# On the Complexity of Learning for a Spiking Neuron

**(Extended Abstract)**

**Wolfgang Maass*** and **Michael Schmitt***

## Abstract

Spiking neurons are models for the computational units in biological neural systems where information is considered to be encoded mainly in the temporal patterns of their activity. They provide a way of analyzing neural computation that is not captured by the traditional neuron models such as sigmoidal and threshold gates (or "Perceptrons").

We introduce a simple model of a spiking neuron that, in addition to the weights that model the plasticity of synaptic strength, also has variable transmission delays between neurons as programmable parameters. For coding of input and output values two modes are taken into account: binary coding for the Boolean and analog coding for the real-valued domain.

We investigate the complexity of learning for a single spiking neuron within the framework of PAC-learnability. With regard to sample complexity, we prove that the VC-dimension is $\Theta(n \log n)$ and, hence, strictly larger than that of a threshold gate. In particular, the lower bound holds for binary coding and even if all weights are kept fixed. The upper bound is valid for the case of analog coding if weights *and* delays are programmable.

With regard to computational complexity, we show that there is no polynomial-time PAC-learning algorithm, unless RP = NP, for a quite restricted spiking neuron that is only slightly more powerful than a Boolean threshold gate: The consistency problem for a spiking neuron using binary coding and programmable delays from $\{0,1\}$ is NP-complete. This holds even if all weights are kept fixed.

The results demonstrate that temporal coding has a surprisingly large impact on the complexity of learning for single neurons.

*Address: Institute for Theoretical Computer Science, Technische Universität Graz, Klosterwiesgasse 32/2, A–8010 Graz, Austria. E-mail: {maass,mschmitt}@igi.tu-graz.ac.at, http://www.cis.tu-graz.ac.at/igi/

## 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 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 ("temporal coding"). Whereas threshold circuits and sigmoidal neural nets provide a suitable model for neural computation in terms of firing rates, they cannot be used for modelling neural computation 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 [14]). 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 input pulses 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" at time $t_i$, this causes a pulse in $v$ of the form $h_i(t - t_i)$ with

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

We assume that the neuron "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 $\theta_v$. 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 $\theta$ 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" $\theta_v$. 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., [19, 7, 13]) 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 [17].

A spiking neuron of this type was called a "spiking neuron of type A" in [14]. 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, similarly as for a threshold gate. 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.

Our results about the VC-dimension of a spiking neuron are complementary to those achieved in [21]. 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 OVERVIEW

In this article we investigate the complexity of learning for a spiking neuron within the PAC-learning framework. (For detailed definitions we refer the reader to [2, 4, 20].) In Section 2 we estimate the computational power and the sample complexity of the spiking neuron model defined above. For binary coding we give upper and lower bounds for the computational power in terms of several classes of Boolean functions. 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 [9], these estimates of the sample complexity in terms of the VC-dimension hold even in the case when the training examples are generated by some arbitrary probability distribution ("agnostic PAC-learning"). In particular, these bounds remain valid when the training examples are not generated by a spiking neuron.

In Section 3 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 is no efficient PAC-learning algorithm for these hypothesis classes unless RP = NP. The intractability results presented in this section have also consequences for the case of agnostic PAC-learning. According to known results [12, 10], polynomial-time agnostic PAC-learning with some hypothesis class $\mathcal{H}$ can be done only if the minimizing disagreement problem for $\mathcal{H}$ is in RP. Now, for each hypothesis class $\mathcal{H}$ the consistency problem can easily be reduced 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. Finally, Section 4 contains some concluding remarks and discussion.

## 2 COMPUTATIONAL POWER AND VC-DIMENSION OF A SPIKING NEURON

We introduce the following notation: The class of Boolean functions that can be computed by a spiking neuron with $n$ binary coded inputs and 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 output. The subclass of $\mathcal{S}_n^{\mathrm{aa}}$ that has its output in $\{0, 1\}$ is denoted by $\mathcal{S}_n^{\mathrm{ab}}$.

A similar notation is used for the threshold and for the sigmoidal gate: The class of Boolean functions computable by a threshold gate is denoted by $\mathcal{T}_n^{\mathrm{bb}}$, and $\mathcal{T}_n^{\mathrm{ab}}$ is the class of halfspaces over $\mathbb{R}^n$. The sigmoidal gate is a neuron model that computes functions from $\mathbb{R}^n$ to $\mathbb{R}$ by applying the sigmoidal function $1/(1+e^{-y})$ to the sum of the weighted inputs. We denote the corresponding function class by $\mathcal{T}_n^{\mathrm{aa}}$.

Finally, $\mu-\mathrm{DNF}_n$ is the class of Boolean functions definable by a DNF formula over $n$ variables where each variable occurs at most once.

### 2.1 COMPUTATIONAL POWER

It is obvious that for binary coding a spiking neuron has at least the computational power of a threshold gate (or "Perceptron"): 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 classes $\mathcal{T}_n^{\mathrm{bb}}$ and $\mu-\mathrm{DNF}_n$ both of which are contained in $\mathcal{S}_n^{\mathrm{bb}}$. The following theorem clarifies the relationships between these classes and gives also an upper bound in terms of disjunctions of threshold gates.

**Theorem 2.1**

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

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

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

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

*e) Each function in $\mathcal{S}_n^{\mathrm{bb}}$ can be computed by an OR of $2n-1$ threshold gates. For $n \geq 2$, there exist functions computable by an OR of two threshold gates that are not in $\mathcal{S}_n^{\mathrm{bb}}$.*

**Proof.** All proofs are straightforward. We therefore restrict the proof to the inequality claims by just giving for each of them a function that separates the two function classes involved.
a) The function $(x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$ can be computed by a threshold gate but not be written as a $\mu-\mathrm{DNF}_3$ formula.
b,c) The function $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$ is in $\mu-\mathrm{DNF}_4$ but cannot be computed by a threshold gate.
d) The function $(x_1 \wedge x_2) \vee (x_2 \wedge x_3)$ is in $\mathcal{S}_3^{\mathrm{bb}}$ but cannot be written as a $\mu-\mathrm{DNF}_3$ formula.
e) The exclusive-OR of two bits can be computed by an OR of two threshold gates but is not in $\mathcal{S}_2^{\mathrm{bb}}$. $\square$

### 2.2 LOWER BOUND FOR THE VC-DIMENSION

The VC-dimension of the classes $\mathcal{T}_n^{\mathrm{bb}}$ and $\mathcal{T}_n^{\mathrm{ab}}$ is known to be $n+1$ (see e.g. [4]). Furthermore, using known results about the pseudo-dimension (see [9]) it is easy to derive that the class $\mathcal{T}_n^{\mathrm{aa}}$ has pseudo-dimension $n+1$. We show now that the VC-dimension of the classes $\mathcal{S}_n^{\mathrm{bb}}$, $\mathcal{S}_n^{\mathrm{ab}}$ and the pseudo-dimension of the class $\mathcal{S}_n^{\mathrm{aa}}$ is strictly larger by a factor of $\Omega(\log n)$. We view in the following results 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 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 [1, 8].

**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 size $|S| = m \cdot k$ 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 $\underline{x}^{i,j}$ for $1 \leq i \leq m$, $1 \leq j \leq k$ where the first $m$ bits of $\underline{x}^{i,j}$ 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 $\underline{x}^{i,j}$ 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

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

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 "0s".

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 used.

### 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 almost optimal (disregarding constant factors) 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. (The argument for the pseudo-dimension is similar.) 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 $\Theta(n \log n)$. The class $\mathcal{S}_n^{\mathrm{aa}}$ has pseudo-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 [5][1] and Proposition A2.1 of [4].

**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 [5], $m$ 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 [4], $2\sum_{k=0}^{n-1}\binom{m-1}{k} \leq 2(e(m-1)/(n-1))^{(n-1)}$ for $m \geq n$. □

**Proof of Theorem 2.4.** The proof is structured as follows: We first estimate the number of dichotomies of an arbitrary finite set $S \subseteq \mathbb{R}^n$ of size $m$ that can be computed by a spiking neuron. This results 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 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

[1] Cover attributes the first proof of this theorem to Schläfli [18].

with regard to the computation of the neuron on input vector $\underline{s}$. If one partitions $\mathbb{R}^n$ by the at most $m \cdot (2n)^2$ 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 $\theta_v$. 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 $\theta_v$ 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 $2(2em)^n$. Hence (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.** For $n \geq 8e^2$, $\text{VC-dim}(\mathcal{S}_n^{\text{ab}}) \leq 8n \log(2n)$.

Assume that $S$ has size $m$ 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. $\square$

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 [16] for $\mathcal{T}_n^{\text{bb}}$ and our Theorem 2.1(c), we get the upper and 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 easier, because the factor $2(4emn)^n$ in (1) that is due to the number of relevant delay vectors can be replaced by a simpler bound: 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 $\mathcal{T}_n^{\text{bb}}$ [15].

## 3 COMPUTATIONAL COMPLEXITY OF PAC-LEARNING FOR A SPIKING NEURON

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 [11].

In this section we consider the complexity of PAC-learning when only hypotheses from $\mathcal{S}_n^{\mathrm{bb}}$ may be used by the learner ("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 $\mathcal{S}_n^{\mathrm{bb}}$ 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 3.1** *The consistency problem for a spiking neuron where each delay is* 0 *or* 1 *is NP-complete.*

The proof is by a reduction from 3SET-SPLITTING [6], a problem which was also used in [3] 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 [3]. 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 3.1.** 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 iff 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 $\theta_v$ 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:

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

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 an EPSP of 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 PSP cannot be larger than 0.

$(\Leftarrow)$ Assume that the spiking neuron is consistent with $S$. Let $g$ be the threshold function which has threshold $\theta_v$, 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 $\beta$ 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

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

[2]Strictly speaking, the restriction of SET-SPLITTING as defined in [6] allows that $|c| \leq 3$. However, it is straightforward to define a reduction that avoids subsets of size 2.

(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 $\beta$ is a set splitting of $(U, C)$. $\square$

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

**Corollary 3.2** *The consistency problem for a spiking neuron with binary delays and certain 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 $\{0, \ldots, k-1\}$ where $k \geq 3$. The reduction is from GRAPH-$k$-COLORABILITY and is basically a modification of the reduction used in [2] for the AND of $k$ threshold gates. Again, the weights and the threshold can also be kept fixed. Combining this with Theorem 3.1 we get the following result.

**Corollary 3.3** *For each $k \geq 2$, the consistency problem for a spiking neuron with delays from $\{0, \ldots, k-1\}$ is NP-complete. This holds also for a spiking neuron with certain fixed weights.*

## 4 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 spiking neuron is superlinear in the number of delays that can be varied, and we have given asymptotically tight bounds for this VC-dimension.

In Section 3 we have 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.

Unfortunately, one cannot hope that the negative learning result of Section 3 is a consequence of the rather simple model for a spiking neuron that we have considered. This negative result, as well as the lower bound on the VC-dimension in Theorem 2.2, can easily be seen to hold also for biologically more realistic shapes of postsynaptic potentials.

## References


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

[2] M. Anthony and N. Biggs. *Computational Learning Theory.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1992.

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

[4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36:929–965, 1989.

[5] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14:326–334, 1965.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

[7] W. Gerstner. Time structure of the activity in neural network models. *Phys. Rev. E*, 51:738–758, 1995.

[8] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner. A neuronal learning rule for submillisecond temporal coding. *Nature*, 383:76–78, 1996.

[9] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

[10] K.-U. Höffgen, H.-U. Simon, and K. S. Van Horn. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50:114–125, 1995.

[11] M. Kearns, M. Li, and L. Valiant. Learning Boolean formulas. *Journal of the ACM*, 41:1298–1328, 1994.

[12] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Proceedings of the Fifth Annual ACM Workshop on Computational

*Learning Theory*, ACM Press, New York, 1992, pp. 341–352.

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

[14] W. Maass. 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*, Canberra, 1996, pp. 1–10.

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

[16] S. Muroga and I. Toda. Lower bound of the number of threshold functions. *IEEE Transactions on Electronic Computers*, 15:805–806, 1966.

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

[18] L. Schläfli. *Gesammelte Mathematische Abhandlungen I.* Birkhäuser, Basel, 1950, pp. 209–212.

[19] H. C. Tuckwell. *Introduction to Theoretical Neurobiology*, vols. 1 and 2. Cambridge University Press, Cambridge, 1988.

[20] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

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