The following statement is an immediate consequence of Theorems 2.2 and 2.4. It summarizes the results of this section in terms of the function classes computed by a spiking neuron.

Corollary 2.5 *The classes* $\mathcal{S}_n^{\mathrm{bb}}$ *and* $\mathcal{S}_n^{\mathrm{ab}}$ *have VC-dimension* $\Theta(n \log n)$.

In the proof of Theorem 2.4 we will use the following result which is a consequence of Theorem 2 in (Cover, 1965)[1] and Proposition A2.1 of (Blumer *et al.*, 1989).

Lemma 2.6 *Let* m *hyperplanes in* $\mathbb{R}^n$ *passing through the origin be given, where* $m \geq n$. *They partition* $\mathbb{R}^n$ *into at most* $2(em/(n-1))^{(n-1)}$ *different regions.*

Proof. By Theorem 2 of (Cover, 1965), m hyperplanes through the origin partition $\mathbb{R}^n$ into at most $2\sum_{k=0}^{n-1}\binom{m-1}{k}$ different regions. By Proposition A2.1(iii) of (Blumer *et al.*, 1989), $2\sum_{k=0}^{n-1}\binom{m-1}{k} \leq 2(e(m-1)/(n-1))^{(n-1)}$ for $m \geq n$. □

Proof of Theorem 2.4. The proof is structured as follows: We first estimate the number of dichotomies induced by a spiking neuron on an arbitrary finite set $S \subseteq \mathbb{R}^n$ of cardinality m. This will result in the upper bound

$$2(4emn)^n \cdot 2(2em)^n. \tag{1}$$

Then the assumption that S is shattered by a spiking neuron, i.e. that all 2^m dichotomies can be computed, will lead to the bound $m = O(n \log n)$ and hence to the claimed result.

The computation of a spiking neuron can be considered in the following way: Given an input vector and a delay vector, the time that is relevant to determine if the neuron fires is divided into at most $2n-1$ intervals which are specified by the starting and ending points of the n pulses. With each interval there is associated a subset of the weights corresponding to the set of pulses that are active during this interval. The neuron fires if within some interval the sum of the weights in the associated subset reaches the threshold.

In order to prove (1), we first estimate the number of different delay vectors that are relevant. For each fixed $\underline{s} \in S$, the space $\mathbb{R}^n$ of delay vectors $\underline{d}$ is partitioned into regions by hyperplanes of the form

$$s_i + d_i + y = s_j + d_j + z,$$

where $y, z \in \{0,1\}$ depending on whether the term corresponds to a starting or ending point of a pulse. There are $(2n)^2$ such hyperplanes for each fixed $\underline{s}$. They partition $\mathbb{R}^n$ into regions of delay vectors that are equivalent with regard to the computation of the neuron on input vector $\underline{s}$. If one partitions $\mathbb{R}^n$ by the at most

[1] This reference is frequently cited when using this result. Cover himself, however, attributes the first proof of this theorem to (Schläfli, 1901).

$m \cdot (2n)^2$ hyperplanes that arise for all $\underline{s} \in S$, the resulting regions consist of delay vectors $\underline{d}$ that are equivalent with regard to all input vectors $\underline{s} \in S$. Estimating the number of different regions, one has to take into account that the hyperplanes not necessarily pass through the origin. But the number of different regions of $\mathbb{R}^n$ generated by $m \cdot (2n)^2$ arbitrary hyperplanes is at most as large as the number of different regions of $\mathbb{R}^{n+1}$ generated by $m \cdot (2n)^2$ hyperplanes that all pass through the origin. By Lemma 2.6 this partition consists of at most $2(4emn)^n$ different regions. Hence, for inputs from S it suffices to consider these many delay vectors.

Now we show that for each fixed delay vector at most $2(2em)^n$ many weight vectors are relevant. The upper bound (1) follows then from this number and the number of different delay vectors. For each fixed input vector $\underline{s} \in S$ and each delay vector $\underline{d}$ there are at most $2n - 1$ hyperplanes that have to be considered corresponding to the intervals during which there are pulses active. Each hyperplane is characterized by a subset of $\{w_1, \ldots, w_n\}$ and by the threshold θ_v. If for the given $\underline{s}$ and $\underline{d}$ two weight vectors of the spiking neuron result in different outputs, then these outputs must be different for one of the intervals and hence, for the hyperplane corresponding to this interval. Consequently, the number of regions of the space $\mathbb{R}^{n+1}$ of weights $w_1, \ldots, w_n$ and threshold θ_v is not larger than the number of regions that arise from the at most $2n - 1$ hyperplanes. Taking into account all $\underline{s} \in S$, the space $\mathbb{R}^{n+1}$ is partitioned by at most $m \cdot (2n - 1)$ hyperplanes that all pass through the origin. By Lemma 2.6 the number of different regions that arise from these hyperplanes is bounded by $2(2em)^n$. From this (1) follows.

Finally, the following claim implies the bound $O(n \log n)$ for the VC-dimension and hence the statement of the theorem.

Claim. The VC-dimension of $\mathcal{S}_n^{\text{ab}}$ is at most $8n \log(2n)$ for all $n \geq 8e^2$.

Assume that S has cardinality m and is shattered by $\mathcal{S}_n^{\text{ab}}$. Hence, all 2^m dichotomies of S can be computed by a spiking neuron. Then (1) implies

$$\begin{aligned} 2^m &\leq 2(4emn)^n \cdot 2(2em)^n \\ &= 4(8e^2m^2n)^n \\ &\leq 4(mn)^{2n}, \end{aligned}$$

where we have used the assumption $n \geq 8e^2$ for the last inequality. Taking logarithms on both sides yields

$$m \leq 2n \log(mn) + 2,$$

which implies

$$m \leq 2n(log(mn) + 1). \tag{2}$$

For any $m \geq \log n$ there is a real number $r \geq 1$ such that $m = r \log(rn)$. (This can easily be seen from the fact that for arbitrary n the function $q_n : [1, \infty) \to [\log n, \infty)$

defined by $q_n(z) = z\log(zn)$ is 1–1 and onto.) Substituting $m = r\log(rn)$ on both sides of (2) yields

$$\begin{aligned} r\log(rn) &\leq 2n(\log(rn\log(rn)) + 1) \\ &= 2n(\log(rn) + \log(\log(rn)) + 1) \\ &\leq 2n(\log(rn) + \log(rn/2) + 1), \end{aligned}$$

where the last inequality follows from $\log(rn) \leq rn/2$. (This requires $rn \geq 4$ which is guaranteed by the assumption $n \geq 8e^2$.) Hence we have

$$r\log(rn) \leq 4n\log(rn).$$

Dividing both sides by $\log(rn)$, which is positive due to $rn \geq 8e^2$, we get

$$r \leq 4n,$$

which implies

$$r\log(rn) \leq 4n\log(4n^2).$$

Resubstituting $m = r\log(rn)$ for the left hand side and rearranging the right hand side yields

$$m \leq 8n\log(2n)$$

as claimed. This completes the proof of Theorem 2.4. □

The bound (1) can also be used to estimate the number of Boolean functions that can be computed by a spiking neuron. Substituting $m = 2^n$ yields the bound $2^{O(n^2)}$. Combining this with the lower bound $2^{\Omega(n^2)}$ of (Muroga and Toda, 1966) for $\mathcal{T}_n^{\mathrm{bb}}$ and our Theorem 2.1(c), we get the upper and the lower bound almost matching.

Corollary 2.7 *There are $2^{\Theta(n^2)}$ many Boolean functions computable by a spiking neuron with binary coding of the inputs.*

For the case of binary coding the analysis can even be made simpler, because the factor $2(4emn)^n$ in (1) that is due to the number of relevant delay vectors can be replaced by a bound that is easier to obtain: One observes that for a set $S \subseteq \{0,1\}^n$ of input vectors at most n^2 many different values have to be considered for each delay. Hence, the number of relevant delay vectors is at most $2^{2n\log n}$. Thus one derives the upper bound $2^{n^2+O(n\log n)}$ for the number of Boolean functions. This result is particularly interesting in the light of the fact that there are at most 2^{n^2} many different functions in $\mathcal{T}_n^{\mathrm{bb}}$ (Muroga, 1971).

2.4 Pseudo-Dimension

When analyzing the PAC-learnability of real-valued function classes the *pseudo-dimension* plays a similar role as the VC-dimension does for binary-valued function classes (Haussler, 1992). Following the terminology of (Macintyre and Sontag, 1993) we say a set $\{\underline{s}^1, \ldots, \underline{s}^m\} \subseteq \mathbb{R}^n$ is *H-shattered* by a class $\mathcal{F}$ of real-valued functions if there exist real numbers $x^1, \ldots, x^m$ such that every dichotomy of $\{(\underline{s}^1, x^1), \ldots, (\underline{s}^m, x^m)\}$ is induced by some function of the form $(\underline{s}, x) \mapsto \mathrm{sign}(f(\underline{s}) - x)$ for some $f \in \mathcal{F}$. (Here $\mathrm{sign}(x)$ denotes the function which outputs 1 if $x \geq 0$, otherwise 0.) Analogously to the VC-dimension, the pseudo-dimension of $\mathcal{F}$ is defined as the largest number m such that there is a set of m elements that is H-shattered by $\mathcal{F}$. Obviously, if $\mathcal{F}$ contains only binary-valued functions then its pseudo-dimension is equal to its VC-dimension.

Using known results about the pseudo-dimension (see, e.g., (Haussler, 1992)) it is easy to derive that the class $\mathcal{T}_n^{\mathrm{aa}}$ has pseudo-dimension $n+1$, which is equal to the number of programmable parameters. From our definitions for binary and analog coding it follows immediately that a set which is shattered by a spiking neuron with binary coding of the output is also H-shattered by a spiking neuron with analog coding of the output. Hence, by Theorem 2.2 the pseudo-dimension of the class $\mathcal{S}_n^{\mathrm{aa}}$ is $\Omega(n \log n)$. Thus, the pseudo-dimension of a spiking neuron is significantly larger than the pseudo-dimension of a sigmoidal gate even when the delays are the only adjustable parameters of the spiking neuron. The following result shows that this lower bound is almost tight.

Theorem 2.8 *The pseudo-dimension of a spiking neuron with n analog coded inputs and analog coded output is $O(n \log n)$.*

Proof. We follow the same lines of reasoning as in the proof of Theorem 2.4, i.e. we first estimate the number of dichotomies that are induced on a set $S = \{(\underline{s}^1, x^1), \ldots, (\underline{s}^m, x^m)\} \subseteq \mathbb{R}^{n+1}$ by the functions $(\underline{s}, x) \mapsto \mathrm{sign}(f(\underline{s}) - x)$ for $f \in \mathcal{S}_n^{\mathrm{aa}}$.

For each fixed $(\underline{s}, x) \in S$, the space $\mathbb{R}^n$ is partitioned into regions of equivalent delay vectors $\underline{d}$ by hyperplanes of the form

$$s_i + d_i + y = s_j + d_j + z$$

and

$$s_i + d_i + u = x,$$

where $u, y, z \in \{0, 1\}$ depending on whether the term corresponds to a starting or ending point of a pulse. For all $(\underline{s}, x) \in S$ together we obtain so at most $m \cdot ((2n)^2 + 2n) = m \cdot 2n(2n+1)$ hyperplanes that define regions of delay vectors that are equivalent with regard to the computation on S. We bound the number of these regions by the number of regions of $\mathbb{R}^{n+1}$ that arise from $m \cdot 2n(2n+1)$ hyperplanes passing through the origin. These are by Lemma 2.6 at most $2(2em(2n+1))^n$

different regions. Hence, for inputs from S it is sufficient to consider these many delay vectors.

Now, for each fixed delay vector the number of relevant weight vectors can be estimated as in Theorem 2.4 by $2(2em)$. Multiplying both bounds, we have that the number of dichotomies that are induced on S by the functions $(\underline{s}, x) \mapsto \text{sign}(f(\underline{s}) - x)$ is at most

$$2(2em(2n+1))^n \cdot 2(2em)^n.$$

If all dichotomies of S are induced then this bound must be greater or equal to 2^m. From this we obtain the claimed result $m = O(n \log n)$ by a calculation which is analogous to that in Theorem 2.4 and omitted here. □

Corollary 2.9 *The class $\mathcal{S}_n^{\text{aa}}$ has pseudo-dimension $\Theta(n \log n)$.*

3 VC-Dimension for Networks of Spiking Neurons

We consider feedforward networks of spiking neurons (SNNs), where the structure, or architecture, of a network is defined in terms of an underlying directed acyclic graph. The network inputs and outputs can encode Boolean or analog variables as in the preceding section. The output of an internal gate is assumed to be an analog variable encoded through the timing of its output spike. The following lower bound for the VC-dimension of an SNN in terms of the number of delays matches the quadratic lower bound in terms of the number of weights due to (Koiran and Sontag, 1997), which holds for sigmoidal neural nets. The agreement between these two bounds is somewhat surprising, since the settings and the constructions are quite different. The result shows in particular that the VC-dimension of an SNN with k adjustable delays can be substantially larger than the VC-dimension of any threshold circuit with k adjustable weights.

Theorem 3.1 *For each n one can construct a feedforward SNN $\mathcal{N}$ with $O(n)$ edges that has VC-dimension $\Omega(n^2)$. This even holds if all weights and thresholds remain fixed.*

Proof. We first build for any given $m \in \mathbb{N}$ a module M_m that can read out (and remove) the most significant bit from any m-bit binary number that is given in the form of a temporal delay of a spike. More precisely, M_m accomplishes the following (see Fig. 2): Assume that T_m and T'_m are constants which will be defined later. When a spike arrives at IN_1 at time $d = \sum_{i=1}^{m} b_i 2^i$ and at IN_2 at time 0, then OUT_1 emits a spike at time $T_m + \sum_{i=1}^{m-1} b_i 2^i$. Further, OUT_2 emits a spike if and only if $b_m = 1$. This conditional spike from OUT_2 is then generated at time $T'_m + \sum_{i=1}^{m-1} b_i 2^i$.

Module M_m is implemented as a feedforward SNN consisting of 5 neurons (see Fig. 3): Neuron v fires if and only if the spike from IN_1 (which arrives at v at time

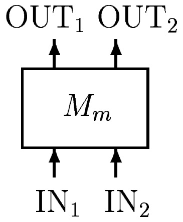

Figure 2: Module M_m.

$d + \Delta - 2^m$) arrives strictly before the spike from IN$_2$ (which arrives at time Δ). Hence, the firing of v is equivalent to $d + \Delta - 2^m < \Delta$, and thus to $d - 2^m < 0$ which holds if and only if $b_m = 0$. This conditional firing of v takes place at time $d + \Delta - 2^m$. Neuron u fires in any case at time $d + \Delta - 2^m$. If v fires, it prevents u' from firing. Hence, u' fires if and only if $b_m = 1$. This conditional firing of u' takes place at time $d + \Delta + \Delta' - 2^m = \Delta + \Delta' + \sum_{i=1}^{m-1} b_i 2^i$. Neuron v' fires if and only if v fires, hence if and only if $b_m = 0$. This conditional firing of v' takes place at time $d + \Delta - 2^m + \Delta' + 2^m = \Delta + \Delta' + \sum_{i=1}^{m-1} b_i 2^i$. (Note that $b_m = 0$.) Thus v'' (which defines OUT$_1$) fires in any case at time $\Delta + \Delta' + \Delta'' + \sum_{i=1}^{m-1} b_i 2^i$, and u' (which defines OUT$_2$) fires if and only if $b_m = 1$. We set $T_m = \Delta + \Delta' + \Delta''$ and $T'_m = \Delta + \Delta'$.

The architecture of $\mathcal{N}$ is shown in Fig. 4. The only programmable parameters are the delays $d_1, \dots, d_n$. A fixed offset $d_0 > 0$ is added to prevent a delay of 0 if $d_i = 0$. The shattered set is $\{\underline{e}_i : i = 1, \dots, n\}^2$ where $\underline{e}_i \in \{0,1\}^n$ is the i-th unit vector. Inputs and output are assumed to be encoded in binary. The weights and delays for the edges that are not programmable are suitably chosen. □

Corollary 3.2 *It is impossible to give an upper bound for the VC-dimension of an SNN with fixed weights in terms of the number n of its delays that are adjustable.*

The proof of Corollary 3.2 follows immediately from the preceding construction. One chooses any $m \in \mathbb{N}$ and constructs a network $\mathcal{N}_{n,m}$ with $O(n + m)$ edges and VC-dimension at least $n \cdot m$ that has just n adjustable delays $d_1, \dots, d_n$ as shown in Fig. 3, so that $\mathcal{N}_{n,m}$ can read out the first m bits of any d_i with the help of m modules $M_1, \dots, M_m$ and additional inputs $y_1, \dots, y_m$. Note that the delays in $M_1, \dots, M_m$ are not required to be adjustable.

The following result, which employs a bound from (Goldberg and Jerrum, 1995), shows that the lower bound of Theorem 3.1 is optimal.

Theorem 3.3 *Consider an SNN $\mathcal{N}$ with rectangular pulses where all delays, weights, and thresholds are programmable parameters, and let l be the number of edges. Then the VC-dimension of $\mathcal{N}$ is $O(l^2)$ even for analog coding of the inputs.*

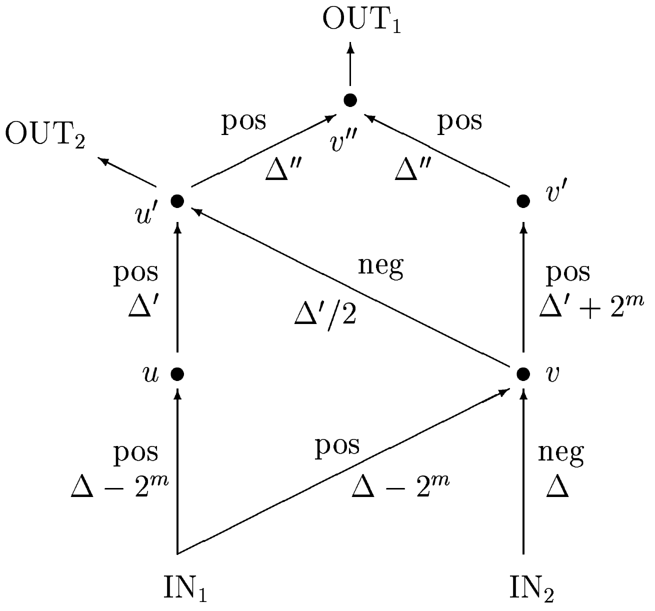

Figure 3: Module M_m and its realization as an SNN. For each edge the sign of its weight (positive or negative) and the value of its delay is indicated.

Proof. Assume without loss of generality that the output of the network is coded in binary. Let l be the number of edges and k be the number of programmable parameters of the network, hence $k = O(l)$. The behavior of the network, i.e. whether its output neuron fires or not, can be decribed by a Boolean formula $\Phi_{k,n}$ that involves as variables the k programmable parameters and the n input variables of the network. To decide whether the output neuron fires we consider all possible paths from an input neuron to the output neuron. There are at most $2^{O(l)}$ such paths. This leads to a Boolean formula $\Phi_{k,n}$ containing $s = 2^{O(l)}$ distinct atomic predicates, where each predicate is a polynomial inequality of degree $d = 1$ over $k + n$ variables. According to Theorem 2.2 of (Goldberg and Jerrum, 1995), the VC-dimension of the class of functions described by this formula is at most $2k \log(8eds) = O(l^2)$. □

4 Computational Complexity of Delay Learning

In order to investigate the computational complexity of learning within the PAC framework one has to specify which class of hypotheses the learner may use. If $\mathcal{S}_n^{\mathrm{bb}}$ were PAC-learnable with some arbitrary polynomial-time computable hypothesis class, then this would imply the same result for DNF (which is one of the major open problems in computational learning theory). This follows from our Theorem 2.1(d) in combination with the corresponding result in (Kearns *et al.*, 1994).

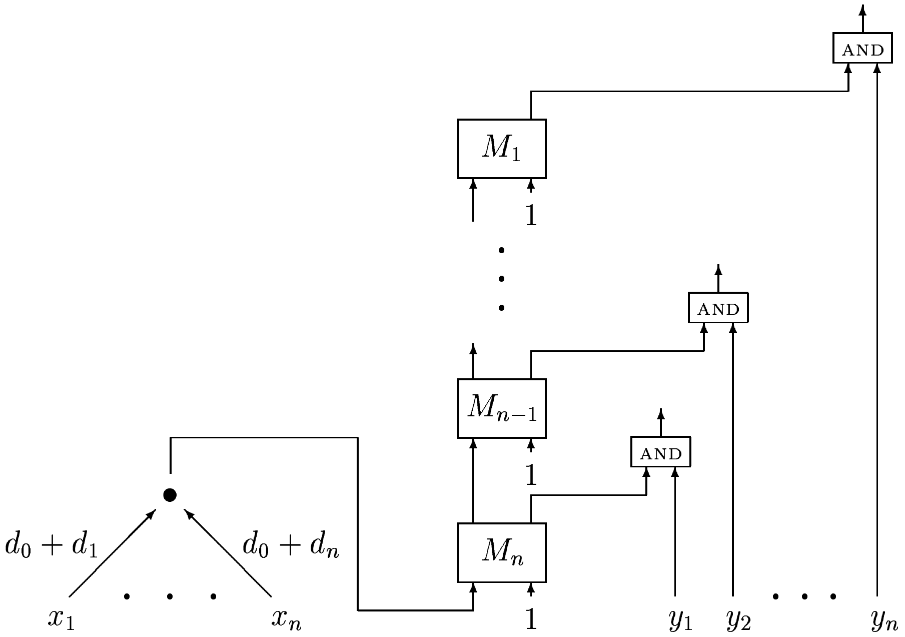

Figure 4: Architecture of the network $\mathcal{N}$. The output gate, which is not shown, fires if and only if any of the n AND-gates fire. An AND-gate fires if and only if it gets pulses from both of its inputs.

In this section we consider the complexity of PAC-learning when only hypotheses from $\mathcal{S}_n^{\text{bb}}$ may be used by the learner, also known as *proper* PAC-learning. This appears to be the more adequate assumption for the analysis of learning for a single spiking neuron.

We investigate the computational complexity of the consistency problem for a spiking neuron which is defined as follows: Given a set of labelled examples from $\{0,1\}^n \times \{0,1\}$, does there exist a function in $\mathcal{S}_n^{\text{bb}}$ that is consistent (i.e., does agree with) all examples?

In the following we show that this problem is NP-complete for a spiking neuron that may choose its delay values only from the set $\{0,1\}$. A spiking neuron with two delay values and binary coding is only slightly more powerful than a Boolean threshold gate, which can be thought of as a spiking neuron with only one delay value. Therefore, this intractability result appears to be optimal in a certain sense. Moreover, the proof shows that the result also holds when the weights and the threshold are kept fixed.

Theorem 4.1 *The consistency problem for a spiking neuron with delays from* $\{0,1\}$ *is NP-complete.*

The proof is by a reduction from 3SET-SPLITTING (Garey and Johnson, 1979), a problem which was also used in (Blum and Rivest, 1992) for intractability results concerning certain two-layer networks of threshold gates. In fact, the problem considered here seems to be closely related to the consistency problem for the AND of two threshold gates analyzed in (Blum and Rivest, 1992). However, their reduction cannot be used here in a straightforward manner (e.g., by flipping the labels to change the AND into an OR), because due to our Theorem 2.1(e) the OR of two threshold gates is not equivalent to a spiking neuron with delays from $\{0,1\}$.

Proof of Theorem 4.1. The problem is in NP because the delay values are binary and the weights can be bounded polynomially in size. The latter is shown similarly as in the case of threshold gates.

To prove NP-hardness we define a polynomial-time reduction from 3SET-SPLITTING, which is the problem to decide for an instance (U, C), where U is a finite set and C is a collection of subsets of U such that $|c| = 3$ for all $c \in C$, if there exists a partition U_0, U_1 of U such that all $c \in C$ satisfy $c \not\subseteq U_0$ and $c \not\subseteq U_1$.[2]

Let (U, C) be given and $n = |U|$. The set of examples is defined as $S = S^+ \cup S^- \subseteq \{0,1\}^{2n} \times \{0,1\}$, where the elements of S^+ and S^- are labelled by "1" and "0", respectively. For a set $I \subseteq \{1, \ldots, 2n\}$ we denote by 1_I the binary vector of length $2n$ that has "1s" exactly at the positions in I.

- Let $1_\emptyset \in S^-$.
- For each $u_i \in U$ let $1_{\{2i-1,2i\}} \in S^+$.
- For each $c \in C$ where $c = \{u_i, u_j, u_k\}$ let $1_{\{2i-1,2i,2j-1,2j,2k-1,2k\}} \in S^-$.

Obviously, there is a function computable in polynomial time that maps each (U, C) to the corresponding S. We show now that (U, C) has a set splitting if and only if there exists a function in $\mathcal{S}_{2n}^{\mathrm{bb}}$ with binary delays that is consistent with S.

$(\Rightarrow)$ Assume that (U, C) has a set splitting $\beta : U \to \{0,1\}$ (i.e., $u_i \in U_j$ iff $\beta(u_i) = j$). Define the weights $w_1, \ldots, w_{2n}$ and threshold θ_v as follows:

$$\left.\begin{array}{lcr} w_{2i-1} & = & 1 \\ w_{2i} & = & -2 \end{array}\right\} \text{ for } i = 1, \ldots, n \quad \text{and} \quad \theta_v = 1/2.$$

Define the delays $d_1, \ldots, d_{2n}$ as:

$$\left.\begin{array}{lcl} d_{2i-1} & = & \beta(u_i) \\ d_{2i} & = & 1 - \beta(u_i) \end{array}\right\} \text{ for } i = 1, \ldots, n.$$

[2] Strictly speaking, the restriction of SET-SPLITTING as defined in (Garey and Johnson, 1979) allows that $|c| \leq 3$. However, it is straightforward to define a reduction that avoids subsets of size 2.

This spiking neuron is consistent with S: For input $1_\emptyset$ it does not fire because $\theta_v > 0$. For each $1_{\{2i-1,2i\}}$ one of the two active inputs generates a pulse of height 1, hence the output is 1. For each $1_{\{2i-1,2i,2j-1,2j,2k-1,2k\}}$ corresponding to a $c \in C$ there is associated with each delay value at least one of w_{2i}, w_{2j}, w_{2k}. Hence, for both delay values the corresponding potential cannot be larger than 0.

($\Leftarrow$) Assume that the spiking neuron is consistent with S. Let g be the threshold function which has threshold θ_v, the weights assigned to delay value 0, and where the weights of delay value 1 are replaced by 0. Define $\beta : U \to \{0,1\}$ as

$$\beta(u_i) = g(1_{\{2i-1,2i\}}).$$

We claim that β is a set splitting of (U, C). Assume the contrary. Then there exists $c \in C, c = \{u_i, u_j, u_k\}$ and $b \in \{0,1\}$ such that

$$\beta(u_i) = \beta(u_j) = \beta(u_k) = b.$$

(i) If $b = 1$ then $g(1_{\{2l-1,2l\}}) = 1$ for each $l \in \{i,j,k\}$. Because $1_\emptyset$ is a negative example and g is a threshold function this implies $g(1_{\{2i-1,2i,2j-1,2j,2k-1,2k\}}) = 1$. Hence, the neuron fires on the input vector corresponding to c, in contradiction to the definition of S.
(ii) If $b = 0$ then consider the threshold function g' consisting of the weights assigned to delay value 1. Accordingly, g' must output 1 on input $1_{\{2l-1,2l\}}$ for each $l \in \{i,j,k\}$ (because the label is 1 and g outputs 0). The label of $1_\emptyset$ then implies that g' outputs 1 on input $1_{\{2i-1,2i,2j-1,2j,2k-1,2k\}}$. It follows that the neuron fires on this input in contradiction to the definition of S.
Finally, (i) and (ii) imply that β is a set splitting of (U, C). □

The fact that the weights need not be modifiable in the previous proof leads to the following stronger result.

Corollary 4.2 *The consistency problem for a spiking neuron with binary delays and fixed weights is NP-complete.*

In a similar way, NP-completeness can be shown for the case that the delays are allowed to take on values from a bounded set $\{0, \dots, k-1\}$ where $k \geq 3$. The reduction is from GRAPH-k-COLORABILITY and is basically a modification of the reduction used in (Anthony and Biggs, 1992) for the AND of k threshold gates. Again, the weights and the threshold can also be kept fixed. Combining this with Theorem 4.1 we get the following result.

Corollary 4.3 *For each $k \geq 2$, the consistency problem for a spiking neuron with delays from $\{0, \dots, k-1\}$ is NP-complete. This holds also for a spiking neuron with fixed weights.*

5 Conclusions

We have investigated a new type of computational model where a set of parameters becomes quite relevant that plays little or no role in other models: transmission delays. We have shown that these new parameters have an even larger impact on the richness of the class of Boolean functions that can be computed by a spiking neuron than those parameters that are traditionally considered to be the main "programmable parameters" of a neuron: the "weights" of its synapses. We have shown that the VC-dimension of a single spiking neuron is superlinear in the number of delays that can be varied, and that the VC-dimension of a network of spiking neurons can grow quadratically with the number of adjustable delays.

Both of these lower bounds hold already for the most simple version of a spiking neuron, respectively network of spiking neurons, where all pulses have a rectangular shape. However, our constructions remain valid for biologically more realistic pulse shapes. These lower bounds are complemented by matching upper bound results, which hold (in terms of the total number of programmable parameters) even if delays *and* weights can be varied simultaneously. Furthermore, these upper bounds hold even in the case of analog network inputs, whereas the lower bounds are valid already in the Boolean case. Hence we get tight bounds for either type of network input.

We have also shown that the learning complexity of a single spiking neuron is surprisingly large, in particular much larger than the learning complexity of a single threshold gate. Similarly as the corresponding result for multi-layer threshold circuits, this should not be interpreted as saying that supervised learning is impossible for a spiking neuron. However it tells us that it will become quite difficult to formulate rigorously provable positive learning results for spiking neurons.

References

Agmon-Snir, H., and Segev, I. (1993), Signal delay and input synchronization in passive dendritic structures. *Journal of Neurophysiology* **70**, 2066–2085.

Anthony, M., and Biggs, N. (1992), "Computational Learning Theory", Cambridge Tracts Theoret. Comput. Sci., Cambridge Univ. Press, Cambridge.

Blum, A. L., and Rivest, R. L. (1992), Training a 3-node neural network is NP-complete, *Neural Networks* **5**, 117–127.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989), Learnability and the Vapnik-Chervonenkis dimension, *J. ACM* **36**, 929–965.

Cover, T. M. (1965), Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Trans. Electron. Comput.* **14**, 326–334.

Garey, M. R., and Johnson, D. S. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman, New York.

Gerstner, W. (1995), Time structure of the activity in neural network models, *Phys. Rev. E* **51**, 738–758.

Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996), A neuronal learning rule for sub-millisecond temporal coding, *Nature* **383**, 76–78.

Goldberg, P. W., and Jerrum, M. R. (1995), Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers, *Machine Learning* **18**, 131–148.

Haussler, D. (1992), Decision theoretic generalizations of the PAC model for neural net and other learning applications, *Inform. and Comput.* **100**, 78–150.

Höffgen, K.-U., Simon, H.-U., and Van Horn, K. S. (1995), Robust trainability of single neurons, *J. Comput. System Sci.* **50**, 114–125.

Kearns, M., Li, M., and Valiant, L. (1994), Learning Boolean formulas, *J. ACM* **41**, 1298–1328.

Kearns, M. J., Schapire, R. E., and Sellie, L. M. (1992), Toward efficient agnostic learning, *in* "Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory," pp. 341–352, ACM Press, New York.

Koiran, P., and Sontag, E. D. (1997), Neural networks with quadratic VC dimension, *J. Comput. System Sci.* **54**, 190–198.

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

Maass, W. (1997b), Networks of spiking neurons: the third generation of neural network models, *Neural Networks*, to appear. (*Extended abstract in* "Proceedings of the Seventh Australian Conference on Neural Networks," pp. 1–10, Canberra, 1996.)

Macintyre, A., and Sontag, E. D. (1993), Finiteness results for sigmoidal "neural" networks, *in* "Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing," pp. 325–334, ACM Press, New York.

Muroga, S. (1971), "Threshold Logic and Its Applications," John Wiley & Sons, New York.

Muroga, S., and Toda, I. (1966), Lower bound of the number of threshold functions. *IEEE Trans. Electron. Comput.* **15**, 805–806.

Murray, A., and Tarassenko, L. (1994), “Analogue Neural VLSI: A Pulse Stream Approach,” Chapman & Hall, London.

Schläfli, L. (1901), “Theorie der vielfachen Kontinuität,” Zürcher & Furrer, Zürich. (*Reprinted in* Schläfli, L. (1950), “Gesammelte Mathematische Abhandlungen,” Band I, Birkhäuser, Basel.)

Tuckwell, H. C. (1988), “Introduction to Theoretical Neurobiology,” Vols. 1 and 2, Cambridge Univ. Press, Cambridge.

Valiant, L. G. (1984), A theory of the learnable, *Communications of the ACM* **27**, 1134–1142.

Zador, A. M., and Pearlmutter, B. A. (1996), VC dimension of an integrate-and-fire neuron model, *Neural Computation* **8**, 611–624.