

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from <http://www.i6doc.com/fr/livre/?GCOI=28001100432440>.

# Spiking Neural Networks: Principles and Challenges

André Grüning<sup>1</sup> and Sander M. Bohte<sup>2</sup>

<sup>1</sup> University of Surrey, United Kingdom

<sup>2</sup> CWI, Amsterdam, The Netherlands

**Abstract.** Over the last decade, various spiking neural network models have been proposed, along with a similarly increasing interest in spiking models of computation in computational neuroscience. The aim of this tutorial paper is to outline some of the common ground in state-of-the-art spiking neural networks as well as open challenges.

## 1 Introduction

Artificial neural networks are modeled after the computational principles of the brain, with the specific aim of understanding and replicating human abilities [2]. Such neural networks are being used for many machine learning tasks such as function approximation and pattern recognition see for example [23, 35, 32]. The field of artificial *spiking* neural networks is an attempt to emphasize the neurobiological aspects of artificial neural computation.

Real biological neurons communicate with each other using electrical pulses, “spikes”. In the 1990s an increasing number of studies showed that real neurons were able to produce precisely timed spikes (examples are [8, 21]) and a synaptic learning rule sensitive to the precise relative spike-timing was found, Spike-Timing-Dependent Plasticity (STDP) [39, 31]. This influenced Maass’ to write his influential paper [38] on “the third generation of neural networks” establishing that the intricate workings of spiking neurons could support the computations required for general function approximation, just like standard artificial neural networks based on analog (rate) neurons. For spiking neural networks, the question *how can models of spiking neurons carry out computations*, is often accompanied by: *does this explain how real biological neurons compute?*

In this paper, we aim to give an overview of the field of artificial neural networks that use spiking neurons, outlining the interwoven factors of spiking neuron models, neural codes and models of learning in such networks.

## 2 Spiking Neuron Models

The general process of spiking signal transmission is well known, and illustrated in Figure 1: action potentials travel along axons and activate synapses. These synapses release a neurotransmitter that quickly diffuses to the post-synaptic neuron. In the post-synaptic neuron, these neurotransmitters affect the neuron’s membrane potential. Excitatory Postsynaptic Potentials (EPSPs) increase the membrane potential (depolarize), and without new inputs, this excitation then

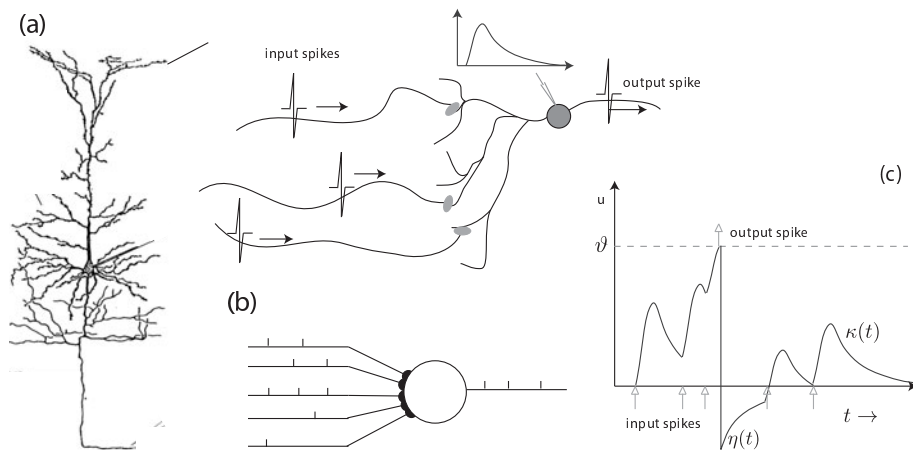


Figure 1: Spiking neurons: real neurons communicate with each other via sequences of pulses – spikes. (a) Dendritic tree, axon and cell body of a neuron. (b) Top: Schematically, spikes arrive from other neurons at the synapses of the postsynaptic neuron. Its membrane potential rises quickly with each incoming spike, and then slowly decays again (inset). If several spikes arrive in a short time window, the membrane potential crosses a threshold, and a spike is fired down the axon. Bottom: Schematically, incoming spikes on various dendrites elicit timed spikes responses as the output. (c) Schematic response of the membrane potential to incoming spikes. If the threshold  $\theta$  is crossed, the membrane potential is reset to a low value, and a spike is fired.

leaks away with a typical time constant. Inhibitory Postsynaptic Potentials (IPSPs) decrease the membrane potential (hyperpolarization). When sufficient EPSPs arrive at a neuron, the membrane potential may depolarise enough to reach a certain threshold, and the neuron generates a spike itself, resetting its membrane potential. The thus generated spike then travels on to other neurons.

The above is a stark simplification, and real neurons display many different spiking behaviours: some respond to input only after a delay, others respond with a burst of spikes. To explain these behaviors, detailed models were developed, with the Hodgkin-Huxley model the most famous [27]. The models are typically phrased as dynamical systems of various complexity, and include models like the Leaky-Integrate-and-Fire (LIF) model, the Quadratic-Integrate-and-Fire model and more complicated models [20, 29], representing different trade-offs between the neuroscientific realism and computational complexity.

An alternative to dynamical systems models are so called Spike Response Models (SRMs) [20]. In SRMs, the membrane potential is not computed through differential equations but as a sum of integral kernels. In principle, SRMs are equivalent to many classes of dynamical systems models, but offer a different insight into the processes in spiking neurons. In this view, a neuron (understood to include its incoming synapses) acts mathematically as *filter* or *operator* in

that it maps incoming sets of spike trains into an output spike train.

Both synapse and dendrite are known to actively contribute to neural computation, but, compared to spiking neurons, much less is known. To capture such extended neural computation, spiking neuron models would have to be complemented by more detailed models of synapses and dendrites.

### 3 Spike-based Neural Codes

The aim of artificial spiking neural networks is to carry out neural computation. This requires that meaning is given to neural spiking: the quantities relevant to the computation have to be expressed in terms of the spikes that spiking neurons communicate with. The challenge is that the nature of *the* neural code (or neural codes) is an unresolved topic of research in neuroscience. However, based on what is known from biology, a number of neural information encodings have been proposed:

**Binary coding** is an all or nothing encoding: a neuron is either active in a given time interval, that is it fires one or several spikes in that time interval, or it is silent. This encoding is motivated from the observation that physiological neurons tend to fire when they receive input (a sensory stimulus, such a light, or external electrical stimuli). This binary abstraction can be used at the level of individual neurons: neurons are modeled as binary units that can only take two values *on/off*. This model was used already in early neural network models, like the ADALINE networks [53]. It can also be used on the level of interpreting spike trains from modern spiking neural network where a binary interpretation of the output spike trains is used in a classification of spike train inputs. As such, binary coding has been used in modern implementations of spiking neural networks [26, 50]. In general, binary coding is appealing because of its simplicity, but it ignores the timed nature and multiplicity of spikes altogether.

**Rate coding** is another abstraction from the timed nature of spikes, in that only the *rate* of spikes in a interval is used as a measure for the information conveyed. Rate encoding is motivated by the observation that physiological neurons tend to fire more often for stronger (sensory or artificial) stimuli. It can again be applied at the level of the single neuron, or at the interpretation of spike trains. In the first case neurons are modeled directly a rate neurons, that transfer at each time step real-valued input numbers –“rates”– into an output “rate”. Rate coding has been the notion behind standard artificial ‘sigmoidal’ neurons in technical contexts and cognitive science. For the interpretation of spike trains, a rate encoding (also called frequency encoding) has also been used to interpret the outputs of spiking neural networks, see for example [51, 52, 22].

**Latency coding** makes use of the timing of spikes, but not of the multiplicity of spikes. Information is encoded as the latency from a certain (internal or external) event until the first spike. This is motivated by the observation that

important sensory stimuli tend to elicit a spike earlier in upstream neurons. This encoding has for example been used in unsupervised learning [43], and supervised learning methods like SpikeProp [6] or the Chronotron [16] and others [51]. Closely related is *rank-order coding*, where information about a stimulus is encoded in the order in which neurons within a group emit their first spikes [49].

**Fully temporal codes** All encodings mentioned so far are special cases of a fully temporal code. In a fully temporal code, the encoding relies on the precise timing of all spikes. They are inspired by evidence in neuroscience that spike-timing can be remarkably precise and reproducible [8, 21].

In a fully temporal code, timings are relevant to a certain given (internal or external) event (such as the onset of a stimulus or spike of a reference neuron). For example, a spike-train with three spikes at 10ms, 17ms and 21ms (relative to e.g. stimulus onset) should have a different meaning than a spike-train with three spikes at 8ms, 17ms and 27ms or a spike train with five spikes at 8ms, 10ms, 17ms, 21ms and 27ms etc. If spike-trains with fixed mutual timings are distributed across a group of neurons, these patterns are referred to as a *polychronous* group [30].

**Predictive spike-coding** is a special case of a fully temporal encoding. Here, the notion is that the spiking mechanism effectively provides a means to carry out an analog-to-digital and digital-to-analog conversion at the soma and the synapses of the neuron, respectively. Examples include greedy solutions like [34, 12, 7, 3, 40, 9], where spike-generation at the soma is associated with the subtraction of a temporal kernel from in the incoming current; at the synapse each spike then causes an EPSP in the target neuron such that the sum of EPSPs approximates the computed signal at the soma. Adaptation mechanisms can change the EPSP, enabling the spiking neuron to adjust to changes in the dynamic range of the inputs [5]. The adaptable dynamics of spiking neurons enables them to compute over multiple timescales [7, 4].

**Probabilistic spike-coding** is concerned with efficiently carrying out inference using spiking neurons, and is mostly considered in a computational neuroscience context. There are two prominent flavours of spiking neural computation here: one where spiking neurons are considered that stochastically generate spikes in response to a given input [37, 1], and one where spiking neurons are essentially deterministic [3]. In both cases, an implicit measure of the inference task is computed, the first in a rate code and the latter in a predictive spike-code.

## 4 Spike-based Learning

Neural codes define the relationship between information and spiking patterns. The challenge remains to carry out useful computations with spiking neural networks. This requires learning methods to adapt weights in the network, and also methods for setting up useful network topologies. As for rate-based neural

networks, for spiking neural networks we can distinguish between *unsupervised learning*, and *supervised learning*:

**Unsupervised Learning** Here data is provided without label, and there is no feedback to the network about its performance. The typical task is to detect and react to statistical correlations in the data. A typical example of this is Hebbian learning, and its spiking generalisations such as STDP. The detection of correlations can be a goal in itself, but it can also be used subsequently, for example to cluster or classify data. In its “standard form” STDP is understood as a process that strengthens a synaptic weight, if the post-synaptic neurons fires shortly *after* the pre-synaptic neuron has fired, and weakened otherwise [31]. However this standard form of STDP is just one out of many physiological forms of STDP [10].

**Supervised Learning** For supervised learning, data (the input) comes with labels (the targets), and the task of the learning device is to associate (classes of) inputs with the target outputs (a mapping or regression between inputs and outputs). An error signal is calculated between the target and the actual output, and is used to update the weights of the network. Supervised learning means that we can use the targets to update parameters directly, while in *reinforcement learning* we only get a general error signal back (“reward”) that describes how well the system is performing. In practice the distinction between those two forms of supervised learning is flexible [24].

**Learning Algorithms** For rate-code neural networks, Back-Propagation Through Time (BPTT) can train any network topology on general pattern classification, regression or transformation tasks – despite some technical weaknesses. To date however, there is no general-purpose algorithm for spiking neural networks. An important challenge here is the discontinuous nature of spiking. Many of the proposed neural spike-codes are thus not amenable to standard neural network learning methods like gradient descent, or work only in approximation and thus have unknown convergence properties. Various supervised learning algorithms have been developed over the past decade, each with their own limitations, such as network topology, adaptability (e.g. reservoir computing), or limited spike encoding (e.g. SpikeProp). Most also focus on classification tasks rather than more challenging tasks like regression, inference and temporal pattern recognition.

Technically, many learning algorithms use the concept of an *eligibility trace* that captures in what way a given (local) synapse contributes to the (global) network output. Typically a (suggested) weight change computed as in STDP is taken as an eligibility trace. The eligibility trace is then converted into a real weight change depending on a global error or reward signal. This is the concept of *reward modulation*, or *reward maximisation* [17], and such approaches can be found in a number of algorithms [13, 36, 15, 18, 25, 42]. Interestingly, reward-modulated learning can be related to minimizing an error functional

[47, 44] where effectively biological (bottom-up) and goal-driven (top-down) ends of research meet.

## 5 Challenges

In our estimation, and compared to “traditional” artificial neural networks, spiking neural networks face a number of – related – challenges: how to reconcile the discontinuous nature of spiking when most error-functions consider real-value and time-continuous quantities? Solutions like SpikeProp [6] linearize the spike-generation process; this however works only for small learning steps. Other solutions, like in the ReSuMe algorithm [45], smear out spikes to achieve a more continuous computation. An alternative is to consider more complex spiking neuron models that are not discontinuous, like Theta Neurons [41], or to use stochastic neurons with an underlying smooth instantaneous firing intensity [17, 18]. Finally, in the predictive spike-coding framework, spiking is just a way to approximate continuous signals, and learning can in principle be defined on those continuous signals. Other methods rely principally on the information that can be extracted from populations of spiking neurons [33, 28].

By definition computation in spiking neural networks is closely related to the challenge of encoding and decoding with spikes. It is important to note that much of the discussion on rate-vs-spike coding in neuroscience does not apply to spiking neural networks: just like biological neurons, computation in spiking neural networks is carried out with spikes. Rate-coding or spike-time coding in such a framework is just a convenient “label” for what an external observer measures in terms of spike-trains [20]. Labels however can be confusing: for example, Florian [16, 14] pointed out that the work in [46] and [26] have common underlying principles. Similarly, to a neuroscientist, predictive spike-codes would qualify as “rate-codes” [20] even though precise spike-times matter for the computation.

While for rate neural networks, temporal dynamics are explicitly induced through recurrent connections and iterative computation of neural activations, an underappreciated feature of spiking neural networks is the inherent notion of time implied by the temporal extension of spike-trains. Exploitation of this temporal feature again depends on the neural coding. For temporal computation, spiking neurons may offer better computational complexity as compared to traditional neural networks: if a spiking neuron substitutes a sigmoidal neuron, communication is sparse and reduced to computing just the activated connections, rather than all connections at every timestep. With both intrinsically temporal dynamics and potential computational efficiency, the venue of temporal computation seems the most promising direction for applications of spiking neural networks, which at this point are relatively lacking.

Another challenge is that of relating sensible models of spiking neural computation to biological spiking neurons. If spiking neural networks are really more powerful computationally than rate neural networks, it seems reasonable to expect beneficial interaction between “computing” driven approaches from

computer science and “data” driven neuroscience. Neuroscience for example informs us about spiking neuron models, whereas efficient spiking neural networks may elucidate on the computational rationale behind such phenomena as STDP and the neural code.

From the preceding, it is clear, that many learning algorithms for spiking neural networks have been developed, and much more work is still needed. We would like to highlight three areas that require further work apart from the neuroscientific challenges mentioned above.

**Comparability.** Due to the differences in neural models, network topology (single layer, multiple layer, recurrent or not etc) and encoding that different learning algorithms require, it is difficult to compare the performance in different spiking neural networks fairly. Hence tasks that *can* in principle be learnt by one algorithm are an unfair challenge for an other and vice versa. This leads over to the next principle challenge:

**Generality.** None of the currently suggested algorithms is general-purpose: Not only are different algorithms suitable only for specific network topologies or neuron models, there is no algorithm either, that is *general purpose*, and able at least in principle to learn an arbitrary task, pattern mapping or classification in the way that backpropagation (through time) and its variants (with all their limitations) do for rate neurons.

**Formulation.** To date no fully satisfactory general-purpose learning algorithm exists for networks of spiking neurons. Most of these approaches stand isolated and do not discuss how they compare or integrate with other similar approaches. This is on the one hand due to different network topologies, neuron models, neural codes and error functions as discussed. However comparison and analysis are also difficult because there is no common structured language to talk about these schemes. Formulation of equations for such algorithms is difficult as the inherent temporal dimension of spike trains introduces a notational load. There is therefore a need for a unified framework for formulating learning algorithms in spiking neural networks in continuous time.

## 6 Advances in Spiking Neural Information Processing

The papers presented in the special session *Advances in Spiking Neural Information Processing Systems* demonstrate novel ideas relating to the challenges outlined above.

In the work presented by Taherkhani et al. [48], a new biologically plausible supervised learning method for spiking neurons is developed. Based on recent insight in the precise details of neural plasticity induced by STDP, experimental results show that the proposed method can effectively map a spatio-temporal input pattern to a target output spike train with a much faster learning speed than previous methods like ReSuMe [45].

Gardner and Grüning [19] quantify the performance of a stochastic neuron model in learning to classify input patterns by precise target responses as outputs, and compare its performance against other learning rules. Performance comparable to alternative neuron models is shown. Importantly, classification of inputs by multiple-spike timings as opposed to a single spike or just a few is demonstrated to both increase performance and reliability of classifications.

Daucé [11] describes how biologically observed spike-based learning, notably STDP, when applied in a large network of spiking neurons causes the network to become sensitive and amplify low levels of periodicity in the input. These results suggest a novel emergent role for STDP in spiking neural networks.

Zambrano et al. [54] demonstrate how adaptive predictive coding enables spiking neurons to efficiently approximate a time-continuous signal, such that a biologically plausible neural network can learn non-linear input-action mappings using reinforcement learning.

## References

- [1] J. Beck, K. Heller, et al. Complex inference in neural circuits with probabilistic population codes and topic models. In *Advances in Neural Inf. Processing Syst. (NIPS) 25*, pp. 3068–3076. 2012.
- [2] C. Bishop. *Neural networks for pattern recognition*. Oxford University Press, USA, 1995.
- [3] M. Boerlin and S. Denève. Spike-based population coding and working memory. *PLoS Comp. Biol.*, 7(2):e1001080, 2011.
- [4] S. Bohte. Error-backpropagation in Fractional Spiking Neural Networks. In *ICANN*, pp. 60–68. Springer, 2011.
- [5] —. Efficient spike-coding with multiplicative adaptation in a spike response model. In *NIPS 25*, pp. 1844–1852. 2012.
- [6] S. Bohte, J. Kok, et al. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48:17–38, 2002.
- [7] S. Bohte and J. Rombouts. Fractionally Predictive Spiking Neurons. In *NIPS 23*, pp. 253–261. 2010.
- [8] C. Carr and M. Konishi. A circuit for detection of interaural time differences in the brain stem of the barn owl. *J. of Neuroscience*, 10(10):3227–3246, 1990.
- [9] D. Chklovskii and D. Soudry. Neuronal spike generation mechanism as an over-sampling, noise-shaping a-to-d converter. In *NIPS 25*, pp. 512–520. 2012.
- [10] C. Clopath et al. Connectivity reflects coding: a model of voltage-based stdp with homeostasis. *Nature Neurosc.*, 13(3):344–352, 2010.
- [11] E. Daucé. Toward STDP-based population action in large networks of spiking neurons. In M. Verleysen, ed., *Proc ESANN'2014*. D-Facto, 2014.
- [12] T. Delbruck. Frame-free dynamic digital vision. In *Proc. of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, pp. 21–26. 2008.
- [13] M. A. Farries and A. L. Fairhall. Reinforcement learning with modulated spike timing-dependent synaptic plasticity. *J Neurophysiol*, 98:3648–3665, 2007.
- [14] R. Florian. Tempotron-like learning with resume. In V. Kurková, R. Neruda, et al., eds., *Artificial Neural Networks - ICANN 2008*, vol. 5164 of *Lecture Notes in Computer Science*, pp. 368–375. Springer Berlin Heidelberg, 2008.



- [15] R. V. Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Computation*, 19(6):1468–1502, 2007.
- [16] —. The chronotron: A neuron that learns to fire temporally precise spike patterns. *PLoS ONE*, 7(8):e40233, 08 2012.
- [17] N. Fremaux, H. Sprekeler, et al. Functional requirements for reward-modulated spike-timing-dependent plasticity. *The Journal of Neuroscience*, 30(40):13326–13337, 10 2010.
- [18] B. Gardner and A. Grüning. Learning temporally precise spiking patterns through reward modulated spike-timing-dependent plasticity. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science. Springer, September 2013.
- [19] —. Classifying patterns in a spiking neural network. In M. Verleysen, ed., *Proc ESANN'2014*. D-Facto, 2014.
- [20] W. Gerstner and W. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [21] W. Gerstner et al. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–78, 1996.
- [22] C. Glackin, L. Maguire, et al. Repetitive field optimisation and supervision of a fuzzy spiking neural network. *Neural Networks*, 24:247–256, 2011.
- [23] A. Graves, M. Liwicki, et al. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, May 2009.
- [24] A. Grüning. Elman backpropagation as reinforcement for simple recurrent networks. *Neural Computation*, 19(11):3108–3131, 2007. ISSN 0899-7667.
- [25] A. Grüning and I. Sporea. Supervised learning of logical operations in layered spiking neural networks with spike train encoding. *Neural Processing Letters*, 36(2):117–134, 2012.
- [26] R. Güttig and H. Sompolinsky. The tempotron: a neuron that learns spike timing-based decisions. *Nature neuroscience*, 9(3):420–428, 2006.
- [27] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.
- [28] Q. Huys, R. Zemel, et al. Fast population coding. *Neural Computation*, 19(2):404–441, 2007.
- [29] E. Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.
- [30] E. M. Izhikevich. Polychronization: computation with spikes. *Neural computation*, 18(2):245–282, 2006.
- [31] W. Kistler and L. J. van Hemmen. Modeling synaptic plasticity in conjunction with the timing of pre- and postsynaptic action potentials. *Neural Computation*, 12:385–405, 2000.
- [32] A. Krizhevsky, I. Sutskever, et al. Imagenet classification with deep convolutional neural networks. In *NIPS*, vol. 1(2), p. 4. 2012.
- [33] A. Lazar. Population encoding with hodgkin-huxley neurons. *Information Theory, IEEE Transactions on*, 56(2):821–837, 2010.
- [34] A. Lazar and L. Toth. Time encoding and perfect recovery of bandlimited signals. In *Proc. ICASSP'03*, pp. VI–709. 2003.
- [35] Q. Le et al. Building high-level features using large scale unsupervised learning. In *Proc. ICML-12*, pp. 81–88. 2012.

- [36] R. Legenstein, D. Pecevski, et al. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLoS Computational Biology*, 3(10):1–27, October 2008.
- [37] W. Ma, J. Beck, et al. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11):1432–1438, 2006.
- [38] W. Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.
- [39] H. Markram et al. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997.
- [40] O. Marre et al. Mapping a complete neural population in the retina. *The Journal of Neuroscience*, 32(43):14859–14873, 2012.
- [41] S. McKennoch, T. Voegtlin, et al. Spike-timing error backpropagation in theta neuron networks. *Neural computation*, 21(1):9–45, 2009.
- [42] A. Mohemmed, S. Schliebs, et al. Span: Spike pattern association neuron for learning spatio-temporal sequences. *Int. J. Neural Systems*, 2011.
- [43] T. Natschläger and B. Ruf. Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems*, 9(3):319–332, 1998.
- [44] J. Pfister, T. Toyoizumi, et al. Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning. *Neural Comput*, 18:1309–1339, 2006.
- [45] F. Ponulak. Supervised learning in spiking neural networks with resume method. *Phd, Poznan University of Technology*, 46:47, 2006.
- [46] F. Ponulak and A. Kasiński. Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification and spike shifting. *Neural Computation*, 22:467–510, 2010.
- [47] I. Sporea and A. Grüning. Supervised learning in multilayer spiking neural networks. *Neural Computation*, 25(2), 2013.
- [48] A. Taherkhani et al. A new biologically plausible supervised learning method for spiking neurons. In M. Verleysen, ed., *Proc. ESANN'2014. D-Facto*, 2014.
- [49] S. Thorpe and J. Gautrais. Rank order coding. *Computational Neuroscience: Trends in Research*, 13:113–119, 1998.
- [50] R. Urbanczik and W. Senn. A gradient learning rule for the tempotron. *Neural Computation*, 21:340–352, 2009.
- [51] —. Reinforcement learning in populations of spiking neurons. *Nature Neuroscience*, 12:250–252, Feb 2009.
- [52] J. J. Wade, L. J. McDaid, et al. Swat: A spiking neural network training algorithm for classification problems. *IEEE Transactions on Neural Networks*, 21(11):1817–1829, 2010.
- [53] B. Widrow, M. E. Hoff, et al. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, vol. Part 4, pp. 96–104. IRE, New York, 1960.
- [54] D. Zambrano et al. Spiking AGREL. In M. Verleysen, ed., *Proc ESANN 2014. D-Facto*, 2014.