Foundations for a Circuit Complexity Theory of Sensory Processing

Robert A. Legenstein & Wolfgang Maass
Institute for Theoretical Computer Science
Technische Universität Graz, Austria
{legi, maass}@igi.tu-graz.ac.at

Abstract

We introduce *total wire length* as salient complexity measure for an analysis of the circuit complexity of sensory processing in biological neural systems and neuromorphic engineering. Furthermore we introduce a set of basic computational problems that apparently need to be solved by circuits for translation- and scale-invariant sensory processing. Finally we exhibit a number of circuit design strategies for these new benchmark functions that can be implemented within realistic complexity bounds, in particular with linear or almost linear total wire length.

1 Introduction

Circuit complexity theory is a classical area of theoretical computer science, that provides estimates for the complexity of circuits for computing specific benchmark functions, such a binary addition, multiplication and sorting (see, e.g. (Savage, 1998)). In recent years interest has grown in understanding the complexity of circuits for early sensory processing, both from the biological point of view and from the point of view of neuromorphic engineering (see (Mead, 1989)). However classical circuit complexity theory has provided little insight into these questions, both because its focus lies on a different set of computational problems, and because its traditional complexity measures are not tailored to those resources that are of primary interest in neuromorphic engineering, especially analog VLSI, and the analysis of neural circuits in biological organisms. This deficit is quite unfortunate since there is growing demand for efficient hardware for sensory processing, and complexity issues become very important since the number n of parallel inputs which such circuits have to handle is typically quite large (for example $n \geq 10^6$ in the case of many visual processing tasks).

We will follow traditional circuit complexity theory in assuming that the underlying graph of each circuit is a directed graph without cycles.[1] The most frequently considered com-

[1]Neural circuits in "wetware" as well as most circuits in analog VLSI contain in addition to feedforward connections also lateral and recurrent connections. This fact presents a serious obstacle for a direct mathematical analysis of such circuits. The standard mathematical approach is to model such circuits by a larger feedforward circuit, where new "virtual gates" are introduced to represent the state of existing gates at later points in time. In addition we will treat analog non-feedforward circuit modules, such as subcircuits that carry out an efficient winner-take-all computation with the help of "lateral inhibition", as "black box modules" (i.e., as gates) in our complexity analysis.

plexity measures in traditional circuit complexity theory are the number (and types) of gates, as well as the depth of a circuit. The latter is defined as the length of the longest directed path in the underlying graph, and is also interpreted as the computation time of the circuit. The focus lies in general on the classification of functions that can be computed by circuits whose number of gates can be bounded by a polynomial in the number n of input variables. This implicitly also provides a polynomial – although typically quite large – bound on the number of "wires" (defined as the edges in the underlying graph of the circuit). We proceed on the assumption that the area (or volume in the case of neural circuits) occupied by wires is a severe bottleneck for physical implementations of circuits for sensory processing. Therefore we will not just count the wires, but consider complexity measures that provide estimates for the total area or volume occupied by wires. We assume that all gates, input- and output-ports are placed on a 2-dimensional plane.[2] We evaluate the required wire length of circuit designs in two closely related models:

(A) Gates, input- and output-ports are placed on different intersection points of a 2-dimensional grid (with unit distance 1 between adjacent intersection points). These units can be connected by (unidirectional) wires that run through the plane in any way that the designer wants, in particular wires may cross and need not run rectilinearly (wires are thought of as running in the 3 dimensional space above the plane, without charge for vertical wire segments)[3]. We refer to the minimal value of the sum of all wire lengths as the *total wire length* of the circuit.

We would like to make this model also applicable to cases where for $k > 2$ some special functions of k inputs – such as the function computed by a threshold gate[4] or a winner-take-all circuit[5] – are computed by neural microcircuits or in analog VLSI by efficient subcircuits that employ a number of transistors, total wire length and area that are all linear in k, with a setting time that is independent of k [6]. In the relatively abstract context of model (A) we model such computational modules as "threshold gates" or "winner-take-all gates" of k inputs, that take one unit of time for their computation like all the other gates, but which occupy each a set of k intersection points of the grid that are all connected by an undirected wire (whose length contributes to the total wire length) in some arbitrary fashion[7].

(B) This is the common circuit model from VLSI-theory (see section 12.2 in (Savage, 1998)), slightly extended to cover also the case of analog VLSI-modules (that internally may contain recurrent connections) with more than 2 inputs, such as modules for winner-take-all. One assumes that gates, input- and output-ports and

[2]This assumption is not fully justified for neural circuits as for example in the cortex, where neurons occupy an about 2 mm thick 3-dimensional sheet of "grey matter". However even in the cortex there exists a strikingly general upper bound of about 10^5 for the number of neurons under any mm^2 of cortical surface (Koch, 1999). Hence if one ignores the contribution of vertical segments to the total wire length and projects these neurons onto a 2-dimensional plane, our model provides a reasonable basis for a rough estimate of the required total wire length even for implementations in "wetware". Note that the symbolic gates or inputs in our circuit analysis may correspond in biological neural circuits to small populations or microcircuits consisting of hundreds of neurons, rather than to single neurons.

[3]We will allow that a wire from a gate may branch and provide input to several other gates. For reasonable bounds on the maximal fan-out (10^4 in the case of neural circuits) this is realistic both for neural circuits and for VLSI.

[4]A threshold gate computes a Boolean function $T : \{0,1\}^k \rightarrow \{0,1\}$ of the form $T(x_1, \ldots, x_k) = 1 \Leftrightarrow \sum_{i=1}^{k} w_i x_i \geq w_0$.

[5]A winner-take-all gate with weights $w_1, \ldots, w_k$ computes a Boolean function $W : \{0,1\}^k \rightarrow \{0,1\}^k$ where for input $x_1, \ldots, x_k$ the ith output is 1 if and only if $w_i x_i > w_j x_j$ for all $j \neq i$.

[6]see (Lazzaro et al., 1989)

[7]Any one of these k nodes may be used to provide one of the k inputs or to extract one of the outputs of the function.

wires cover rectilinear areas with a minimal width and separation λ. Areas occupied by different gates, input- and output-ports are not allowed to intersect with one another. A computational module for winner-take-all with k inputs is represented by a rectilinear gate area of size $k \cdot \lambda$. A threshold gate with k inputs is modeled by $k + 1$ gates (k of them for multiplying a binary input with a weight) that are connected by a wire. Areas occupied by wires may intersect with areas occupied by gates, input- and output-ports and also with other wires, but there is a constant bound ν on the number of wire areas to which a point of the plane may belong. The complexity measure induced by the model is the *area* of the smallest rectangle that encloses the circuit.

The attractiveness of model (A) lies in its mathematical simplicity, and in its generality. It provides a rough estimate for the cost of connectivity both in artificial (basically 2-dimensional) circuits and in neural circuits, where 2-dimensional wire crossing problems are apparently avoided (at least on a small scale) since dendritic and axonal branches are routed through 3-dimensional cortical tissue. Model (B) is useful for testing whether some specific circuit architecture can be ported from neural to artificial cirucits. In either model we follow (Savage, 1998) in assuming that one unit of time is needed to transmit a bit across a wire (of any length), and one unit of time for each gate switching. However in contrast to (Savage, 1998) we always assume that all inputs are presented in parallel.

There exist quite reliable estimates for the order of magnitudes for the number n of inputs, the number of neurons and the total wire length of biological neural circuits for sensory processing, see (Abeles, 1998; Koch, 1999; Shepherd, 1998; Braitenberg and Schüz, 1998).[8] Collectively they suggest that only those circuit architectures for sensory processing are biologically realistic that employ an almost linear number of gates and a quadratic or subquadratic total wire length – provided that the constant factor in front of the asymptotic complexity bound has a value close to 1. Since most asymptotic bounds in circuit complexity theory have constant factors in front that are much larger than 1, one really has to focus on circuit architectures with clearly subquadratic bounds for the total wire length.

[8]The number of neurons that transmit information from the retina (via the thalamus) to the cortex is estimated to be around 10^6 (all estimates given are for primates, and they only reflect the order of magnitude). The total number of neurons that transmit sensory (mostly somatosensory) information to the cortex is estimated to be around 10^8. In the subsequent sections we assume that these inputs represent the outputs of various local feature detectors for n locations in some 1- or 2-dimensional map. Thus, if one assumes for example that on average there are 10 different feature detectors for each location on this map, one arrives at biologically realistic estimates for n that lie between 10^5 and 10^7.

The total number of neurons in the primary visual cortex of primates is estimated to be around 10^9, occupying an area of roughly 10^4 mm^2 of cortical surface. There are up to 10^5 neurons under one mm^2 of cortical surface, which yields a value of $10^{-5/2}$ mm for the distance between adjacent grid points in our model (A). The total length of axonal and dendritic branches below one mm^2 of cortical surface is estimated to be between 1 and 10 km, yielding up to 10^{11} mm total wire length for primary visual cortex. Thus if one assumes that 100 separate circuits are implemented in primary visual cortex, each of them can use 10^7 neurons and a total wire length of 10^9 mm. Hence realistic bounds for the complexity of a single one of these circuits for visual pattern recognition are $10^7 = n^{7/5}$ neurons (for $n = 10^5$), and a total wire length of at most $10^{11.5} = n^{2.3}$ in the framework of model (A).

The whole cortex receives sensory input from about 10^8 neurons. It processes this input with about 10^{10} neurons and less than 10^{12} mm total wire length. If one assumes that 10^3 separate circuits process this sensory information in parallel, each of them processing about $1/10$th of the input (where again 10 different local feature detectors report about every location in a map), one arrives at $n = 10^6$ for each circuit, and each circuit can use on average $n^{7/6}$ neurons and a total wire length of $10^{11.5} < n^2$ in the sense of model (A). The actual resources available for sensory processing are likely to be substantially smaller, since cortical neurons and circuits have many other functions.

The complexity bounds for circuits that can realistically be implemented in VLSI are typically even more severe than for "wetware", and linear or almost linear bounds for the total wire length are desirable for that purpose. In this article we begin the investigation of algorithms for basic sensory pattern recognition tasks that can be implemented within this low-level complexity regime. The architecture of such circuits has to differ strongly from most previously proposed circuits for sensory processing, since already complete connectivity between just two linear size 2-dimensional layers of a feedforward neural net requires a total wire length of $\Omega(n^{5/2})$. Furthermore a circuit which first selects a salient input segment consisting of a block of up to m adjacent inputs in some 2-dimensional map, and then sends this block of $\leq m$ inputs in parallel to some central "pattern template matcher", typically requires a total wire length of $\Omega(n^{3/2} \cdot m)$ – even without taking the circuitry for the "selection" or the template matching into account.

2 Global Pattern Detection in 1-Dimensional Maps

Assume that local feature detectors are arranged in a 1-dimensional map, and the computational task is to detect global patterns in the form of specific spatial arrangements of local features in a translation- and scale-invariant manner. The 1-dimensional map could for example represent time and the local feature detectors could be detectors for certain phonemes, or the 1-dimensional map could represent frequency-bands in a (static) spectrogram and the local feature detectors might for example detect specific intensity changes in certain frequency bands, or the 1-dimensional map could represent one spatial dimension in visual space (for example left-right or up-down).

We employ the following framework for analyzing the circuit complexity of 1-dimensional pattern detection problems. Assume that one has arrays $a_1, \ldots, a_n; b_1, \ldots, b_n; \ldots$ of binary variables, where $a_i = 1$ signals that the local feature a has been detected at location i, and $b_i = 1$ signals that the local feature b has been detected at location i, etc. An interesting prototype for a global pattern detecting problem is whether feature a occurs at a location i that lies to the left of a location j where feature b occurs:

$$P^n_{LR}(a_1, \ldots, a_n, b_1, \ldots, b_n) = \begin{cases} 1, & \text{if } \exists i, j (i < j \text{ and } a_i = b_j = 1) \\ 0, & \text{else .} \end{cases}$$

We assume in the layout-analysis of all circuits for P^n_{LR} that its $2n$ inputs are provided on one row of the grid, possibly with space in between. One can easily construct a feedforward circuit based on AND/OR gates of fan-in 2, that computes P^n_{LR} with $O(n)$ gates in depth $O(\log n)$ with total wire length and area $O(n \log n)$. The following lower bound result shows that *any* circuit for P^n_{LR} that is based on a balanced binary tree has a superlinear total wire length of $\Omega(n \log n)$.

Theorem 2.1 *Any circuit that implements a balanced binary tree with n leafs that lie on one row of a grid has total wire length $\geq \frac{n \log n}{4}$.*

Idea of the proof: Show by induction on m that any circuit that implements a balanced binary tree for $n = 2^m$ leafs lying on one row has total wire length at least $\frac{n \log n}{4} + x$, where x denotes the horizontal distance of the location of the root of the tree from the midpoint of the leafs. ∎

In view of this lower bound result it comes as a pleasant surprise that many basic global pattern detection problems can nevertheless be solved in time $O(\log n)$ by circuits with *linear* total wire length – as we will show in this article.

The following result shows that by using threshold gates one can drastically improve the computation time in comparison with the circuit based on AND/OR gates of fan-in 2, without increasing the total wire length.

Theorem 2.2 *P^n_{LR} can be computed by a feedforward circuit of depth 2, consisting of $2\log n + 1$ threshold gates with total wire length $O(n \log n)$ in model (A) and area $O(n \log n)$ in model (B).*

The following lower bound result shows that the number of threshold gates used by the circuit of Theorem 2.2 is almost optimal:

Theorem 2.3 *Any feedforward circuit consisting of threshold gates needs to have at least $\Omega(\log n)$ gates for computing P^n_{LR}.*

In analog VLSI the area occupied by a subcircuit that implements a winner-take-all gate is comparable to that for a threshold gate. Hence the next theorem demonstrates a drastic gain in efficiency if one employs modules for computing winner-take-all in addition to threshold gates. It combines the minimal possible computation time of 2 with a *linear* total wire length.

Theorem 2.4 *P^n_{LR} can be computed by a feedforward circuit of depth 2, consisting of two winner-take-all gates and one threshold gate, with total wire length and area $O(n)$.*

In contrast to the threshold circuit of Theorem 2.2 just linear size integer weights are needed for this circuit.

3 Global Pattern Detection in 2-Dimensional Maps

For many important sensory processing tasks – such as for visual or somatosensory input – the input variables are arranged in a 2-dimensional map whose structure reflects spatial relationship in the outside world. We assume that local feature detectors are able to detect the presence of salient local features in their specific "receptive field", such as for example a center which emits higher (or lower) intensity than its immediate surrounding, or a high-intensity line segment in a certain direction, the end of a line, a junction of line segments, or even more complex local visual patterns like an eye or a nose. The ultimate computational goal is to detect specific *global* spatial arrangements of such local patterns, such as the letter "T", or in the end also a human face, in a translation- and scale-invariant manner.

We formalize 2-dimensional global pattern detection problems by assuming that the input consists of arrays $\underline{a} = \langle a_1, \ldots, a_n \rangle, \underline{b} = \langle b_1, \ldots, b_n \rangle$, etc. of binary variables that are arranged on a 2-dimensional square grid[9]. Each index i can be thought of as representing a location within some $\sqrt{n} \times \sqrt{n}$-square in the outside world. In our formal model a subsquare within the 2-dimensional grid is reserved for each index i, where the input variables a_i, b_i, etc. are given on adjacent nodes of this grid[10]. Since we assume that this spatial arrangement of input variables reflects spatial relations in the outside world, salient examples for global pattern detection problems require the computation of functions such as

$$P^n_D(\underline{a}, \underline{b}) = \begin{cases} 1, & \text{if there exist } i \text{ and } j \text{ so that } a_i = b_j = 1 \text{ and input location } j \\ & \text{is above and to the right of input location } i \\ 0, & \text{else} \end{cases}$$

[9]Whenever needed we assume for simplicity that n is such that $\sqrt{n}, \log n$ etc. are natural numbers. The square grid for the input variables may leave some nodes empty, which can be occupied by gates in model (A).

[10]To make this more formal one can assume that indices i and j represent pairs $\langle i_1, i_2 \rangle, \langle j_1, j_2 \rangle$ of coordinates. Then "input location j is above and to the right of input location i" means: $i_1 < j_1$ and $i_2 < j_2$. The circuit complexity of variations of the function P^n_D where one or both of the "$<$" are replaced by "$\leq$" is the same.

Theorem 3.1 *The function P_D^n can be computed – and witnesses i and j with $a_i = b_j = 1$ can be exhibited if they exist – by a circuit with total wire length $O(n)$ and area $O(n)$, consisting of $O(n)$ Boolean gates of fan-in 2 (and fan-out 2) in depth $O(\log n \cdot \log\log n)$. The depth of the circuit can be reduced to $O(\log n)$ if one employs threshold gates with fan-in $\log n$. This can also be done with total wire length and area $O(n)$.*

The linear total wire length and area of this circuit – which is up to a constant factor *optimal* for any circuit whose output depends on all of its n inputs – is achieved with the help of a suitable modification of an H-tree. The square where the inputs are received is iteratively divided into 4 subsquares. For each subsquare one computes recursively the locations of the highest, lowest, leftmost, and rightmost occurrences of the two features a and b. ∎

The next theorem shows that one can compute P_D^n faster if one can afford a somewhat larger total wire length. This circuit construction, that is based on AND/OR gates of limited fan-in Δ, has the additional advantage that it can not just exhibit *some* pair $\langle i, j\rangle$ as witness for $P_D^n(\underline{a}, \underline{b}) = 1$ (provided such witness exists), but it can exhibit in addition *all* j that can be used as witness together with some i. This property allows us to "chain" the global pattern detection problem formalized through the function P_D^n, and to decide within the same complexity bound whether for any fixed number k of input vectors $\underline{a}^{(1)}, \ldots, \underline{a}^{(k)}$ from $\{0,1\}^n$ there exist locations $i^{(1)}, \ldots, i^{(k)}$ so that $a_{i^{(m)}}^{(m)} = 1$ for $m = 1, \ldots, k$ and location $i^{(m+1)}$ lies to the right and above location $i^{(m)}$ for $m = 1, \ldots, k-1$. In fact, one can also compute a k-tuple of witnesses $i^{(1)}, \ldots, i^{(k)}$ within the same complexity bounds, provided it exists. This circuit design is based on an efficient layout for prefix computations.

Theorem 3.2 *For any given n and $\Delta \in \{2, \ldots, \sqrt{n}\}$ one can compute the function P_D^n in depth $O(\frac{\log n}{\log \Delta})$ by a feedforward circuit consisting of $O(n)$ AND/OR gates of fan-in $\leq \Delta$, with total wire length $O(n \cdot \Delta \cdot \frac{\log n}{\log \Delta})$ and area $O(n \cdot (\Delta \cdot \frac{\log n}{\log \Delta})^2)$.* ∎

Another essential ingredient of translation- and scale-invariant global pattern recognition is the capability to detect whether a local feature c occurs (roughly) in the middle between locations i and j where the local features a and b occur. This global pattern detection problem is formalized through the following function P_I^n:

If $\sum \underline{a} = \sum \underline{b} = 1$ then

$$P_I^n(\underline{a}, \underline{b}, \underline{c}) = \begin{cases} 1, & \text{if there exist } i, j, k \text{ so that input location } k \text{ lies on the} \\ & \text{middle of the line between locations } i \text{ and } j, \text{ and} \\ & a_i = b_j = c_k = 1 \\ 0, & \text{else} \quad . \end{cases}$$

This function P_I^n can be computed very fast by circuits with the least possible total wire length and area (up to constant factors), using threshold gates of fan-in up to $\sqrt{n}$:

Theorem 3.3 *The function P_I^n can be computed – and witnesses can be exhibited – by a circuit with total wire length and area $O(n)$, consisting of $O(n)$ Boolean gates of fan-in 2 and $O(\sqrt{n})$ threshold gates of fan-in $\sqrt{n}$ in depth 5.*

The design of the circuit exploits that the computation of P_I^n can be reduced to the solution of two closely related 1-dimensional problems. ∎

4 Discussion

There exists a very large literature on neural circuits for translation-invariant pattern recognition see http://www.cnl.salk.edu/~wiskott/Bibliographies/Invariances.html. However there exists substantial disagreement regarding the interpretation of existing approaches see http://www.ph.tn.tudelft.nl/PRInfo/shift/maillist.html. Virtually all positive results are based on computer simulations of small circuits, or of learning algorithms for concrete neural networks with a fixed input size n on the order of 20 or 30, without an analysis how the required number of gates and the area or volume occupied by wires scale up with the input size. The computational performance of these networks is often reported in an anecdotical manner.

The goal of this article is to show that circuit complexity theory may become a useful ingredient for understanding the computational strategies of biological neural circuits, and for extracting from them portable principles that can be applied to novel artificial circuits. Therefore we have introduced in model (A) the total wire length as an abstract complexity measure that appears to be among the most salient ones in this context, and which can in principle be applied both to neural circuits in the cortex and to artificial circuitry. We would like to argue that only those computational strategies that can be implemented with subquadratic total wire length have a chance to reflect aspects of cortical information processing, and only those with almost linear total wire length are implementable in special purpose VLSI-chips for real-world sensory processing tasks.[11] We have formalized some basic computational problems, that appear to underly various translation- and scale-invariant sensory processing tasks, as a first set of benchmark functions for a circuit complexity theory of sensory processing. We have presented designs for circuits that compute these benchmark functions with small – in most cases linear or almost linear – total wire length. The computational strategies of these circuits differ strongly from those that have been considered in previous approaches, which failed to take the limitations imposed by the realistically available amount of total wire length into account[12].

Complete proofs of the Theorems in this extended abstract are available for the referees on http://www.tu-graz.ac.at/igi/maass/a1 .

References

Abeles, M. (1998). *Corticonics: Neural Circuits of the Cerebral Cortex*, Cambridge Univ. Press.

Braitenberg, V., Schüz, A. (1998). *Cortex: Statistics and Geometry of Neuronal Connectivity*, 2nd ed., Springer Verlag.

Koch, C. (1999). *Biophysics of Computation*, Oxford Univ. Press.

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.

Mead, C. (1989). *Analog VLSI and Neural Systems*. Addison-Wesley (Reading, MA, USA).

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

Shepherd, G. M. (1998). *The Synaptic Organisation of the Brain*, 2nd ed., Oxford Univ. Press.

[11] Of course there are other important complexity measures for circuits – such as energy consumption – besides those that have been addressed in this article.

[12] We do *not* want to argue that learning plays no role in the design and optimization of circuits for specific sensory processing tasks; on the contrary. But one of the few points where the discussion from http://www.ph.tn.tudelft.nl/PRInfo/shift/maillist.html agreed is that translation- and scale-invariant pattern recognition is a task which is so demanding, that learning algorithms have to be supported by pre-existing circuit structures. The goal of this article is to contribute to a more systematic understanding of circuit structures with realistic complexity bounds that are suitable to be trained by a learning algorithm for more specific sensory processing tasks.