

Improved Spike-Timed Mappings using a Tri-Phasic Spike Timing-Dependent Plasticity Rule

Scott V. Notley
Department of Computing
University of Surrey
Guildford, Surrey, U.K.
Email: S.Notley@surrey.ac.uk

André Grüning
Department of Computing
University of Surrey
Guildford, Surrey, U.K.
Email: A.Gruning@surrey.ac.uk

Abstract—Reservoir computing and the liquid state machine model have received much attention in the literature in recent years. In this paper we investigate using a reservoir composed of a network of spiking neurons, with synaptic delays, whose synapses are allowed to evolve using a tri-phasic spike timing-dependent plasticity (STDP) rule. The networks are trained to produce specific spike trains in response to spatio-temporal input patterns. The results of using a tri-phasic STDP rule on the network properties are compared to those found using the more common exponential form of the rule. It is found that each rule causes the synaptic weights to evolve in significantly different fashions giving rise to different network dynamics. It is also found that the networks evolved with the tri-phasic rule are more capable of mapping input spatio-temporal patterns to the output spike trains.

Index Terms—Reservoir Computing, Liquid State Machine, Spike Timing-Dependent Plasticity, Tri-Phasic, Spiking Neurons.

I. INTRODUCTION

There has recently been an interest in recurrent networks and especially reservoir computing as the solution to complicated computational tasks [1]. The Liquid State Machine (LSM) [2] was introduced as a biologically plausible model of cortical microcircuits that is capable of real-time computations on continuous streams of data, such as spike trains [3], [4], and is a highly recurrent network of spiking neurons coupled with linear-readout neurons. The recurrent network or reservoir is viewed as a generic structure and learning takes place by the supervised training of linear readout neurons with no training taking place in the reservoir.

Since learning only takes place at the linear readout neurons training is relatively simple. However, a more biologically realistic model would also contain some form of learning within the reservoir such as Spike-Timing Dependent Plasticity (STDP). The effects of intrinsic plasticity on the properties of reservoirs have been studied by Steil [5] and Schrauwen *et al* [6]. Triesch [7] has also studied the effects of intrinsic plasticity and its synergies with synaptic plasticity in reservoirs. However, in each of the studies the networks were composed of rate based neurons instead of spiking neurons. Norton and Ventura [8] have presented results investigating spike-timing dependent plasticity (STDP) in reservoirs using spiking neurons and found that in the presence of non-random inputs

there was an improvement in the separation properties of the reservoirs. This work was extended to a learning scheme called separation driven synaptic modification [9] that is related to Hebbian and reinforcement learning, however, the biological plausibility of this method is not clear.

The work of Izhikevich [10], while not motivated from the reservoir paradigm, investigated the effects of STDP on random recurrent networks of spiking neurons. This work introduced synaptic delays to the recurrent network and the concept of time-locked neural firing patterns called polychronous groups. Polychronous groups arise due to the network topology coupled with the ongoing synaptic plasticity and were cited as a possible mechanism for networks to store large amounts of information and to exhibit working memory [11]. The work also demonstrated that polychronous groups may be activated in response to specific spatio-temporal input patterns although there has not been much significant advances reported in the literature exploiting this phenomenon.

In this work we investigate using a network of spiking-neurons with synaptic delays as the reservoir for a LSM. The networks are allowed to evolve using two different STDP rules: the common exponential form and a tri-phasic form (TP-STDP). The performance of the networks is assessed in terms of its ability to produce two different spike-timed sequences in response to two different spatio-temporal inputs. Network properties are also investigated and it is found that the network dynamics and the evolution of the synaptic weights differs dependent on both the initial weights and the STDP rule that is used. The ability of the network to map spatio-temporal input patterns to specific output spike trains is assessed and it is found that the the TP-STDP rule gives improved results over the STDP rules.

II. NETWORK AND TRAINING

A. Architecture

The reservoir for the liquid state machine was based on the network described by Izhikevich [10], implemented as a network of 1000 neurons and fully connect to a single read-out neuron. The neurons of the reservoir were randomly connected with 100 efferent connections from each pre-synaptic neuron to 100 post-synaptic neurons giving a connection probability

of 0.1. Each neuron was defined as either inhibitory or excitatory with a ratio of 80% excitatory to 20% inhibitory as found in the mammalian cortex [12]. Each neuron was modelled using Izhikevich neurons [13] [14] with excitatory neurons being of the regular spiking type and inhibitory neurons being of the fast spiking type. Only efferent synapses from excitatory neurons are plastic and the weight allowed to evolve according to the STDP rule used (weights are constrained to be excitatory or inhibitory and thus may not change type). Inhibitory synapses were initialised to -5mV^{-1} ; initialisation of excitatory weights is discussed in section II-D. Each excitatory synaptic connection had an associated axonal delay initialised uniformly over the range of 1ms to 20ms and each inhibitory synapse had an associated 1ms axonal delay.

The read-out neuron was an Izhikevich neuron of the fast spiking type and was fully connected to the liquid neurons. The efferent synapses were initialised to a uniform distribution with a range 4 to 8. The weight of these efferent synapses are not dependent on the type of pre-synaptic neuron and are allowed to change type from excitatory to inhibitory and vice-versa.

B. Spike-Timing Dependent Rules

The basis of most models of learning in neural networks is based on Hebbian plasticity [15]. Spike-Timing Dependent Plasticity (STDP) is a form of Hebbian learning where changes in synaptic efficacy is sensitive to the precise timing of pre- and post-synaptic action potentials [16], [17]. The precise form of the STDP rule depends on the neuron type but it is generally assumed that a pre-synaptic spike preceding a post-synaptic spike i.e. a causal relationship, leads to long term potentiation and that post- preceding pre- leads to long term depression.

In this paper we consider two forms of STDP as shown in Figure 1(a) & 1(b): (a) An exponentially decaying form and, (b) A tri-phasic form of STDP (TP-STDP) [18], [19], [20]. Both forms of STDP were implemented as additive ($w_t = w_{t-1} + \Delta w$) with a nearest neighbour approach [21]. Synaptic weights were constrained to be within the range $w_{min} = 0$ and $w_{max} = 10$.

The change in synaptic efficacy, Δw , for the exponential STDP rule is given by:

$$\Delta w = \begin{cases} A^+ e^{-\frac{\Delta t}{\tau_m}} & \text{if } \Delta t \geq 0 \\ -A^- e^{\frac{\Delta t}{\tau_m}} & \text{if } \Delta t \leq 0 \end{cases} \quad (1)$$

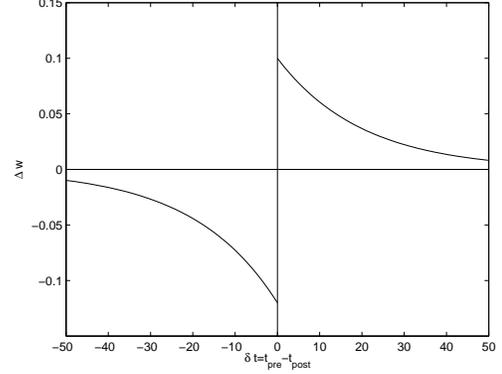
where $A^+ = 0.1$, $A^- = 0.12$, $\tau_m = 20\text{ms}$ and Δt is the time between pre- and post-synaptic spikes given by $\Delta t = t_{post} - t_{pre}$.

For the tri-phasic STDP rule the change in synaptic efficacy is given by:

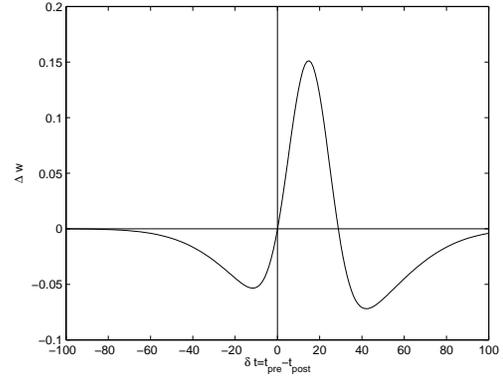
$$\Delta w = 0.25e^{-\frac{(\Delta t - 15)^2}{200}} - 0.1e^{-\frac{-(\Delta t - 20)^2}{2000}} \quad (2)$$

¹In the work of Izhikevich an incoming spike causes an instantaneous increase in the membrane potential determined by the value of the efferent synaptic weight, thus units are measured in millivolts

again where $\Delta t = t_{pre} - t_{post}$. As can be seen from figure 1(b) this rule leads to depression of synapses even with causal spikes if they have a time interval of between approximately 30ms and 100ms. For both rules, time intervals of greater than 100ms cause no modification of the synapses.



(a) Exponential STDP



(b) Tri-Phasic STDP

Fig. 1. Two forms of STDP rule used in this paper

C. Read-Out Training

The read-out neuron was trained using the ReSuMe algorithm presented by Ponulak and Kasinski [22]. This algorithm is a biologically plausible form of supervised learning that allows a neuron to adapt its efferent synaptic weights to enable it to learn arbitrary spike patterns in response to a given synaptic stimuli. The algorithm is independent of the neuron model.

The work of Ponulak and Kasinski shows that for single input spikes on each synapse the algorithm is guaranteed to converge. However, with multiple spikes this is not the case and they suggest that a low learning rate is employed to ensure good performance. The ReSuMe learning equation is given by:

$$\frac{d}{dt} w_{oi}(t) = [S_d(t) - S_o(t)] \left[a_d + \int_0^\infty a_{di}(s) S_i(t-s) ds \right] \quad (3)$$

where $w_{oi}(t)$ is the i -th efferent synaptic weight of neuron o , $S_d(t)$ is the desired output spike train, S_o is the actual output spike train of neuron o , a_d is a learning rate that matches the output firing rate to that of the desired spike train, $S_i(t)$ is the i -th input spike train and a_{di} is a kernel given by:

$$a_{di}(s) = A_{di}e^{-\frac{s}{\tau_{di}}} \quad (4)$$

where A_{di} is a learning rate constant and τ_{di} is a time constant. In the simulations presented in this paper the parameters used were $a_d = 5mV$, $A_{di} = 1mV$ and $\tau_{di} = 5ms$.

D. Weight Initialisation and the STDP Rule

It has been shown that for additive exponential STDP the final distribution of the synaptic weights in a single neuron forms a bi-modal distribution [23], [21]. Cateau and Fukai have also shown the same for a tri-phasic rule [18]. These authors analysis shows this to be due to an unstable fixed point in the Δw vs w phase space which drives weights towards their maximum or minimum. In this sense there is a dependency of the final synaptic weights on the initial conditions. If the weights are all initialised well below the fixed point then *all* of the weights will be driven towards the minimum producing a uni-modal weight distribution and a network of minimal activity. Similarly, if weights are initialised well above the fixed point then *all* of the weights are driven toward their maximum.

For the case of weights that are initialised near to the fixed point then a bi-modal distribution of weights is produced. As such the performance of the neuron is affected by the initialisation of the weights with respect to the fixed point of the STDP rule used. For example, even if the weights are initialised to a uniform distribution across the full range of weights, the position of the fixed point within the weight range determines the proportion of weights being driven towards either the maximum or minimum.

The fixed point analysis discussed above is for single neurons with no form of feedback from output to input. For the networks discussed in this article the situation is more complex with multiple recurrent connections amongst a large number of neurons and a network fixed point may not even exist. In practice the fixed points of the networks for each rule were found experimentally by varying the initial weights of the network across the maximum and minimum range and observing the evolution of the network weight distributions as described below.

1) *Simulations:* Based on the work of Izhikevich [10] networks were allowed to 'settle' for a period of 1 hour model time. During this period the network was stimulated with a random input giving each neuron a random firing rate of approximately 1Hz. This random input allows the network to self-organise and form groups of neurons that fire in a polychronous fashion. During this settling period the network was also repeatedly and evenly stimulated with two arbitrarily chosen spatio-temporal input patterns via a single set of 10 input neurons.

Figure 2 shows examples of the final weight distributions for varying initial weights for both the exponential STDP rule and the TP-STDP rule. For the STDP rules, figures 2(a) & 2(c), the weights are driven towards a bi-modal distribution. For initial weights below 5mV the distributions were unimodal with the weights trending towards the minimum. For the TP-STDP rule and initial weight of 5mV, figure 2(b), leads to a unimodal distribution and an initial weight of 6mV, figure 2(d), leads to a bi-modal distribution. This shows that the evolution of the network weights is dependent on the STDP rule used and on the initialisation of the network. Further to this it may also be seen, from figure 2, that even when the TP-STDP evolves to a bi-modal distribution the balance of the number of synapses at maximum to minimum is different to that found for the STDP rule. This suggests that there may be an unstable network fixed point that is determined by the initial weights and the STDP rule used.

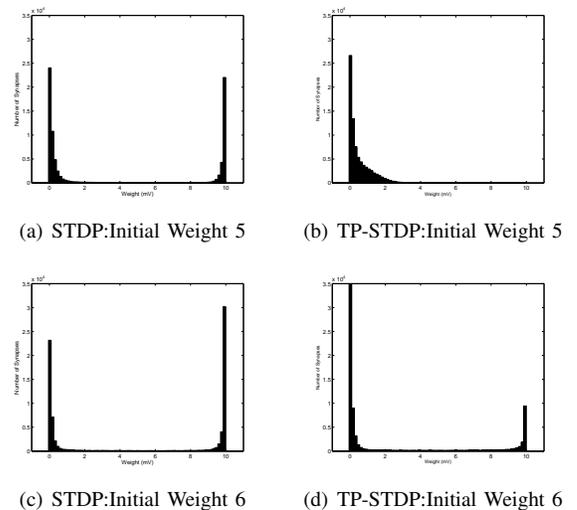


Fig. 2. Final weight distributions for varying weight initialisations and each STDP rule

Figure 3 shows the average synaptic weight in the network plotted as a function of synaptic delay for both STDP rules for networks with initial weights of 6mV. It should be noted that for each synaptic delay the distribution of weights is still bi-modal but the ratio of maximum to minimum weight is varying with synaptic delay. From this figure it is quite clear that there is a distinct difference in the evolution of the synaptic weights for each STDP rule. For the TP-STDP rule the ratio of maximum to minimum weight reduces significantly with delay with a delay of 1ms having a ratio of approximately 0.5. The ratio for the STDP rule is small for middle delay values but approximately 0.5 at the extremes of the delay range.

III. NETWORK PERFORMANCE

After settling each network was again repeatedly stimulated with the two input patterns (used during settling) but with no STDP and no random input. The read-out neuron was then trained using the ReSuMe algorithm to produce two different

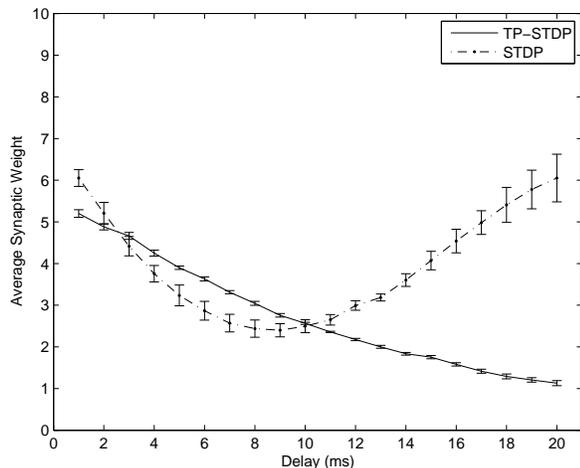


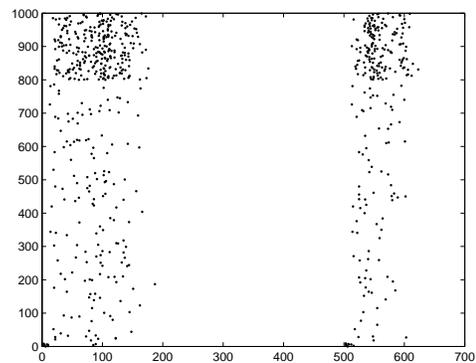
Fig. 3. Average synaptic weight as a function of synaptic delay. Curves averaged over 10 networks, error bars show the Standard Error.

but arbitrary spike trains based on the response of the network to each input pattern. The ReSuMe algorithm was applied over 4000 one second epochs of input stimulus to allow a slow convergence to take place.

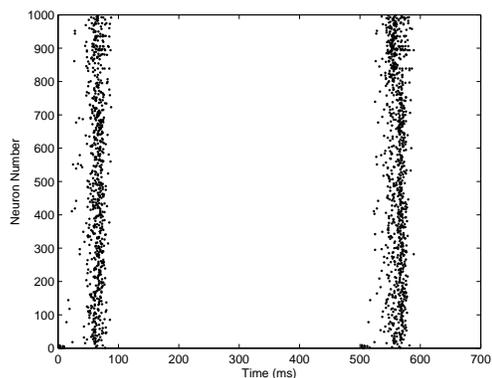
A. Network Reponse

Figure's 4(a) & 4(b) shows network responses to two input patterns for a network allowed to settle with exponential STDP and TP-STDP respectively. It can be seen that the network allowed to settle with STDP produces a longer but more sparse response than the network allowed to settle using TP-STDP (the total number of spikes in the responses were generally found to be of the same order). A possible explanation of this may be found with respect to figure 3. The weight/delay curve for STDP achieves a balance of weights where there is significant proportion of long delays at maximum efficacy and enables the network to produce responses that are longer in duration. The network response for the STDP network also shows that there is a higher firing rate for the inhibitory neurons (neurons greater than 799) which may be leading to the sparseness in the excitatory neurons.

For network response of the TP-STDP may again be explained in terms of figure 3 as in this case there is a greater proportion of excitable synapses for short delays and at long delays there is only a small number of synapses at maximum efficacy. Thus, the network produces a response that is constrained to shorter time periods but with a higher firing rate producing a dense response. Since read-out neurons are only capable of producing output spikes when there is sufficient input energy, it follows that a more dense network response may be more capable of driving the read-out neuron to produce output spikes at arbitrary time intervals with a high level of accuracy. In contrast to this and by the same argument, the period of time for which the short dense responses may produce output spikes is limited.



(a) STDP



(b) TP-STDP

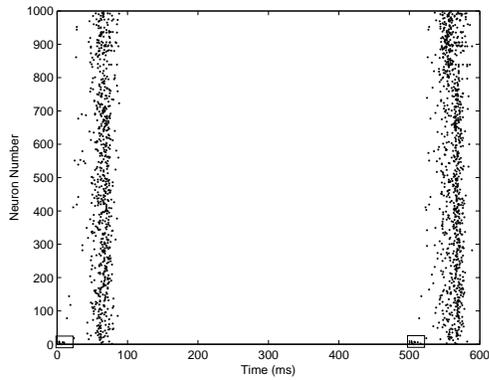
Fig. 4. Typical network responses for networks initialised with weights=6. (a) STDP and (b) TP-STDP

B. Read-Out Neuron Training

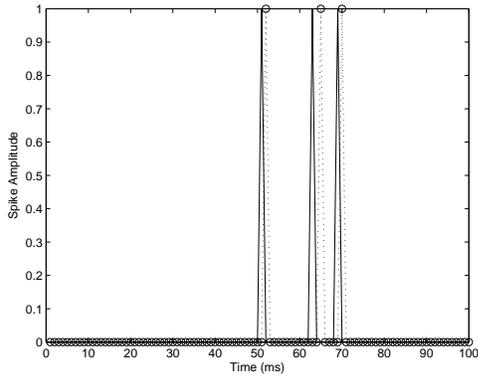
Figure 5 shows an example of the results found with a network allowed to settle using the tri-phasic rule and a readout neuron trained using the ReSuMe algorithm. Figure 5(a) shows the network response to each of the input patterns at 0ms and 500ms. Figures 5(b) & 5(c) show the response of the read-out neuron to each input pattern after training. In this case the desired responses are close to each other in that they are both close in time and are composed of three spikes. The main difference between the spike patterns is the relative timing of the spikes. In both cases the read-out neuron was able to produce a response that closely matches the desired output patterns.

The average performance of the networks was found by calculating the distance between the desired responses and the actual responses. In this work the spike distance suggested by van Rossum [24] was used.

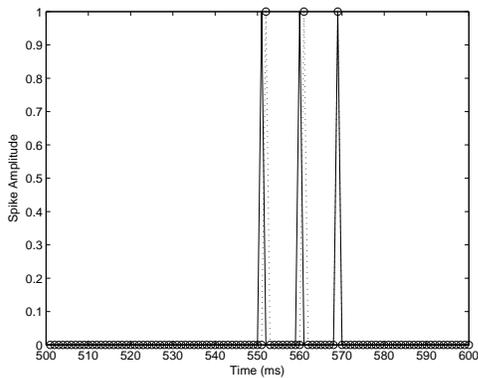
Figure 6 shows the results of applying the van Rossum distance to the responses produced from networks allowed to settle with STDP and TP-STDP for a range of initial starting weights. For initial weights of 4mV and below both STDP rules produce similar results in terms of the average van Rossum distance and do not perform well and corresponds the regime



(a) Network response



(b) Read out response to input 1



(c) Read out response to input 2

Fig. 5. Example of results found for training the network to produce two arbitrary spike trains in response to two spatio-temporal input patterns. (a) The network response to each input. The boxes show the spatio-temporal input, (b) & (c) The read-out responses to the first input and second input patterns respectively (solid lines show the desired responses, the dotted lines and circles show the actual responses)

where the initial weights are unimodal and decreasing towards the minimum value. For an initial weight of 5mV both sets of networks perform at a similar level with low van Rossum measures and it may be noted that at this value the networks

produce a weight distribution that is now bi-modal and thus some weights have 'gravitated' to the maximum. For initial weights of 6mV and above it may be seen that networks settled with the TP-STDP rule significantly outperforms the networks settled using the standard STDP rule. It is also apparent that the TP-STDP rule produces networks that perform well for a greater range of initial starting weights.

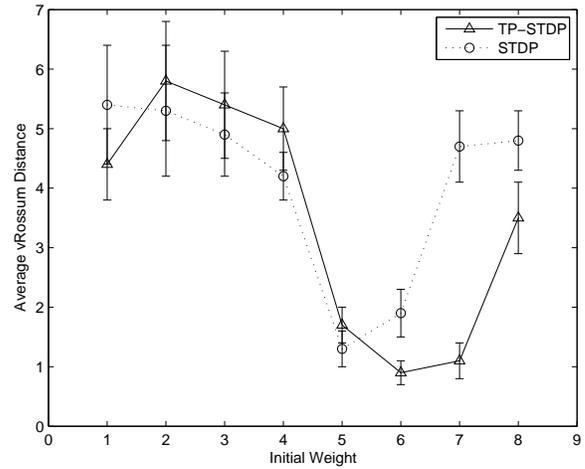


Fig. 6. Average synaptic weight as a function of synaptic delay. Curves averaged over 10 networks, error bars show the Standard Error.

IV. DISCUSSION

This paper has investigated the use of two different STDP rules applied to a reservoir of neurons with synaptic delays and has further investigated the networks ability to produce specific output spike trains in response to spatio-temporal inputs. It was found that the evolution of the reservoir weights, and thus the network dynamics, was dependent on the STDP rule used and the initial conditions of the network. The STDP rule was found to produce a set of weights that gave an even balance for both short and long synaptic delays. In contrast, the TP-STDP rule was found to give more weighting to the shorter synaptic delays and reduce the amount of long synaptic delays. As a consequence of this the STDP rule produces networks that have long but sparse responses whereas the TP-STDP rule generates networks that gives short dense responses.

It was also found that networks evolved with the TP-STDP rule, in terms of the van Rossum distance between actual output and desired output, performed significantly better than those networks evolved with the STDP rule. One possible reason for this is that the read-out neuron is only capable of producing output spikes when there is sufficient input energy. Thus a dense network response would more likely to be able to drive the the read-out neuron at the required times. A sparse response on the other hand, although capable of driving the read-out neuron, may not be able to do this with the same level of accuracy in time.

It should also be noted that the networks were driven with two different spatio-temporal inputs and the single read-out neuron trained to produce two different spike trains in response to these inputs. This places a further condition on the networks when using a linear read-out neuron to produce more than one input-output mapping; the network dynamics in response to each input pattern must be linearly separable from each other. The production of network responses that are linearly separable is referred to as the separation property [2]. Maass *et al* also show that for the LSM to be able to produce robust outputs the networks must have a fading memory property. The fading memory property ensures that current network state at the time of the input patterns arrival does not cause network responses that are vastly different for similar spatio-temporal input patterns. The weight/delay curves suggest that the TP-STDP rule produces networks that may have shorter fading memory properties enabling them to be robust to the residual effects of previous inputs.

ACKNOWLEDGMENT

Scott Notley (fully) and André Grüning (part) were supported by EPSRC grant EP/I014934/1

REFERENCES

- [1] D. Verstraeten, B. Schrauwen, M. D’Haene, and D. Stroobandt, “An experimental unification of reservoir computing methods,” *Neural Networks*, vol. 20, pp. 391–403, 2007.
- [2] W. Maass, T. Nätzlager, and H. Markram, “Real-time computing without stable state: A new framework for neural computation based on perturbations,” *Neural Comp.*, vol. 14, pp. 2531–2560, 2002.
- [3] H. Jaeger, W. Maass, and J. C. Principe, “Editorial: special issue on echo state networks and liquid state machines,” *Neural Networks*, vol. 20, pp. 287–289, 2007.
- [4] W. Maass, *Computability in Context: Computation and Logic in the Real World*. Imperial College Press, 2010, ch. 1, pp. 275–296.
- [5] J. J. Steil, “Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning,” *Neural Networks*, vol. 20, pp. 353–364, 2007.
- [6] B. Schrauwen, M. Wardermann, D. Verstraeten, J. J. Steil, and D. Stroobandt, “Improving reservoirs using intrinsic plasticity,” *Neurocomputing*, vol. 71, pp. 1159–1171, 2008.
- [7] J. Triesch, “Synergies between intrinsic and synaptic plasticity mechanisms,” *Neural Comp.*, vol. 19, pp. 885–909, 2007.
- [8] D. Norton and D. Ventura, “Preparing more effective liquid state machines using hebbian learning.” *Int. Joint Conf. on Neural Networks*, 2006.
- [9] —, “Improving liquid state machines through iterative refinement of the reservoir,” *Neurocomputing*, vol. 73, pp. 2893–2904, 2010.
- [10] E. N. Izhikevich, “Polychronization: computation with spikes,” *Neural Computation*, vol. 18, pp. 245–282, 2006.
- [11] B. Szatmáry and E. M. Izhikevich, “Spike-timing theory of working memory,” *PLoS Comput. Biol.*, vol. 6, no. 8, pp. 1–11, 2010.
- [12] V. Braitenberg and A. Schuz, *Anatomy of the cortex: statistics and geometry*. Springer, 1991.
- [13] E. N. Izhikevich, “Simple model of spiking neurons,” *IEEE Trans. Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.
- [14] —, “Which model to use for cortical spiking neurons?” *IEEE Trans. Neural Networks*, vol. 15, pp. 1063–1070, 2004.
- [15] L. F. Abbott and S. B. Nelson, “Synaptic plasticity: taming the beast,” *Nature Neuroscience Supplement*, vol. 3, pp. 1178–1183, 2000.
- [16] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, “Regulation of synaptic efficacy by coincidence of postsynaptic epsps and epsps,” *Science*, vol. 275, no. 5297, pp. 213–215.
- [17] G. Bi and M. Poo, “Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type,” *J. Neuroscience*, vol. 18, no. 24, pp. 10464–10472, 1998.
- [18] H. Câteau and T. Fukai, “A stochastic method to predict the consequence of arbitrary forms of spike-timing-dependent plasticity,” *Neural Computation*, vol. 15, no. 3, pp. 597–620, 2003.
- [19] G. M. Wittenber and S. S.-H. Wang, “Malleability of spike-timing-dependent plasticity at the ca3-ca1 synapse,” *J. Neuroscience*, vol. 26, no. 24, pp. 6610–6617, 2006.
- [20] H. Z. Shouval, S. S.-H. Wang, and G. M. Wittenberg, “Spike timing dependent plasticity: a consequence of more fundamental learning rules,” *Frontiers in Computational Neuroscience*, vol. 4, no. 19, pp. 1–13, 2010.
- [21] A. Morrison, M. Diesmann, and W. Gerstner, “Phenomenological models of synaptic plasticity based on spike timing,” *Biol. Cybern.*, vol. 98, pp. 459–478, 2008.
- [22] F. Ponulak and A. Kasiński, “Supervised learning in spiking neural networks with resume: Sequence learning, classification and spike shifting,” *Neural Computation*, vol. 22, pp. 467–510, 2010.
- [23] M. C. W. van Rossum, G. Q. Bi, and C. G. Turrigiano, “Stable hebbian learning from spike timing-dependent plasticity,” *J. Neuroscience*, vol. 20, no. 23, pp. 8812–8821, 2000.
- [24] M. C. W. van Rossum, “A novel spike distance,” pp. 751–763, 2001.