Electronic Colloquium on Computational Complexity, Report No. 30 (2000)

A Simple Model for Neural Computation with Firing Rates and Firing Correlations

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

Abstract

A simple extension of standard neural network models is introduced that provides a model for neural computations that involve both firing rates and firing correlations. Such extension appears to be useful since it has been shown that firing correlations play a significant computational role in many biological neural systems. Standard neural network models are only suitable for describing neural computations in terms of firing rates.

The resulting extended neural network models are still relatively simple, so that their computational power can be analyzed theoretically. We prove rigorous separation results, which show that the use of firing correlations in addition to firing rates can drastically increase the computational power of a neural network.

On the side one of our separation results also throws new light on a question that involves just standard neural network models: We prove that some high-order sigmoidal neural nets can compute boolean functions which require for their computation with first-order sigmoidal units a substantially larger neural net, without imposing restrictive conditions on the architecture, parameters, or activation functions of the first-order sigmoidal neural nets.

1 Introduction

A large number of results in experimental neurophysiology show that in addition to firing rates also *correlations* between firing times of neurons are relevant for neural coding and computation in various biological neural systems (see for example [deCharms, Merzenich, 1996], [Engel et al., 1992],

ISSN 1433-8092

[Eggermont, 1990], [Hatsopoulos et al.,1996], [Kreiter, Singer, 1996], [Krüger, 1991], [MacLeod, Laurent, 1996], [Sillito et al., 1994], [Singer, 1995], [Vaadia et al., 1995]). The standard neural network models, such as threshold circuits or sigmoidal neural nets (MLP's), are not suitable for evaluating the potential capabilities and limitations of such neural systems, since they are tailored to modeling neural computation just in terms of firing rates.

There exist of course substantially more detailed mathematical models for biological neural systems, where one can study correlations between firing times, such as networks of integrate-and-fire neurons or "spiking neurons" (see for example [Bower, Beeman, 1995], [Gerstner, 1995], [Maass, 1996a], [Maass, 1997a], [Tuckwell, 1988]). However these models have the disadvantage that they keep track of every single firing of each neuron in the system, which makes them less suitable for exploring analytically those types of neural computations where only the high-level statistical correlations of firing times are relevant for the neural computation, whereas individual firing times are corrupted by various sources of noise.

We initiate in this article a new approach towards modeling neural computation with firing rates and correlations in firing times. We introduce a simple extension of the familiar abstract neural network models (i.e., of threshold-circuits and sigmoidal neural nets) that allows us to model also salient computational features of correlations between firing times in this simplified setting. From a more general point of view one may view this approach as an attempt to project one particularly important new computational aspect of the "third generation of neural network models" (i.e., of networks of spiking neurons, see [Maass, 1996b]) into the simplified "symbolic" setting of the first and second generation of neural network models (threshold circuits and sigmoidal neural nets).

There are certainly advantages to studying salient aspects of computations and learning in biological neural systems in a simplified setting. In fact, a large portion of our present insight into the functioning of biological neural systems has been obtained by building on ideas that have emerged from the investigation of *extremely simplified* models for biological neural systems, such as networks of McCulloch-Pitts neurons (i.e., threshold circuits) and sigmoidal neural nets. However, one has to make sure that such simplified models capture the *essence* of the organization of computation and learning in the more complex biological models. In this regard the usefulness of standard neural network models has reached a limit. They are not able to reflect the computational role of transient correlations in firing times of biological neurons, although it has become clear during the last few years that such correlations play an important role for computations in biological neural systems.

Several models that reflect computational effects of firing correlations in a simplified setting have already previously been proposed ([Milner, 1974], [von der Malsburg, 1981], [Eckhorn et al., 1990], [Abeles, 1991], [Lumer, Huberman, 1992], [Shastri, Ajjanagadde, 1993], [Phillips, Singer, 1997]). The model in [Shastri, Ajjanagadde, 1993] is of a somewhat different nature, since it models on an abstract level possible effects of the structure of *phases* among neurons that fire at the same frequency. One interesting difference is that the relation of having the same phase is a transitive relation, whereas partial correlation between spike trains - which is reflected in the other models and in our new model - is in general a non-transitive relation. All these preceding models focus on special purpose computations related to selective input enhancement, or "binding" of different features of an object. One has not been able to prove separation results that distinguish the computational power of these models from that of standard neural network models. Since for example sigmoidal neural nets are "universal approximators", it is clear that there *exist* sigmoidal neural nets that compute the same functions as the previously mentioned more complex models that involve also computational effects of firing correlations. Therefore it is impossible to demonstrate a computational advantage of firing correlations by showing that they allow the computation of certain functions that cannot be computed without firing correlations. Instead, the essential question is a *quantitative* one: for example, by how much *larger* would a standard neural network model have to be that computes the same function as some network that also models firing correlations. This question has not been answered for the preceding models. In this sense the conjectured superiority of computational models that also involve firing correlations has remained an open question.

In contrast to these preceding approaches we consider in the article not just special purpose computations, but we model the role of firing correlation within the context of a *general computational model*. Furthermore we *prove* rigorous *quantitative* results about the computational power of this augmented computational model in comparison with standard neural network models.

Similarly as previous general purpose neural network models, such as sigmoidal neural nets, the new model introduced in this article may provide a useful simplified reference model for the investigation of the organization of computation and learning in cortical neural systems. We would like to point out two possible directions for such applications.

So far one has mainly tried to understand the use of firing correlations in sensory cortical systems in the context of symbolic concepts such as "binding" of features. It has remained open whether the organization of pattern recognition in sensory cortical neural systems can be understood in such terms, in spite of their intuitive plausibility. One has been able to build systems based on symbolic concepts such as "binding" that work well for artificial pattern recognition problems with very few inputs. It has remained open whether such systems can be scaled up to handle realistic numbers of inputs, and "real world" noisy pattern recognition tasks. On the other hand there exist many artificial systems that work quite well for "real world" noisy pattern recognition tasks with very large numbers of diverse inputs, but all these systems are based on quite different approaches (see e.g. [Weiss, Kulikowski, 1991], [Ripley, 1996], [Mel, 1997]). Hence there is a chance that sensory neural systems in the cortex also involve complex computations involving firing correlation that cannot be adequately understood in psychological terms such as "binding". This argument raises the question which other computational role could firing correlations possibly play for pattern recognition in complex sensory neural systems. The model introduced in this paper provides a framework for approaching this question.

Another problem area concerns the possible use of firing correlations in higher cortical systems. It may become even more difficult to assign conceptually tractable simple roles such as "binding" to the use of firing correlations in these substantially more complex systems. Therefore it appears to be useful to explore also other possible computational uses of firing correlations in the context of general purpose computational models in order to generate more sophisticated hypotheses for their possible computational role in cortical neural systems. A related question that apparently requires this more abstract level of analysis is the question how a neural system can *compute* in terms of firing correlations. On the basis of the results of [Eckhorn et al., 1990], [Kreiter, Singer, 1996] one may conjecture that every projection of a moving bar or edge on the retina generates in the visual cortex an independent local firing correlation among neurons whose receptive fields are crossed by that bar of edge. In a real world situation where the retina-images of dozens of visual objects simultaneously move, especially if the subject is also moving himself, this mechanism would create simultaneously a large number of *different* local firing correlations in the visual cortex. On the other hand experimental results (see [Traub et al., 1996] and the references therein) also report the occurrence of firing correlations in cortical systems on a much larger level, that are apparently related to the response of the whole cortex to an input. Hence the question arises which computational operations are executed by cortical neural systems to generate from a complex multitude of local firing correlations in the visual cortex and other sensory neural systems a few large-scale firing correlations that represent a system-response of the cortex to the most salient aspects of a complex sensory input. Therefore one needs to explore computational models for computations on firing correlations, i.e. for computations which receive certain firing correlations (in addition to firing rates) as input and produce other firing correlations as part of their output. The model introduced in this article provides a first platform for this type of computational analysis.

An independent motivation for investigating the possible computational role of firing correlations arises from recent experiments with new electronic hardware. In the context of attempts to build reliable devices for signal processing and computation in the micro-watt range one has started to explore silicon-implementations of networks of spiking neurons ("pulsed VLSI"), see e.g. [Murray, Tarassenko, 1994], [Zaghloul et al., 1994]. Very recently (see [van Schaik et al., 1997]) one has also started to explore the computational use of correlations among different pulse streams. However the question how correlations between pulse streams can best be used in a technological context remains wide open. The model introduced in this paper provides a framework where possible uses of pulse correlations in pulsed VLSI can be analyzed and compared with other computational models and known mechanisms of biological neural systems.

In the next section we define our model for computations with firing rates and firing correlations. We explore the computational power of a single unit of this model in section 3. In section 4 we analyze the computational power of multi-layer versions of this model. Conclusions are given in section 5.

2 Definition of the Model

In the usual neural network models one reserves for each gate (or "unit") u of the network a formal variable $o(u)$ which denotes the output of u . In a biological interpretation this variable $o(u)$ models the current firing rate of a neuron u . We will now consider a new type of neural network model $\mathcal{N}$ where one has in addition to the formal variables $o(u)$ for each gate u of $\mathcal{N}$ a second type of formal variables $c(S)$ for various *sets* S of gates in $\mathcal{N}$. In a biological interpretation the formal variable $c(S)$ models the current correlation in the firing times of neurons in this set S . A characteristic feature of this new type of variables is that no additional computational units are needed to compute their value. Instead, their values are determined through the collective activity of the gates according to the subsequent equation (2.2). In particular, a computation may involve many more variables $c(S)$ than there are gates in the network.

We assume that an arbitrary directed graph is given that describes the architecture of $\mathcal{N}$. Each gate v of the network $\mathcal{N}$ receives both types of variables as input, i.e. in addition to the variables $o(u)$ for immediate predecessors u it also receives the variables $c(S)$ for subsets S of its set of immediate predecessors. The output $o(v)$ of v depends on both types of inputs, reflecting the biological property that the firing rate of a neuron may depend both on the firing rates of its predecessors *and* on the firing correlations among subsets of its predecessors. Dually, in addition to the variable $o(v)$ the input of v also contributes to the values of the variables $c(T)$ for sets T with $v \in T$.

Thus, besides the "real" variables $o(u)$, gates in $\mathcal{N}$ also handle a second type of "imaginary" variables $c(S)$. Furthermore, the computational operation of a gate v cannot be decomposed into its operation on "real" variables and its operation on "imaginary" variables, since both its "real" output $o(v)$ and its "imaginary" output $c(T)$ depend on *both* types of input variables. Hence the computational operation of such gate v is reminiscent of a *complex* function in mathematics (for example $z \mapsto e^z$, where the real and imaginary component of its output e^z for an input $z = x + iy$ depend on both x and y). Because of this loose analogy we refer to the new type of gates for neural networks that are investigated in this article as *complex* gates. In contrast we will refer to traditional gates of neural network models as *standard* gates. A *standard neural net* is in our terminology a neural net consisting of standard gates.

The "real" component $o(v)$ of the output of a complex gate v is described by the equation

$$o(v) \;=\; \sigma(\sum_{u \in U} \alpha_{vu} \cdot o(u) + \sum_{S \subseteq U} \alpha_{vS} \cdot c(S) \cdot \prod_{u \in S} o(u) + \alpha_v) \qquad (2.1)$$

where U is the set of immediate predecessors of v in the directed graph that describes the architecture of the network $\mathcal{N}$. The parameters α_{vu}, α_{vS} and α_v may have arbitrary real values. We refer to parameters of the form α_{vu} and α_{vS} as *weights*, and to α_v as the *bias* of the complex gate v . The letter σ in (2.1) denotes some arbitrary activation function $\sigma : \mathbf{R} \to \mathbf{R}$, that can be chosen as in traditional neural network models. If σ is the Heaviside-function (defined by $\sigma(y) = 1$ if $y \geq 0$ and $\sigma(y) = 0$ if $y < 0$) we refer to such gate as a *complex threshold gate.* If σ is a smooth squashing function (such as the logistic sigmoid $\sigma(y) = \frac{1}{1+e^{-y}}$) we refer to the gate defined by (2.1) as a *complex sigmoidal gate.*

The "imaginary" part of the output of the same complex gate v , i.e. the values of the variables $c(T)$ with $v \in T$, is determined by the following equation

$$c(T) = \sigma_c(\sum_{u \in U} \alpha_{Tu} \cdot o(u) + \sum_{S \subseteq U} \alpha_{TS} \cdot \; c(S) \cdot \prod_{u \in S} o(u) \; + \; \alpha_T) \tag{2.2}$$

This equation is not relevant for the computational problems discussed in this section or section 3. Hence we defer its discussion to section 4, where we will look at *multi-layer* computations with complex gates.

The computation of the "real" component $o(v)$ of the output of a complex gate v from inputs $o(u)$ and $c(S)$ is carried out according to equation (2.1) by a high-order sigmoidal gate, where the *order* refers to the degree of the polynomial in the variables $o(u)$ and $c(S)$ to which the activation function σ is applied. High-order gates have previously been discussed in various contexts in the neural network literature (see for example [Durbin, Rumelhart, 1989], [Koch, Poggio, 1992], [Bruck, Smolenski, 1992], [Omlin, Giles, 1996], [Maass, 1997b]), although apparently not in the context of modeling firing correlations. The product $c(S) \cdot \prod_{u \in S} o(u)$ in (2.1) models the following effect: a firing correlation of the neurons in S can increase the firing rate of v if simultaneously all neurons in S fire above their resting firing rate (see [Bernander et al., 1994] for detailed simulation results).

If one assumes that the firing rate $o(u)$ of each input neuron u is proportional to the *probability* $p_u(I)$ that neuron u fires during some fixed short time-window I (say of length 2 ms), and if the spike trains from a set S of such neurons u can be modeled by *independent* stochastic processes, then the probability that all neurons $u \in S$ fire during this time-window I is given by $\prod_{u \in S} p_u(I)$. If the spike trains from these neurons $u \in S$ are *not* generated by independent stochastic processes, one can interpret the value of the variable $c(S)$ as

$$\frac{Pr[\text{ all neurons } u \in S \text{ fire during } I \text{]}}{\prod_{u \in S} p_u(I)} \; . \tag{2.3}$$

Then the term $c(S) \cdot \prod_{u \in S} o(u)$ in equation (2.1) is equal to $Pr[$ all neurons $u \in S$ fire during I $]$. The actual impact that a larger than "usual" (i.e. larger than $\prod_{u \in S} p_u(I))$ probability of synchronous firing of input neurons $u \in S$ may have on the firing rate of v is scaled in equation (2.1) by some generalized "weight" α_{vS} .

In a biological interpretation the value α_{vS} depends on the locations of the synapses between neurons $u \in S$ and v on the axonal tree of the presynaptic neurons u and on the dendritic tree or soma of the postsynaptic neuron v . In addition it depends on the biochemical structure of these synapses, and on the distribution of voltage-dependent channels on the dendritic tree of v . For example, if the transmission delays between the neurons $u \in S$ and their synapses on v all have roughly the same values, if all neurons in u have synapses close together on the dendritic tree of v , and if there exists an accumulation of voltage-dependent channels in a close-by branching point on the way to the soma of v (so that a "dendritic spike" can be generated at such "hot spot"), then an increase in the firing correlation of the neurons in S is likely to have a significant impact on the firing rate of neuron v , and α_{vS} should be given a relatively large value. In a similar way the parameters α_{Tu} and α_{TS} in equation (2.2) also provide a possibility to reflect salient computational aspects of the specific geometrical and biochemical structure of biological neurons (see the discussion at the beginning of section 4). We refer to [Mel, 1994] for further information on biological details of dendritic integration.

Our simple model gives rise to a number of interesting new research questions regarding computations with biological neurons. One question is whether all features of concrete biological neurons that are relevant for their computations in terms of firing rates and firing correlations can be captured through suitable choices of the parameters involved in equations (2.1) and (2.2). In addition the questions arise how many of these parameters can be chosen independently by a biological neuron, which specific constraints limit their independence, which of these parameters can be tuned through a learning process in a developing or mature neuron, and which learning algorithms govern the determination of these parameters in biological neural systems.

Unfortunately there does not yet exist much knowledge about the way in which the firing rate of a neuron v depends on the incoming spike trains from its presynaptic neurons $u \in S$. A quantitative experimental investigation of this dependency would require to record in vivo or in vitro the output spike train of a neuron v simultaneously with the spike trains from *all* presynaptic neurons. It is dubious whether such experiment will become feasible for a typical cortical neuron for all sets S of presynaptic neurons, since a cortical neuron may have as many as 10 000 synapses.

Computer simulations of dendritic integration in a detailed compartmental model for a biological neuron may be viewed as the next best source of information. We refer to [Bernander et al., 1994] for some first results in this direction.

Apart from these biological questions, our abstract computational model also gives rise to the question which ranges of values for the parameters in equations (2.1) and (2.2) can be achieved through silicon-implementations of networks of spiking neurons, and how many of these parameters can be tuned independently.

An interpretation of $c(S)$ according to (2.3) suggest that $c(S)$ may assume arbitrary real values in the range $[0, \infty]$. However, similarly as the variables $o(u)$ in standard neural network models, we allow that these new variables $c(S)$ are interpreted in a more abstract way. Their actual range depends in our model on the specific activation functions σ_c that are employed.

3 The Computational Power of a Complex Threshold Gate

In this section we exhibit concrete computational problems for which one can *prove* that they can be computed more efficiently by a *complex* threshold gate. More precisely, we identify a class of boolean functions F that can be computed by a *single* complex threshold gate, whereas *any* feedforward threshold circuit or sigmoidal neural net computing the same function F needs to have *many* gates. We will discuss after Theorem 3.1 four concrete instances of such functions F which are assumed to be related to fundamental computational tasks of biological neural systems.

Let $\mathcal{C}$ be some arbitrary class of nonempty subsets of $\{u_1, \ldots, u_n\}$. We consider the boolean function $F_{\mathcal{C}} : \{0,1\}^m \rightarrow \{0,1\}$ for $m := n + |\mathcal{C}|$. $F_{\mathcal{C}}$ gives for input vectors $\langle x_i \rangle_{i=1,\ldots,n} {}^\frown \langle x_S \rangle_{S \in \mathcal{C}}$ from $\{0,1\}^m$ the output 1 if and only if $x_S \cdot \prod_{u_i \in S} x_i = 1$ for some $S \in \mathcal{C}$. The operation " $^\frown$ " denotes the concatenation of vectors.

Theorem 3.1 *The function $F_{\mathcal{C}} : \{0,1\}^m \rightarrow \{0,1\}$ can be computed by a single complex threshold gate.*

Let $\tilde{\mathcal{C}} \subseteq \mathcal{C}$ be an arbitrary collection of sets in $\mathcal{C}$ so that no $S, S' \in \tilde{\mathcal{C}}$ exist with $S \subsetneq S'$. Then any standard feedforward threshold circuit computing $F_{\mathcal{C}}$ needs to have at least $|\tilde{\mathcal{C}}|/log(|\tilde{\mathcal{C}}|+1)$ gates, and any standard feedforward sigmoidal neural net with piecewise rational activation functions needs to have $\Omega(|\tilde{\mathcal{C}}|^{1/2})$ gates. For standards sigmoidal neural nets whose activation function involves exponentiation one gets a lower bound of $\Omega(|\tilde{\mathcal{C}}|^{1/4})$ for the required number of gates.

Proof: The computation of $F_{\mathcal{C}}$ by a single complex threshold gate v is straightforward. We assume that v receives the input variables x_i in the form of activations $o(u_i) = x_i$ of its input neurons $u_1, \ldots, u_n$, and the input variables x_S in the form of correlations $c(S)$ of the corresponding set S of input variables.

One sets $\alpha_{vu_i} = 0$ for $i = 1, \ldots, n$, $\alpha_{vS} = 1$ if $S \in \mathcal{C}$ and $\alpha_{vS} = 0$ if $S \notin \mathcal{C}$, and $\alpha_v = -\frac{1}{2}$. With this parameter assignment the complex threshold gate v computes the given function $F_{\mathcal{C}}$.

Let $\mathcal{N}$ be some arbitrary standard feedforward threshold circuit or sigmoidal neural net with m binary input variables $\langle x_i \rangle_{i=1,\ldots,n} {}^\frown \langle x_S \rangle_{S \in \mathcal{C}}$ that computes $F_{\mathcal{C}}$. In the case of a sigmoidal neural net we assume that the real-valued output of its output-gate is *rounded* to 0 or 1 to yield a *boolean*-valued output.

Assume that ℓ gates of $\mathcal{N}$ have a direct edge to at least one of the input nodes for the input variables $x_S,\ S \in \mathcal{C}$. Obviously ℓ is not larger than the *total* number of gates (i.e. computation nodes) in $\mathcal{N}$.

Let $\tilde{\mathcal{N}}$ be a variation of $\mathcal{N}$ whose only input variables are $x_1, \ldots, x_n$, and where the biases of those ℓ gates in $\mathcal{N}$ that have a direct edge from one of the input nodes for the input variables $\{x_S : S \in \mathcal{C}\}$ are the only "programmable parameters" of $\tilde{\mathcal{N}}$. All other weights and biases in $\tilde{\mathcal{N}}$ have the same values as in $\mathcal{N}$. The input nodes for the variables $x_S,\ S \in \mathcal{C}$, are no longer present in $\tilde{\mathcal{N}}$. For arbitrary $\mathbf{c} \in \mathbf{R}^\ell$ we write $\tilde{\mathcal{N}}^{\mathbf{c}}$ for the network that results if the values in $\mathbf{c}$ are assigned to the ℓ programmable parameters of $\tilde{\mathcal{N}}$.

Consider the set

$$D :\ = \ \{\langle a_1, \ldots, a_n \rangle \in \{0,1\}^n :\ \exists S\, (S \in \tilde{\mathcal{C}} \wedge S = \{u_i : a_i = 1\})\ .$$

This is the set of all inputs for $\tilde{\mathcal{N}}$ that encode the "characteristic function" of some set $S \in \tilde{\mathcal{C}}$.

We show that $\tilde{\mathcal{N}}$ – or more precisely the class of all boolean functions from $\{0,1\}^n$ into $\{0,1\}$ that are computed by $\tilde{\mathcal{N}}^{\mathbf{c}}$ for different $\mathbf{c} \in \mathbf{R}^\ell$ – "shatters" the set D . In other words, we show that for every $A \subseteq D$ there exists some $\mathbf{c} \in \mathbf{R}^\ell$ such that $\tilde{\mathcal{N}}^{\mathbf{c}}$ outputs the value 1 for an input $\mathbf{a} \in D$ if and only if $\mathbf{a} \in A$.

Thus, we fix some arbitrary $A \subseteq D$. Let $\mathbf{b}_A \in \{0,1\}^{|\mathcal{C}|}$ be a corresponding assignment to the input variables $\{x_S : S \in \mathcal{C}\}$ of $\mathcal{N}$ which assigns to x_S the value 1 if and only if $S = \{u_i : a_i = 1\}$ for some $\langle a_1, \ldots, a_n \rangle \in A$. Let $\mathbf{c}_A$ be the values of the ℓ programmable parameters of $\tilde{\mathcal{N}}$ which result if one assigns $\mathbf{b}_A$ to the input variables $\langle x_S \rangle_{S \in \mathcal{C}}$ of $\mathcal{N}$. Obviously $\tilde{\mathcal{N}}^{\mathbf{c}_A}$ computes the same function from $\{0,1\}^n$ into $\{0,1\}$ as the net $\mathcal{N}$ with this fixed assignment $\mathbf{b}_A$ to its input variables $\langle x_S \rangle_{S \in \mathcal{C}}$. Since $\mathcal{N}$ computes $F_{\mathcal{C}}$ and since no set in $\tilde{\mathcal{C}}$ is contained in another set in $\tilde{\mathcal{C}}$, it outputs by definition of $\mathbf{b}_A$ the value 1 for input $\langle a_1, \ldots, a_n,\ \mathbf{b}_A \rangle$ from $\{0,1\}^m$ if and only if $\langle a_1, \ldots, a_n \rangle \in A$. Hence $\tilde{\mathcal{N}}^{\mathbf{c}_A}$ with input $\langle a_1, \ldots, a_n \rangle \in D$ outputs 1 if and only if $\langle a_1, \ldots, a_n \rangle \in A$.

The preceding argument shows that the neural net $\tilde{\mathcal{N}}$ with its ℓ programmable parameters shatters the set D of size $|\tilde{\mathcal{C}}|$. Hence the VC-dimension of $\tilde{\mathcal{N}}$ is at least $|\tilde{\mathcal{C}}|$. Since $\tilde{\mathcal{N}}$ has ℓ programmable parameters, any *upper* bound

$B(\ell)$ for its VC-dimension induces for ℓ a *lower* bound of the form

$$\text{least } \ell \text{ such that } B(\ell) \geq |\tilde{\mathcal{C}}| \ .$$

By the definition of ℓ this argument yields the same lower bound for the number of gates in $\mathcal{N}$. We would like to point out that such argument has first been used (in a different context) by Koiran [Koiran, 1996].

In the case of a sigmoidal neural net $\mathcal{N}$ with piecewise rational activation functions the upper bound $O(\ell^2)$ for the VC-dimension of $\tilde{\mathcal{N}}$ from [Goldberg, Jerrum, 1995] yields $\ell = \Omega(|\tilde{\mathcal{C}}|^{1/2})$. In the case of a sigmoidal neural net with the sigmoid activation functions $\sigma(y) = 1/(1 + e^{-y})$ (or other activation functions involving exponentiation) the upper bound $O(\ell^4)$ for the VC-dimension of $\tilde{\mathcal{N}}$ from [Karpinski, Macintyre, 1996] yields $\ell = \Omega(|\tilde{\mathcal{C}}|^{1/4})$.

In the case of a first order threshold circuit $\mathcal{N}$ one can show that $\ell \geq |\tilde{\mathcal{C}}| / \log(|\tilde{\mathcal{C}}| + 1)$ by using a more direct counting argument from the proof of Theorem 1 a) in [Maass, 1996b]. An almost equivalent bound $\ell \geq |\tilde{\mathcal{C}}|/(1 + \log |\tilde{\mathcal{C}}|)$ follows from the estimates in [Cover, 1968]. Note that one cannot use here the upper bound from [Baum, Haussler, 1989] for the VC-dimension of a threshold circuit, since that bound assumes that instead of our ℓ chosen biases *all* weights and biases of the threshold circuit are programmable parameters. ∎

We will now discuss applications of Theorem 3.1 to concrete classes $\mathcal{C}$ of subsets of $\{u_1, \ldots, u_n\}$, for which the computation of the associated boolean function $F_{\mathcal{C}} : \{0,1\}^{n+|\mathcal{C}|} \to \{0,1\}$ appears to be of interest in the context of biological neural systems.

Application 1. Kreiter and Singer [Kreiter, Singer, 1996] have shown that the firing of pairs of neurons in the motion-sensitive area MT of the visual cortex is synchronized when their individual receptive fields are crossed by a single moving bar. In contrast, if the same cells are activated by two different moving bars, responses show no or much fewer synchronous epochs.

Assume that a neuron in a higher cortical area wants to extract from these neurons in area MT the information whether a single moving object crosses at least k vertically adjacent receptive fields, where k is some given parameter. On the basis of the neural coding that is suggested by the results of [Kreiter, Singer, 1996], this neuron has to compute the function $F_{\mathcal{C}}$ for $\mathcal{C} := \{S \subseteq \{u_1, \ldots, u_n\} :$ the neurons $u_i \in S$ represent at least k vertically adjacent locations$\}$, for some fixed spatial map of the neurons $u_1, \ldots, u_n$. For simplicity we assume that the neurons $u_1, \ldots, u_n$ represent some $\ell \times m$ grid under their spatial map.

If $k \leq \ell/2$, this class $\mathcal{C}$ contains a subclass $\tilde{\mathcal{C}}$ of $\geq n/2$ sets with no set contained in another set. Hence a network of $\Omega(n/\log n)$ standard threshold gates or $\Omega(n^{1/2})$ standard sigmoidal neurons with piecewise rational activation functions would be needed to compute this function $F_{\mathcal{C}}$. On the other hand a single complex threshold gate can compute the function $F_{\mathcal{C}}$.

Application 2. Closely related to the previously discussed special cases is the socalled "binding problem" (see e.g. [Milner, 1974], [von der Malsburg, 1981], [Singer, 1995], [Shastri, Ajjanagadde, 1993], [Phillips, Singer, 1997]). Assume that objects in the outside world are analyzed by sensory systems in terms of m basic properties $P_1, \ldots, P_m$ from different categories (e.g. blue, round, large, approaching, honking, ...). Furthermore assume that a collection $\mathcal{C}$ of subsets H of $\{P_1, \ldots, P_m\}$ has been singled out because the existence of an object in the outside world that satisfies a combination $\bigwedge\limits_{P_j \in H} P_j$ of basic properties in H (e.g. large $\wedge$ approaching $\wedge$ honking) is of particular relevance for the system. On the other hand it may be of no relevance for the systems if the same properties $P_j \in H$ are satisfied by *different* objects in the outside world (for example, if the human sensory systems detects in a busy street scene simultaneously a *large* house, an *approaching* child, and a *honking* car in a distant traffic jam).

It has been conjectured that neurons $v_1, \ldots, v_m$, where the firing rate $o(v_i)$ of neuron v_i indicates the absence or presence of an object with basic property P_i , communicate through their firing correlation which basic properties are shared by the same object. Obviously a single complex threshold gate can then decide whether some combination $\bigwedge\limits_{P_j \in H} P_j$ of basic properties is satisfied by a single object, since this amounts to computing the function $F_{\mathcal{C}}$ for the previously described class $\mathcal{C}$. On the other hand, according to Theorem 3.1 neural networks of substantial size are needed to compute the same function $F_{\mathcal{C}}$ with standard sigmoidal neural nets.

The same example will be analyzed further at the beginning of section 4 in the context of multi-layer computations.

Application 3. Many biological neural systems can select particular parts of their input for processing by higher layers, whereas the remainder of their input is ignored for the moment. This mechanism is referred to as "attention" or "awareness" and we refer to [Koch, Crick, 1994] for a detailed discussion of this problem in the context of neurophysiology.

One possible mechanism that has been put forward as a possible neurophysiological correlate of attention is a transient firing correlation of those sensory neurons whose receptive fields fall into the current region of attention. In this context a firing correlation would occur even among neurons that encode a *static* stimulus. Hence it has been conjectured that firing correlation in the context of attention is induced by internal mechanisms of the network, rather then by the stimulus itself.

Within the mathematical framework considered in this article one can model the use of firing correlation for mediating attention in the following way. We assume that there are n input neurons $u_1, \ldots, u_n$, whose firing rates $o(u_1), \ldots, o(u_n)$ constitute the network input. In addition we assume that an additional neuron u_0 fires at a constant rate, and that the firing times of u_0

are correlated with the firing times of those input-neurons u_i which currently receive "attention". In other words, we assume that $c(\{u_0, u_i\})$ is increased for those neurons u_i that currently receive attention through some internal mechanism. This can take place without changing the firing rates $o(u_i)$ of these neurons. Under these assumptions a subsequent complex gate that computes $\sigma(\sum_{i=1}^{n} \alpha_i \cdot c(\{u_0, u_i\}) \cdot o(u_0) \cdot o(u_i))$ can simulate any standard sigmoidal neuron that computes $\sigma(\sum_{i \in A} \alpha_i \cdot o(u_i))$ for the set $A \subseteq \{u_1, \ldots, u_n\}$ of neurons that currently receive attention.

As a particularly simple special case consider the computational task to output 1 if and only if $o(u_i) = 1$ for some $u_i \in A$. This amounts to the computation of the function $F_{\mathcal{C}}$ for the class $\mathcal{C} := \{\{u_0, u_i\} : i = 1, \ldots, n\}$, where u_0 is an additional input neuron with constant output value $o(u_0) = 1$. According to Theorem 3.1 a single complex threshold gate can carry out this task, whereas a neural network consisting of standard sigmoidal neurons needs to have a substantial number of gates to carry out the same computation. One noteworthy feature of the previously described computational task is that a single complex gate v can solve this task which has parameter values $\alpha_{vS} \neq 0$ only for sets S of size 2.

Larger separation results can easily be shown by Theorem 3.1 for more complex computational tasks involving attention.

The last result of this section is not related to the new neural network models that are discussed in the other parts of this paper. It concerns the computational power of two different types of standard sigmoidal gates: high-order sigmoidal gates (also called $\Sigma\Pi$-units [Durbin, Rumelhart, 1989]) and first-order sigmoidal gates. The "order" of a sigmoidal gate is defined as the degree of the polynomial in the input variables that provides the argument for the acitivation function of the gate. In the most common case of a first-order gate this polynomial is simply a weighted sum.

There exist various preceding results on the difference in computational power between high-order and first-order sigmoidal gates (see [Koch, Poggio, 1992] and [Siu et al., 1994] for a survey). Most of them require for the first-order gates the restriction that the activation function is the Heaviside-function (then a first-order sigmoidal gate is simply a threshold gate) and that the depth of the network of first-order sigmoidal gates is bounded by some constant. In [Hajnal et al., 1993] it was shown that the function INNER PRODUCT MOD2 can be computed by a second-order threshold circuit (with linearly many product terms) of depth 2 with polynomially many gates and polynomial size integer weights, whereas any first-order threshold circuit of depth 2 with polynomial size weights needs to have exponentially many gates to compute the same function. Other separation results follow from lower bounds for bounded depth circuits computing PARITY (see [Bruck, Smolenski, 1992] and the survey in [Siu et al., 1994]). These lower bounds provide nice separation results between first-and second-order threshold circuits if one encodes bits by $-1, 1$ instead of $0, 1$. In this case PARITY with n input bits can be computed by a single threshold gate of order n with polynomially many product terms. The lower bounds for first-order threshold circuits of bounded depth can be extended to first-order sigmoidal neural nets of bounded depth with *smooth* activation functions if one assumes that their analog circuit-outputs have some fixed separation between outputs for accepted inputs and outputs for rejected inputs (see [Maass et al., 1994]).

In contrast to the preceding separation results the following Corollary provides a separation between high-order and first-order sigmoidal nets without any restriction on the architecture of the first-order networks for gates with any common activation function, without any restriction on their weights, without a required separation at the output, and without a required encoding of bits by $-1, 1$.

Corollary 3.2 *A high-order sigmoidal gate has strictly more computational power than a first-order sigmoidal gate:*

There exist for any $m \in \mathbf{N}$ boolean functions $F : \{0,1\}^m \rightarrow \{0,1\}$ that can be computed by a single high-order sigmoidal gate with $m/2$ product terms of degree ≤ 2 , but where F requires for its computation with first-order sigmoidal gates a multi-layer circuit with a substantial number of gates. The precise lower bounds depend on the activation function of the first-order sigmoidal gates: F requires

- $\Omega(m/\log m)$ *first-order threshold gates*
- $\Omega(m^{1/2})$ *first-order sigmoidal gates with piecewise rational activation functions*
- $\Omega(m^{1/2})$ *first-order sigmoidal gates with activation functions that involve exponentiation.*

We always assume that the analog output of a sigmoidal neural net is rounded to yield a boolean output.

Proof: Choose $n \in \mathbf{N}$ maximal so that $2n \leq m$. Set $\mathcal{C} := \{S \subseteq \{u_1, \ldots, u_n\} : |S| = 1\}$ in Theorem 3.1. Then $F_\mathcal{C}$ is a function from $\{0,1\}^{2n}$ into $\{0,1\}$, and $|\mathcal{C}| = \Omega(m)$. Obviously $F_\mathcal{C}$ can be computed by a single standard threshold gate with second-order terms.

Theorem 3.1 yields lower bounds for the size of arbitrary feedforward first-order sigmoidal neural nets that compute $F_\mathcal{C}$. These lower bounds are formulated in terms of $|\mathcal{C}|$, but since $|\mathcal{C}| = \Omega(m)$ they yield the same lower bounds in terms of m .

One can make $F_\mathcal{C}$ formally into a function F from $\{0,1\}^m$ into $\{0,1\}$ by adding $m - 2n$ dummy input-variables. ∎

4 Modeling Multilayer Computations with Firing Times and Firing Correlations

In the preceding section we had considered the case where the firing correlations $c(S)$ for sets $S \subseteq U$ of input neurons were given as part of the network input. In this section we consider the dual case where firing correlations among input neurons play no role, but where internally generated firing correlations $c(T)$ for sets $T \subseteq \{v_1, \ldots, v_m\}$ of "hidden" neurons become improtant for the computation. It turns out that the computational power of the network can be tremendously increased through the use of this new set of internal variables $c(T)$.

According to (2.2) the values of these correlation variables $c(T)$ are determined as follows:

$$c(T) = \sigma_c(\sum_{u \in U} \alpha_{Tu} \cdot o(u) + \sum_{S \subseteq U} \alpha_{TS} \cdot \ c(S) \cdot \prod_{u \in S} o(u) \ + \ \alpha_T) \tag{4.1}$$

U is in this equation the set of all neurons which are immediate predecessor of some neuron in the set T . The function σ_c is some suitable activation function which scales the value of $c(T)$ into the desired range of this variable. For simplicity we assume in the following that $c(T)$ ranges over $[0,1]$, and that σ_c is either the Heaviside function or some sigmoidal function (like the activation function σ in equation (2.1)).

In a biological interpretation the parameters α_{Tu} and α_{TS} in equation (4.1) can be used to model details of the geometrical and biochemical structure of the neurons in T and U . The first term $\sum\limits_{u \in U} \alpha_{Tu} \cdot o(u)$ in (4.1) reflects the fact that the firing correlation of the neurons in T can be increased through common input. Hence the value of α_{Tu} should be chosen positive if neuron u has excitatory synapses to all neurons in T, with roughly the same transmission delays to the soma of neurons in T . In fact, one may expect that the value of α_{Tu} is in many cases proportional to $min\{\alpha_{vu} : v \in T\}$.

The second term $\sum\limits_{S \subseteq U} \alpha_{TS} \cdot c(S) \cdot \prod\limits_{u \in S} o(u)$ in (4.1) reflects an alternative way in which firing correlation among neurons in T can be achieved: if each neuron in T receives input from some neuron $u \in S$ where S is a set of presynaptic neurons that fire with a fairly large firing correlation (i.e., $c(S) \cdot \prod\limits_{u \in S} o(u)$ is large). The parameter α_{TS} depends on the connectivity structure between the neuron sets S and T, and on biochemical details of their synapses and of the dendritic trees of the neurons in T .

The computational model that is defined by the equations (2.1) and (4.1) makes it very easy to solve computational problems related to the socalled "binding problem" for scenes with different objects (see our discussion in application 2 at the end of section 3). Assume again that there exists a set of m basic properties $P_1, \ldots, P_m$ of objects, and a collection $\mathcal{C}$ of subsets H of $\{P_1, \ldots, P_m\}$. Furthermore assume that for $H \in \mathcal{C}$ the combination

$\bigwedge H := \bigwedge_{P_j \in H} P_j$ of basic properties P_j is of particular relevance for the neural system if it occurs in a *single* object. Finally assume that the computational goal of this system is to output 1 if the input indicates the presence of objects O_i and $O_{\tilde{i}}$ so that O_i satisfies $\bigwedge H$ and $O_{\tilde{i}}$ satisfies $\bigwedge \tilde{H}$, where H and $\tilde{H}$ are two arbitrary but different sets in $\mathcal{C}$.

This concrete computational problem involves the "binding problem", since it is required that *one* object O_i satisfies *all* properties from some combination H in $\mathcal{C}$, and *one* object $O_{\tilde{i}}$ satisfies *all* properties from some combination $\tilde{H}$ in $\mathcal{C}$. It would not suffice if both O_i and $O_{\tilde{i}}$ each satisfy some properties in $H \cup \tilde{H}$, even if altogether all properties in $H \cup \tilde{H}$ get satisfied by at least one of the objects O_i and $O_{\tilde{i}}$.

We will now construct a network $\mathcal{N}$ that solves this computational problem with computations that involve firing rates and firing correlations according to equation (2.1) and (4.1). We assume that the network $\mathcal{N}$ has n input units $u_1, \ldots, u_n$, and that $o(u_i) = 1$ indicates that object O_i is present (else $o(u_i) = 0$), $i = 1, \ldots, n$. On the second layer of the network $\mathcal{N}$ we have m units $v_1, \ldots, v_m$ so that $o(v_j) = 1$ if and only if there is some object O_i present that has property P_j . Assume that the weights α_{ji} encode the knowledge which objects O_i have which basic properties P_j : $\alpha_{ji} = 1$ if O_i has property P_j , else $\alpha_{ji} = 0$. If the output unit v of the network receives information about the input only through the activations $o(v_1), \ldots o(v_m)$ of the units $v_1, \ldots, v_m$ on its first hidden layer, it cannot solve the computational problem that we consider. But if one takes into account that in addition the *firing correlations* $c(T)$ for subsets $T \subseteq \{v_1, \ldots, v_m\}$ can carry information about the input, one can easily solve this problem.

This can be carried out by the following network $\mathcal{N}$ (see Figure 1). We compute $c(T)$ according to equation (4.1) with $U = \{u_1, \ldots, u_n\}, \alpha_{Tu_i} = min\{\alpha_{ji} : v_j \in T\}$, $\alpha_{TS} = 0$ for all S , and bias $\alpha_T = -1/2$. If one employs for simplicity the Heaviside activation function for σ and σ_c , one has for $T_H := \{v_j : P_j \in H\}$ that $c(T_H) = 1$ if and only if $o(u_i) = 1$ for some $i \in \{1, \ldots, n\}$ so that $\alpha_{ji} = 1$ for all $P_j \in H$ (i.e., if an object O_i is present that has property P_j for *all* $P_j \in H$). Furthermore one has $\prod_{P_j \in H} o(v_j) = 1$ in this case.

We employ a complex gate v as output gate on the third layer of $\mathcal{N}$. Its bias α_v receives the value 1.5 . Its parameter α_{vS} has value 1 if $S = \{v_j : P_j \in H\}$ for some set $H \in \mathcal{C}$, otherwise $\alpha_{vS} = 0$. We set $\alpha_{vu} = 0$ for all preceding neurons u .

According to (2.1) we have $o(v) = 1$ if and only if $c(T_H) \cdot \prod_{P_j \in H} o(v_j) = 1$ for at least two different sets H in $\mathcal{C}$. Hence the network $\mathcal{N}$ outputs 1 if and only if its input $\langle o(u_1), \ldots, o(u_n) \rangle \in \{0,1\}^n$ satisfies $o(u_i) = 1$ and $o(u_{\tilde{i}}) = 1$ for some $i, \tilde{i} \in \{1, \ldots, n\}$ so that O_i satisfies $\bigwedge H$ and $O_{\tilde{i}}$ satisfies $\bigwedge \tilde{H}$, where H and $\tilde{H}$ are two arbitrary but different sets in $\mathcal{C}$. Therefore the network $\mathcal{N}$

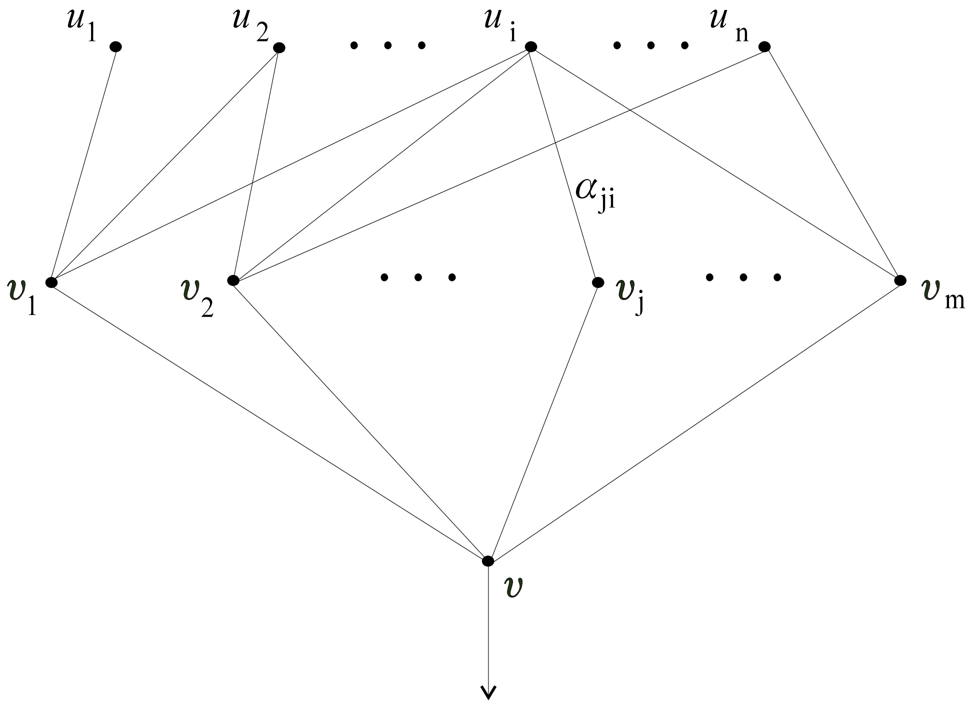

Figure 1: Construction of a network $\mathcal{N}$ of complex gates for the solution of a computational problem which is related to the "binding problem".

solves the computational problem that we have considered.

One can easily see that a very similar network of complex gates can also handle the more realistic situation where the input units $u_1, \ldots, u_n$ represent n "sensors", and the presence of an object O_i is signaled through correlated firing of a characteristic set $S_i \subseteq \{u_1, \ldots, u_n\}$ of these n neurons. Formally, one has then k subsets $S_1, \ldots, S_k \subseteq \{u_1, \ldots, u_n\}$, input variables $o(u_1), \ldots o(u_n)$ and $c(S_i)$ for $S_i \subseteq \{u_1, \ldots, u_n\}$, and one says that "object O_i is present" for some $i \in \{1, \ldots, k\}$ if $c(S_i) \cdot \prod_{u_j \in S_i} o(u_j) = 1$. The architecture of a network $\mathcal{N}$ of complex gates that solves the same computational problem as before for this "distributed" representation of objects O_i can be chosen exactly as in the previous case. In this case the information which object O_i has which basic property P_j is encoded in the parameters α_{vS} for $S \subseteq \{u_1, \ldots, u_n\}$ and $v \in \{v_1, \ldots, v_m\}$, rather than in the parameters α_{ji} . Furthermore the computation of $c(T)$ for sets $T \subseteq \{v_1, \ldots, v_m\}$ of hidden neurons according to equation (4.1) depends now on a proper choice of the parameters α_{TS} for sets $S \subseteq \{u_1, \ldots, u_n\}$. It no longer depends on the parameters $\alpha_{T_{u_i}}$, which can be chosen to be 0 in this more complex case. The construction of the output neuron v remains unchanged. The preceding construction provides an example for a computation that involves two different roles of firing correlation within one computation (binding of sensors u_i , and binding of features P_j), as it might occur for example for visual processing in different areas of the visual cortex.

Finally we would like to point out that the computational goal of the preceding network $\mathcal{N}$ has been chosen to be simple in order to illustrate the basic mechanisms. More complex computations involving different objects can easily be carried out if one replaces the output neuron v of $\mathcal{N}$ by a more sophisticated subsequent network.

The preceding construction indicates in an informal way the new expressive possibilities that arise if a network can internally use firing correlations for its computation, even if the network-input and -output is given in terms of firing rates. In the following rigorous result we also consider neural computations where network-inputs and -outputs are given exclusively in terms of firing rates. We prove in Theorem 4.1 and Corollary 4.2 by straightforward arguments within the framework of computational complexity theory that the computational power of a neural net of some given size becomes substantially larger if it can internally employ auxiliary variables $c(T)$ that model firing correlations. In fact, it turns out that without constraining rules on possible assignments of values to the parameters of a complex gate their computational power is extremely large.

Theorem 4.1 *Most functions $F : \{0,1\}^n \to \{0,1\}$ require exponentially in n many gates for their computation on a feedforward network of standard threshold gates.*

On the other hand all functions $F : \{0,1\}^n \to \{0,1\}$ can be computed by a neural net $\mathcal{N}$ with a fixed architecture that consists of just $2n+2$ complex threshold gates.

Proof: The *first* part of the claim is well-known. It follows immediately from [Muroga et al., 1961], where it was shown that the weights and the bias of any threshold gate with k binary inputs can be chosen to be integers of absolute value $2^{O(k\cdot\log k)}$. The sharpest bound known is $2^{\frac{k+1}{2}\log(k+1)-k}$ ([Schmitt, 1994]). This implies that feedforward networks with arbitrary architecture and arbitrary real weights and biases consisting of n input nodes and m standard threshold gates can compute at most $2^{(nm+\frac{m^2}{2})(\frac{n+m}{2})\cdot\log(n+m)}$ different boolean functions $F : \{0,1\}^n \to \{0,1\}$. The logarithm of this estimate has value $o(2^n)$ for $m := \frac{2^{n/3}}{n}$. Therefore the fraction of functions $F : \{0,1\}^n \to \{0,1\}$ that can be computed with $\leq \frac{2^{n/3}}{n}$ standard threshold gates goes to 0 for $n \to \infty$.

The proof of the *second* part of Theorem 4.1 is provided by the following construction of a network $\mathcal{N}$ that computes F . We construct a 3-layer feedforward network $\mathcal{N}$ which employs complex gates on layers 2 and 3. The computation of $\mathcal{N}$ procedes according to equations (2.1) and (4.1). The architecture of $\mathcal{N}$ and the parameters of the gates on its first two layers are independent of the function F . Only the parameters of its output gate depend on the given function $F : \{0,1\}^n \to \{0,1\}$.

The first layer of $\mathcal{N}$ consists of n input gates $u_1,\ldots,u_n$ that encode the network input $\underline{x} = \langle x_1,\ldots,x_n\rangle \in \{0,1\}^n$ through their firing rates (i.e., $o(u_i) =$

x_i for $i = 1, \ldots, n$). In addition $\mathcal{N}$ contains a gate u_0 that outputs $o(u_0) = 1$, independendly of the network input $\underline{x}$.

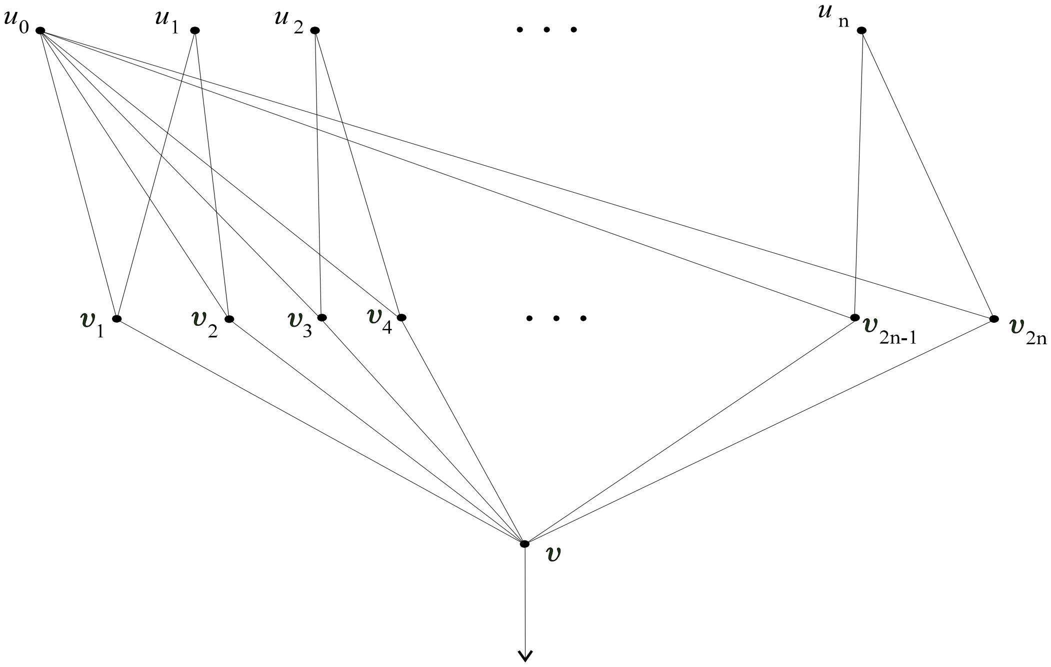

Figure 2: Construction of a small network $\mathcal{N}$ of complex gates for the proof of Theorem 4.1.

The second layer of $\mathcal{N}$ consists of $2n$ complex gates $v_1, \ldots v_{2n}$. The weight on the edge from u_0 to v_j has value 2 for $j = 1, \ldots, 2n$. For $i \in \{1, \ldots, n\}$ the weight on the edge from u_i to v_{2i-1} also has value 2. The bias of v_{2i-1} is set equal to -3 . The weight on the edge from u_i to v_{2i} has value -2, and the bias of v_{2i} is set to -1. The weights α_{ji} on edges from u_i to v_j for $j \notin \{2i-1, 2i\}$ have value 0. Then we have $o(u_i) = o(v_{2i-1}) = 1 - o(v_{2i})$ for every network input $\underline{x} \in \{0,1\}^n$. Hence exactly n gates on layer 2 output a "1" for every network input $\underline{x}$.

There are many ways to achieve that $c(T) = 1$ according to equation (4.1) for any set $T \subseteq \{v_1, \ldots, v_{2n}\}$ of size n . The only role of the auxiliary neuron u_0 in $\mathcal{N}$ is to make this possible with an assignment of parameters in equation (4.1) which is consistent with an interpretation of $c(T)$ as firing correlation, and with the assignment of values to the weights α_{ji} . Since u_0 is the only neuron that has edges with nonzero weights to all $v_j \in T$, it is plausible to set $\alpha_{Tu_0} = 2$ and $\alpha_{Tu_i} = 0$ for $i \in \{1, \ldots, n\}$. With $\alpha_{TS} = 0$ and $\alpha_T = -1$ one then has $c(T) = 1$ according to equation (4.1).

Hence for any network input $\underline{x} \in \{0,1\}^n$ one has $c(T) \cdot \prod\limits_{v_j \in T} o(v_j) = 1$ for exactly one set T of size n (with $v_{2i-1} \in T$ if $x_i = 1$ and $v_{2i} \in T$ if $x_i = 0$). We will denote this set by $T_{\underline{x}}$ in the following. Note that $c(T) \cdot \prod\limits_{v_j \in T} o(v_j) = 0$ for all sets T of size $\geq n$ with $T \neq T_{\underline{x}}$. The values of $c(T)$ for sets T of size less than n will be irrelevant for the following (since $\alpha_{vT} = 0$ for these sets T).

The computation of the complex gate v on layer 3 of $\mathcal{N}$ procedes according to equation (2.1). Its bias is set to -1/2, and its weights on edges from preceding gates v_i are all set to 0 ($i = 1, \ldots, 2n$). Its weight α_{vS} for a set $S \subseteq \{v_1, \ldots, v_{2n}\}$ is set to 1 if $S \supseteq T_{\underline{x}}$ for some $\underline{x} \in \{0,1\}^n$ with $F(\underline{x}) = 1$, otherwise $\alpha_{vS} = 0$.

The output gate v of this network $\mathcal{N}$ outputs $o(v) = F(\underline{x})$ for any network input $\underline{x} \in \{0,1\}^n$. Hence the network $\mathcal{N}$ consisting of $2n + 2$ gates (in addition to the input gates $u_1, \ldots, u_n$) computes the arbitrarily given function $F : \{0,1\}^n \to \{0,1\}$.

On the side we would like to point out that the parameters of $\mathcal{N}$ were chosen in such a way that for all network inputs the input of the Heaviside activation function of any gate in $\mathcal{N}$ is bounded away from zero. Therefore one can apply standard transformations to carry out the same computations also with smooth activation functions, such as the logistic sigmoid, instead of the Heaviside function. ∎

Corollary 4.2 *If a function $F : \{0,1\}^n \to \{0,1\}$ can be defined by a boolean formula in disjunctive normal form with k terms, then F can be computed by a neural net $\mathcal{N}$ consisting of at most $2n+2$ complex gates which have nonzero coefficients for just k product terms.*

Proof: For the output gate v of the neural net $\mathcal{N}$ one sets $\alpha_{vS} = 1$ if S corresponds to a monomial in the disjunctive normal form of F , otherwise one sets $\alpha_{vS} = 0$. The rest of the construction of Theorem (4.1) remains unchanged. ∎

Remark 4.3 Some recent experimental results (for example [deCharms, Merzenich, 1996], [Vaadia et al., 1995]) show that some biological neural systems change their firing correlations in response to a stimulus, whereas the firing rates of the neurons involved do not change. This computational mode corresponds to the special case of our model where all firing rates in (2.1) and (4.1) (except for the firing rates of input- and -output-neurons) are constants. It turns out that this special case of our model, which no longer contains any high-order terms if $c(S) = 0$ for any set S of input neurons, is computationally still very powerful.
For example, one can compute in this way the NP-complete decision problem CLIQUE on a linear-size feedforward circuit of complex threshold gates with one hidden layer. This is possible because the computation of this circuit can employ super-polynomically many correlation variables. We assume here that the absence or presence of each edge in the input-graph G with n nodes is indicated by a separate binary variable, and that the output of CLIQUE is 1 if and only if G has a clique of size $\geq n/2$.

We assume that $c(S) = 0$ for any set S of input neurons, and that the input is given through the firing rates of the input neurons. We arrange that the firing rates $o(u)$ of all *hidden* neurons u are constants. Therefore all product terms $\prod_{u \in S} o(u)$ for sets S of hidden neurons that occur in the computation of this circuit can be replaced by constants. Hence all terms $c(S) \cdot \prod_{u \in S} o(u)$ for *arbitrary* sets S of neurons on any layer of this network can be replaced by *linear* terms. In the hidden layer of the network we employ one complex gate v for each node of the input-graph G. For any set T of $n/2$ of these hidden units v one arranges that $c(T) = 1$ if and only if T is a clique in G (set $\alpha_{Tu} = 1$ for an input node u if u represents an edge whose incident vertices both belong to T , otherwise set $\alpha_{Tu} = 0$; the bias α_T is set to $-\binom{n/2}{2}$).

5 Conclusions

We have introduced a simple framework for modeling neural computations with firing rates and firing correlations. In contrast to networks of spiking neurons, these new models need not keep track of individual firing times. Therefore they provide a suitable framework for a theoretical investigation of *large-scale* features of neural computations with firing rates and firing correlations. In contrast to previous models for modeling computational effects of firing correlations, our new models are *general purpose computational models*, whose function is not restricted to special types of computations.

We have presented several constructions of computationally powerful networks in this framework. Our constructions show that besides possible uses in solutions of the "binding problem", firing correlations may also play other roles in the context of complex computations. Furthermore we present for the first time a *proof* that a neural network model that reflects salient aspects of neural computations with firing rates and firing correlations has strictly more computational power than a network of the same size that just models neural computations in terms of firing rates.

On the side one of our separation results also has implications for standard neural network models. Corollary 3.2 provides the first *proof* that a high-order sigmoidal neural net can have strictly more computational power than any first-order sigmoidal neural net of a substantially larger size, without imposing constraints on the architecture, parameters or (reasonable) choice of activation functions employed by the first-order sigmoidal neural net.

We hope that the notion of a complex gate that is introduced in this paper provides a useful reference model for evaluating the computational capabilities of biological neurons. On the other hand our model is sufficiently abstract to also provide a reference model for evaluating the computational capabilities of related computational units in pulsed VLSI. In both cases our model gives rise to interesting new questions regarding the possible values of parameters of a complex gate which can be realized in that setting. In fact, the model introduced in this paper may turn out to be a useful "transmission tool" for moving computational mechanisms and principles from biological neurons to silicon-implementations. In addition it appears to be useful for quantifying specific advantages of silicon-implementations of computations with correlated pulse trains - for example new parameter ranges that can be achieved for the parameters of a complex gate in silicon but not in wetware.

Finally, we would like to point out that the new computational model that has been presented in this paper also has consequences for the investigation of *learning* in neural systems. The function that is computed by a network of complex gates depends on new parameters, and the question arises how these parameters can be tuned by suitable learning rules. It should be noted that according to Theorem 4.1 a fixed network architecture of linear size with just *one* layer of "adjustable" complex gates provides already a universal computational model (although only for binary inputs and outputs). This points to the possibility that instead of biologically dubious learning rules for multilayer networks it may suffice to explore learning rules for tuning a larger set of parameters for a single layer of complex gates.

Acknowledgment: I would like to thank Thomas Natschläger for helpful comments.

References

[Abeles, 1991] M. Abeles. *Corticonics.* Cambridge University Press, Cambridge, 1991.

[Baum, Haussler, 1989] E. B. Baum, D. Haussler. What size net yields valid generalization? *Neural Computation*, vol. 1, 151–160, 1989.

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

[Bernander et al., 1994] Ö. Bernander, C. Koch, M. Usher. The effect of synchronized inputs at the single neuron level. *Neural Computation*, vol. 6, 622–641, 1994.

[Bruck, Smolenski, 1992] J. Bruck, R. Smolenski. Polynomial threshold functions, AC^0 functions, and spectral norms. *SIAM J. Computing*, vol. 21, 33–42, 1992.

[Cover, 1968] T. M. Cover. Capacity problems for linear machines. In: *Pattern Recognition*, L. Kanal, ed., Thompson Book Co., 283–289, 1968.

[deCharms, Merzenich, 1996] R. C. deCharms, M. M. Merzenich. Primary cortical representation of sounds by the coordination of action-potential timing. *Nature*, 381:610-613, 1996.

[Durbin, Rumelhart, 1989] R. E. Durbin, D. E. Rumelhart. Product units: a computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, vol. 1, 133–142, 1989.

[Eckhorn et al., 1990] R. Eckhorn, H. J. Reitboeck, M. Arndt, P. Dicke. Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex. *Neural Computation*, vol. 2, 293–307, 1990.

[Eggermont, 1990] J. J. Eggermont. *The Correlative Brain: Theory and Experiment in Neural Interaction.* Springer (Berlin), 1990.

[Engel et al., 1992] A. K. Engel, P. König, A. K. Kreiter, T. B. Schillen, W. Singer. Temporal encoding in the visual cortex: new vistas on integration in the nervous system. *Trends in Neurosciences*, 15(6):218–226, 1992.

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

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

[Hajnal et al., 1993] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, G. Turan. Threshold circuits of bounded depth. *J. Comput. System Sci. 46*, 129–154, 1993.

[Hatsopoulos et al.,1996] N. G. Hatsopoulos, J. N. Sanes, J. P. Donoghue. Dynamics correlations in unit firing of motor cortical neurons related to movement preparation and action. *Proc. of the 5th Annual Conference on Computational Neuroscience*, 1996, Academic Press (San Diego), 1997, to appear.

[Karpinski, Macintyre, 1996] M. Karpinski, A. Macintyre. Polynomial bounds for VC-dimension of sigmoidal and general Pfaffian neural networks. To appear in the *J. of Comp. and System Sciences.*

[Koch, Crick, 1994] C. Koch, F. Crick. Some further ideas regarding the neuronal basis of awareness. In: *Large-Scale Neuronal Theories of the Brain.* C. Koch and J. L. Davis, eds., MIT-Press, Cambridge, 93–110, 1994.

[Koch, Poggio, 1992] C. Koch, T. Poggio. Multiplying with synapses and neurons. In: *Single Neuron Computation*, Academic Press, 315–345, 1992.

[Koiran, 1996] P. Koiran. VC-dimension in circuit complexity. *Proc. of the 11th IEEE Conference on Computational Complexity*, 81–85, 1996.

[Kreiter, Singer, 1996] A. K. Kreiter, W. Singer. Stimulus-dependent synchronization of neuronal responses in the visual cortex of the awake macaque monkey. *The Journal of Neuroscience*, vol. 16(7), 2381–2396, 1996.

[Krüger, 1991] Krüger, ed. *Neuronal Cooperativity.* Springer, Berlin, 1991.

[Lumer, Huberman, 1992] E. D. Lumer, B. A. Huberman. Binding hierarchies: a basis for dynamic perceptual grouping. *Neural Computation*, vol. 4, 341–355, 1992.

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

[Maass, 1996b] W. Maass. Networks of spiking neurons: the third generation of neural network models. Appears in *Neural Networks*. Extended abstract (with a different title) appears in: *Advances in Neural Information Processing Systems*, vol. 9, MIT Press, Cambridge, 1997.

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

[Maass, 1997b] W. Maass. Bounds for the computational power and learning complexity of analog neural nets. *SIAM J. on Computing*, 1997, to appear.

[Maass et al., 1994] W. Maass, G. Schnitger, E. D. Sontag. A comparison of the computational power of sigmoid and boolean threshold circuits. In: *Theoretical Advances in Neural Computation and Learning*, V. Roychowdhury, K.-Y. Siu, A. Orlitsky, eds., Kluwer Academic Publishers, Boston, 127–151, 1994.

[MacLeod, Laurent, 1996] K. MacLeod, G. Laurent. Distinct mechanisms for synchronization and temporal patterning of odor-encoding neural assemblies. *Science*, 274:976-979, 1996.

[Mel, 1994] B. W. Mel. Information processing in dendritic trees. *Neural Computation*, vol. 6, 1031–1085, 1994.

[Mel, 1997] B. W. Mel. SEEMORE: Combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, to appear.

[Milner, 1974] P. M. Milner. A model for visual shape recognition. *Psychological Review*, vol. 81(6), 521–535, 1974.

[Muroga et al., 1961] S. Muroga, I. Toda, S. Takasu. theory of majority decision elements. *J. Franklin Institute*, vol. 271, 376–418, 1961.

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

[Omlin, Giles, 1996] C. W. Omlin, C. L. Giles. Constructing deterministic finite-state automata in recurrent neural networks. *J. Assoc. Comput. Mach.*, vol. 43, 937–972, 1996.

[Phillips, Singer, 1997] W.A. Phillips, W. Singer. In search of common foundations for cortical computation. *Behavioral and Brain Sciences*, 1997, in press.

[Ripley, 1996] B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press (Cambridge), 1996.

[Rumelhart et al., 1986] D. E. Rumelhart, G. E. Hinton, J. L. McClelland. A general framework for parallel distributed processing. In: *Parallel Distributed Processing*, vol. 1, 45–76, MIT-Press, Cambridge, 1986.

[Schmitt, 1994] M. Schmitt. On the size of weights for McCulloch-Pitts neurons. *Proc. of the 6th Italian Workshop on Neural Networks*, E. Caianello, ed., World Scientific, Singapore, 241–246, 1994.

[Shastri, Ajjanagadde, 1993] L. Shastri, V. Ajjanagadde. From simple associations to systematic reasoning: a connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioural and Brain Sciences*, vol. 16, 417–494, 1993.

[Sillito et al., 1994] A. M. Sillito, H. E. Jones, G. C. Gerstin, D. C. West. Feature-linked synchronization of thalamic relay cell firing induced by feedback. *Nature*, vol. 369, 479–482, 1994.

[Singer, 1995] 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.

[Siu et al., 1994] K.-Y. Siu, V. Roychowdhury, T. Kailath. *Discrete Neural Computation: A Theoretical Foundation*, Prentice Hall, 1995.

[Traub et al., 1996] R. D. Traub, M. A. Whittington, I. M. Stanford, J. G. R. Jefferys. A mechanism for generation of lang-range synchronous fast oscillations in the cortex. *Nature*, vol. 383, 621–624, 1996.

[Tuckwell, 1988] H. C. Tuckwell. *Introduction to Theoretical Neurobiology*, volume 1 and 2. Cambridge Univ. Press, Cambridge, 1988.

[Vaadia et al., 1995] E. Vaadia, A. Aertsen, I. Nelken. Dynamics of neuronal interactions cannot be explained by neuronal transients. *Proc. Royal Soc. of London B*, vol. 261, 407–410, 1995.

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

[van Schaik et al., 1997] A. van Schaik, E. Fragnière, E. Vittoz. A silicon model of amplitude modulation detection in the auditory brainstem. *Advances in Neural Information Processing Systems*, vol. 9, MIT-Press (Cambridge), 1997.

[von der Malsburg, 1981] C. von der Malsburg. The correlation theory of brain function. Internal Report 81-2 of the Dept. of Neurobiology of the Max Planck Institute for Biophysical Chemistry in Göttingen, Germany, 1981.

[Weiss, Kulikowski, 1991] S. M. Weiss, C. A. Kulikowski. *Computer Systems that Learn.* Morgan Kaufmann (San Mateo), 1991.

[Winder, 1963] R. O. Winder. Bounds on threshold gate realizability. *IEEE Trans. on Electronic Computers*, vol. 12, 561–564, 1963.

[Zaghloul et al., 1994] M. L. Zaghloul, J. L. Meador, R. W. Newcomb, eds., *Silicon Implementations of Pulse Coded Neural Networks*, Kluwer Academic Publishers, 1994.