On the Computational Power of Winner-Take-All

Wolfgang Maass*
Institute for Theoretical Computer Science
Technische Universität Graz
Klosterwiesgasse 32/2
A-8010 Graz, Austria
email: maass@igi.tu-graz.ac.at
http://www.cis.tu-graz.ac.at/igi/maass

Abstract

This article initiates a rigorous theoretical analysis of the computational power of circuits that employ modules for computing winner-take-all. Computational models that involve competitive stages have so far been neglected in computational complexity theory, although they are widely used in computational brain models, artificial neural networks, and analog VLSI. Our theoretical analysis shows that winner-take-all is a surprisingly powerful computational module in comparison with threshold gates (= McCulloch-Pitts neurons) and sigmoidal gates. We prove an optimal quadratic lower bound for computing winner-take-all in any feedforward circuit consisting of threshold gates. In addition we show that arbitrary continuous functions can be approximated by circuits employing a single soft winner-take-all gate as their only nonlinear operation.

Our theoretical analysis also provides answers to two basic questions that have been raised by neurophysiologists in view of the well-known asymmetry between excitatory and inhibitory connections in cortical circuits: how much computational power of neural networks is lost if only positive weights are employed in weighted sums, and how much adaptive capability is lost if only the positive weights are subject to plasticity.

*Research for this article was partially supported by the ESPRIT Working Group NeuroCOLT, No. 8556, and the Fonds zur Förderung der wissenschaftlichen Forschung (FWF), Austria, project P12153.

1 Introduction

Computational models that involve competitive stages are widely used in computational brain models, artificial neural networks, and analog VLSI (see (Arbib, 1995)). The simplest competitive computational module is a hard winner-take-all gate that computes a function $\mathrm{WTA}_n : \mathbb{R}^n \to \{0,1\}^n$ whose output $\langle b_1, \ldots, b_n \rangle = \mathrm{WTA}_n(x_1, \ldots, x_n)$ satisfies

$$b_i = \begin{cases} 1 & \text{, if} \quad x_i > x_j \quad \text{for all} \quad j \neq i \\ 0 & \text{, if} \quad x_j > x_i \quad \text{for some} \quad j \neq i \end{cases} .$$

Thus in the case of pairwise different inputs $x_1, \ldots, x_n$ a single output bit b_i has value 1, which marks the position of the largest input x_i.[1]

In this article we also investigate the computational power of two common variations of winner-take-all: k-winner-take-all, where the ith output b_i has value 1 if and only if x_i is among the k largest inputs, and soft winner-take-all, where the ith output is an analog variable r_i whose value reflects the rank of x_i among the input variables.

Winner-take-all is ubiquitous as a computational module in computational brain models, especially in models involving computational mechanisms for attention (Niebur and Koch, 1998). Biologically plausible models for computing winner-take-all in biological neural systems exist both on the basis of the assumption that the analog inputs x_i are encoded through firing rates – where the most frequently firing neuron exerts the strongest inhibition on its competitors and thereby stops them from firing after a while –, and on the basis of the assumption that the analog inputs x_i are encoded through the temporal delays of single spikes – where the earliest firing neuron (that encodes the largest x_i) inhibits its competitors before they can fire (Thorpe, 1990).

We would like to point to another link between results of this article and computational neuroscience. There exists a notable difference between the computational role of weights of different signs in artificial neural network models on one hand, and anatomical and physiological data regarding the interplay of excitatory and inhibitory neural inputs in biological neural systems on the other hand (see (Abeles, 1991) and (Shepherd, 1998)). Virtually any artificial neural network model is based on an assumed symmetry between positive and negative weights. Typically weights of either sign occur on an equal footing as coefficients in weighted sums that represent the input to an artificial neuron. In contrast to that, there exists a strong asymmetry regarding positive (excitatory) and negative (inhibitory) inputs to biological neurons. At

[1] Different conventions are considered in the literature in case that there is no unique "winner" x_i, but we need not specify any convention in this article since our lower bound result for WTA_n holds for all of these versions.

most 15% of neurons in the cortex are inhibitory neurons, and these are the only neurons that can exert a negative (inhibitory) influence on the activity of other neurons. Furthermore, the location and structure of their synaptic connections to other neurons differ drastically from those formed by excitatory neurons. Inhibitory neurons are usually just connected to neurons in their immediate vicinity, and it is not clear to what extent their synapses are changed through learning. Furthermore the location of their synapses on the target neurons (rarely on spines, often close to the soma, frequently with multiple synapses on the target neuron) suggest that their computational function is not symmetric to that of excitatory synapses. Rather, such data would support a conjecture that their impact on other neurons may be more of a local regulatory nature, that inhibitory neurons do not function as computational units per se like the (excitatory) pyramidal neurons, whose synapses are subject to fine-tuning via various learning mechanisms. These observations from anatomy and neurophysiology have given rise to the question, whether a quite different style of neural circuit design may be feasible, that achieves sufficient computational power without requiring symmetry between excitatory and inhibitory interaction among neurons. The circuits constructed in section 3 and 4 of this article provide a positive answer to this question. It is shown there that neural circuits that use inhibition *exclusively* for lateral inhibition in the context of winner-take-all have the same computational power as multi-layer perceptrons that employ weighted sums with positive and negative weights in the usual manner. Furthermore one can, if one wants to, keep the inhibitory synapses in these circuits fixed, and just modify the excitatory synapses in order to program the circuits so that they adopt a desired input/output behavior. It should be noted that it has long been known that winner-take-all can be implemented via inhibitory neurons in biologically realistic circuit models ((Elias and Grossberg, 1975), (Amari and Arbib, 1977), (Coultrip et al., 1992), (Yuille and Grzywacz, 1989)). The only novel contribution of this article is the result that in combination with neurons that compute weighted sums (with positive weights only) such winner-take-all modules have universal computational power, both for digital and for analog computation.

A large number of efficient implementations of winner-take-all in analog VLSI have been proposed, starting with (Lazzaro et al., 1989). The circuit of (Lazzaro et al., 1989) computes an approximate version of WTA_n with just $2n$ transistors and wires of total length $O(n)$, with lateral inhibition implemented by adding currents on a single wire of length $O(n)$. Its computation time scales with the size of its largest input. Numerous other efficient implementations of winner-take-all in analog VLSI have subsequently been produced, see for example (Andreou et al., 1991), (Choi and Sheu, 1993), (Fang et al., 1996). Among them are circuits based on silicon spiking neurons ((DeYong et al., 1992), (Meador and Hylander, 1994), (Indiveri, 1999)) and circuits that emulate attention in artificial sensory processing ((DeWeerth and Morris, 1994), (Horiuchi et al., 1997), (Brajovic and Kanade, 1998), (Indiveri, 1999)). In spite of these numerous hardware implementations of winner-take-all and numerous practical applications, we are not aware of any theoretical results on the general computational power of these modules. It is one goal of this article to place these novel circuits into the context of other circuit models that are commonly studied in computational complexity theory, and to evaluate their relative strength.

There exists an important structural difference between those circuits that are commonly studied in computational complexity theory, and those circuits that one typically encounters in hardware or wetware. Almost all circuit models in computational complexity theory are *feedforward* circuits, i.e. their architecture constitutes a directed graph without cycles ((Wegener, 1987), (Savage, 1998)). In contrast to that, typical physical realizations of circuits contain besides feedforward connections also *lateral* connections (i.e. connections among gates on the same layer), and frequently also *recurrent* connections (i.e. connections from a higher layer backwards to some lower layer of the circuit). This gives rise to the question whether this discrepancy is relevant, for example whether there are practically important computational tasks that can be implemented substantially more efficiently on a circuit with lateral connections than on a strictly feedforward circuit. In this article we address this issue in the following manner: we view modules that compute winner-take-all (whose implementation usually involves lateral connections) as "black boxes" of which we only model their input/output behavior. We refer to these modules as *winner-take-all gates* in the following. We study possible computational uses of such winner-take-all gates in circuits where these gates (and other gates) are wired together in a feedforward fashion.[2] We charge one unit of time for the operation of each gate. We will show that in this framework the new modules, which may internally involve lateral connections, do in fact add substantial computational power to a feedforward circuit.

The arguably most powerful gates that have previously been studied in computational complexity theory are *threshold gates* (also referred to as McCulloch-Pitts neurons or perceptrons, see (Minsky and Papert, 1969), (Siu et al., 1995)) and sigmoidal gates (which may be viewed as soft versions of threshold gates). A threshold gate with weights $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and threshold $\Theta \in \mathbb{R}$ computes the function $G : \mathbb{R}^n \to \{0, 1\}$ defined by $G(x_1, \ldots, x_n) = 1 \Leftrightarrow \sum_{i=1}^{n} \alpha_i x_i \geq \Theta$. Note that AND and OR of n bits as well as NOT are special cases of threshold gates. A *threshold circuit* (also referred to as multi-layer perceptron) is a feedforward circuit consisting of threshold

[2]We only examine computations in such a circuit for a single batch-input, not for streams of varying inputs. Hence it does not matter for this analysis whether one implements a winner-take-all gate by a circuit that needs to be re-initialized before the next input, or by a circuit that immediately responds to changes in its input.

gates. The *depth* of a threshold circuit C is the maximal length of a directed path from an input node to an output gate. The circuit is called *layered* if all such paths have the same length. Note that a circuit with k hidden layers has depth $k+1$ in this terminology. Except for Theorem 2.1 we will discuss in this article only circuits with a single output. The other results can be extended to networks with several outputs by duplicating the network so that each output variable is formally computed by a separate network.

We will prove in section 2 that WTA_n is a rather expensive computational operation from the point of view of threshold circuits, since any such circuit needs quadratically in n many gates to compute WTA_n. We will show in section 3 that the full computational power of two layers of threshold gates can be achieved by a single k-winner-take-all gate applied to positive weighted sums of the input variables. Furthermore we will show in section 4 that by replacing the single k-winner-take-all gate by a single soft winner-take-all gate, these extremely simple circuits become universal approximators for arbitrary continuous functions.

2 An Optimal Quadratic Lower Bound for Hard Winner-Take-All

We show in this section that *any* feedforward circuit consisting of threshold gates needs to consist of quadratically in n many gates for computing WTA_n. This result also implies a lower bound for any circuit of threshold gates involving lateral and/or recurrent connections that computes WTA_n (provided one assumes that each gate receives at time t only outputs from other gates that were computed before time t): One can simulate *any* circuit with s gates whose computation takes t discrete time steps by a *feedforward* circuit with $s \cdot t$ gates whose computation takes t discrete time steps. Hence for example if there exists some circuit consisting of $O(n)$ threshold gates with lateral and recurrent connections that computes WTA_n in t discrete time steps, then WTA_n can be computed by a feedforward circuit consisting of $O(t \cdot n)$ threshold gates. Therefore the subsequent Theorem 2.1 implies that WTA_n cannot be computed in sublinear time by any linear size circuit consisting of threshold gates with *arbitrary* (i.e. feedforward, lateral, and recurrent) connections. In case that linear size implementations of (approximations of) WTA_n in analog VLSI can be built whose computation time grows sublinearly in n, then this negative result would provide theoretical evidence for the superior computational capabilities of analog circuits with lateral connections.

For $n = 2$ it is obvious that WTA_n can be computed by a threshold circuit of size 2, and that this size is optimal. For $n \geq 3$ the most straightforward design of a (feedforward) threshold circuit C that computes WTA_n uses $\binom{n}{2}+n$ threshold gates: For each pair $\langle i,j\rangle$ with $1 \leq i < j \leq n$ one employs a threshold gate G_{ij} that outputs 1 if and only if $x_j \geq x_i$. The ith output b_i of WTA_n for a circuit input $\underline{x}$ is computed by a threshold gate G_i with $G_i = 1 \Leftrightarrow \sum\limits_{j<i} G_{ji}(\underline{x}) + \sum\limits_{j>i} -G_{ij}(\underline{x}) \geq i-1$.

This circuit design appears to be sub-optimal, since most of its threshold gates – the $\binom{n}{2}$ gates G_{ij} – do not make use of their capability to evaluate weighted sums of *many* variables, with *arbitrary weights* from $\mathbb{R}$. However the following result shows that no feedforward threshold circuit (not even with an arbitrary number of layers, and threshold gates of arbitrary fan-in with arbitrary real weights) can compute WTA_n with fewer than $\binom{n}{2}+n$ gates.

Theorem 2.1. *Assume that $n \geq 3$ and WTA_n is computed by some arbitrary feedforward circuit C consisting of threshold gates with arbitrary weights. Then C consists of at least $\binom{n}{2}+n$ threshold gates.*

Proof of Theorem 2.1. Let C be any threshold circuit that computes WTA_n for all $\underline{x} = \langle x_1,\ldots,x_n\rangle \in \mathbb{R}^n$ with pairwise different $x_1,\ldots,x_n$. We say that a threshold gate in the circuit C *contains* an input variable x_k if there exists a direct "wire" (i.e., an edge in the directed graph underlying C) from the kth input node to this gate, and the kth input variable x_k occurs in the weighted sum of this threshold gate with a weight $\neq 0$. Note that this threshold gate may also receive outputs from other threshold gates as part of its input, since we do not assume that C is a layered circuit.

Fix any $i,j \in \{1,\ldots,n\}$ with $i \neq j$. We will show that there exists a gate G_{ij} in C that contains the input variables x_i and x_j, but no other input variables. The proof proceeds in 4 steps.

Step 1: Choose $h \in (0.6, 0.9)$ and $\rho \in (0, 0.1)$ such that no gate G in C that contains the input variable x_j, but no other input variable, changes its output value when x_j varies over $(h-\rho, h+\rho)$, no matter which fixed binary values $\underline{a}$ have been assigned to the other inputs of gate G.

For any fixed $\underline{a}$ there exists at most a single value $t_{\underline{a}}$ for x_j so that G changes its output for $x_j = t_{\underline{a}}$ when x_j varies from $-\infty$ to $+\infty$. It suffices to choose h and ρ so that none of these finitely many values of $t_{\underline{a}}$ (for arbitrary binary $\underline{a}$ and arbitrary gates G) falls into the interval $(h-\rho, h+\rho)$.

Step 2: Choose a closed ball $B \subseteq (0, 0.5)^{n-2}$ with a center $\underline{c} \in \mathbb{R}^{n-2}$ and a radius $\delta > 0$ so that no gate G in C changes its output when we set $x_i = x_j = h$ and let the vector of the other $n-2$ input variables vary over B (this is required to hold for any fixed binary values of the inputs that G may receive from other threshold gates).

We exploit here that for fixed $x_i = x_j = h$ the gates in C (with inputs from other gates replaced by all possible binary values) partition $(0, 0.5)^{n-2}$ into finitely many sets S, each of which can be written as intersection of half-spaces, so that no gate in C changes its output when we fix $x_i = x_j = h$ and let the other $n-2$ input variables of the circuit vary over S (while keeping inputs to C from other gates artificially fixed).

Step 3: Choose $\gamma \in (0, \rho)$ so that in every gate G in C that contains besides x_j some other input variable x_k with $k \notin \{i, j\}$ a change of x_j by an amount γ causes a smaller change of the weighted sum at this threshold gate G than a change by an amount δ of any of the input variables x_k with $k \notin \{i, j\}$ that it contains.

This property can be satisfied because we can choose for γ an arbitrarily small positive number.

Step 4: Set $x_i := h$, $\langle x_k \rangle_{k \notin \{i,j\}} := \underline{c}$, and let x_j vary over $[h - \frac{\gamma}{2}, h + \frac{\gamma}{2}]$. By assumption, the output of C changes since the output of $\mathrm{WTA}_n(x_1, \ldots, x_n)$ changes (x_i is the winner for $x_j < h$, x_j is the winner for $x_j > h$). Let G_{ij} be some gate in C that changes its output, whereas no gate that lies on a path from an input variable to G_{ij} changes its output. Hence G_{ij} contains the input variable x_j. The choice of h and $\rho > \gamma$ in step 1 implies that G_{ij} contains besides x_j some other input variable.

Assume for a contradiction that G_{ij} contains an input variable x_k with $k \notin \{i, j\}$. By the choice of γ in step 3 this implies that the output of G_{ij} changes when we set $x_i = x_j = h$ and move x_k by an amount up to δ from its value in $\underline{c}$, while keeping all other input variables and inputs that G_{ij} receives from other threshold gates fixed (even if some preceding threshold gates in C would change their output in response to this change in the input variable x_k). This yields a contradiction to the definition of the ball B in step 2.

Thus we have shown that $k \in \{i, j\}$ for any input variable x_k that G_{ij} contains. Therefore G_{ij} contains exactly the input variables x_i and x_j.

So far we have shown that for any $i, j \in \{1, \ldots, n\}$ with $i \neq j$ there exists a gate G_{ij} in C that contains the two input variables x_i, x_j, and no other input variables.

It remains to be shown that apart from these $\binom{n}{2}$ gates G_{ij} the circuit C contains n other gates $G_1, \ldots, G_n$ that compute the n output bits $b_1, \ldots, b_n$ of WTA_n. It is impossible that $G_k = G_l$ for some $k \neq l$, since $b_k \neq b_l$ for some arguments of WTA_n. Assume for a contradiction that $G_k = G_{ij}$ for some

$k, i, j \in \{1, \ldots, n\}$ with $i \neq j$. We have $k \neq i$ or $k \neq j$. Assume without loss of generality that $k \neq i$. Let l be any number in $\{1, \ldots, n\} - \{k, i\}$. Assign to $x_1, \ldots, x_n$ some pairwise different values $a_1, \ldots, a_n$ so that $a_l > a_{l'}$ for all $l' \neq l$. Since $l \neq k$ and $i \neq k$ we have that for any $x_i \in \mathbb{R}$ the kth output variable b_k of $\mathrm{WTA}_n(a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots a_n)$ has value 0 (a_k cannot be the maximal argument for any value of x_i since $a_l > a_k$). On the other hand, since G_{ij} contains the variable x_i, there exist values for x_i (move $x_i \to \infty$ or $x_i \to -\infty$) so that G_{ij} outputs 1, no matter which binary values are assigned to the inputs that G_{ij} receives in C from other threshold gates (while we keep x_j fixed at value a_j). This implies that G_{ij} does not output b_k in circuit C, i.e., $G_k \neq G_{ij}$. Therefore the circuit C contains in addition to the gates G_{ij} n other gates that provide the circuit outputs $b_1, \ldots, b_n$.

∎

3 Simulating Two Layers of Threshold Gates with a Single k-Winner-Take-All Gate as the only Nonlinearity

A popular variation of winner-take-all (which has also been implemented in analog VLSI (Urahama and Nagao, 1995)) is k-winner-take-all. The output of k-winner-take-all indicates for each input variable x_i whether x_i is among the k largest inputs. Formally we define for any $k \in \{1, \ldots, n\}$ the function $k\text{-WTA}_n : \mathbb{R}^n \to \{0,1\}^n$ where $k\text{-WTA}_n\ (x_1, \ldots, x_n) = \langle b_1, \ldots, b_n \rangle$ has the property that

$$b_i = 1 \Leftrightarrow (x_j > x_i \text{ holds for at most } k-1 \text{ indices } j).$$

We will show in this section that a single gate that computes k-WTA_n can absorb all nonlinear computational operations of any two-layer threshold circuit with one output gate and any number of hidden gates.

Theorem 3.1. *Any two-layer feedforward circuit C (with m analog or binary input variables and one binary output variable) consisting of threshold gates can be simulated by a circuit consisting of a single k-winner-take-all gate k-WTA_n applied to n weighted sums of the input variables with positive weights, except for some set $S \subseteq \mathbb{R}^m$ of inputs that has measure 0.*

In particular, any boolean function $f : \{0,1\}^m \to \{0,1\}$ can be computed by a single k-winner-take-all gate applied to weighted sums of the input bits.

If C has polynomial size and integer weights, whose size is bounded by a polynomial in m, then n can be bounded by a polynomial in m and all weights *in the simulating circuit are natural numbers whose size is bounded by a polynomial in m.*

Remark 3.2. The exception set of measure 0 in this result is a union of up to n hyperplanes in $\mathbb{R}^m$. This exception set is apparently of no practical significance, since for *any* given finite set $\tilde{D}$, not just for $\tilde{D} \subseteq \{0,1\}^m$ but for example for $\tilde{D} := \{\langle z_1, \ldots, z_m \rangle \in \mathbb{R}^m$: each z_i is a rational number with bit-length $\leq 1000\}$, one can move these hyperplanes (by adjusting the constant terms in their definitions) so that no point in $\tilde{D}$ belongs to the exception set. Hence the k-WTA circuit can simulate the given threshold circuit C on *any* finite set $\tilde{D}$, with an architecture that is independent of $\tilde{D}$.

On the other hand it should be noted that the proof of the lower bound result from Theorem 2.1 requires that the circuit C computes WTA everywhere, not just on a finite set.

Remark 3.3. One can easily show that the exception set S of measure 0 in Theorem 3.1 is *necessary*: The set of inputs $\underline{z} \in \mathbb{R}^m$ for which a k-WTA$_n$ gate applied to weighted sums outputs 1 is *always* a closed set, whereas the set of inputs for which a circuit C of depth 2 consisting of threshold gates outputs 1 can be an open set. Hence in general a circuit C of the latter type *cannot* be simulated for *all* inputs $\underline{z} \in \mathbb{R}^m$ by a k-WTA$_n$ gate applied to weighted sums.

Corollary 3.4. *Any layered threshold circuit C of arbitrary even depth d can be simulated for all inputs except for a set of measure 0 by a feedforward circuit consisting of ℓ k-WTA gates on $\frac{d}{2}$ layers (each applied to positive weighted sums of circuit inputs on the first layer, and of outputs from preceding k-WTA gates on the subsequent layers), where ℓ is the number of gates on even-numbered layers in C. Thus one can view the simulating circuit as a d-layer circuit with alternating layers of linear and k-WTA gates.*

Alternatively one can simulate the layers 3 to d of circuit C by a single k-WTA gate applied to a (possibly very large) number of linear gates. This yields a simulation of C (except for some input set S of measure 0) by a 4-layer circuit with alternating layers of linear gates and k-WTA gates. The number of k-WTA gates in this circuit can be bounded by $\ell_2 + 1$, where ℓ_2 is the number of threshold gates on layer 2 of C.

Proof of Theorem 3.1: Since the outputs of the gates on the hidden layer of C are from $\{0,1\}$, we can assume without loss of generality that the weights $\alpha_1, \ldots, \alpha_n$ of the output gate G of C are from $\{-1,1\}$ (see for example (Siu et al., 1995) for details; one first observes that it suffices to use integer weights for threshold gates with binary inputs, one can then normalize these weights to values in $\{-1,1\}$ by duplicating gates on the hidden layer of C). Thus for any $\underline{z} \in \mathbb{R}^m$ we have $C(\underline{z}) = 1 \Leftrightarrow \sum\limits_{j=1}^{n} \alpha_j \hat{G}_j(\underline{z}) \geq \Theta$, where $\hat{G}_1, \ldots, \hat{G}_n$ are the threshold gates on the hidden layer of C, $\alpha_1, \ldots, \alpha_n$ are from $\{-1, 1\}$, and Θ is the threshold of the output gate G. In order to eliminate the negative weights in G we replace each gate $\hat{G}_j$ for which $\alpha_j = -1$ by another threshold gate G_j so that $G_j(\underline{z}) = 1 - \hat{G}_j(\underline{z})$ for all $\underline{z} \in \mathbb{R}^m$ except on some hyperplane. We utilize here that $\neg \sum_{i=1}^m w_i z_i \geq \Theta \Leftrightarrow \sum_{i=1}^m (-w_i) z_i > -\Theta$ for arbitrary $w_i, z_i, \Theta \in \mathbb{R}$. We set $G_j := \hat{G}_j$ for all $j \in \{1, \ldots, n\}$ with $\alpha_j = 1$. Then we have for all $\underline{z} \in \mathbb{R}^m$, except for $\underline{z}$ from some exception set S consisting of up to n hyperplanes,

$$\sum_{j=1}^n \alpha_j \hat{G}_j(\underline{z}) = \sum_{j=1}^n G_j(\underline{z}) - |\{j \in \{1, \ldots, n\} : \alpha_j = -1\}| \ .$$

Hence $C(\underline{z}) = 1 \Leftrightarrow \sum\limits_{j=1}^n G_j(\underline{z}) \geq k$ for all $\underline{z} \in \mathbb{R}^m - S$, for some suitable $k \in \mathbb{N}$.

Let $w_1^j, \ldots, w_m^j \in \mathbb{R}$ be the weights and $\Theta^j \in \mathbb{R}$ be the threshold of gate $G_j, j = 1, \ldots, n$. Thus $G_j(\underline{z}) = 1 \Leftrightarrow \sum\limits_{i:w_i^j > 0} |w_i^j| z_i - \sum\limits_{i:w_i^j < 0} |w_i^j| z_i \geq \Theta^j$. Hence with

$$S_j := \sum_{i:w_i^j < 0} |w_i^j| z_i + \Theta^j + \sum_{\ell \neq j} \sum_{i:w_i^\ell > 0} |w_i^\ell| z_i \qquad \text{for } j = 1, \ldots, n$$

and

$$S_{n+1} := \sum_{j=1}^n \sum_{i:w_i^j > 0} |w_i^j| z_i$$

we have for every $j \in \{1, \ldots, n\}$ and every $\underline{z} \in \mathbb{R}^m$:

$$S_{n+1} \geq S_j \Leftrightarrow \sum_{i:w_i^j > 0} |w_i^j| z_i - \sum_{i:w_i^j < 0} |w_i^j| z_i \ \ \geq \ \ \Theta^j \Leftrightarrow G_j(\underline{z}) = 1 \ .$$

This implies that the $(n+1)$st output b_{n+1} of the gate $(n-k+1)-\mathrm{WTA}_{n+1}$ applied to $S_1, \ldots, S_{n+1}$ satisfies

$$\begin{aligned} b_{n+1} = 1 \ \ & \Leftrightarrow |\{j \in \{1, \ldots, n+1\} : S_j > S_{n+1}\}| \leq n - k \\ & \Leftrightarrow |\{j \in \{1, \ldots, n+1\} : S_{n+1} \geq S_j\}| \geq k + 1 \\ & \Leftrightarrow |\{j \in \{1, \ldots, n\} : S_{n+1} \geq S_j\}| \geq k \\ & \Leftrightarrow \sum_{j=1}^n G_j(\underline{z}) \geq k \\ & \Leftrightarrow C(\underline{z}) = 1 \ . \end{aligned}$$

Note that all the coefficients in the sums $S_1, \ldots, S_{n+1}$ are positive. ∎

Proof of Corollary 3.4. Apply the preceding construction separately to the first two layers of C, then to the second two layers of C, etc. Note that the later layers of C just receive boolean inputs from the preceding layer. Hence one can simulate the computation of layer 3 to d also by a two-layer threshold circuit, that requires just a single $k-$WTA gate for its simulation in our approach from Theorem 3.1. ∎

4 Soft Winner-Take-All Applied to Positive Weighted Sums is an Universal Approximator

We consider in this section a soft version soft-WTA of a winner-take-all gate. Its output $\langle r_1, \ldots, r_n \rangle$ for real valued inputs $\langle x_1, \ldots, x_n \rangle$ consists of n analog numbers r_i, whose value reflects the relative position of x_i within the ordering of $x_1, \ldots, x_n$ according to their size. Soft versions of winner-take-all are also quite plausible as computational function of cortical circuits with lateral inhibition. Efficient implementations in analog VLSI are still elusive, but in (Indiveri, 1999) an analog VLSI circuit for a version of soft winner-take-all has been presented where the *time* when the ith output unit is activated reflects the rank of x_i among the inputs.

We show in this section that single gates from a fairly large class of soft winner-take-all gates can serve as the only nonlinearity in universal approximators for arbitrary continuous functions. The only other computational operations needed are weighted sums with positive weights. We start with a basic version of a soft winner-take-all gate soft-WTA$_{n,k}$ for natural numbers k, n with $k \leq \frac{n}{2}$, that computes the following function $\langle x_1, \ldots, x_n \rangle \mapsto \langle r_1, \ldots, r_n \rangle$ from $\mathbb{R}^n$ into $[0,1]^n$:

$$r_i := \pi\left(\frac{|\{j \in \{1, \ldots, n\} : x_i \geq x_j\}| - \frac{n}{2}}{k}\right),$$

where $\pi : \mathbb{R} \to [0,1]$ is the piecewise linear function defined by

$$\pi(x) = \begin{cases} 1, & \text{if} \quad x > 1 \\ x, & \text{if} \quad 0 \leq x \leq 1 \\ 0, & \text{if} \quad x < 0 \ . \end{cases}$$

If one implements a soft winner-take-all gate via lateral inhibition, one can expect that its ith output r_i is lowered (down from 1) by every competitor x_j that wins the competition with x_i (i.e., $x_j > x_i$). Furthermore one can expect that the ith output r_i is completely annihilated (i.e., is set equal to 0) once the number of competitors x_j that win the competition with x_i reaches a certain critical value c. This intuition has served as guiding principle in the previously described formalization. However for the sake of mathematical simplicity we have defined the outputs r_i of soft-WTA$_{n,k}$ not in terms of the number of competitors x_j that win over x_i (i.e., $x_j > x_i$), but rather in terms of the number of competitors x_j that do *not* win over x_i (i.e., $x_i \geq x_j$). Obviously one has $|\{j \in \{1, \ldots, n\} : x_i \geq x_j\}| = n - |\{j \in \{1, \ldots, n\} : x_j > x_i\}|$. Hence the value of r_i goes down if more competitors x_j win over x_i – as desired. The critical number c of winning competitors that suffice to "annihilate" r_i is formalized through the "threshold" $\frac{n}{2}$. Our subsequent result shows that in principle it suffices to set the annihilation-threshold always equal to $\frac{n}{2}$. But in applications one may of course choose other values for the annihilation threshold.

It is likely that an analog implementation of a soft-winner-take-all gate will not be able to produce an output that is really *linearly* related to $|\{j \in \{1, \ldots, n\} : x_i \geq x_j\}| - \frac{n}{2}$ over a sufficiently large range of values. Instead, an analog implementation is more likely to output a value of the form

$$g\left(\frac{|\{j \in \{1, \ldots, n\} : x_i \geq x_j\}| - \frac{n}{2}}{k}\right),$$

where the piecewise linear function π is replaced by some possibly rather complicated nonlinear warping of the target output values. The following result shows that such gates with any *given* nonlinear warping g can serve just as well as the competitive stage (and as the only nonlinearity) in universal approximators. More precisely, let g be *any* continuous function $g : \mathbb{R} \to [0, 1]$ that has value 1 for $x > 1$, value 0 for $x < 0$, and which is strictly increasing over the interval $[0, 1]$. We denote a soft winner-take-all gate that employs g instead of π by soft-WTA$^g_{n,k}$.

Theorem 4.1. *Assume that $h : D \to [0, 1]$ is an arbitrary continuous function with a bounded and closed domain $D \subseteq \mathbb{R}^m$ (for example: $D = [0, 1]^m$). Then for any $\varepsilon > 0$ and for any function g (satisfying above conditions) there exist natural numbers k, n, biases $\alpha^j_0 \in \mathbb{R}$, and nonnegative coefficients α^j_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, so that the circuit consisting of the soft winner-take-all gate soft-WTA$^g_{n,k}$ applied to the n sums $\sum_{i=1}^{m} \alpha^j_i z_i + \alpha^j_0$ for $j = 1, \ldots, n$ computes a function[3] $f : D \to [0, 1]$ so that $|f(\underline{z}) - h(\underline{z})| < \varepsilon$ for all $\underline{z} \in D$.*

Thus circuits consisting of a single soft WTA-gate applied to positive weighted sums of the input variables are universal approximators for continuous functions.

Remark 4.2. The proof shows that the number n of sums that arise in the circuit constructed in Theorem 4.1 can essentially be bounded in terms of the number of hidden gates in a one-hidden-layer circuit C of sigmoidal gates that approximates the given continuous function h (more precisely: the function $g^{-1} \circ h$), the maximal size of weights (expressed as multiple of the smallest

[3]more precisely: we set $f(\underline{z}) := r_n$ for the n-th output variable r_n of soft-WTA$^g_{n,k}$

nonzero weight) on the second layer of C, and $1/\varepsilon$ (where ε is the desired approximation precision). Numerous practical applications of backprop suggest that at least the first one of these three numbers can be kept fairly small for most functions h that are of practical interest.

Proof of Theorem 4.1: The proof has the following structure. We first apply the regular universal approximation theorem for sigmoidal neural nets in order to approximate the continuous function $g^{-1}(h(\underline{z}))$ by a weighted sum $\sum_{j=1}^{\tilde{n}} \tilde{\beta}_j G_j^1(\underline{z})$ of π-gates, i.e. of sigmoidal gates G_j^1 that employ the piecewise linear activation function π that was defined at the beginning of this chapter. As the next step we simplify the weights and approximate the weighted sum $\sum_{j=1}^{\tilde{n}} \tilde{\beta}_j G_j^1(\underline{z})$ by another weighted sum $\sum_{j=1}^{\hat{n}} \frac{\alpha_j}{\hat{k}} G_j^2(\underline{z})$ of π-gates G_j^2 with $\alpha_j \in \{-1, 1\}$ for $j = 1, \ldots, \hat{n}$ and some suitable $\hat{k} \in \mathbb{N}$. We then eliminate all negative weights by using that one can rewrite $-G_j^2(\underline{z})$ as $G_j^3(\underline{z}) - 1$ with another π-gate G_j^3, for all j with $\alpha_j = -1$. By setting $G_j^3 := G_j^2$ for all j with $\alpha_j = 1$ we then have $\sum_{j=1}^{\hat{n}} \frac{\alpha_j}{\hat{k}} G_j^2(\underline{z}) = \frac{\sum_{j=1}^{\hat{n}} G_j^3(\underline{z}) - \hat{T}}{\hat{k}}$ for all $\underline{z} \in D$, with some $\hat{T} \in \{0, \ldots, \hat{n}\}$. As the next step we approximate each π-gate G^3 by a sum $\frac{1}{\ell} \sum\limits_{i=1}^{\ell} G_{(i)}$ of ℓ threshold gates $G_{(i)}$. Thus altogether we have constructed an approximation of the continuous function $g^{-1}(h(\underline{z}))$ by a weighted sum $\sum_{j=1}^{n'} \frac{G_j(\underline{z}) - T}{k}$ of threshold gates with a uniform value $\frac{1}{k}$ for all weights.[4] By expanding our technique from the proof of Theorem 3.1 we can replace this simple sum of threshold gates by a single soft-WTA$_{n,k}^g$ gate applied to weighted sums in such a way that the resulting WTA-circuit approximates the given function h.

We now describe the proof in detail. According to (Leshno et al., 1993) there exist $\tilde{n} \in \mathbb{N}$, $\tilde{\beta}_j \in \mathbb{R}$, and π-gates G_j^1 for $j = 1, \ldots, \tilde{n}$ so that

$$|\sum_{j=1}^{\tilde{n}} \tilde{\beta}_j G_j^1(\underline{z}) - g^{-1}(h(\underline{z}))| < \frac{\tilde{\varepsilon}}{3} \qquad \text{for all } \underline{z} \in D \ ,$$

where g^{-1} is the inverse of the restriction of the given function g to $[0,1]$, and $\tilde{\varepsilon} > 0$ is chosen so that $|g(x) - g(y)| < \varepsilon$ for any $x, y \in \mathbb{R}$ with $|x - y| < \tilde{\varepsilon}$. We use here that $g^{-1} \circ h$ is continuous.

As the next step we simplify the weights. It is obvious that there exist

[4] One could of course also apply the universal approximation theorem directly to threshold gates (instead of π-gates) in order to get an approximation of $g^{-1} \circ h$ by a weighted sum of threshold gates. But the elimination of negative weights would then give rise to an exception set S like in Theorem 3.1.

$\hat{k} \in \mathbb{N}$ and $\beta_1, \ldots, \beta_{\tilde{n}} \in \mathbb{Z}$ so that $\sum_{j=1}^{\tilde{n}} |\tilde{\beta}_j - \frac{\beta_j}{\hat{k}}| < \frac{\tilde{\varepsilon}}{3}$. We then have

$$|\sum_{j=1}^{\tilde{n}} \frac{\beta_j}{\hat{k}} G_j^1(\underline{z}) - g^{-1}(h(\underline{z}))| < \frac{2}{3}\tilde{\varepsilon} \qquad \text{for all } \underline{z} \in D\ .$$

We set $\hat{n} := \sum\limits_{j=1}^{\tilde{n}} |\beta_j|$, and for all $j \in \{1, \ldots, \tilde{n}\}$ we create $|\beta_j|$ copies of the π-gate G_j^1. Let $G_1^2, \ldots, G_{\hat{n}}^2$ be the resulting sequence of π-gates G_j^2. Then there exist $\alpha_j \in \{-1, 1\}$ for $j = 1, \ldots, \hat{n}$ so that

$$|\sum_{j=1}^{\hat{n}} \frac{\alpha_j}{\hat{k}} G_j^2(\underline{z}) - g^{-1}(h(\underline{z}))| < \frac{2}{3}\tilde{\varepsilon} \qquad \text{for all } \underline{z} \in D\ .$$

We now eliminate all negative weights from this weighted sum. More precisely, we show that there are π-gates $G_1^3, \ldots, G_{\hat{n}}^3$ so that

$$\sum_{j=1}^{\hat{n}} \frac{\alpha_j}{\hat{k}} G_j^2(\underline{z}) = \frac{\sum_{j=1}^{\hat{n}} G_j^3(\underline{z}) - \hat{T}}{\hat{k}} \qquad \text{for all } \underline{z} \in D\ , \tag{1}$$

where $\hat{T}$ is the number of $j \in \{1, \ldots, \hat{n}\}$ with $\alpha_j = -1$. Consider some arbitrary $j \in \{1, \ldots, \hat{n}\}$ with $\alpha_j = -1$. Assume that G_j^2 is defined by $G_j^2(\underline{z}) = \pi(\underline{w} \cdot \underline{z} + w_0 + \frac{1}{2})$ for all $\underline{z} \in \mathbb{R}^m$, with parameters $\underline{w} \in \mathbb{R}^m$ and $w_0 \in \mathbb{R}$. Because of the properties of the function π we have

$$-\pi(\underline{w} \cdot \underline{z} + w_0 + \frac{1}{2}) \ = \ -1 + \pi(-\underline{w} \cdot \underline{z} - w_0 + \frac{1}{2})$$

for all $\underline{z} \in \mathbb{R}^m$. Indeed, if $|\underline{w} \cdot \underline{z} + w_0| \leq \frac{1}{2}$ we have $-\pi(\underline{w} \cdot \underline{z} + w_0 + \frac{1}{2}) = -\underline{w} \cdot \underline{z} - w_0 - \frac{1}{2} = -1 + \pi(-\underline{w} \cdot \underline{z} - w_0 + \frac{1}{2})$. If $\underline{w} \cdot \underline{z} + w_0 < -\frac{1}{2}$ then $-\pi(\underline{w} \cdot \underline{z} + w_0 + \frac{1}{2}) = 0 = -1 + \pi(-\underline{w} \cdot \underline{z} - w_0 + \frac{1}{2})$, and if $\underline{w} \cdot \underline{z} + w_0 > \frac{1}{2}$ then $-\pi(\underline{w} \cdot \underline{z} + w_0 + \frac{1}{2}) = -1 = -1 + \pi(-\underline{w} \cdot \underline{z} - w_0 + \frac{1}{2})$. Thus, if we define the π-gate G_j^3 by

$$G_j^3(\underline{z}) \ = \ \pi(-\underline{w} \cdot \underline{z} - w_0 + \frac{1}{2}) \qquad \text{for all } \underline{z} \in \mathbb{R}^m\ ,$$

we have $-G_j^2(\underline{z}) = -1 + G_j^3(\underline{z})$ for all $\underline{z} \in \mathbb{R}^m$. Besides transforming all π-gates G_j^2 with $\alpha_j = -1$ in this fashion, we set $G_j^3 = G_j^2$ for all $j \in \{1, \ldots, \hat{n}\}$ with $\alpha_j = 1$. Obviously equation (1) is satisfied with these definitions.

Since the activation function π can be approximated arbitrary closely by step functions, there exists for each π-gate G^3 a sequence $G_{(1)}, \ldots, G_{(\ell)}$ of threshold gates so that

$$|G^3(\underline{z}) - \frac{\sum_{i=1}^{\ell} G_{(i)}(\underline{z})}{\ell}| < \frac{\tilde{\varepsilon} \cdot \hat{k}}{3 \cdot \hat{n}} \qquad \text{for all } \underline{z} \in \mathbb{R}^m.$$

By applying this transformation to all π-gates $G_1^3, \ldots, G_{\hat{n}}^3$ we arrive at a sequence $G_1, \ldots, G_{n'}$ of threshold gates with $n' := \hat{n} \cdot \ell$ so that one has for $k := \hat{k} \cdot \ell$ and $T := \hat{T} \cdot \ell$ that

$$|\frac{\sum_{j=1}^{n'} G_j(\underline{z}) - T}{k} - g^{-1}(h(\underline{z}))| < \tilde{\varepsilon} \qquad \text{for all } \underline{z} \in D \ .$$

According to the choice of $\tilde{\varepsilon}$ this implies that

$$|g(\frac{\sum_{j=1}^{n'} G_j(\underline{z}) - T}{k}) - h(\underline{z})| < \varepsilon \qquad \text{for all } \underline{z} \in D \ .$$

By adding dummy threshold gates G_j that give constant output for all $\underline{z} \in D$ one can adjust n' and T so that $n' \geq 2k$ and $T = \frac{n'-1}{2}$, without changing the value of $\sum\limits_{j=1}^{n'} G_j(\underline{z}) - T$ for any $\underline{z} \in D$.

It just remains to show that

$$g(\frac{\sum_{j=1}^{n'} G_j(\underline{z}) - \frac{n'-1}{2}}{k})$$

can be computed for all $\underline{z} \in D$ by some soft winner-take-all gate soft-WTA$_{n,k}^g$ applied to n weighted sums with positive coefficients. We set $n := n' + 1$. For $j = 1, \ldots, n'$ and $i = 1, \ldots, m$ let $w_i^j, \Theta^j \in \mathbb{R}$ be parameters so that

$$G_j(\underline{z}) = 1 \Leftrightarrow \sum_{i:w_i^j > 0} |w_i^j| z_i - \sum_{i:w_i^j < 0} |w_i^j| z_i \geq \Theta^j \ .$$

Then $\sum\limits_{j=1}^{n'} G_j(\underline{z}) = |\{j \in \{1, \ldots, n'\} : S_{n'+1} \geq S_j\}|$ for $S_{n'+1} := \sum\limits_{j=1}^{n'} \sum\limits_{i:w_i^j>0} |w_i^j| z_i$ and $S_j := \sum\limits_{i:w_i^j<0} |w_i^j| z_i + \Theta^j + \sum\limits_{\ell \neq j} \sum\limits_{i:w_i^\ell > 0} |w_i^\ell| z_i$. This holds because we have for every $j \in \{1, \ldots, n'\}$

$$S_{n'+1} \geq S_j \Leftrightarrow \sum_{i:w_i^j > 0} |w_i^j| z_i - \sum_{i:w_i^j < 0} |w_i^j| z_i \geq \Theta^j \ .$$

Since $n = n' + 1$ and therefore $\frac{n'-1}{2} + 1 = \frac{n}{2}$, the preceding implies that the nth output variable r_n of soft-WTA$_{n,k}^g$ applied to $S_1, \ldots, S_n$ outputs

$$g(\frac{|\{j \in \{1, \ldots, n\} : S_n \geq S_j\}| - \frac{n}{2}}{k}) = g(\frac{\sum_{j=1}^{n'} G_j(\underline{z}) - \frac{n'-1}{2}}{k})$$

for all $\underline{z} \in D$. Note that all weights in the weighted sums $S_1, \ldots, S_n$ are positive. ∎

5 Conclusions

We have established the first rigorous analytical results regarding the computational power of winner-take-all.

The lower bound result of section 2 shows that the computational power of hard winner-take-all is already quite large, even if compared with the arguably most powerful gate commonly studied in circuit complexity theory: the threshold gate (also referred to a McCulloch-Pitts neuron or perceptron). Theorem 2.1 yields an optimal quadratic lower bound for computing hard winner-take-all on any feedforward circuit consisting of threshold gates. This implies that no circuit consisting of linearly many threshold gates with arbitrary (i.e., feedforward, lateral, and recurrent) connections can compute hard winner-take-all in sublinear time. Since approximate versions of winner-take-all can be computed very fast in linear size analog VLSI chips (Lazzaro et al., 1989), this lower bound result may be viewed as evidence for a possible gain in computational power that can be achieved via lateral connections in analog VLSI (and apparently also in cortical circuits).

It is well known ((Minsky and Papert, 1969)) that a single threshold gate is not able to compute certain important functions, whereas circuits of moderate (i.e., polynomial) size consisting of two layers of threshold gates with polynomial size weights have remarkable computational power (Siu et al., 1995). We have shown in section 3 that any such 2-layer (i.e., 1 hidden layer) circuit can be simulated by a single competitive stage, applied to polynomially many weighted sums with positive integer weights of polynomial size.

In section 4 we have analyzed the computational power of soft winner-take-all gates in the context of *analog* computation. It was shown that a single soft winner-take-all gate may serve as the only nonlinearity in a class of circuits that have universal computational power in the sense that they can approximate any continuous functions. In addition we have shown that this result is robust with regard to any continuous nonlinear warping of the output of such soft winner-take-all gate, which is likely to occur in an analog implementation of soft winner-take-all in hardware or wetware. Furthermore our novel universal approximators require only *positive* linear operations besides soft winner-take-all, thereby showing that in principle no computational power is lost if inhibition is used exclusively for unspecific lateral inhibition in neural circuits, and no flexibility is lost if synaptic plasticity (i.e., "learning") is restricted to excitatory synapses.

This result appears to be of interest for the understanding of the function of biological neural systems, since typically only 15% of synapses in the cortex are inhibitory, and plasticity for those synapses is somewhat dubious.

Our somewhat surprising results regarding the computational power and universality of winner-take-all point to further opportunities for low-power analog VLSI chips, since winner-take-all can be implemented very efficiently in this technology. In particular, our theoretical results of section 4 predict that efficient implementations of soft winner-take-all will be useful in many contexts. Previous analog VLSI implementations of winner-take-all have primarily been used for special purpose computational tasks. In contrast, our results show that a VLSI implementation of a single soft winner-take-all in combination with circuitry for computing weighted sums of the input variables yields devices with universal computational power for analog computation.

Altogether the theoretical results of this article support the viability of an alternative style of neural circuit design, where complex multi-layer perceptrons, i.e., feedforward circuits consisting of threshold gates or sigmoidal gates with positive and negative weights, are replaced by a single competitive stage applied to positive weighted sums. One may argue that this circuit design is more compatible with anatomical and physiological data from biological neural circuits.

Acknowledgment:
I would like to thank Moshe Abeles, Peter Auer, Rodney Douglas, Timmer Horiuchi, Giacomo Indiveri, and Shih-Chii Liu for helpful discussions, and the anonymous referees for helpful comments.

References

Abeles, M. (1991). *Corticonics: Neural Circuits of the Cerebral Cortex.* Cambridge University Press (Cambridge, UK).

Amari, S. and Arbib, M. (1977). Competition and cooperation in neural nets. In: *Systems Neuroscience*, J. Metzler, Ed., Academic Press (San Diego), 119–165.

Andreou, A. G., Boahen, K. A., Pouliquen, P. O, Pavasovic, A., Jenkins, R. E., and Strohbehn, K. (1991). Current-mode subthreshold MOS circuits for analog VLSI neural systems. *IEEE Trans. on Neural Networks*, 2:2, 205–213.

Arbib, M. A. (1995). *The Handbook of Brain Theory and Neural Networks.* MIT Press (Cambridge, MA, USA).

Brajovic, V., and Kanade, T. (1998). Computational sensor for visual tracking with attention, *IEEE Journal of Solid State Circuits* , vol. 33, 8, 1199-1207.

Choi, J. and Sheu B. J. (1993). A high precision VLSI winner-take-all circuit for self-organizing neural networks. *IEEE J. Solid-state circuits*, 28:5, 576–584.

Coultrip, R., Granger R., and Lynch, G. (1992). A cortical model of winner-take-all competition via lateral inhibition. *Neural Networks*, vol. 5, 47–54.

DeWeerth, S. P., and Morris, T. G. (1994). Analog VLSI circuits for primitive sensory attention. *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 6, 507–510.

DeYong, M., Findley, R., Fields, C. (1992). The design, fabrication, and test of a new VLSI hybrid analog-digital neural processing element. *IEEE Trans. on Neural Networks*, vol. 3, 363–374.

Elias, S. A. and Grossberg, S. (1975). Pattern formation, contrast control, and oscillations in the short term memory of shunting on-center off-surround networks. *Biol. Cybern.*, 20:69–98.

Fang, Y., Cohen, M., and Kincaid, M. (1996). Dynamics of a winner-take-all neural network. *Neural Networks*, 9:7, 1141–1154.

Horiuchi, T. K., Morris, T. G., Koch, C., DeWeerth, S. P. (1997). Analog VLSI circuits for attention-based visual tracking. *Advances in Neural Information Processing Systems*, vol. 9, 706–712.

Indiveri, G. (1999). A 2D neuromorphic VLSI architecture for modeling selective attention, submitted for publication.

Lazzaro, J., Ryckebusch, S., Mahowald, M. A., Mead, C. A. (1989). Winner-take-all networks of $O(n)$ complexity. *Advances in Neural Information Processing Systems*, vol. 1, Morgan Kaufmann (San Mateo), 703-711.

Leshno, M., Lin, V., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867.

Meador, J. L., and Hylander, P. D. (1994). Pulse coded winner-take-all networks. In: *Silicon Implementation of Pulse Coded Neural Networks*, Zaghloul, M. E., Meador, J., and Newcomb, R. W., eds., Kluwer Academic Publishers (Boston), 79–99.

Minsky, M. C., Papert, S. A. (1969). *Perceptrons*, MIT Press (Cambridge).

Niebur, E., and Koch, C. (1998). Computational architectures for attention. In: *The Attentive Brain*, Parasuraman, R., ed., MIT Press (Cambridge), 163–186.

Savage, J. E. (1998). *Models of Computation: Exploring the Power of Computing.* Addison-Wesley (Reading, MA, USA).

Shepherd, G. M., ed. (1998). *The Synaptic Organization of the Brain.* Oxford University Press (Oxford, UK).

Siu, K.-Y., Roychowdhury, V., Kailath, T. (1995). *Discrete Neural Computation: A Theoretical Foundation.* Prentice Hall (Englewood Cliffs, NJ, USA).

Thorpe, S. J. (1990). Spike arrival times: a highly efficient coding scheme for neural networks. In: *Parallel Processing in Neural Systems and Computers*, Eckmiller, R., Hartmann, G., and Hauske, G., eds., Elsevier Science Publishers B. V., (Amsterdam), 91–94.

Urahama, K., and Nagao, T. (1995). k-winner-take-all circuit with $O(N)$ complexity. *IEEE Trans. on Neural Networks*, vol.6, 776–778.

Yuille, A. L. and Grzywacz, N. M. (1989). A winner-take.all mechanism based on presynaptic inhibition. *Neural Computation*, 1:334-347.

Wegener, I. (1987). *The Complexity of Boolean Functions.* Wiley-Teubner (Chichester).