Networks of Spiking Neurons: The Third Generation of Neural Network Models

Wolfgang Maass

Institute for Theoretical Computer Science
Technische Universitaet Graz
Klosterwiesgasse 32/2
A-8010 Graz, Austria
e-mail: maass@igi.tu-graz.ac.at

April 18, 1996

Abstract

The computational power of formal models for networks of spiking neurons is compared with that of other neural network models based on McCulloch Pitts neurons (i.e. threshold gates) respectively sigmoidal gates. In particular it is shown that networks of spiking neurons are computationally more powerful than these other neural network models. A concrete biologically relevant function is exhibited which can be computed by a single spiking neuron (for biologically reasonable values of its parameters), but which requires hundreds of hidden units on a sigmoidal neural net.

This article does not assume prior knowledge about spiking neurons, and it contains an extensive list of references to the currently available literature on computations in networks of spiking neurons and relevant results from neurobiology.

1 Definitions and Motivations

If one classifies neural network models according to their computational units, one can distinguish three different generations. The *first generation* is based on *McCulloch-Pitts neurons* as computational units. These are also referred to as perceptrons or threshold-gates. They give rise to a variety of neural network models such as multilayer perceptrons (also called threshold circuits), Hopfield nets, and Boltzmann machines. A characteristic feature of these models is that they can only give *digital* output. In fact they are *universal* for computations with digital input and output, and every boolean function can be computed by some multi-layer perceptron with a single hidden layer.

The *second generation* is based on computational units that apply to a weighted sum (or polynomial) of the inputs an "activation function" with a continuous set of possible output values, such as the *sigmoid function* $\sigma(y) = 1/(1 + e^{-y})$ or the linear saturated function π with $\pi(y) = y$ for $0 \leq y \leq 1$, $\pi(y) = 0$ for $y < 0$, $\pi(y) = 1$ for $y > 1$. Besides piecewise polynomial activation functions we consider in this paper also "piecewise exponential" activation functions, whose pieces can be defined by expressions involving exponentiation (such as the definition of σ). Typical examples for networks from this second generation are feedforward and recurrent sigmoidal neural nets, as well as networks of radial basis function units. These nets are also able to compute (with the help of rounding at the network output) arbitrary boolean functions. Actually it has been shown that neural nets from the second generation can compute certain boolean functions with *fewer* gates than neural nets from the first generation ([41], [11]). In addition, neural nets from the second generation are able to compute functions with *analog* input and output. In fact they are *universal* for analog computations in the sense that any continuous function with a compact domain and range can be approximated arbitrarily well (with regard to uniform convergence, i.e. the L_∞-norm) by a network of this type with a single hidden layer. Another characteristic feature of this second generation of neural network models is that they support learning algorithms that are based on gradient descent such as backprop.

For a biological interpretation of neural nets from the second generation one views the output of a sigmoidal unit as a representation of the current firing rate of a biological neuron. Since biological neurons, especially in higher cortical areas, are known to fire at various intermediate frequencies between their minimum and maximum frequency, neural nets from the second generation are with regard to this "firing rate interpretation" biologically more realistic than models from the first generation.

However at least with regard to *fast* analog computations by networks of neurons in the cortex, the "firing rate interpretation" itself has become questionable. Perrett et al. ([49]) and Thorpe et al. ([62]) have demonstrated that visual pattern analysis and pattern classification can be carried out by humans in just 100 msec, in spite of the fact that it involves a minimum of 10 synaptic stages from the retina to the temporal lobe. The same speed of visual processing has been measured by Rolls et al. ([53]) in macaque monkeys. Furthermore they have shown that a single cortical area involved in visual processing can complete its computation in just 20-30 msec ([52], [53]). On the other hand the firing rates of neurons involved in these computations are usually below 100 Hz, and hence at least 20-30 msec would be needed just to sample the current firing rate of a neuron. Thus a coding of analog variables by firing rates is quite dubious in the context of fast cortical computations.

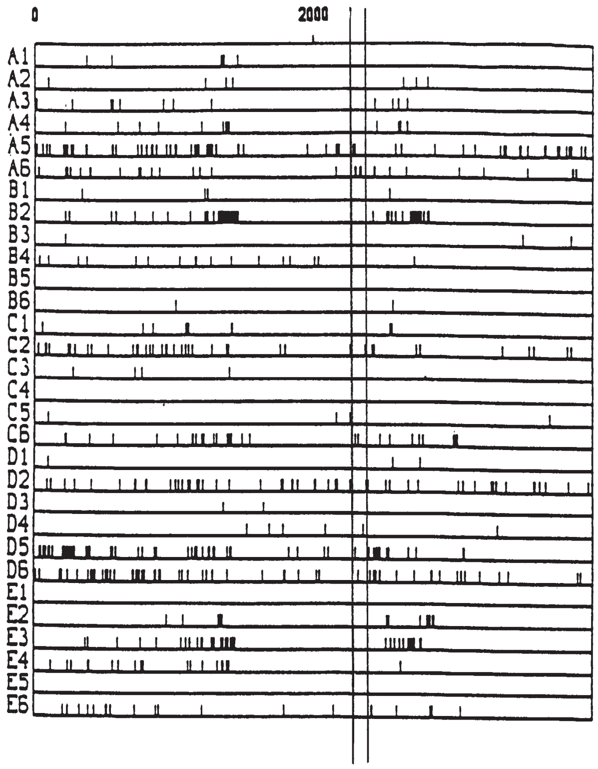

Figure 1: Simultaneous recordings (over 4 seconds) of the firing times of 30 neurons from monkey striate cortex by Krüger and Aiple [30]. Each firing is denoted by a short vertical bar, with a separate row for each neuron.
For comparison we have marked the length of an interval of 100 msec by two vertical lines. This time span is known to suffice for the completion of some complex multilayer cortical computations.

On the other hand experimental evidence has accumulated during the last few years which indicates that many biological neural systems use the timing of single action potentials (or "spikes") to encode information ([1], [2], [6], [3], [4], [5], [13], [20], [27], [33], [54], [58], [59], [62]).

These experimental results from neurobiology have lead to the investigation of a *third generation* of neural network models wich employ *spiking neurons* (or "integrate and fire neurons") as computational units. Recently, one has also started to carry out experiments with related new types of electronic hardware such as pulse stream VLSI (see e.g. [46], [50], [22], [43], [44], [48], [47], [12], [23]). In these new chips one can encode analog variables by *time differences* between pulses, which has practical advantages over other encoding methods. The goal of understanding the capabilities and limitations of this new type of analog neural hardware provides additional motivation for theoretical investigation of the third generation of neural network models.

Mathematical models for "integrate and fire neurons" (or "spiking neurons" as they have been called more recently) can be traced back to [31] (see [63]). There exist a number of variations of this model, which are described and compared in the recent survey (see [15]). With regard to the relationship of these mathematical models to the known behaviour of biological neurons we refer to [1], [3], [4], [8], [9], [20], [24], [56], [57], and [63]. These mathematical models for spiking neurons do not provide a complete description of the extremely complex computational function of a biological neuron. Rather, like the computational units of the previous two generations of neural network models, these are *simplified* models that focus on just a few aspects of biological neurons. However in comparison with the previous two models they are substantially more realistic. In particular they describe much better the actual *output* of a biological neuron, and hence they allow us to investigate on a theoretical level the possiblities for using *time* as a resource for computation and communication. Whereas the timing of computation steps is usually "trivialized" in the models from the preceding two generations (either through an assumed synchronization, or through an assumed stochastic asynchronicity), the timing of individual computation steps plays a key-role for computations in networks of spiking neurons. In fact, the output of a spiking neuron v *consists* of the set $F_v \subseteq \mathbf{R}^+$ of points in time when v "fires" (where $\mathbf{R}^+ = \{x \in \mathbf{R} : x \geq 0\}$).

In the simplest (deterministic) model of a spiking neuron one assumes that a neuron v fires whenever its "potential" P_v (which models the electric membrane potential at the "trigger zone" of neuron v) reaches a certain threshold Θ_v. This potential P_v is the sum of socalled excitatory postsynaptic potentials ("EPSP's") and inhibitory postsynaptic potentials ("IPSP's"), which result from the firing of other neurons u that are connected through a "synapse" to neuron v .The firing of a "presynaptic" neuron u at time s contributes to the potential P_v at time t an amount that is modelled by the term $w_{u,v} \cdot \varepsilon_{u,v}(t-s)$, which consists of a *"weight"* $w_{u,v} \geq 0$ and a *response-function* $\varepsilon_{u,v}(t-s)$. Biologically realistic shapes of such response functions are indicated in Figure 2.

The "weight" $w_{u,v} \geq 0$ in the term $w_{u,v} \cdot \varepsilon_{u,v}(t-s)$ reflects the "strength" (called "efficacy" in neurobiology) of the synapse between neuron u and neuron v . In the context of *learning* one can replace $w_{u,v}$ by a *function* $w_{u,v}(t)$. In addition it has been conjectured that rapid changes of the value of $w_{u,v}(t)$ are also essential for *computations* in biological neural systems. However for simplicity we view here $w_{u,v}$ just as a constant.

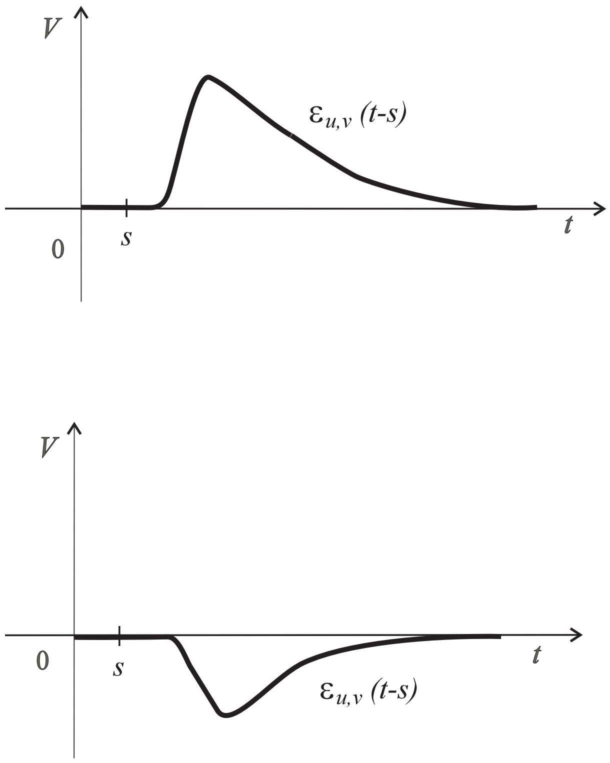

Figure 2: Typical shape of response functions (EPSP and IPSP) of a biological neuron.

The restriction of $w_{u,v}$ to non-negative values is motivated by the assumption that a biological synapse is either "excitatory" or "inhibitory", and that it does not change its "sign" in the course of a "learning-process". Another biologically motivated constraint is that apparently for most biological neurons u either all response-functions $\varepsilon_{u,v}(t-s)$ are "excitatory" (i.e. positive), or all of them are "inhibitory" (i.e. negative). Obviously these constraints have basically no impact on theoretical complexity investigations (just consider pairs of excitatory and inhibitory neurons instead of single neurons), unless one cares about small constant factors in the size of networks, or one wants to model the actual architecture of cortical circuits (see [12], [57]).

It is mathematically more convenient to assume that the potential P_v has value 0 in the absence of postsynaptic potentials, and that the threshold value Θ_v is always > 0 . In a "typical" biological neuron the resting membrane potential is around -70 mV, the firing threshold is around -50 mV, and a postsynaptic potential (i.e. EPSP or IPSP) changes the membrane potential temporarily by at most a few mV.

If a neuron v has fired at time t' , it will not fire again for a few msec after t' , no matter how large its current potential $P_v(t)$ is (*"absolute refractory period"*). Then for a few further msec it is still "reluctant" to fire, i.e. a firing requires a larger value of $P_v(t)$ than usual (*"relative refractory period"*). Both of these refractory effects are modelled by a suitable *"threshold function"* $\Theta_v(t-t')$, where t' is the time of the most recent firing of v. In the deterministic (i.e. noise free) version of the spiking neuron model one assumes that v fires whenever $P_v(t)$ crosses from below the function $\Theta_v(t-t')$. A typical shape of the function $\Theta_v(t-t')$ for a biological neuron is indicated in Figure 3.

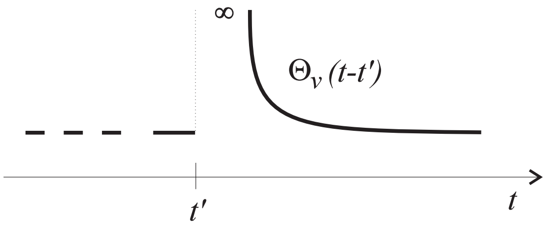

Figure 3: Typical shape of the threshold function of a biological neuron.

Thus formally a *Spiking Neuron Network* (SNN) consists of a finite set V of *spiking neurons*, a set $E \subseteq V \times V$ of *synapses*, a *weight* $w_{u,v} \geq 0$ and a *response function* $\varepsilon_{u,v} : \mathbf{R}^+ \to \mathbf{R}$ for each synapse $\langle u, v\rangle \in E$ (where $\mathbf{R}^+ := \{x \in \mathbf{R} : x \geq 0\}$), and a *threshold function* $\Theta_v : \mathbf{R}^+ \to \mathbf{R}^+$ for each neuron $v \in V$.

If $F_u \subseteq \mathbf{R}^+$ is the set of *firing times* of a neuron u , then the *potential* at the trigger zone of neuron v at time t is given by

$$P_v(t) := \sum\nolimits_{u:\langle u,v\rangle \in E} \ \sum\nolimits_{s \in F_u : s<t} w_{u,v} \cdot \varepsilon_{u,v}(t-s) \ .$$

In a noise-free model a neuron v fires at time t as soon as $P_v(t)$ reaches $\Theta_v(t-t')$, where t' is the time of the most recent firing of v .

For some specified subset $V_{in} \subseteq V$ of *input neurons* one assumes that the firing times ("spike trains") F_u for neurons $u \in V_{in}$ are not defined by the preceding convention, but are given from the outside. The firing times F_v for all other neurons $v \in V$ are determined by the previously described rules, and the output of the network is given in the form of the spike trains F_v for the neurons v in a specified set of *output neurons* $V_{out} \subseteq V$.

Experiments have shown that in vitro biological neurons fire with slightly varying delays in response to repetitions of the same current injection ([3]). Only under certain conditions neurons are known to fire in more reliable manner ([45]). Therefore one also considers the *stochastic* or *noisy* version of the SNN model, where the difference $P_v(t)-\Theta_v(t-t')$ just governs the *probability* that neuron v fires at time t. The choice of the exact firing times is left up to some unknown stochastic processes, and it may for example occur that v does *not* fire in a time interval I during which $P_v(t)-\Theta_v(t-t') > 0$, or that v fires "spontaneously" at a time t when $P_v(t) - \Theta_v(t-t') < 0$.

The previously described noisy version of the SNN model is basically identical with the *spike response model* in [15], [16], and with the other common mathematical models for networks of spiking neurons (see e.g. [1], [4], [63]). Subtle differences exist between these models with regard to their treatment of the refractory effects and the "reset" of the membrane potential after a firing. But these differences will be irrelevant for the results that are considered in this article.

For theoretical results about stable states, synfire chains, associative memory etc. in networks of spiking neurons we refer to [10], [1], [14], [16], [17], [51], [7], [19], [21]. Results about computations with stochastic spiking neurons in firing rate coding can be found in [28], [55], and results about the information transmitted by spiking neurons in [61]. Computations with a somewhat different model of a stochastic spiking neuron are studied in [25] (see also the discussion in [36]), and in [55], [65].

We use in this article the terms *analog, numerical* and *real-valued* interchangeably to denote variables that range over $\mathbf{R}$ or an interval of $\mathbf{R}$. For simplicity we assume that all neural nets from the first two generations that are considered in the following have a feedforward architecture.

2 Simulation and Separation Results

The mathematically simplest one within the range of SNN-models is the one where the firing is deterministic, and both the response functions and the threshold functions are piecewise constant (i.e. "step functions") as indicated in Figure 4. We refer to this version as *type A* in the following. This version of the SNN-model actually captures quite well the intended capabilities of artificial spiking neurons in pulse stream VLSI.

2.1 Computation of Boolean Functions

We first observe that for the case of *boolean* input this model is computationally at least as powerful as neural nets from the first generation. We assume that n input bits $x_1, \ldots, x_n$ are given to the SNN via n input neurons $a_1, \ldots, a_n$, where a_i fires at a specific time T_{in} if $x_i = 1$, and a_i does not fire at all if $x_i = 0$. We assume that the output bit of the SNN is given by the firing or non-firing of a specified *output neuron* during some specified time-window. One can then simulate any layered feedforward neural net $\mathcal{N}$ from the first generation by an SNN $\mathcal{N}'$ of type A which has basically the same architecture as $\mathcal{N}$. Only if one wants to respect in $\mathcal{N}'$ the biologically motivated constraint that each neuron in $\mathcal{N}'$ should only trigger EPSP's, or only IPSP's, then each gate of $\mathcal{N}$ has to be simulated by a *pair* consisting of an excitatory and an inhibitory spiking neuron that both get the same input. In $\mathcal{N}'$ one need not

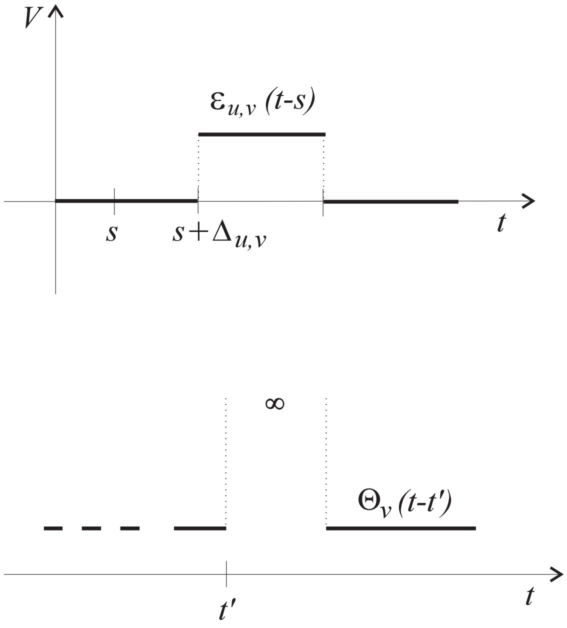

Figure 4: Response- and threshold functions of a spiking neuron of type A.

make use of the possibility to assign for a neuron v different values to the delays $\Delta_{u,v}$ (which model the time that passes until a firing of u has an effect on $P_v(t)$, see Figure 4) for different neurons u with $\langle u, v\rangle \in E$. For a biological neuron these delays $\Delta_{u,v}$ may very well be different, depending on the length of the axon of u and the distance from the synapse to the trigger zone of v.

If one makes use of the possibility to employ for certain neurons v *different* delays $\Delta_{u,v}$ for different neurons u , then one can show that an SNN of type A is in fact computationally *more powerful* than neural nets of the same or similar size from the first or second generation. For that purpose we consider the concrete boolean function $CD_n : \{0,1\}^{2n} \rightarrow \{0,1\}$, which is defined by

$$CD_n(x_1, \ldots, x_n, y_1, \ldots, y_n) = \begin{cases} 1 , & \text{if } x_i = y_i = 1 \\ & \text{for some } i \in \{1, \ldots, n\} \\ 0 , & \text{else .} \end{cases}$$

This function appears to be relevant in a biological context, since it formalizes some form of pattern-matching respectively *coincidence-detection.* A single spiking neuron v of type A (or of any other "reasonable" type) can compute CD_n . One just has to choose the delays to v from the input nodes $a_1, \ldots, a_n$ (for $x_1, \ldots, x_n$) and the input nodes $b_1, \ldots, b_n$ (for $y_1, \ldots, y_n$) in such a way that $\Delta_{a_i,v} = \Delta_{b_i,v}$ for $i = 1, \ldots, n$, and $\Delta_{a_j,v}$ is so much larger than $\Delta_{a_i,v}$ for $j > i$ that the nonzero parts of the response functions $\varepsilon_{a_j,v}$ and $\varepsilon_{a_i,v}$ do not overlap if a_j and a_i fire simultaneously.

Theorem 1

a) Any threshold circuit $\mathcal{N}$ that computes CD_n has at least $n/log(n+1)$ gates.

b) Any sigmoidal neural net $\mathcal{N}$ with piecewise polynomial activation functions that computes CD_n has $\Omega(n^{1/2})$ gates. For the case of piecewise exponential activation functions (such as σ) one gets a lower bound of $\Omega(n^{1/4})$.

Proof: Let $a_1, \ldots, a_n,\ b_1, \ldots, b_n$ be the input nodes of $\mathcal{N}$ where it receives the values $x_1, \ldots, x_n,\ y_1, \ldots, y_n$ of its $2n$ input variables. We show in fact a slightly stronger result than claimed: The lower bounds hold already for the numbers of those gates in $\mathcal{N}$ that have a direct edge from at least one of the input nodes $b_1, \ldots, b_n$. Thus in the case of *layered neural nets* these are lower bounds for the number of gates on the first hidden layer.

We consider computations of $\mathcal{N}$ where some "fixed" vector $\underline{q} \in \{0,1\}^n$ is assigned to the input nodes $b_1, \ldots, b_n$, so that the output of $\mathcal{N}$ may be viewed as a function of the assignments to the input nodes $a_1, \ldots, a_n$. We only consider the set S of those n assignments $\underline{e}_1, \ldots, \underline{e}_n \in \{0,1\}^n$ to $a_1, \ldots, a_n$ where exactly one of the n input variables has the value 1 . Since $\mathcal{N}$ computes CD_n , it is obvious that for the 2^n different choices of $\underline{q} \in \{0,1\}^n$ the network computes 2^n different functions from S into $\{0,1\}$.

For the proof of *part a)* we fix a linear order $\prec$ on the computation nodes in $\mathcal{N}$ so that each computation node g receives (apart from input nodes $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$) only edges from other computation nodes in $\mathcal{N}$ that *precede* g in this linear order. Consider some arbitrary computation node g in $\mathcal{N}$, and a set Q of assignments $\underline{q} \in \{0,1\}^n$ so that every computation node before g computes a function from S into $\{0,1\}$ (with regard to assignments of inputs from S to the input nodes $a_1, \ldots, a_n$), which does not depend on the assignment of $\underline{q} \in Q$ to $b_1, \ldots, b_n$. Note that for the first computation node in $\mathcal{N}$ we can set $Q := \{0,1\}^n$.

Then for assignments from S to $a_1, \ldots, a_n$ the values which gate g receives from other computation nodes do not depend on the chosen assignment $\underline{q} \in Q$ to $b_1, \ldots, b_n$. Hence the weighted sum of the values which g receives via direct edges from the input nodes $a_1, \ldots, a_n$ and from computation nodes that precede g in $\prec$ assumes at most n different values $r_1 \leq \ldots \leq r_n$ for arbitrary assignments from S to $a_1, \ldots, a_n$ and arbitrary assignments from Q to $b_1, \ldots, b_n$. Obviously the output of g depends only on the value of this weighted sum and on the weighted sum r of those values that g receives via direct edges from input nodes $b_1, \ldots, b_n$. If Θ is the threshold of the threshold gate g , then the minimal i such that $r_i + r \geq \Theta$ can assume at most $n+1$ different values (including the value $i = n+1$ if $r_n + r < \Theta$) . Consequently with different fixed assignments of $\underline{q} \in Q$ to $b_1, \ldots, b_n$ the node g can compute at most $n+1$ different functions from S into $\{0,1\}$. This yields a partition of Q into

$n+1$ equivalence classes, and one can apply the same argument for each of these equivalence classes to the *next* node in $\mathcal{N}$ (with regard to the linear order $\prec$).

If one starts this construction for the first computation node in $\mathcal{N}$ with $Q = \{0,1\}^n$, one gets a partition of Q into at most $(n+1)^k$ equivalence classes after the k-th node. On the other hand the fact that $\mathcal{N}$ computes CD_n implies that the output node of $\mathcal{N}$ computes for each assignment to $b_1, \ldots, b_n$ a *different* function from S into $\{0,1\}$, i.e. it partitions $\{0,1\}^n$ into 2^n different equivalence classes Q . Hence the number s of computation nodes in $\mathcal{N}$ that have a direct edge from at least one of the input nodes $b_1, \ldots, b_n$ satisfies $(n+1)^s \geq 2^n$, i.e. $s \geq n/\log(n+1)$.

The proof of *part b)* is very similar to an argument in Koiran [29]. If one considers just $a_1, \ldots, a_n$ as input nodes of $\mathcal{N}$, then different fixed assignments to $b_1, \ldots, b_n$ can only shift the threshold of those s computation nodes in $\mathcal{N}$ that have direct edges from $b_1, \ldots, b_n$. We now consider a variation $\mathcal{N}'$ of $\mathcal{N}$ where the input nodes $b_1, \ldots, b_n$ are deleted, and the *thresholds* of the abovementioned s gates in $\mathcal{N}$ are viewed as the only "programmable parameters" (or "weights") in the usual sense of VC-dimension theory for neural networks (for a brief survey see [34]). The fact that $\mathcal{N}$ computes CD_n implies that $\mathcal{N}'$ shatters S (with regard to different assignments to these s programmable parameters). Thus $\mathcal{N}'$ has a VC-dimension of at least n . On the other hand the results of Goldberg and Jerrum [18] and Karpinski and Macintyre [26] imply that in this case the number s of programmable parameters in $\mathcal{N}$ satisfies $n = O(s^2)$ in the case of piecewise polynomial activation functions, respectively $n = O(s^4)$ in the case of piecewise exponential activation functions. ∎

2.2 Computation of Functions with Analog Input

We have already shown that for *boolean* inputs a network of spiking neurons of type A has the full computational power of a neural net from the first generation of the same size, and is in fact more powerful. However neural nets from all three generations are also able to process *numerical* inputs from $\mathbf{R}^n$ or $[0,1]^n$, instead of just boolean inputs from $\{0,1\}^n$. For networks of spiking neurons it is natural to encode a numerical input variable $x_i \in \mathbf{R}$ by the firing time $T_{in} - x_i \cdot c$ of input neuron a_i (see also [20]), where $c > 0$ is some constant. Similarly one expects that a numerical output $y \in \mathbf{R}$ is realized in an SNN by the firing of a certain "output neuron" at time $T_{out} - y \cdot c$ where $T_{out} > T_{in}$ is independent from the values $x_1, \ldots, x_n$ of the input variables. For the computation of functions with *boolean* output one can either employ the same output convention as before, or apply rounding (i.e. one considers a firing of the output neuron *before* a certain fixed time T as an output of "1").

An interesting function with regard to separation results is in this context the *"element distinctness function"* $ED_n \ : \ (\mathbf{R}^+)^n \to \{0,1\}$ defined by

$$ED_n(x_1, \ldots, x_n) \ = \ \begin{cases} 1 \ , & \text{if } x_i = x_j \ \text{ for some } i \neq j \\ 0 \ , & \text{if } |x_i - x_j| \geq 1 \text{ for all } i, j \text{ with } i \neq j \\ \text{arbitrary}, & \text{else} \ . \end{cases}$$

If one encodes the value of input variable x_i by a firing of input neuron a_i at time $T_{in} - x_i \cdot c$, then for sufficiently large values of the constant $c > 0$ a single spiking neuron v can compute ED_n (even with $\Delta_{a_i,v} = \Delta_{a_j,v}$ for all $i, j \in \{1, \ldots, n\}$) . This holds for any reasonable type of response functions, e.g. type A, or the type B considered below.

Theorem 2 *Any layered threshold circuit $\mathcal{N}$ that computes ED_n has $\Omega(n \cdot log \ n)$ gates on its first hidden layer.*

Proof: Let k be the number of gates in $\mathcal{N}$ on the first hidden layer. The corresponding k halfspaces partition the input space $\mathbf{R}^n$ into at most 2^k different polytopes (i.e. intersections of halfspaces) so that $\mathcal{N}$ gives the same output for all inputs from the same polytope. For this consideration one has to allow polytopes that are intersections of closed and open halfspaces.

We now consider those $n!$ inputs $\underline{x}_\pi = \langle x_1, \ldots, x_n \rangle \in \{1, \ldots, n\}^n$ that represent all $n!$ permutations π of $\{i, \ldots, n\}$. It suffices to show that each $\underline{x}_\pi$ lies in a different polytope, since this implies that $2^k \geq n!$. Thus assume for a contradiction that two permutations $\underline{x}_\pi$ and $\underline{x}_{\hat{\pi}}$ lie in the same polytope P . By construction the threshold circuit $\mathcal{N}$ gives the same output for all $\underline{x} \in P$. Since P is convex, $\mathcal{N}$ gives not only the same output for $\underline{x}_\pi$ and $\underline{x}_{\hat{\pi}}$, but also for all points on the line L that connects these two points. This yields a contradiction, since $ED_n(\underline{x}_\pi) = ED_n(\underline{x}_{\hat{\pi}}) = 0$, but $ED_n(\underline{x}) = 1$ for some point $\underline{x}$ on this line L . ∎

In order to analyze the complexity of functions with *boolean* output on sigmoidal neural nets, one needs to fix a suitable convention for *rounding* the real-valued output of such net. In order to make our subsequent lower bound result as strong as possible, one may assume here the *weakest* possible rounding convention, where for some arbitrary parameter Θ the real-valued output r of the output node of the net is rounded to 1 if $r \geq \Theta$. No separating interval is required between outputs that are rounded to 0 respectively 1 .

In the same way as for CD_n one can show that any neural net from the second generation that computes ED_n needs to have $\Omega(n^{1/4})$ gates. This lower bound will be improved to $(n-1)/4$ in the following theorem. The proof of this stronger separation result exploits instead of a bound for the VC-dimension Sontag's better upper bound of $2w+1$ [60] for the maximal number d such that *every* set of d different inputs can be shattered by a sigmoidal neural net with w programmable parameters. In order to apply his result in our lower bound argument one has to construct from an arbitrary sigmoidal neural net which computes ED_n a related net that shatters *every* set of $n-1$ inputs.

Theorem 3 *Any sigmoidal neural net $\mathcal{N}$ that computes ED_n has at least $\frac{n-4}{2}-1$ hidden units.*

Proof: Let $\mathcal{N}$ be an arbitrary sigmoidal neural net with k gates that computes ED_n.

Consider *any* set $S \subseteq \mathbf{R}^+$ of size $n-1$. Let $\lambda > 0$ be sufficiently large so that the numbers in $\lambda \cdot S$ have pairwise distance ≥ 2 . Let A be a set of $n-1$ numbers $>$ max $(\lambda \cdot S) + 2$ with pairwise distance ≥ 2 .

By assumption $\mathcal{N}$ can decide for n arbitrary inputs from $\lambda \cdot S \ \cup A$ whether they are all different. Let $\mathcal{N}_\lambda$ be a variation of $\mathcal{N}$ where all weights on edges from the first input variable are multiplied with λ. Then by assigning suitable fixed sets of $n-1$ pairwise different numbers from $\lambda \cdot S \ \cup A$ to the other $n-1$ input variables, $\mathcal{N}_\lambda$ computes any characteristic function over S .

Thus if one considers as *programmable* parameters of $\mathcal{N}$ the $\leq k$ weights on edges from the first input variable of $\mathcal{N}$ and the $\leq k$ thresholds of gates that are connected to some of the other $n-1$ input variables, then $\mathcal{N}$ shatters S with $2k$ programmable parameters. Actually in the more general setting of the subsequent argument we arrive at just $k+1$ programmable parameters, since the occurrences of the factor λ is up to k weights may be counted as a *single* programmable parameter.

Since $S \subseteq \mathbf{R}^+$ of size $n-1$ was chosen *arbitrarily*, we can now apply the result from Sontag [60], which implies that $n-1 \leq 2(k+1)+1$, hence $k \geq (n-4)/2$. Thus $\mathcal{N}$ has at least $(n-4)/2$ computation nodes, and therefore at least $(n-4)/2-1$ hidden units. ∎

Remark: The result of section 4 in Sontag [60] implies that his upper bound , and hence the lower bound of the preceding Theorem 3, remain valid if the neural net $\mathcal{N}$ that computes ED_n employs besides sigmoidal gates also threshold gates.

Apparently for most neurons v in the cortex it is not likely that the "weights"

$w_{u,v}$ of its synapses can be so large that just two synchronous EPSP's suffice to move the potential P_v over the firing threshold $\Theta_v(0)$. In that regard the common mathematical model for a spiking neuron "overestimates" the computational capabilities of a biological neuron. It is more realistic to assume that 6 simultaneously arriving EPSP's can cause a neuron to fire (see the discussion in [64]). Therefore we consider the following variation $\widetilde{ED}_n : (\mathbf{R}^+)^n \to \{0,1\}$ of the function ED_n:

$$\widetilde{ED}_n(x_1,\ldots,x_n) =$$

$$\begin{cases} 1\,, & \text{if there exists some } k \geq 1 \text{ such that } x_1, x_2, x_3, \\ & x_{3k+1}, x_{3k+2}, x_{3k+3} \text{ all have the same value} \\ 0\,, & \text{if every interval I} \subseteq \mathbf{R}^+ \text{ of length 1 contains the values} \\ & \text{of at most 3 input variables } x_i \\ \text{arbitrary,} & \text{else .} \end{cases}$$

In the common model of a spiking neuron the membrane potential $P_v(t)$ is assumed to be a *linear* sum of the postsynaptic potentials. This is certainly an idealization, since isolated EPSP's that arrive at synapses far away from the trigger zone (which is located at the beginning of the axon) are subject to an exponential decay on their way to the trigger zone. Hence such isolated EPSP's have hardly any impact on the membrane potential $P_v(t)$ at the trigger zone. On the other hand EPSP's that arrive *synchronously* at *adjacent* synapses are "boosted" at "hot spots" of the dendritic tree, and hence may have a significant impact on the membrane potential $P_v(t)$ at the trigger zone [56]. We have defined $\widetilde{ED}_n$ in such a way that in spite of these nonlinear effects in the integration of EPSP's it is quite plausible that a biological neuron can compute $\widetilde{ED}_n$ in temporal coding for a fairly large value of n . To compute $\widetilde{ED}_n$ a neuron is required to fire only when two "blocks" consisting of 3 *adjacent* synapses all receive *synchronous* EPSP's. Furthermore a "hair-trigger" situation is avoided, since *no* requirements are made for the case that the neuron receives just 4 or 5 synchronous (or almost synchronous) EPSP's. Only in the case where the neuron receives at most 3 EPSP's during any time interval of "unit-length" (which can have any value) a non-firing of the neuron is assumed.

In order to prove a lower bound for the number of hidden units in arbitrary neural nets $\mathcal{N}$ that compute $\widetilde{ED}_n$ with sigmoidal gates and threshold gates, one proceeds as in the proof of Theorem 3. One now considers arbitrary sets $S \subseteq \mathbf{R}^+$ of size $\lfloor (n-3)/3 \rfloor$. One divides the remaining $n-3$ input variables into $\lfloor (n-3)/3 \rfloor$ blocks of 3 variables that always receive a common value. In a variation $\mathcal{N}_\lambda$ of $\mathcal{N}$ one identifies the first 3 input variables, and multiplies all their weights with a common factor λ . Since $\mathcal{N}$ computes $\widetilde{ED}_n$, the network $\mathcal{N}_\lambda$ with k computation nodes shatters S with the help of $k+1$ programmable parameters. Hence Sontag's result [60] yields $\lfloor (n-3)/3 \rfloor \leq 2(k+1)+1$, i.e. $k \geq (n-14)/6$.

If one plugs in a common estimate for the number n of synapses at a biological neuron, such as $n = 10000$, the preceding inequality yields a lower bound of 1663 for the number $k-1$ of hidden units in $\mathcal{N}$. Hence even if one prefers to plug in somewhat different values for some of the abovementioned constants, the preceding proof for $\widetilde{ED}_n$ (respectively for a variation of $\widetilde{ED}_n$ that reflects different choices of the parameters involved) still yields a lower bound of several hundred for the minimal size of a sigmoidal neural net which computes the same function. Thus we have demonstrated a *substantial* difference between the computational power of *biological neurons* and *sigmoidal "neurons"* (i.e. computational units from the second generation).

For numerical inputs our previously sketched simulation of threshold circuits (i.e. neural nets from the first generation) by a network of spiking neurons of type A fails. More surprisingly, one can even show that there *exists no way* of simulating for numerical inputs an arbitrary threshold circuit with s gates by a network of $f(s)$ spiking neurons of type A, in fact this is impossible for *any* function $f : \mathbf{N} \to \mathbf{N}$. Consider a threshold circuit that outputs 1 for inputs $x_1, x_2, x_3 \in [0,1]$ if $x_1+x_2 = x_3$, and 0 else. Obviously this can be achieved by a circuit with just 3 threshold gates: the circuit outputs 1 if $(x_1+x_2 \geq x_3$ AND $x_1+x_2 \leq x_3)$. However it has been shown that this function from $[0,1]^3$ into $\{0,1\}$ (as well as any restriction to $[0,\gamma]^3$ for some $\gamma > 0$) *cannot* be computed by *any* network of spiking neurons of type A, no matter how many neurons and how much computation time it employs. This follows from a general characterization of the computational power of networks of spiking neurons of type A for numerical inputs in terms of the computational power of a restriction called N^--RAM of the common model of a random access machine (RAM) that is given in [40].

Thus we have arrived here at a limit for the computational power of spiking neurons of type A for numerical inputs. The question arises whether this limitation indicates a weakness of spiking neurons in general, or just a weakness of the extremely simple response- and threshold functions of type A. For that purpose we consider now spiking neurons with *continuous piecewise linear* (instead of piecewise constant) response- and threshold functions, to which we will refer as *spiking neurons of type B*. Examples for the simplest nontrivial response functions of type B are indicated in Figure 5.

With regard to the computational power of spiking neurons of type B it does not make much difference whether one allows here piecewise constant, piecewise linear, or more general types of threshold functions Θ_v , as long as we consider only feedforward computations and the threshold functions Θ_v have the value "∞" for small arguments. Also the concrete shape of the response functions of type B will be irrelevant for the following.

One can show that in contrast to the abovementioned negative result about neural

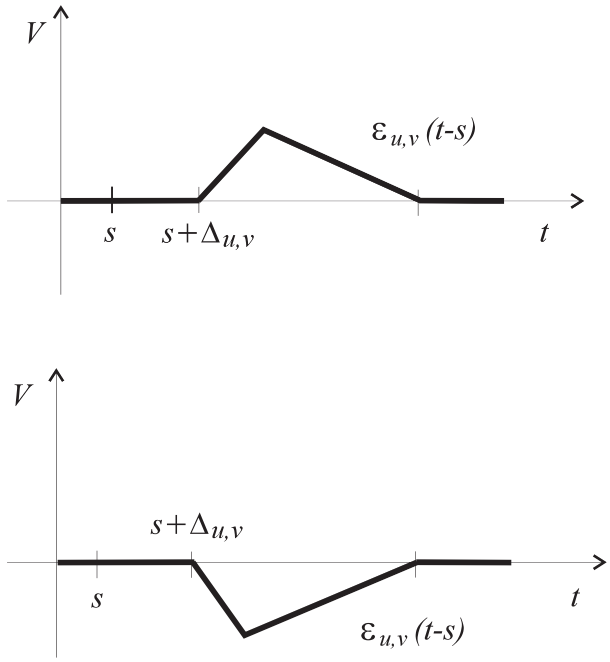

Figure 5: Response functions (EPSP and IPSP) of a spiking neuron of type B. The particular shape of the "triangle"is not important for the results in this article.

nets of type A a network of $O(1)$ spiking neurons with response functions of type B (e.g. as indicated in Figure 5) can simulate any threshold gate even for n real valued input variables. This simulation exploits an important effect of spiking neurons of type B that cannot be realized with spiking neurons of type A: incoming EPSP's and IPSP's can *shift* the firing time of a neuron in a *continuous* manner ([39]). More precisely, for a certain range of the parameters involved, the firing time t_v of a neuron v in response to the firings of presynaptic neurons u at times $T_{in} - x_u \cdot c$ can be written in the form

$$t_v = T_{out} - \sum_{u:\langle u,v\rangle \in E} sign(\varepsilon_{u,v}) \cdot w_{u,v} \cdot x_u \ , \tag{1}$$

where T_{out} does not depend on the values of the x_u , and where $sign(\varepsilon_{u,v}) = 1$ in the case of an EPSP and $sign(\varepsilon_{u,v}) = -1$ in the case of an IPSP. Thus neuron v outputs the *weighted sum*

$$\sum_{u:\langle u,v\rangle \in E} sign(\varepsilon_{u,v}) \cdot w_{u,v} \cdot x_u$$

in *temporal coding* (in response to analog inputs x_u given in temporal coding).

The equation (1) reveals the somewhat surprising fact that the "weights" $w_{u,v}$ of synapses of spiking neurons are able to play in the context of *temporal coding* the same role as for computational units of the first two generations of neural network.

All later layers after the first hidden layer in a layered neural net from the first generation receive just *boolean* input, even if the network input is real-valued. Hence these later layers can easily be simulated by spiking neurons of type A (as indicated before). However a subtle but serious problem arises if one wants to simulate threshold circuits with *boolean* inputs and outputs (or any other type of boolean circuit) with spiking neurons of type B, e.g. with response functions as in Figure 5, which are substantially closer to the biological prototypes in Figure 2 than any response functions of type A. It is obvious that a spiking neuron of type B can simulate a *boolean* gate only if it receives *synchronized* input spikes. The problem is that even if a layer of spiking neurons of type B receives boolean input via *synchronized* input spikes (e.g. in a coding where a spike corresponds to "1" and no spike corresponds to "0"), those neurons on this layer which fire will not fire in a *synchronized* manner, but at slightly different times that depend on their concrete input "bits". The root of this problem (which does not arise for spiking neurons of type A) is the fact that a potential $P_v(t)$ that is the sum of several EPSP's and IPSP's of type B will itself be continuous and piecewise linear, and that the slopes of its linear pieces will depend in particular on the *number* of EPSP's that it receives simultaneously (hence on the concrete "boolean" input in our interpretation). Thus the precise time when $P_v(t)$ crosses the threshold $\Theta_v(0)$ will in general depend on the "boolean" input of the spiking neuron. This causes a serious problem for the simulation of multilayer threshold circuits (or other multilayer boolean circuits) by SNN's of type B, because if those neurons v on the considered layer that fire (and hence represent a "1" in the simulation of a boolean circuit) do not fire in a *synchronized* manner, the simulation of threshold gates or other boolean gates (such as AND) by the *next* layer of spiking neurons of type B becomes impossible.

Theorem 4 *Any threshold circuit with s gates can be simulated for real valued inputs from $[0,1]^n$ by a network of $O(s)$ spiking neurons of type B.*

Proof: Consider first an arbitrary threshold gate G with inputs $\langle x_1, \ldots, x_n \rangle$ from $[0,1]^n$ that outputs 1 if $\sum_{i=1}^{n} \alpha_i x_i \geq \alpha_0$, and 0 else. We show that G can be simulated by a network of a fixed number (i.e. $O(1)$) of spiking neurons of type B with regard to temporal coding of network inputs $x_1, \ldots, x_n$ (for a sufficiently small value of the constant c). One employs here the same construction as for the simulation of a linear (respectively sigmoidal) gate in [39], which yields a spiking neuron v whose firing time represents the weighted sum $\sum_{i=1}^{n} \alpha_i x_i$ in temporal coding. In particular v fires at the latest by a fixed time T (which does not depend on $x_1, \ldots, x_n$) if $\sum_{i=1}^{n} \alpha_i x_i \geq \alpha_0$, and else after time T . We arrange that the resulting EPSP from v arrives at a subsequent spiking neuron v' , which receives in addition an EPSP from an auxiliary spiking neuron whose firing time depends on T_{in} , but not on $x_1, \ldots, x_n$. With a suitable choice of weights and delays of v' one can achieve that v' fires if and only if v fires not later than time T .

Obviously one can simulate in the same way the whole first layer of any given threshold circuit C . In order to simulate the subsequent layers of C with spiking neurons of type B one can employ the construction from [36]. The previously described spiking neurons v' represent the outputs of gates on the first layer of C by firing if and only if the corresponding gate in C outputs 1 . However the precise time at which v' fires in this case depends on $x_1, \ldots, x_n$. Hence before one can use the "boolean" outputs of these gates v' as inputs for other spiking neurons of type B which simulate the subsequent layers of C according to the construction in [36], one has to employ a synchronization module as constructed in the proof of Theorem 2.1 in [36]. ∎

Thus in contrast to SNN's of type A, networks of spiking neurons of type B can simulate neural nets from the *first* generation even for the case of *real-valued* network input. Hence the question arises whether networks of spiking neurons of type B can also simulate (respectively approximate) neural nets from the *second* generation which have real-valued input *and output*. This question is answered affirmatively in [39], by showing that with regard to temporal coding of real-valued variables x from a sufficiently small range $[0, \gamma]$ *any continuous function* $F \; : \; [0, \gamma]^n \to [0, \gamma]^k$ can be approximated arbitrary closely (with regard to uniform convergence, i.e. L_∞) by a network of spiking neurons of type B with just one hidden layer.

Furthermore if F is computed by a neural net with s gates from the second generation, which employ as activation function the linear saturated function π_γ : $[0, \gamma] \to [0, \gamma]$ defined by

$$\pi_\gamma(y) = \begin{cases} \gamma \, , & \text{if } \gamma < y \\ y \, , & \text{if } 0 \leq y \leq \gamma \\ 0 \, , & \text{if } y < 0 \; , \end{cases}$$

then the approximating network of spiking neurons requires only $O(s)$ neurons. The result of [32] implies that *any* continuous function $F : [0, \gamma]^n \to [0, \gamma]^k$ can be approximated by such net.

Thus one may say that with regard to circuit complexity for computing analog functions, networks of spiking neurons of type B are at least as powerful as neural nets from the second generation (with the linear saturated activation function). Furthermore our previously described *lower* bounds for the size of neural nets from the first two generations (for nets that compute the functions CD_n , ED_n or $\widetilde{ED_n}$) imply that networks of spiking neurons of type B are in fact *strictly more powerful* than neural nets from the first two generations: in order to achieve separation results between SNN's of *type B* and neural nets from the first two generations it just remains to verify that a single spiking neuron of type B can compute CD_n , ED_n and $\widetilde{ED_n}$.

We refer to [36] and [39] for details of the proofs of several of the abovementioned simulation results. It can be seen from these proofs that they do not actually require for positive results about the computational power of SNN's of type B that the response- or threshold functions are piecewise linear (i.e. of type B). Rather it suffices to assume that they just have a small linearly increasing respectively decreasing segment, a property which is approximately satisfied by EPSP's and IPSP's of biological neurons (see Figure 2). In [48], [37] a complete characterization of the computational power of SNN's of type B is given in terms of a restriction (called N-RAM) of the familiar model of a random access machine.

In addition it is shown in [39] that the simulation of sigmoidal neural nets by SNN's can also be carried out with the biologically more realistic model of a *stochastic* or *noisy* spiking neuron. It is also easy to see that the here considered functions CD_n , ED_n and $\widetilde{ED}_n$ can be computed by a single *noisy* spiking neuron. Furthermore it is shown in [38] that even with *very noisy* spiking neurons one can in principle carry out arbitrary *digital* computations with any desired degree of reliability. However noise certainly affects the computational power of networks of spiking neurons, and we refer to [42] with regard to methods for proving *lower* bounds for networks of noisy spiking neurons.

References

[1] M. Abeles, *Corticonics: Neural Circuits of the Cerebral Cortex*, Cambridge University Press, 1991.

[2] M. Abeles, H. Bergman, E. Margalit, E. and Vaadia, "Spatiotemporal firing patterns in the frontal cortex of behaving monkeys", *J. of Neurophysiology*, vol. 70, pp 1629–1638, 1993.

[3] A. Aertsen, ed., *Brain Theory: Spatio-Temporal Aspects of Brain Function*, Elsevier, 1993.

[4] M. A. Arbib, ed., *The Handbook of Brain Theory and Neural Networks*, MIT-Press, Cambridge, 1995.

[5] W. Bair, C. Koch, W. Newsome, K. and Britten, "Reliable temporal modulation in cortical spike trains in the awake monkey", *Proc. of the Symposium on Dynamics of Neural Processing*, Washington, USA, 1994.

[6] W. Bialek, and F. Rieke, "Reliability and information transmission in spiking neurons", *Trends in Neuroscience*, vol. 15, pp 428–434, 1992.

[7] E. Bienenstock, "A model of neocortex", *Network: Computation in Neural Systems*, vol. 6, pp 179–224, 1995.

[8] J. M. Bower, and D. Beeman, *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*, Springer, New York, 1995.

[9] P. S. Churchland, and T. J. Sejnowski, *The Computational Brain*, MIT-Press, 1993.

[10] M. C. Crair, and W. Bialek, "Non-Boltzmann dynamics in networks of spiking neurons", *Advances in Neural Information Processing Systems*, vol. 2, Morgan Kaufmann, San Mateo, pp 109–116, 1990.

[11] B. DasGypta, G. Schnitger, "Analog versus discrete neural networks", *preprint*, (1996).

[12] R. J. Douglas, C. Koch, M. Mahowald, K. A. C. Martin, and H. H. Suarez, "Recurrent excitation in neocortical circuits", *Science*, vol. 269, pp 981–985, 1995.

[13] D. Ferster, and N. Spruston, N., " Cracking the neuronal code", *Science*, vol. 270, p 756–757, 1995.

[14] W. Gerstner, "Associative memory in a network of 'biological' neurons", *Advances in Neural Information Processing Systems*, vol. 3, Morgan Kaufmann, San Mateo, pp 84–90, 1991

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

[16] W. Gerstner, and J. L. van Hemmen, "How to describe neuronal activity: spikes, rates, or assemblies?", *Advances in Neural Information Processing Systems*, vol. 6, Morgan Kaufmann, San Mateo, pp 463–470, 1994.

[17] W. Gerstner, R. Ritz, and J. L. van Hemmen, "A biologically motivated and analytically soluble model of collective oscillations in the cortex: I. Theory of weak locking", *Biol. Cybern.*, vol. 68, pp 363–374, 1993.

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

[19] M. Herrmann, J. A. Hertz, and A. Prügel-Bennett, "Analysis of synfire chains", *Nordita preprint*, 95/5 S.

[20] J. J. Hopfield, "Pattern recognition computation using action potential timing for stimulus representations", *Nature*, vol. 376, pp 33–36, 1995.

[21] J. J. Hopfield, and A. V. M. Herz, "Rapid local synchronization of action potentials: Towards computation with coupled integrate-and-fire neurons", *Proc. Natl. Acad. Sci.*, vol. 92, pp 6655–6662, 1995.

[22] T. Horinchi, J. Lazzaro, A. Moore, and C. Koch, "A delay-line based motion detection chip",*Advances in Neural Information Processing Systems*, vol. 3, Morgan Kaufmann, San Mateo, pp 406–412, 1991.

[23] A. Jahnke, U. Roth, and H. Klar, "Towards efficient hardware for spike-processing neural networks", *Proc. of the World Congress on Neural Networks*, Washington, 1995.

[24] D. Johnston, and S. M. Wu, *Foundations of Cellular Neurophysiology*, MIT-Press, Cambridge, 1995.

[25] K. T. Judd, and K. Aihara, "Pulse propagation networks: A neural network model that uses temporal coding by action potentials", *Neural Networks*, vol. 6, pp 203–215, 1993.

[26] M. Karpinski, and A. Macintyre, "Polynomial bounds for VC-dimension of sigmoidal and general Pfaffian neural networks", to appear in the *J. of Comp. and System Sciences*.

[27] R. Kempter, W. Gerstner, J. L. van Hemmen, H. and Wagner, "Temporal coding in the sub-millisecond range: model of barn owl auditory pathway", *Advances in Neural Information Processing Systems*, vol. 8, MIT-Press, Cambridge, 1996, to appear.

[28] C. Koch, and T. Poggio, "Multiplying with synapses and neurons", in: *Single Neuron Computation*, T. McKenna, J. Davis, and S. F. Zornetzer, eds., Academic Press, Boston, 1992.

[29] P. Koiran, "VC-dimension in circuit complexity", *preprint* (1995).

[30] J. Krüger, and F. Aiple, "Multielectrode investigation of monkey striate cortex: spike train correlations in the infragranular layers, *J. Neurophysiology*, vol. 60, pp 798–828, 1988.

[31] L. Lapique, "Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization", *J. Physiol. Pathol. Gen.*, vol. 9, pp 620–635, 1907.

[32] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function", *Neural Networks*, vol. 6, pp 861–867, 1993.

[33] R. Lestienne, "Determination of the precision of spike timing in the visual cortex of anaesthetised cats", *Biol Cybernetics*, vol. 74, pp 55–61, 1996.

[34] W. Maass, "Vapnik-Chervonenkis dimension of neural nets", in: *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, ed., MIT Press (Cambridge, 1995), pp 1000–1003.

[35] W. Maass, "On the computational complexity of networks of spiking neurons", *Proc. of the 1994 Conference on Neural Information Processing Systems 7, NIPS '94*, G. Tesauro, D. S. Touretzky, and T. K. Leen, eds., MIT Press (Cambridge), pp 183–190, 1995.

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

[37] W. Maass, "Analog computations on networks of spiking neurons", *Proc. of the 7th Italian Workshop on Neural Nets*, World Scientific Press, 1995.

[38] W. Maass, "On the computational power of noisy spiking neurons", *Advances in Neural Information Processing Systems*, vol. 8, MIT-Press (Cambridge), 1996, to appear.

[39] W. Maass, "An efficient implementation of sigmoidal neural nets in temporal coding with noisy spiking neurons", 1996, submitted for publication.

[40] W. Maass, and B. Ruf, "On the relevance of the shape of postsynaptic potentials for the computational power of Spiking Neurons", *Proc. of the International Conference on Artificial Neural Networks, ICANN'95*, EC2&Cie, Paris, pp 515–520, 1995.

[41] W. Maass, G. Schnitger, and E. Sontag, "On the computational power of sigmoid versus boolean threshold circuits", *Proc. of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pp 767-776, 1991. extended version appeared in: *Theoretical Advances in Neural Computation and Learning*, V. P. Roychowdhury, K. Y. Siu, A. Orlitsky, editors, Kluwer Academic Publishers (Boston), pp 127–151, 1994.

[42] W. Maass, and P. Orponen, "The computational power of noisy and probabilistic analog neural nets", preprint, 1996.

[43] M. Mahowald, *VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function*, Phd-dissertation at the Cal. Inst. of Technology, 1992.

[44] M. Mahowald, "An Analog VLSI System for Stereoscopic Vision", *Kluwer Academic Publishers*, Boston, 1994.

[45] Z. F. Mainen, and T. J. Sejnowski, "Reliability of spike timing in neocortical neurons", *Science*, vol. 268, pp 1503–1506, 1995.

[46] C. Mead, *"Analog VLSI and Neural Systems"*, Addison-Wesley (Reading), 1989.

[47] J. L. Meador, A. Wu, C. Cole, N. Nintunze, and P. Chintrakulchai, "Programmable impulse neural circuits", IEEE Trans. on Neural Networks, vol. 2, pp 101–109, 1991.

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

[49] D. I. Perrett, E. T. Rolls, and W. C. Caan, "Visual neurons responsive to faces in the monkey temporal cortex", *Experimental Brain Research*, vol. 47, pp 329–342, 1982.

[50] G. A. Pratt, *Pulse Computation*, Phd-thesis in the Dept. of Elec. Eng. and Comp. Sci., MIT, Cambridge, USA, 1989.

[51] R. Ritz, W. Gerstner, U. Fuentes, and L. van Hemmen, "A biologically motivated and analytically soluble model of collective ascillations in the cortex: II. Applications to binding and pattern segmentation", *Biol. Cybernetics*, vol. 71, pp 349–358, 1994.

[52] E. T. Rolls, "Brain mechanisms for invariant visual recognition and learning", *Behavioural Processe*, vol. 33, pp 113–138, 1994.

[53] E. T. Rolls, and M. J. Tovee, "Processing speed in the cerebral cortex, and the neurophysiology of visual backward masking", *Proc. Roy. Soc. B.*, vol. 257, pp 9–15, 1994.

[54] T. J. Sejnowski, "Time for a new neural code?", *Nature*, vol. 376, pp 21–22, 1995.

[55] J. Shawe-Taylor, P. Jeavons, and M. Van Daalen, "Probabilistic bit stream neural chip: theory.", *preprint*, 1995.

[56] G. M. Shepherd, *Neurobiology*, 3rd ed., Oxford University Press, New York, 1994.

[57] G. M. Shepherd, ed., *The Synaptic Organization of the Brain*, 3rd ed., Oxford University Press, New York, 1990.

[58] W. Singer, "Synchronization of Neuronal Responses as a Putative Binding Mechanism", In: *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, ed., MIT-Press, Cambridge, pp 960–964, 1995. 1995.

[59] W. Softky, "Sub-millisecond coincidence detection in active dendritic tree", *Neuroscience*, vol. 58, pp 13–41, 1994.

[60] E. D. Sontag, "Shattering all sets of k points in 'general position' requires $(k-1)/2$ parameters", *preprint*, (Feb. 1996).

[61] C. F. Stevens, and A. Zador, "Information through a spiking neuron", *Advances in Neural Information Processing Systems*, vol. 8, MIT Press (Cambridge), 1996, to appear.

[62] S. T. Thorpe, and M. Imbert, "Biological constraints on connectionist modelling", In: *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman-Soulié, and L. Steels, eds., Elsevier, North-Holland, 1989.

[63] H. C. Tuckwell, *"Introduction to Theoretical Neurobiology"*, vol. 1 and 2, Cambridge University Press, Cambridge, 1988.

[64] L. G. Valiant, *"Circuits of the Mind"*, Oxford University Press, 1994.

[65] J. Zhao, *"Stochastic Bit Stream Neural Networks: Theory, Simulations and Applications"*, Phd. Thesis in the Dept. of Comp. Sci., Royal Holloway, University of London, 1995.