Neural Computation: A Research Topic for Theoretical Computer Science? Some Thoughts and Pointers.

Wolfgang Maass

Institute for Theoretical Computer Science
Technische Universität Graz
A-8010 Graz, Austria
email: maass@igi.tu-graz.ac.at
http://www.tu-graz.ac.at/igi/maass

Abstract

We address difficulties and opportunities for research contributions from theoretical computer science in the area of neural computation. In addition some pointers to sources for further information are provided.

1 Introduction

One of the most interesting scientific developments during the next few decades will be the unraveling of the structure of computation in the nervous systems of living organisms. Since the information processing capabilities of living organisms are in many aspects superior to those of our current artificial computing machinery, this is likely to have significant consequences for the way in which computers and robots will be designed in the year 2020. Learning from successful computational strategies of neural systems will be especially desirable for computer science because:

- The customary speedup of computers resulting from decreases in component sizes on computer chips is likely to reach hard physical limits during the next 10-20 years, and new architectural ideas are needed to achieve further improvements in computational speed.
- A large fraction of computational machinery is going to leave our desks, and will enter handhold consumer products (we are just seeing the beginning of this trend in the current evolution of mobile phones). In this context the power consumed by computational devices represents a major bottleneck. Neural organisms have developed ingenious strategies to compute with little energy. For example the energy

consumption of a biological supercomputer – the human brain – is estimated to be around 12 W.

- The days when it suffices for a computer to be good at number crunching are over. In conjunction with the previously described trend where computers leave the offices and enter our everyday life we want them to give people without computer expertise intelligent advice, and to adapt to their personal preferences and habits.

Traditionally theoretical computer science has played the role of a scout that explores novel approaches towards computing well in advance of other sciences. This did also occur in the case of neural computation. Von Neumann's book *The Computer and the Brain* [von Neumann, 1958] raised over 40 years ago already several of the essential issues for understanding neural computation from the point of view of a theoretical computer scientist. Even earlier, the seminal paper *A logical calculus of the ideas immanent in nervous activity* [McCulloch and Pitts, 1943] provided an abstract circuit model[1] that reflects some essential aspects of computation in biological neural systems, but which is at the same time sufficiently simple and formally precise so that it can be investigated theoretically.[2] The paper *A Theory of the Learnable* by Valiant [Valiant, 1984] initiated research in theoretical computer science on another essential ingredient of neural systems: learning capabilities. This created the new area of computational learning theory in theoretical computer science. It made a number of important contributions to applied machine learning, but so far had little impact on the investigation of learning in neural systems[3]. Another mathematically rigorous approach towards understanding learning, reinforcement learning (see [Bertsekas and Tsitsiklis, 1996, Sutton and Barto, 1998]), has been more successful in this regard. But so far reinforcement learning has attracted little attention in theoretical computer science.

Altogether we would be very happy if we could discern a slow but steady development where theoretical computer science is gaining increasing impact on the investigation of computing and learning in neural systems. Unfortunately there is little support for such optimism , and concerted efforts would be needed to change this situation. An inspection of any recent issue of leading journals (e.g. *Neural Computation*[4], *Network: Computation in Neural Systems*) or conference proceedings in this area (e.g. of the Annual NIPS[5], with proceedings published by MIT Press under the title *Advances in Neural Information Processing Systems*) shows that there exists a large amount of interdisciplinary work on neural computation, with a fair number of theoretical contributions. But so far this theoretical work has been dominated by approaches from theoretical physics, information theory, and statistics. One might speculate that this results from the fact that theoretical computer science has become to a large extent "method-driven", i.e., we typically look

[1]This model is nowadays called a *threshold circuit*. The books [Siu et al., 1995] and [Roychowdhury et al., 1994] provide good surveys.

[2]Curiously enough it is reported in [Hopcroft and Ullman, 1979] that one of the main concepts of theoretical computer science, the finite automation, had been conceived by Kleene [Kleene, 1956] when he investigated the model proposed by McCulloch and Pitts.

[3]A program for applying results from this area towards understanding neural systems is outlined in Valiant's book *Circuits of the Mind* [Valiant, 1994]

[4]see `http://neco.mitpress.org/`

[5]see `http://www.cs.cmu.edu/afs/cs/project/cnbc/nips/NIPS.html`

for new problems that can be solved by variations and extensions of a body of fascinating mathematical tools that we have come to like, and that form the heart of current theoretical computer science. In contrast, to have a serious impact on research in neural computation, a theoretical researcher has to become also "problem-driven", i.e., we have to employ and develop those mathematical concepts and tools that are most adequate for the problem at hand.

One of the main obstacles for a theoretical computer scientist who is ready to tackle theoretical problems about computing and learning in biological neural systems is the diversity of models, and the diversity of opinions among leading neuroscientists regarding the right way to understand computations in the brain.[6] This concerns especially the first questions that a theoretical computer scientist is likely to ask:

- How is information encoded in biological neural systems?
- What are the computational units of biological neural systems,
  and what functions can they compute?
- What is signal and what is noise in neural computation?
- How is learning in neural systems organized and implemented?

Yet another difficulty arises from the fact that neuroscientists employ different models for different levels of detail, starting from models for the whole brain, going down to neural columns and circuits, further down to neurons and synapses, and finally down to the molecular "switches", e.g. the diverse variety of ion – channels and receptors that control the flow of charged particles in neurons and synapses. Unfortunately it is still not known, which of these levels is the right one for understanding neural computation and learning. It is quite possible that none of these levels is "the right one" and that the interplay of several of these levels of modeling is essential. On the other hand this family of models with different levels of temporal and spatial resolution provides a rich source of interesting research problems for theoretical computer scientists:

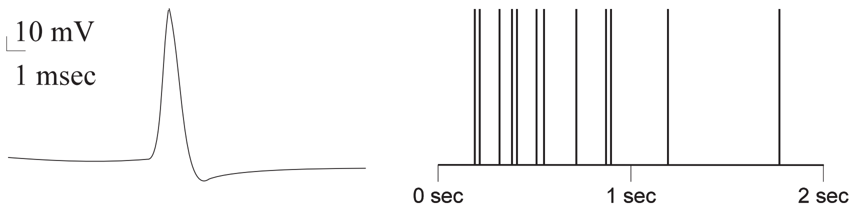
- How are these models related?
- Which computation-related phenomena in large scale models can be caused by coordinated activities of the components of more detailed models?[7]

2 What is Different in Neural Computation in Comparison with Computation in a Digital Computer?

The short answer to this question is: basically everything. More specifically one can point to three basic aspects where the structure of neural computation differs from the structure

[6] This is a consequence of the fact that most of the basic questions about computing and learning in living organisms cannot be answered directly through suitable experiments. Usually one has to rely on indirect empirical evidence, that often depends on details of the experimental setup, the specific neural system and species that is studied, and the methods for data-analysis that are employed.

[7] An example of such investigation is for example [Maass and Natschläger, 2000].



(a) (b)

Figure 1: *a) Typical action potential (spike). b) A typical spike train produced by a neuron (each firing time marked by a bar)*

of computation in present-day computers:

- the role of time
- the role of space
- the role of the program.

One may argue that these three are among the most important aspects of a computation. Hence a computer scientists working on neural computation should be ready to revise fundamental methods and assumptions of his discipline. Various approaches got stuck because they were too timid in this regard. For example threshold circuits (and more generally: most common artificial neural network models) employ computational units that have some aspects in common with biological neurons, but they handle time in the traditional way of computer science (assuming a global clock). Therefore neuroscientists have a hard time relating these models to biological neural systems. There are a few basic facts to which one can point in order to highlight the different role that *time* plays in biological neural systems. Common biological neural systems have no central clock that marks those points in time when computational units should change their output, and when signals are to be read. Rather, the temporal dynamics of neural systems is input driven and salient aspects of the input and internal state representations are *encoded* in the *temporal relationships between different computational events.* Furthermore inputs and outputs of neural systems frequently consist of time series of analog values, rather than of a single batch of numbers or bits like in the common computational models of theoretical computer science.

A closer look at the computational units of neural systems immediately reveals their particular capability for carrying out computations on time series. The *output* of a biological neuron consists of "action potentials" or *"spikes"* (see Fig. 1). More precisely: a *biological neuron* does not output any number or bit, instead it *marks points in time.* The *input* to a biological neuron v consists of trains of pulses, socalled excitatory postsynaptic potentials (EPSP's) and inhibitory postsynaptic potentials (IPSP's) of a shape as indicated in Fig. 2. More precisely: About 1000 to 10000 other neurons u are each connected to v by a *synapse.* The synapse from neuron u to neuron v transforms the output spike train of neuron u (which is of a type as illustrated in Fig. 1 b)) into a train of EPSP's or IPSP's in neuron v. The neuron v "fires" and emits a spike whenever the current value of the

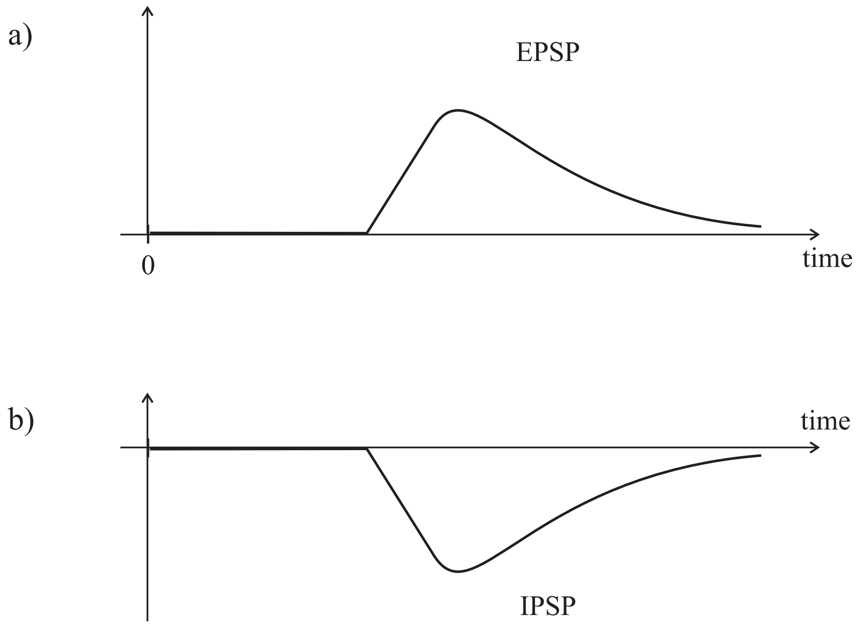


Figure 2: *a) Typical time course of an excitatory postsynaptic potential (EPSP). b) Typical time course of an inhibitory postsynaptic potential (IPSP). The vertical axis denotes the membrane voltage of the neuron v*

sum of EPSP's and IPSP's from all presynaptic neurons u exceeds a certain threshold. A simulator of a small circuit of spiking neurons (which requires no background knowledge) is available from the homepage of our institute[8]. This simulator gives already an idea of the complicated changes of the dynamics of a neural circuit that result from changes in the synaptic "weights". A glimpse at the temporal dynamics of a *real* neural system is provided in Fig. 3.

The understanding of the temporal dynamics of neural computation is further complicated by the fact that the "weight" of a biological synapse (i.e. the amplitude of the resulting EPSP or IPSP) changes from spike to spike, even if there is currently no "learning" involved (see [Maass and Zador, 1999] for a survey and [Maass and Sontag, 1999] for mathematical models). Hence biological synapses resemble more finite automata, rather than just multiplicative factors (the synaptic "weights"). In other words: synapses represent besides neurons a second type of computational unit in neural systems, that are specialized for processing time series. This biological insight has drastic consequences for understanding learning in neural systems, since it has become quite likely that learning should be viewed on the synaptic level as changes in the transition functions of the finite automata that encode the dynamic behavior of the synapses involved (see [Natschläger et al., 2000] for computer simulations). We refer to [Maass and Bishop, 1999] for further information on the role of temporal dynamics in neural computation and first attempts to employ similar mechanisms in artificial computing machinery. There exists substantial room for new ideas from computer science for exploiting the timing of computation steps as a resource for coding, communication and computation.

[8] see DEMOS on `http://www.tu-graz.ac.at/igi/`

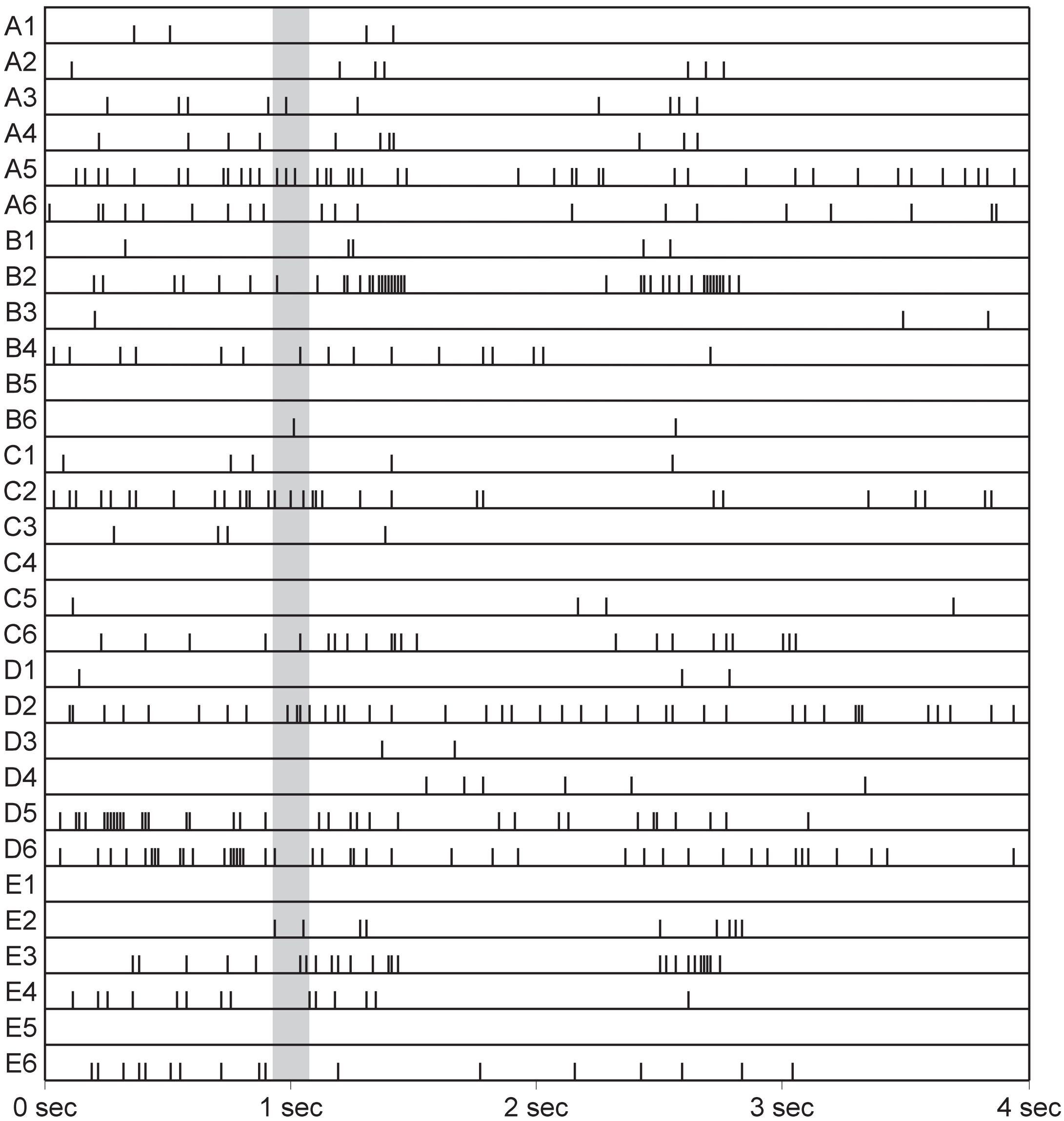


Figure 3: *Simultaneous recordings (over 4 seconds) of the firing times of 30 neurons from monkey striate cortex by Krüger and Aiple [Krüger and Aiple, 1988]. Each firing is denoted by a vertical bar, with a separate row for each neuron. For comparison we have shaded an interval of 150 msec. This time span is known to suffice for the completion of some very complex computations, such as the recognition of a face, that involve several cortical areas.*

The special role of *space* in neural computation becomes clear if one takes into account that the total length of wires is one of the most severe bottlenecks for neural circuits (see [Braitenberg and Schuez,1998, Chklovskii and Stevens, 1998]). From the point of view of theoretical computer science this is quite fortunate, since the known upper bounds for the total length of wires in neural circuits severely restrict the types of circuit architectures that are biologically realistic. For example most threshold circuits and artificial neural networks require full connectivity between two layers of (at least) linear size. But this connectivity structure would require too much total wire length, so that this fact alone makes them already unrealistic as models for commonly studied neural systems such as primary visual cortex. Thus we arrive at the challenging problem to come up with alternative circuit architectures that make more efficient use of wire length by employing more sophisticated spatial arrangements of computational units and data. This topic had already been investigated in theoretical computer science quite a while ago in the context of VLSI-design. The precise complexity constraints on wire length are somewhat different for neural circuits, since here the 3rd dimension can be used more extensively for the routing of wires. We refer to [Legenstein and Maass, 2000] for a precise formulation of total wire length as a complexity measure for neural circuits, and first results on circuit architectures that appear to be realistic from this point of view. Such top-down approach towards finding realistic design strategies for neural circuits is a rich source of research problems where methods from theoretical computer science can be applied. Another attractive goal is understanding the computational role of recurrent connections in neural circuits.

The *program* of a neural circuit consists of those aspects of its architecture that are fixed in the genetic code and of the results of *learning*. A straightforward calculation shows that only a relatively small portion of the parameters that are needed to describe a neural circuit can possibly be contained in the genetic code. Consequently one can understand neural circuits only if one also manages to understand the learning algorithms that shape and maintain them. Unfortunately empirical data on the learning mechanisms employed by neural systems are quite complex. For example, a few years ago it was discovered that the relative timing of the firing of the pre- and postsynaptic neuron is essential for the "learning" of a synapse [Markram et al., 1997], thereby placing many earlier learning theories based on "static" learning rules in jeopardy. We can expect to see during the next few years more empirical data regarding the way in which the different parameters that control the inherent dynamics of a synapse (or in the language of computer science: the transition function of the finite automaton implemented by the synapse) change during learning. It is widely believed by neuroscientists that learning is implemented simultaneously by a variety of different mechanisms that act on different time scales. A challenging goal for a theoretical computer scientist is to provide plausible models for the global organization of these learning mechanisms. Such models are urgently needed, but they can hardly be inferred directly from empirical data. Here it will be necessary to "guess" reasonable models, and to look for predictions of such models which can be tested empirically.

3 Outlook

There exist a few quite readable survey articles and books that make the problems of neural computation accessible to a theoretical computer scientist: [Churchland and Sejnowski, 1992, Arbib, 1995, Ballard, 1997, Koch, 1998, Maass and Bishop, 1999]. [9]

Concurrently with the investigation of neural computation in living organisms one has started to design electronic hardware that captures particular aspects of the spe-

[9] For online information we refer to `http://www.hirnforschung.net/cneuro/`, `http://neuro.med.cornell.edu/`, `http://tcw2.ppsw.rug.nl/~tjeerd/genesis/beeman/cnslecs.html`.

cial role that time and space play in biological neural computation (see [Mead, 1989] and [Murray, 1999]). Examples are artificial retinas [Mead, 1989], schemes for low power analog communication between chips via pulses (address-event-representation, see [Douglas et al., 1999, Mortara and Venier, 1999]) and cellular neural networks [Roska, 1997]. This approach is sometimes referred to as *Neuromorphic Engineering.*[10] Obviously this area is still at a very early stage, and one might hope that theoretical computer science will play a role in its future development.

Within the concert of different scientific disciplines that investigate neural computation theoretical computer science has the chance to provide abstract models that capture essential aspects of biological neural systems in a simplified mathematical framework, thereby providing a platform for extracting "portable" computational mechanisms and principles that can potentially be transported to novel *artificial* computing machinery. In spite of the important contributions made by theoretical physicists, cognitive scientists and experts for information theory one can easily detect the specific shortcomings of these approaches in the current state of theoretical knowledge about neural computation: Statistical physics provides wonderful tools for modeling large homogeneous systems, but it is less suitable for analyzing computations in specific circuits made up of diverse units. Approaches from cognitive science often neglect to ask how the complexity of a proposed circuit scales up with the input size. Approaches from information theory are good at analyzing where and how information about an input is represented in neural circuits, but it is not clear whether they can also be used for analyzing efficient computations (whose goal is to produce an output, rather than preserving the input).

These observation show that there is still a need for contributions to neural computation that make use of specific strengths of theoretical computer science, such as expertise in the design and comparison of computational models, the investigation of the computational power of specific computational models, and the analysis of the computational complexity of specific computational tasks. On the other hand such contributions to neural computation require a strong effort towards interdisciplinary collaboration. There are very few problems arising from neural computation on which a theoretical computer scientist can start to work without further interaction. In most cases an ongoing interchange with experimental neuroscientists (and/or experts for neuromorphic engineering) is necessary in order to avoid that one focuses on questions or parameter ranges that are less relevant.

If the theoretical computer science community decides that it wants to include neural computation among its research topics, some organized efforts appear to be necessary. Beneficial short term measures would be the inclusion of tutorials on neural computation in conferences for theoretical computer science, as well as the organization of informal workshops jointly with neuroscientists. One of the primary goals of such workshops should be the identification of specific topics in neural computation and learning where contributions from theoretical computer science can be expected to be of value. The next step would be the formation of research networks where research on these specific topics can be carried out by theoretical computer scientists in collaboration with neuroscientists and experts for neuromorphic engineering.

[10] see `http://www.ini.unizh.ch:80/telluride99/` for a sample of research topics and [Smith, 1998] for the Proceedings of the first European Workshop on Neuromorphic Engineering.

References

[Arbib, 1995] Arbib, M. A., editor (1995). *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge. See `http://www.dsclab.ece.ntua.gr/~kblekas/brain-theo.html`.

[Ballard, 1997] Ballard, D. H. (1997). *An Introduction to Natural Computation.* MIT-Press.

[Bertsekas and Tsitsiklis, 1996] Bertsekas, D. P., Tsitsiklis, J. N., (1996). *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA.

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

[Chklovskii and Stevens, 1998] Chklovskii, D. B., Stevens, C. F., (1998). Wiring the brain optimally; see `http://www.sloan.salk.edu/~mitya/`.

[Churchland and Sejnowski, 1992] Churchland, P. and Sejnowski, T. (1992). *The Computational Brain.* MIT Press, Cambridge.

[Douglas et al., 1999] Douglas, R. J., Deiss, S. R., and Whatley, A. M. (1999). A pulse-coded communications infrastructure for neuromorphic systems. In Maass, W. and Bishop, C., editors, *Pulsed Neural Networks.* MIT-Press, Cambridge.

[Hopcroft and Ullman, 1979] Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to automata theory, languages and computation.* Addison-Wesley, Reading Mas.

[Kleene, 1956] Kleene, S. C. (1956). Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–42. Princeton University Press, Princeton N.J.

[Koch, 1998] Koch, C. (1998). *Biophysics of Computation: Information Processing in Single Neurons.* Oxford University Press, Oxford.

[Krüger and Aiple, 1988] Krüger, J. and Aiple, F. (1988). Multielectrode investigation of monkey stritate cortex: Spike train correlations in the infragranular layers. *Neurophysiology*, 60:798–828.

[Legenstein and Maass, 2000] Legenstein, R. A., and Maass, W. (2000). Foundations for a circuit complexity theory of sensory processing. Appears in: *Advances in Neural Information Processing Systems 2000 (NIPS '2000)*, vol. 13, MIT Press, Cambridge, 2001. Online: #122 on `http://www.tu-graz.ac.at/igi/maass/publications.html`.

[Maass and Bishop, 1999] Maass, W. and Bishop, C., editors (1998). *Pulsed Neural Networks.* MIT-Press, Cambridge. See `http://www.tu-graz.ac.at/igi/maass/PNN.html`.

[Maass and Natschläger, 2000] Maass, W. and Natschläger, T. (2000). A model for fast analog computation based on unreliable synapses, *Neural Computation* 12(7), 1679-1704. Online: #102 on `http://www.tu-graz.ac.at/igi/maass/publications.html`.

[Maass and Sontag, 1999] Maass, W. and Sontag, E.D. 1999 Neural systems as nonlinear filters. *Neural Computation* in press. Online: #107 on `http://www.tu-graz.ac.at/igi/maass/PNN.html`.

[Maass and Zador, 1999] Maass, W. and Zador, A. (1999). Computing and learning with dynamic synapses, pages 321–336, in: Maass, W. and Bishop, C., editors, *Pulsed Neural Networks*. MIT-Press, Cambridge. Online: #101 on `http://www.tu-graz.ac.at/igi/maass/publications.html`.

[Markram et al., 1997] Markram, H., Lubke, J., Frotscher, M., and Sakmann, B., (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275:213–215.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophysics*, 5:115–133.

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

[Mortara and Venier, 1999] Mortara, A. and Venier, P. (1999). Analog VLSI pulsed networks for perceptive processing. In Maass, W. and Bishop, C., editors, *Pulsed Neural Networks*. MIT-Press, Cambridge.

[Murray, 1999] Murray, A. F. (1998). Pulse-based computation in VLSI neural networks. In Maass, W. and Bishop, C., editors, *Pulsed Neural Networks*. MIT-Press, Cambridge.

[Natschläger et al., 2000] Natschläger, T., Maass, W., Sontag, E. D., Zador, A. M., (2000). Processing of time series by neural circuits with biologically realistic synaptic dynamics. Appears in: *Advances in Neural Information Processing Systems 2000 (NIPS '2000)*, vol. 13, MIT Press, Cambridge, 2001. Online: #111 on `http://www.tu-graz.ac.at/igi/maass/publications.html`.

[Roska, 1997] Roska, T. (1997). Implementation of cnn computing technology. In W. Gerstner, A. Germond, M. H. and Nicoud, J.-D., editors, *Proc. of ICANN 1997*, pages 1151–1155. Springer Verlag, Berlin.

[Roychowdhury et al., 1994] Roychowdhury, V. P., Siu, K. Y., Orlitsky, A., editors. (1994). *Theoretical Advances in Neural Computation and Learning.* Kluwer Academic Publishers, Boston.

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

[Smith, 1998] Smith, L. (1998). *Neuromorphic Systems: Engineering Silicon from Neurobiology.* World Scientific.

[Sutton and Barto, 1998] Sutton, R. S., Barto, A. G., (1998). *Reinforcement Learning.* MIT Press, Cambridge.

[Valiant, 1984] Valiant, L. G., (1984). A theory of the learnable. *Comm. of the ACM*, vol. 27, 1134–1142.

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

[von Neumann, 1958] von Neumann, J. (1958). *The Computer and the Brain.* Yale University Press, New Haven.