
Litterature summaries

Contents

1 Spiking Neurons	1
1.1 Simple model of spiking neurons [4]	1
2 Spiking Synapses	1
2.1 Spike-timing-dependent plasticity (STDP) [2]	1
2.2 Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning [12]	2
2.3 ReSuMe - new supervised learning method for Spiking Neural Networks [13]	2
2.4 Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks [6]	3
2.5 Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting [15]	3
2.6 Apparatus and methods for gating analog and spiking signals in artificial neural networks [14]	4
2.7 Gradient Descent for Spiking Neural Network [3]	4
2.8 Comparison of supervised learning methods for spike time coding in spiking neural networks [7]	5
2.9 Tempotron-like learning with ReSuMe [1]	5
2.10 Span: Spike pattern association neuron for learning spatio-temporal spike patterns [10]	5
3 Spiking microcircuits	7
3.1 Real-time computing without stable states: A new framework for neural computation based on perturbations [9]	7
3.2 Polychronization: Computation with spikes [5]	8
3.3 Supervised learning in spiking neural networks with FORCE training [11]	8
3.4 Design of the spiking neuron having learning capabilities based on FPGA circuits [8] .	9
4 Speech synthesis	9
4.1 A Neural Model to Predict Parameters for a Generalized Command Response Model of Intonation [16]	9
5 Other	9
5.1 Software for brain network simulations: a comparative study [17]	9

1 Spiking Neurons

1.1 Simple model of spiking neurons [4]

Abstract : A model is presented that reproduces spiking and bursting behavior of known types of cortical neurons. The model combines the biological plausibility of Hodgkin-Huxley-type dynamics and the computational efficiency of integrate-and-fire neurons. Using this model, one can simulate tens of thousands of spiking cortical neurons in real time (1 ms resolution) using a desktop PC.

Equation :

$$\dot{v} = 0.04v^2 + 5v + 140 - u + I \quad (1)$$

$$\dot{u} = a(bv - u) \quad (2)$$

with the auxiliary after-spike resetting

$$\text{if } v \geq 30 \text{ mV, then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (3)$$

v represents the membrane potential of the neuron and u represents a membrane recovery variable. a and b describe respectively the time scale and the sensitivity of the recovery variable u . c and d describes the after-spike reset value respectively of the membrane potential v and the recovery variable u .

Typical values are : $a = 0.02$ $b = 0.2$, $c = -65 \text{ mV}$, $d = 2$

2 Spiking Synapses

2.1 Spike-timing-dependent plasticity (STDP) [2]

The synapse is the biological element present in between two connected neurons called presynaptic and postsynaptic neurons. The synapse regulates the amount of current from the presynaptic neuron that is transmitted to the postsynaptic neuron. The presynaptic current can be both reduced or amplified by the synapse and then injected in the postsynaptic neuron. The synaptic plasticity includes the mechanisms that involve changes of the synaptic weight : the factor from presynaptic current to postsynaptic current. Hebbian learning is a general mathematical formulation of Hebb's postulate based on two particular aspects : the change of the synaptic efficacy depend only on (spatially) local variables and depend on the joint activity of pre- and postsynaptic neurons. Numerous learning rules have been derived from rate-based models of synaptic plasticity. Pair-based models of STDP consider the temporal difference between pre- and postsynaptic spikes for the change in weight of a synapse. If we introduce $S_j = \sum_f \delta(t - t_j^f)$ and $S_i = \sum_f \delta(t - t_i^f)$ for the spike trains of pre- and postsynaptic neurons respectively, the update rule is :

$$\frac{dw_{ij}}{dt} = -A_-(w_{ij})e^{-\frac{|\Delta t|}{\tau_-}} S_j + A_+(w_{ij})e^{-\frac{|\Delta t|}{\tau_+}} S_i \quad (4)$$

where $A_{\pm}(w)$ describes the dependence of the update on the current weight of the synapse, $\Delta t = |t_{post} - t_{pre}|$ the temporal difference and τ_{\pm} the time constant of the exponential decay.

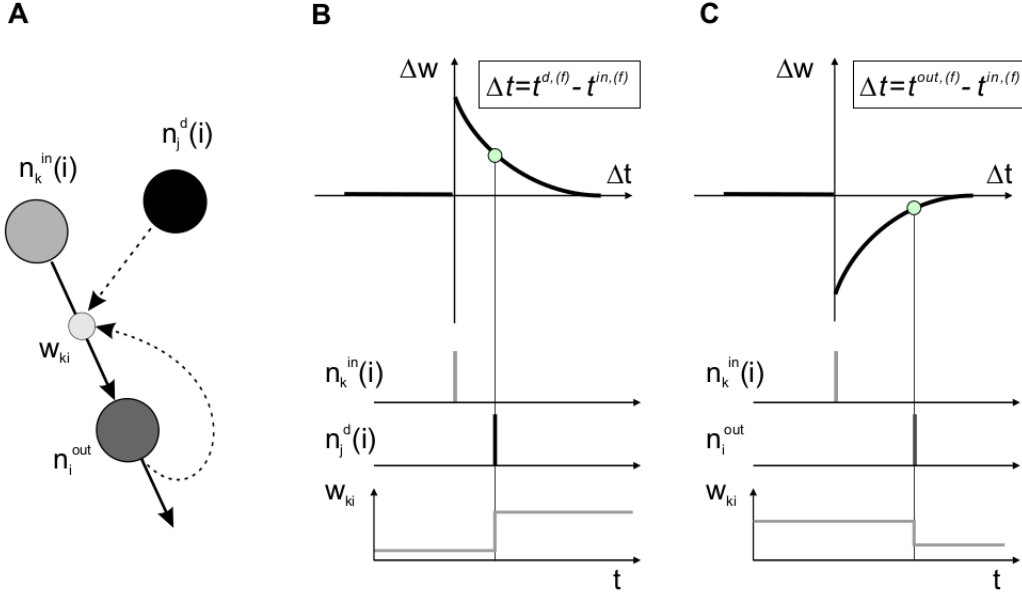


Fig. 1. (A) Basic learning neural structure consists of a presynaptic n_k^{in} and a post-synaptic neuron n_i^{out} as well as a teacher neuron n_j^d . The neurons n_k^{in} and n_i^{out} are connected via synapse w_{ki} (solid lines). The synaptic efficacy is influenced by the firing of the teacher and output neurons (dashed lines). (B),(C) Illustration of the learning rules modifying the synaptic weights. The rules are exponential functions of the time difference Δt between the desired $t^{d,(f)}$ and input $t^{in,(f)}$ spike times (B), as well as between output $t^{out,(f)}$ and input $t^{in,(f)}$ spike times (C).

Figure 1: ReSuMe learning rule

2.2 Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning [12]

Supervised learning paradigm to derive a synaptic update rule that optimizes by gradient ascent the likelihood of postsynaptic firing at one or several desired firing times. Define the probability density of an entire spike train. It allows to describe explicitly the likelihood of emitting a set of spikes. Since the likelihood is a smooth function (density dist.) it is differentiable with respect to any variable. Thus learning by gradient ascent.

Strong mathematical proof. Relation to STDP. Only single spike, or limited to few spikes.

2.3 ReSuMe - new supervised learning method for Spiking Neural Networks [13]

Original paper, presenting ReSuMe learning rule (first time) for movement control. Combined with a Liquid State Machine. See figure 1

2.4 Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks [6]

Experimental demonstration that ReSuMe has the following properties:

1. it can learn temporal sequences of spikes
2. it can learn model object's I/O properties
3. it is scalable
4. it is computationally simple
5. it is fast converging

See figure 1

2.5 Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting [15]

Robust theory describing ReSuMe (Remote Supervised Method) learning rule. **how do neurons learn to reproduce template signals encoded in precisely timed sequence of spikes?**

Supervised learning is believed to be utilized by the neural motor centres to form the internal representation of the body and the environment (cerebellum and cerebellar cortex)

It (ReSuMe) is based on the interaction between two spike-timing dependent plasticity (STDP) processes. It appears that ReSuMe represents a spiking analogy to the classical Widrow-Hoff algorithm proposed for the rate-based neuron model (Widrow and Hoff, 1960; Widrow, 1962)

It has been demonstrated that ReSuMe enables effective learning of complex temporal and spatio-temporal firing patterns with an arbitrary high accuracy

Due to the ability of the method to operate online and due to its fast convergence the method is suitable for real-life applications. [...] SNN trained with ReSuMe become efficient neurocontrollers for movement generation and control

Widrow-Hoff learning rule (weight change Δw_{oi}):

$$\Delta w_{oi} = \alpha x_i (y_d - y_o) \quad (5)$$

α : learning rate, x_i : input signal, y_o : output signal, y_d : teaching signal

ReSuMe learning rule (weight change $\frac{d}{dt}w_{oi}(t)$):

$$\frac{d}{dt}w_{oi}(t) = [S_d(t) - S_o(t)] \left[a_d + \int_0^\infty a_{di}(s) S_i(t-s) ds \right] \quad (6)$$

$S_i(t)$, $S_o(t)$ and $S_d(t)$: presynaptic, postsynaptic and target spike trains. The role of the non-correlative factor (Non-hebbian term) a_d is to adjust the average strength of the synaptic inputs so to impose on a neuron a desired level of activity such that the actual mean firing rate of $S_o(t)$ approaches the mean firing rate of signal $S_d(t)$. Thus the task of setting up the precise timing of spikes, is attributed mainly to the Hebbian term $a_{id}(s)$. The kernel $a_{id}(s)$ define the shape of a learning window:

$$a_{di}(s) = A_{di} \exp\left(-\frac{s}{\tau_{di}}\right) \quad (7)$$

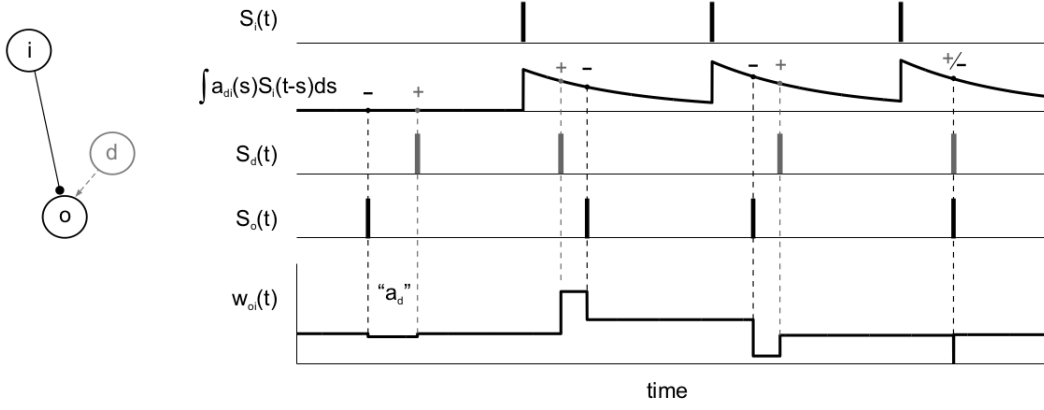


Figure 1: Illustration of the ReSuMe learning rule. For any excitatory synaptic connection from neuron i to neuron o , a synaptic strength w_{oi} is potentiated whenever a target spike ($S_d(t)$) is observed, and depressed whenever the trained neuron fires ($S_o(t)$). Inhibitory synapses are driven by the opposite rule. The amount of the resulting weight change is defined by the non-associative term a_d and by the kernel $\int a_{di}(s) S_i(t-s) ds$. Potentiation and depression compensate each other whenever the trained neuron fires exactly at the target time.

Figure 2: ReSuMe learning rule

1. Learning: sequence of spikes / with various neuron models / in the presence of noise and uncertainty
2. Classification
3. Spike shifting (by setting different parameters)

(The main advantage of VLSI (Maass and Bishop, 1999) systems over processor-based designs lies in their truly parallel mode of operation which fully utilizes the essential property of parallelism found in spiking neural networks.) Advantages: simplicity, scalability, online training suitability, and fast convergence.

See figure 2

2.6 Apparatus and methods for gating analog and spiking signals in artificial neural networks [14]

Universal node (neuron) design implementing a universal learning rule in a mixed (analog and spiking) signal spiking network.

2.7 Gradient Descent for Spiking Neural Network [3]

Introduce a differentiable synaptic model, which is based on the presynaptic membrane potential. The synaptic current is thus activated gradually throughout the active zone (margin below the threshold potential), this allows to compute a gradient. Uses a regularizer. Predictive coding task, auto-encoder task, delayed-memory XOR task.

2.8 Comparison of supervised learning methods for spike time coding in spiking neural networks [7]

See figure 3

2.9 Tempotron-like learning with ReSuMe [1]

description of the tempotron as a special case of ReSuMe learning rule. Classification task : categorize spike trains not only by firing or not, but also by firing given spike trains.

Despite its limitations, the tempotron has been proved quite efficient for spoken digit recognition, outperforming with only 15 spiking neurons complex state-of-the-art HMM word recognition systems (Gutig and Sompolinsky, 2008). This shows that spiking neural networks are quite powerful, and using appropriate learning methods for training them might reveal even more of their potential

2.10 Span: Spike pattern association neuron for learning spatio-temporal spike patterns [10]

Transform spike trains during the learning phase into analog signals (filtered spike trains), based on Widrow-Hoff rule. Compared with ReSuMe and Chronotron.

Some pattern recognition tasks such as the recognition of colors, visual patterns, odors and sound qualities can not be easily solved by rate-based neural models. Temporal information encoding can also reduce the number of neurons that are necessary to perform a given task.

SNN has proved itself adequate for a number of computation and engineering problems. It is considered as a suitable tool to perform temporal pattern recognition and real-time computation

Learning rule Δw_i :

1. with α -kernel: $\tilde{s}(t) = \sum_{t^f} e\tau^{-1}(t - t^f)e^{-\frac{(t-t^f)}{\tau}}H(t - t^f)$

$$\Delta w_i = \lambda\left(\frac{e}{2}\right)^2 \left[\sum_g \sum_f (|t_i^f - t_d^g| + \tau)e^{-\frac{|t_i^f - t_d^g|}{\tau}} - \sum_h \sum_f (|t_i^f - t_a^h| + \tau)e^{-\frac{|t_i^f - t_a^h|}{\tau}} \right] \quad (8)$$

2. with exponential kernel: $\tilde{s}(t - t^f) = H(t - t^f)e^{-\frac{t-t^f}{\tau}}$

$$\delta w_i = \frac{\tau\lambda}{2} \left[\sum_g \sum_f e^{-\frac{|t_d^g - t_i^f|}{\tau}} - \sum_h \sum_f e^{-\frac{|t_a^h - t_i^f|}{\tau}} \right] \quad (9)$$

Illustration of the learning rule on figure 4

The paper explores: Output spike sequence learning, learning with noise, memory capacity (number of patterns memorized depending on the number of synapses), classification.

Table 1. Supervised learning methods for spike time coding in SNN

No	Author/ Method	Approach	Presented Tasks	Coding Scheme	References
1	S. Bohte/ <i>SpikeProp</i>	Gradient estimation	Classification	Time-to-first-spike	(Bohte, 2003; Bohte <i>et al.</i> , 2000; 2002; Moore, 2002; Schrauwen and Van Campenhout, 2004; Xin and Embrechts, 2001)
2	J. Pfister, D. Barber, W. Gerstner	Statistical approach	Spike sequence learning, Classification	Precise spike timing	(Barber, 2003; Pfister <i>et al.</i> , 2003; 2005)
3	A. Carnell, D. Richardson	Linear algebra formalisms	Spike sequence learning	Precise spike timing	(Carnell and Richardson, 2004)
4	A. Belatreche, L. Maguire, T. McGinnity, Q. Wu	Evolutionary strategy	Classification	Time-to-first-spike	(Belatreche <i>et al.</i> , 2003)
5	J. Sogune	Learning in synfire chains	Single temporal interval learning	Relative spike time	(Sogune, 2001)
6	B. Ruf, M. Schmitt	Supervised Hebbian learning	Classification	Time-to-first-spike	(Ruf, 1998; Ruf and Schmitt, 1997)
7	R. Legenstein, Ch. Naeger, W. Maass	Supervised Hebbian learning	Spike sequence learning, Input-output mapping	Precise spike timing	(Legenstein <i>et al.</i> , 2005)
8	F. Ponulak/ <i>ReSuMe</i>	Remote supervision	Spike sequence learning, Input-output mapping, Neuron model independence, Real-life applications	Precise spike timing	(Kasiński and Ponulak, 2005; Ponulak, 2005; Ponulak and Kasiński, 2005)

Figure 3: Comparison of supervised learning methods for SNN

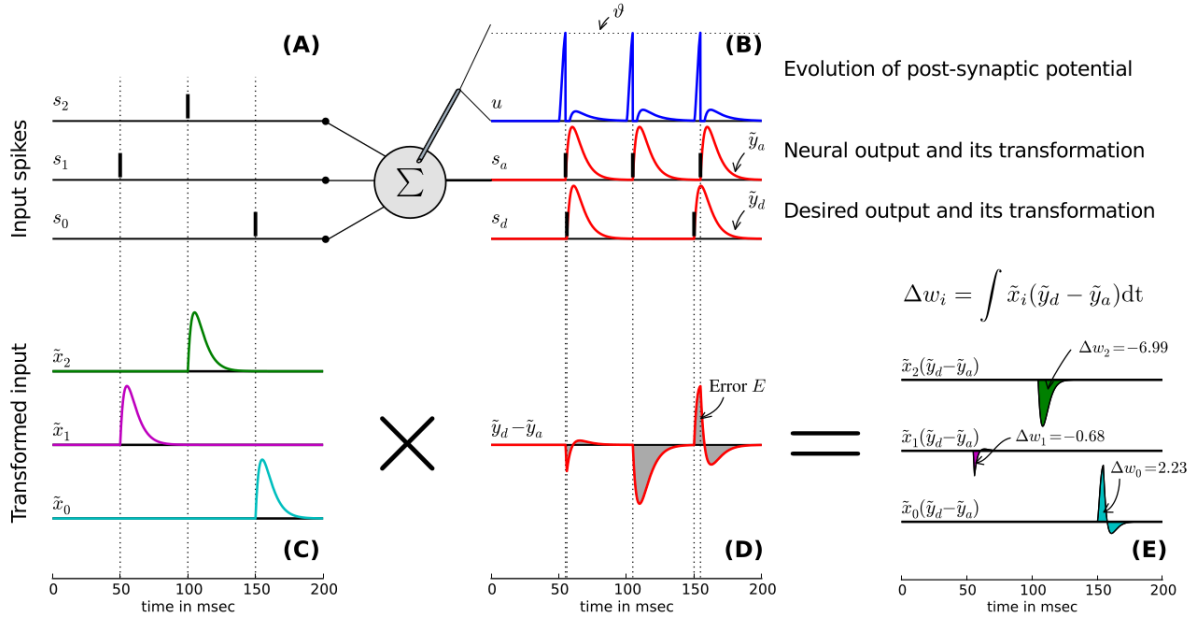


Figure 1: Illustration of the proposed learning rule SPAN. See text for detailed explanations of the figure.

Figure 4: SPAN learning rule

3 Spiking microcircuits

3.1 Real-time computing without stable states: A new framework for neural computation based on perturbations [9]

- alternative to paradigms based on Turing machines or attractor neural networks
- high-dimensional dynamical system (task-independent)
- universal analog fading memory and universal computational power
- readouts neurons can learn to extract in real-time from the current neural circuit information about current and past inputs
- from observation of neocortex's circuits: extremely complex but surprisingly stereotopic microcircuits that can perform a wide spectrum of tasks
- multiple readouts can be trained to perform different tasks on the same state trajectory of a recurrent neural circuit → parallel real-time computing.

The liquid has only one attractor state: the resting state. Perturbed state of the liquid represents present and past inputs → potentially providing the information needed for an analysis of various dynamic aspects of the environment.

1. **Separation property (SP)**: amount of separation between the trajectories of internal states of the system that are caused by two different input streams.
2. **Approximation property (AP)**: resolution and recoding capabilities of the readout mechanisms (capability to distinguish and transform different internal states of the liquid into given target outputs)

The liquid state $x^M(t)$ is simply the current output of some operator or filter L^M (called liquid filter or liquid circuit) that maps input function $u(\cdot)$ onto functions $x^M(t)$:

$$x^M(t) = (L^M u)(t) \quad (10)$$

The second component of an LSM M is a memoryless readout map f^M that transforms, at every time t , the current liquid state $x^M(t)$ into the output:

$$y(t) = f^M(x^M(t)) \quad (11)$$

3.2 Polychronization: Computation with spikes [5]

Spiking neural network with STDP synaptic weights evolution (1000 Izhikevich neurons). It describes the different rhythms observed, the balance of excitation and inhibition, the rate to spike-timing conversion.

The **originality of the paper** is the consideration of a large range of synaptic delays, it enables the emergence of polychronous groups (PG), subsets of interconnected neurons activated by a precise spike timing sequence.

PGs appear in the the context of gamma frequencies. the number of PGs far exceeds the number of neurons in the network, resulting in a large memory capacity of the system.

The paper provides a way to detect PGs from an anatomical point of view.

3.3 Supervised learning in spiking neural networks with FORCE training [11]

Examples include the FORCE method, a novel technique that harnesses chaos to perform computations. We demonstrate the direct applicability of FORCE training to spiking neurons by training networks to mimic various dynamical systems in addition to reproduce more elaborate tasks such as input classification, storing sequences, reproducing the singing behavior of songbirds, and recalling a scene from a movie.

The synaptic weight matrix is the sum of a set of static weights and a set of dynamically varying weights. The time dependent weights, , are learned using a supervised learning method called Recursive Least Mean Squares (RLMS) so that the squared error between the target dynamics (i.e. the task, $x(t)$) and the network dynamics ($\hat{x}(t)$) is minimized. For the dynamics under consideration, the Izhikevich model had the greatest accuracy and fastest training times. This is partially due to the fact that the Izhikevich neuron has the spike frequency adaptation variable (u_i) which operates on a longer time scale (i.e. 100 ms). We set the parameters such that the average firing rate was low (< 60 Hz). To test whether a network of spiking neurons can function as a statistical classifier, we trained networks of Izhikevich neurons to classify inputs into two classes separated by either linear or nonlinear boundaries.

The weight matrix :

$$\omega_{ij} = G\omega_{ij}^0 + Q\eta_i\phi_j^T \quad (12)$$

ω_{ij}^0 the static weight matrix that induces chaos, G chaos factor, Q factor increasing desired recurrent dynamics, η_i encoding variables, ϕ_j decoding variables that are learned with recursive least mean squares:

$$\phi(t) = \phi(t - \Delta t) - e(t)P(t)r(t) \quad (13)$$

$e(t) = \hat{x}(t) - x(t)$ the error, $P(t)$ estimate for the inverse of the correlation matrix (learning rule) and $r(t)$ the firing rate.

3.4 Design of the spiking neuron having learning capabilities based on FPGA circuits [8]

Hardware real-time implementations of Spiking Neuron Networks (SNN) with ReSuMe learning rule.

4 Speech synthesis

4.1 A Neural Model to Predict Parameters for a Generalized Command Response Model of Intonation [16]

Bastian paper. Presents Generalised Command Response (GCR) model, which is extracted from an intonation contour by a matching pursuit algorithm. Constituted by atoms.

GCR also enables analysis of the underlying physiological process (approximate muscle response).

Given that GCR approximate muscle responses to neural spikes, it would make sense to use a Spiking Neural Network to generate these atoms. The generated spikes would be filtered by muscle responses to generate the pitch contour.

We generalize the Spike Pattern Association Neuron (SPAN) algorithm from the literature to construct a loss from GCR atoms, and show that it can be backpropagated to allow the system to generate natural prosody.

5 Other

5.1 Software for brain network simulations: a comparative study [17]

References

- [1] Răzvan V Florian. Tempotron-like learning with resume. In *International Conference on Artificial Neural Networks*, pages 368–375. Springer, 2008.
- [2] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [3] Dongsung Huh and Terrence J Sejnowski. Gradient descent for spiking neural networks. In *Advances in Neural Information Processing Systems*, pages 1440–1450, 2018.

- [4] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.
- [5] Eugene M Izhikevich. Polychronization: computation with spikes. *Neural computation*, 18(2):245–282, 2006.
- [6] Andrzej Kasinski and Filip Ponulak. Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks. In *International Conference on Artificial Neural Networks*, pages 145–152. Springer, 2005.
- [7] Andrzej Kasiński and Filip Ponulak. Comparison of supervised learning methods for spike time coding in spiking neural networks. *International Journal of Applied Mathematics and Computer Science*, 16:101–113, 2006.
- [8] Marek Kraft, Andrzej Kasiński, and Filip Ponulak. Design of the spiking neuron having learning capabilities based on fpga circuits. *IFAC Proceedings Volumes*, 39(17):301–306, 2006.
- [9] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.
- [10] Ammar Mohemmed, Stefan Schliebs, Satoshi Matsuda, and Nikola Kasabov. Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International journal of neural systems*, 22(04):1250012, 2012.
- [11] Wilten Nicola and Claudia Clopath. Supervised learning in spiking neural networks with force training. *arXiv preprint arXiv:1609.02545*, 2016.
- [12] Jean-Pascal Pfister, Taro Toyozumi, David Barber, and Wulfram Gerstner. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural computation*, 18(6):1318–1348, 2006.
- [13] Filip Ponulak. Resume-new supervised learning method for spiking neural networks. *Institute of Control and Information Engineering, Poznan University of Technology*, 42, 2005.
- [14] Filip Ponulak. Apparatus and methods for gating analog and spiking signals in artificial neural networks, December 15 2015. US Patent 9,213,937.
- [15] Filip Ponulak and Andrzej Kasiński. Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural computation*, 22(2):467–510, 2010.
- [16] Bastian Schnell and Philip N Garner. A neural model to predict parameters for a generalized command response model of intonation. *Proc. Interspeech 2018*, pages 3147–3151, 2018.
- [17] Ruben A Tikidji-Hamburyan, Vikram Narayana, Zeki Bozkus, and Tarek A El-Ghazawi. Software for brain network simulations: a comparative study. *Frontiers in neuroinformatics*, 11:46, 2017.