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.

# Classifying Patterns in a Spiking Neural Network

Brian Gardner*and André Grüning

Department of Computing, University of Surrey,
Guildford, Surrey, GU2 7XH, United Kingdom

**Abstract**. Learning rules for spiking neural networks have emerged that can classify spatio-temporal spiking patterns as precise target spike trains, although there remains uncertainty in which rule to select that offers the greatest performance. Here, we 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. We achieve a level of performance that is comparable with that found previously for alternative neuron models, and demonstrate the advantages of classifying inputs by multiple-spike timings: both by increasing the performance and the reliability of classifications.

## 1 Introduction

A Spiking Neural Network (SNN) represents a third generation neural network model, being a significant improvement in terms of biological realism over its predecessors. Despite this, there exist relatively few learning rules for SNN's over more traditional neural network models, despite theoretically being capable of superior computational power by utilising a temporal coding scheme [1].

Several learning rules have been developed that enable the learning of multiple and precise spike times: important examples include Remote Supervised learning Method (ReSuMe), the Chronotron and a statistical method developed by Pfister et al. [2, 3, 4]. ReSuMe is a biologically realistic and empirically derived rule, allowing the supervised learning of target spike trains in response to spiking input patterns. This rule is ideal in classifying input patterns by temporally precise spike trains, although it was shown to have an inferior memory capacity in comparison with the E-learning form of the Chronotron learning rule [3]. Additionally, ReSuMe was further extended to learning in a multilayered network [5], where it was shown to converge in learning more rapidly on a pattern classification task as the number of hidden neurons increased. The rule developed by Pfister et al. [4] optimises the probability of a postsynaptic neuron generating a target spike train in response to a spiking input pattern. Unlike ReSuMe and the Chronotron, few attempts in applying this rule to classifying multiple input patterns by temporally precise target spike trains have been made, making the rule's efficiency difficult to estimate.

Therefore, we explore the performance of Pfister et al.'s learning rule on a pattern classification task, with the aim of quantifying its memory capacity in terms of the number of input patterns that can be reliably classified by precise

*BG was fully supported by EPSRC grant EP/J500562/1.

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.

output spike trains. We further examine the potential benefits that can be gained from target spike trains that consist of multiple spikes as opposed to single-spike outputs; both in terms of the absolute performance and the reliability of responses.

## 2 Model

A single readout neuron receives input from $m$ presynaptic neurons. The list of firing times due to the $j^{\text{th}}$ input, $1 \leq j \leq m$, is denoted by the spike train $X_j$. If the readout neuron generates the output spike train $Y$ in response to an input pattern $\mathbf{X} = \{X_j\}$, then its membrane potential at time $t$ is defined by the Spike Response Model [6]:

$$u(t|\mathbf{X}, Y) := U_{rest} + \sum_j w_j \sum_{s \in X_j} \epsilon(t-s) + \sum_{s \in Y} \kappa(t-s) \,, \tag{1}$$

with $U_{rest} = -70$ mV the resting membrane potential and $w_j$ the $j^{th}$ afferent synaptic weight. The Postsynaptic Potential (PSP) kernel is taken as $\epsilon(s) = \frac{\epsilon_0}{\tau_m - \tau_s}(\mathrm{e}^{-s/\tau_m} - \mathrm{e}^{-s/\tau_s})$ , where $\epsilon_0 = 20$ mV·ms is a scaling constant, $\tau_m = 10$ ms the membrane time constant and $\tau_s = 2$ ms the synaptic rise time. The reset kernel is $\kappa(s) = \kappa_0 \mathrm{e}^{-s/\tau_m}$ , with $\kappa_0 = -15$ mV. Both kernels are set to 0 for $s < 0$. Given the neuron's membrane potential $u(t)$, spike times are distributed according to the instantaneous firing rate:

$$\rho(u(t)) = \rho_0 \exp\{\beta(u(t) - \vartheta)\} \,, \tag{2}$$

where $\rho_0 = 0.01$ ms$^{-1}$ and $\beta = 5$ mV$^{-1}$ are stochasticity parameters, and $\vartheta = -55$ mV is the firing threshold. The probability of the neuron firing at each moment in time is $\rho(t)\,\delta t$, where we set the simulation time step $\delta t = 0.2$ ms.

Using a stochastic model for neuronal spike generation allows us to determine the likelihood of producing a desired postsynaptic spike train $Y^{\text{ref}}$ in response to $\mathbf{X}$. As shown by Pfister et al. [4], the log-likelihood of generating $Y^{\text{ref}}$ is:

$$\log P(Y^{\text{ref}}|\mathbf{X}) = \sum_{s \in Y^{\text{ref}}} \log \rho(u(s)) - \int_0^T \rho(u(t))\,\mathrm{d}t \,, \tag{3}$$

given an input pattern lasting duration $T$. Taking the gradient of the above and combining with equations (1) and (2) determines the direction of synaptic weight updates:

$$\begin{aligned} \Delta w_j &= \eta \, \frac{\partial \log P(Y^{\text{ref}}|\mathbf{X})}{\partial w_j} \\ &= \eta \int_0^T \Big[ \sum_{s \in Y^{\text{ref}}} \delta(t-s) - \rho(u(t)) \Big] \sum_{s \in X_j} \epsilon(t-s)\,\mathrm{d}t \,, \end{aligned} \tag{4}$$

where the factor $\beta$ is incorporated into the learning rate $\eta$.

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.

The network must learn to classify $p$ input patterns into $c$ classes, where in our simulations an equal number of input patterns were assigned to each class. Each class had an associated target spike train $Y^{\text{ref}}$, that the network readout neuron learnt to reproduce in response to all input patterns belonging to the same class. Learning took place on an episodic basis, where an input pattern was randomly selected at the start of each episode to be presented to the network, lasting duration $T = 500$ ms.

In determining which target spike train (and class) most closely matched the output spike train $Y^{\text{out}}$ on each episode, we used the van Rossum Distance (vRD) [7]. The vRD measures the distance between two spike trains, giving the metric $\mathcal{D}(Y^{\text{out}}, Y^{\text{ref}})$ , where we set the coincidence time constant $\tau_c = 10$ ms. In more detail, the vRD between $Y^{\text{out}}$ and each $Y^{\text{ref}}$ was computed at the end of each episode, giving the set of distances: $\mathbf{D} = \{\mathcal{D}_1, ..., \mathcal{D}_c\}$, for $c$ different classes. The matching class number $i$ corresponded to the smallest distance $\mathcal{D}_i$, that is $i = \arg\min_i \mathbf{D}$. Additionally, we added the constraint that the difference between $\mathcal{D}_i$ and the next smallest distance exceeded 0.1 for any classification to be made, thereby removing any ambiguity in selecting the matching class.

## 3 Results

The network consisted of $m = 400$ presynaptic neurons as the inputs, and 1 postsynaptic neuron as the readout. Input neurons were fully connected to the readout neuron, where synaptic weights were initialised to 0. Input patterns presented to the network consisted of independent Poisson spike trains at each input neuron with a mean firing rate of 6 Hz, with a random realization for each pattern. Input patterns were equally assigned between 3 classes, where each class was associated with a target spike train that contained between 1 and 5 spikes, depending on the learning task. Target spike times for each class were randomly selected from a uniform distribution over the interval [40, T], with a minimum interspike interval of 10 ms. Target spike trains differed from each other by $\mathcal{D} > 1 - e^{-1}$, corresponding to a vRD between two spike times that are separated by at least $\tau_c$, to ensure classes were unique.

The firing activity of the readout neuron was defined by equation (2) and afferent synaptic weights were updated according to equation (4) at the end of each learning episode. Synaptic weights were hard-bound to the interval $-10 \leq w_j \leq 10$. For the learning rate, an exponential dependence on the number of input patterns gave optimal performance: $\eta = c_1 \exp(-c_2\, p)$, with fitted parameters $c_1 = 0.022$ and $c_2 = 0.025$. To ensure convergence in learning, the total number of episodes on each learning task was $1000p$.

To measure the performance $P$ of the network with the episode number $n$ we took a moving average: $\tilde{P}(n) = (1 - \lambda)\tilde{P}(n-1) + \lambda P(n)$, where $P(n) = 100\%$ if the correct pattern was classified and $P(n) = 0$ otherwise. The timing parameter was set to $\lambda = 2/(1 + 100p)$. A moving average was necessary, given that the readout neuron's spike-timing responses fluctuated between episodes.

Figure 1 shows the performance of the network after convergence in learning

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.

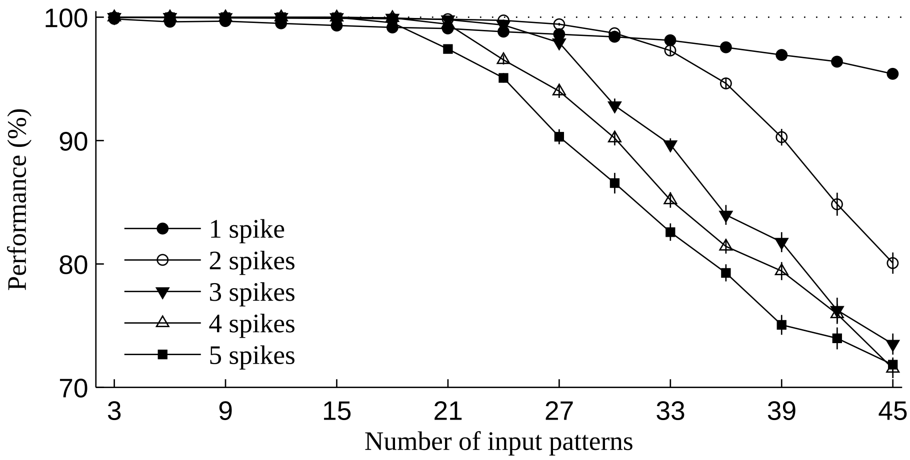


Fig. 1: Classification performance $\tilde{P}$ after convergence in learning, as a function of the number of input patterns and the number of target spike times. Input patterns are equally assigned between 3 classes. Performance values are averaged over 40 realizations, each with a random initialization. Error bars show the standard error of the mean. The performance for 1 target spike remains below 99.9% over all the number of input patterns.

when tasked with classifying between 3 and 45 input patterns that are equally assigned between 3 classes. Between 1 and 5 target spike times belonged to each class. The performance of the network for 1 target spike was found to decrease slowly, reaching $95.4 \pm 0.2\%$ by 45 input patterns. By contrast, The performance decreased more rapidly for a greater number of target spikes (reaching $80.1 \pm 0.9\%$, $73.5 \pm 0.9\%$, $71.5 \pm 0.8\%$ and $71.8 \pm 0.6$ % for 2, 3, 4 and 5 target spikes respectively by 45 input patterns). Despite this, multiple target spikes demonstrated superiority over a single target spike when learning fewer input patterns: approaching 100% performance for more than 2 target spikes and less than 18 input patterns. Furthermore, the standard deviation of the performance was smaller for multiple rather than single target spikes when learning fewer input patterns: with a standard deviation $\sigma < 0.05\%$ for greater than 2 target spikes and $\sigma \approx 0.3\%$ for 1 target spike, for less than 15 input patterns. Such a reduction in the variation of the performance was attributed to the readout neuron only having to reproduce a fraction of the target spike times belonging to each class, for sufficiently reliable pattern classifications. For example: when classifying patterns into classes with 3 target spikes, an output spike train matching only 2 of 3 target spike times would be sufficiently close for a correct input classification. For classes with 1 target spike however, there must be at least 1 output spike for any classification to be made.

Conversely, for a larger number of input patterns, the standard deviation of the performance increased with the number of target spikes: with $3\% < \sigma < 6\%$

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.

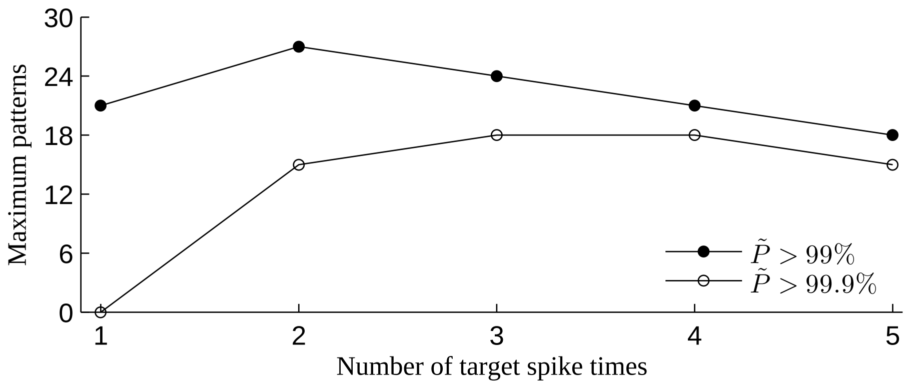


Fig. 2: Maximum number of input patterns that can be learnt as a function of the number of target spike times belonging to each class. Different curves correspond to the minimum allowable performance $\tilde{P}$ after convergence in learning. Each point is accurate to within 2 patterns.

for greater than 1 target spike and $\sigma \approx 1\%$ for 1 target spike, by 45 input patterns. This was due to the increased load on the network in having to learn to fire at multiple times, in addition to learning a large number of input patterns.

In finding the optimal number of target spikes that should belong to each class for reliable classifications, we determined the maximum number of input patterns that could be learnt at a given level of performance (figure 2). For example, a performance greater than 99% and 99.9% corresponded to a probability of failure on each episode that was less than $10^{-2}$ and $10^{-3}$ respectively. As expected, the maximum number of patterns that could be learnt decreased as the minimum performance level increased, with a shift towards a larger number of target spikes belonging to each class.

According to [8, 3] the maximum number of patterns $p$ that can be learnt in a network increases with the number of afferent synapses $m$, by the ratio $\alpha = p/m$; also being the capacity of the network. From figure 2, we determined the capacity as $\alpha = 0.045 \pm 0.005$ for $\tilde{P} > 99.9\%$, corresponding to classes containing 3 or 4 target spikes. In comparison with previous work, this capacity is superior to that of ReSuMe and I-learning (previously determined in [3] as between 0.02 and 0.04), whilst being an order of magnitude less than E-learning (0.22) [3].

## 4 Discussion

In this paper, we demonstrated the capability of a stochastic neuron model in learning to classify input patterns by the precise timing of output spikes. We considered both single and multiple target spikes that belonged to each class, and compared their performance when classifying between 3 and 45 input patterns.

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.

We found classes containing 3 or 4 target spikes were optimal when classifying input patterns: both by increasing the level of performance, and reducing the variation in the performance. Our method was also found to be comparable with that found previously for alternate neuron models.

The performance of the network dropped off significantly after learning more than 18 input patterns, that might impact on its potential for real applications. Since the maximum number of input patterns that can be classified scales with the size of the network [3], a solution could be to simply increase the number of input neurons for a desired level of performance. Alternatively, more than one readout neuron could be implemented, with the load on the network being equally divided between them.

We assumed a supervised signal was available during learning to provide target spike trains to the network. There remains much uncertainty in the origin of such a signal however, with reinforcement learning representing a more biologically realistic alternative. In a previous paper [9], we demonstrated how a target spike train could be learnt with a delayed reward signal.

This paper was concerned primarily with establishing the potential applicability of a stochastic neuron model in learning to classify patterns, where we only determined the capacity of the network for 400 afferent synapses, with input patterns equally divided between 3 classes. It is beyond the scope of this paper to explore various network setups that might have an impact on learning, although future work could aim to further quantify the capacity over a varying number of afferent synapses and classes, as well as to provide a more direct comparison against another learning rule such as ReSuMe.

## References

[1] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[2] 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.

[3] Răzvan V Florian. The chronotron: A neuron that learns to fire temporally precise spike patterns. *PloS one*, 7(8):e40233, 2012.

[4] Jean-Pascal Pfister, Taro Toyoizumi, 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.

[5] Ioana Sporea and André Grüning. Supervised learning in multilayer spiking neural networks. *Neural computation*, 25(2):473–509, 2013.

[6] W. Gerstner and W.M. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge University Press, Cambridge, 2002.

[7] Mark CW van Rossum. A novel spike distance. *Neural Computation*, 13(4):751–763, 2001.

[8] Robert Gütig and Haim Sompolinsky. The tempotron: a neuron that learns spike timing–based decisions. *Nature neuroscience*, 9(3):420–428, 2006.

[9] Brian Gardner and André Grüning. Learning temporally precise spiking patterns through reward modulated spike-timing-dependent plasticity. In *Artificial Neural Networks and Machine Learning–ICANN 2013*, pages 256–263. Springer, 2013.