Estimating numerosity of visual concepts with a single spiking neuron.

Hannes Rapp[a,*], Martin Paul Nawrot[a], Merav Stern[b]

[a] *Computational Systems Neuroscience, Institute of Zoology, University of Cologne, Zülpicher Straße 47b, Cologne, Germany*
[b] *Department of Applied Mathematics, University of Washington, Lewis Hall 201, Box 353925, Seattle, WA 98195-3925*

Abstract

It has been shown that insects exhibit behaviour of basic numerical cognition such as counting. How insects develop these abilities is not yet known. Here we show that numerosity estimation can be realized and learned by a single spiking neuron with an appropriate plasticity rule. We introduce a modified synapse-based learning algorithm implemented in an integrate and fire neuron model. We demonstrate that this brain-inspired algorithm performs on par with state-of the art machine learning approaches and requires considerably less training epochs. In addition to numerosity estimation, we also show that our model is able to efficiently categorize realistic sensory input patterns on a noisy background with high precision. We propose that using action potentials to represent numerosity and other concepts within a single spiking neuron is beneficial for organisms with small brains and limited neuronal resources.

Keywords: numerical cognition, insect cognition, spiking neuron, synaptic learning rule, credit assignment, machine learning

1. Introduction

Insects have been shown to possess cognitive abilities (Chittka and Niven, 2009; Avarguès-Weber et al., 2011, 2012; Avarguès-Weber and Giurfa, 2013; Pahl

*Corresponding author
Email address: `hannes.rapp@smail.uni-koeln.de` (Hannes Rapp)

et al., 2013). These include estimating numerosity (Rose, 2018; Skorupski et al., 2018), counting (Chittka and Geiger, 1995; Dacke and Srinivasan, 2008; Menzel et al., 2010) and other basic arithmetical concepts (Howard et al., 2018). How insects succeed in these cognitive tasks is not yet clear. A recent model study by Vasas and Chittka (2019) suggests that a small circuit of only four rate-based neurons can implement the basic cognitive ability of counting visually presented items. The study implies that their minimal circuits can recognize concepts such as a "higher" or "lower" item number than zero (Howard et al., 2018) or "same" and "different" number of items (Avarguès-Weber et al., 2012) when combined with a sequential inspection strategy that mimics the behavioural strategy of insects during detection (Dacke and Srinivasan, 2008). The neural circuit studied in Vasas and Chittka (2019) was shown to successfully predict whether the numerosity of a particular feature (e.g. yellow) has been present below or above some threshold value, despite been presented within a sequence of other features and distractors. The circuit was hand-tuned in order to perform successfully in a given task. This poses the question how an efficient network connectivity can be learned by means of synaptic plasticity.

Numerosity estimation tasks (that use the number of detected instances for counting), have also been considered in the field of computer vision, in particular in the study of object recognition. Many resources have been devoted to train artificial neural networks to perform these tasks, especially with the recent rise of deep learning based methods (Schmidhuber, 2015). These methods where shown to be successful at object detection and allow for counting by detecting the multiple relevant objects within a scene, explicitly (Ren et al., 2015) or implicitly (Lempitsky and Zisserman, 2010). However, these model classes are costly and often require cloud-computing clusters and millions of samples to be trained (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014). It was recently shown by OpenAI that the amount of computation needed by such artificial systems has been growing exponentially since 2012. Clearly, insect brains with their limited neuronal resources cannot afford similar strategies but have to employ fundamentally different algorithms to achieve basic numerical cognition. These

biological algorithms might proof highly efficient and thus have the potential to inform the development of machine learning (ML) approaches.

Here we present a biologically realistic spiking neuron model with a synaptic learning rule that allows the development of basic numerical cognition abilities while overcoming noisy input. Our model approach takes advantage of the discrete nature of action potentials generated by a single spiking output neuron. The number of emitted spikes within a short time period represent a plausible biological mechanism for representing numbers. In a virtual experiment we train our neuron model to estimate the numerosity of handwritten digits (LeCun and Cortes, 2010) that represent even numbers. The synaptic weights are learned from the observations and thus our model overcomes the problem of hand-tuning a single-purpose neuronal circuit. After training the model is tested in a counting task of sequentially presented features similar to the task explored in Vasas and Chittka (2019).

A number of recent studies managed to train spiking neural networks with gradient-based learning methods. To overcome the discontinuity problem due to the discrete nature of action potentials, some studies evaluated the post-synaptic currents created by spikes in the receiving neurons (Nicola and Clopath, 2017) and (Huh and Sejnowski, 2017) for the training procedures. Other studies used the timing of the spikes as a continuous parameter (Bohte et al., 2000; O'Connor et al., 2017), which lead to synaptic learning rules that rely on the exact time intervals between the spikes of the sending and receiving neurons (pre- and post-synaptic). These Spike Timing Dependent (STDP) rules had first been observed experimentally and hence much of attention is given to them in neuroscientific studies (Bi and ming Poo, 2001; Caporale and Dan, 2008; Song and Abbott, 2000). A few recent studies tried to approximate or relax the discontinuity problem (Zenke and Ganguli, 2018; Bengio et al., 2013) to enable error backpropagation learning with spiking neural networks. Training spiking neurons as classifiers, perceptron-like machines has also been proposed by Gütig and Sompolinsky (2006) and Memmesheimer et al. (2014) as well as using closely related binary neurons to perform classification in olfactory systems (Huerta et al., 2004).

Since in the present work we are interested in estimating numerosity, the teaching signal in our model represents only a single integer value, which is the total number of even digits in each image presented. In order to teach our model neuron this value we introduce an improvement to the implementation of Gütig (2016) . This approach over comes the spiking discontinuity problem by considering the membrane potential for the gradient-based learning. We show that our improved approach allows to train the even digits counting quicker with better generalization capabilities and also supports better the reliability of numerosity estimation under inputs with complex distributions, including noise distributions, as naturally present in the brain.

2. Results

Our neuron model follows the principles of the Multi-Spike Tempotron (MST) (Gütig, 2016), a leaky integrate-and-fire neuron model with a gradient-based local learning rule. The neuron is taught to respond to a sequence of pre-synaptic spatio-temporal spike input patterns (features) with a particular number of output spikes (for details see Methods). The original MST algorithm used Momentum (Polyak, 1964) to boost its capabilities. We improve the MST performances by using a synaptic specific adaptive update approach similar to RMSprop as proposed by Tieleman and Hinton (2012). We show that this adaptive learning approach generates significantly better and stable generalization and more efficient performance of the learning capabilities. We review both the Momentum and the RMSprop in the method section.

We first apply the MST model to the problem of counting even digits within an image composed of several randomly scattered handwritten MNIST digits (LeCun and Cortes, 2010). For reference we compare the performance to a conventional computer vision model that uses a convolutional neural network (ConvNet)(Krizhevsky et al., 2012; Seguí et al., 2015). Next we consider the problem of estimating numerosity of arbitrary features present in the presynaptic spike trains. This allows us to study the robustness of numerosity estimation

with the MST when the input and features follow different input distributions that deviate from the homogeneous Poisson distribution as studied in Gütig (2016).

2.1. Numerosity of even handwritten digits

Numerostiy estimation, as a conceptual problem, is similar to a regression problem. In both cases we have no a-priori knowledge of the maximum number of desired features present in an input. For this reason, we choose *root mean-squared error* (RMSE) of wrongly counted even digits to evaluate the model performance. Note that for this criterion lower values mean better performance and that this criterion especially penalizes large deviations from the underlying true value. It is also important to note that the ConvNet model used for benchmark comparison is built using prior knowledge about the distribution of the training set. The ConvNet is constrained to learn a categorical distribution over $[0, 6]$, where 6 is the maximum possible count of even digits in our set of images (an example of an image in the set is given in figure 1). This has two implications; First, the ConvNet model will be unable to predict the number of even digits in images that include more than 6 even digits. While for this particular task the data-set is constructed such that this is not possible, in general regression problems the prediction targets are usually not bounded. Second, the counting error a ConvNet can make is constrained by the training bound, i.e. the maximum error is 6. In contrast, the MST model does not have any need for this prior knowledge or constraints. In principle it is capable of solving the general, true regression problem and can also make predictions for images that contain more than 6 even digits. It thus solves a more difficult learning problem when solving the task. The maximum prediction error in this case is unbound rendering the MST more vulnerable, resulting in a possibility for higher RMSE values as compared to the ConvNet.

The results are summarized in table 1. The best performing model, MST with adaptive learning rate (see methods for further details), is highlighted in bold. We find that generally the MST with our adaptive learning rate performs

better compared to the Momentum (Gütig, 2016), independent of the choice of a particular spike-encoding method. Interestingly the spike-based single neuron MST model was able to outperform the rate-based ConvNet. In order for the ConvNet to achieve better RMSE (~ 0.95), similar to the MST model, the ConvNet needs to be trained for about 5-10x more epochs than the MST (data not shown). If the model's complexity in terms of free parameters is taken into account (adjusted RMSE), the MST model is incomparable more computationally efficient. Further we find that using FoCal (Bhattacharya and Furber, 2010) as spike-encoding method (inspired by the encoding done at the fovea; see Methods) works significantly better compared to a naive encoding. This was an expected behaviour since FoCal exploits local and spatial correlations (Bhattacharya and Furber, 2010). It is a paradigm known to be more effective compared to pixel-by-pixel consideration. This is in agreement with artificial neural networks where the success of ConvNets over regular, multilayer perceptron (MLP) networks is mostly due to the learned spatial filters by its convolutional layers.

Since we've shown that encoding can impact the model's performance, it is possible that by applying additional efficient encoding algorithms, the performance of the MST model can be improved even farther than the results presented here. We leave the exploration of different types of encodings open for future research.

2.2. Generic task-related numerosity estimation

While speed of convergence is similar, we find that using adaptive learning results in significantly less variant training error (fig. 2). This also holds for variance of test error on an independent validation data-set and results in better generalization capabilities to previously unseen inputs (*patterns only* fig. 3). The adaptive, per synapse learning rate, combined with smoothing over past gradients has a regularizing effect and prevents the model from over-fitting to the training data. We further conclude, that the modified algorithm is able to find better and wider optima of the spike-threshold surface loss function as compared to learning with Momentum. More importantly this behaviour is consistent and

independent of the input spike train's distribution and noise level (fig. 3).

3. Discussion

Numerical cognition is a general term that covers several subproblems, for example numerosity, counting, basic arithmetical operations and many more. While each individual subproblem might appear to be fairly trivial for us humans, it's not clear yet how this could be realized computationally on the level of spiking neurons or networks thereof. Despite the simplicity of these subproblems they do provide a foundation for more complex concepts that humans make heavy use of and are relevant for behaviour like decision making. For example, if one is able to count entities it might only be a small step towards combining that information to perform more advanced concepts such as empirical statistics and estimating (discrete) probabilities. While the specific symbolic math concepts are unavailable to the brain, animals and even insects are still able to show basic numerical cognition abilities and evaluating basic probabilities. While insects and other small-brained species are not specifically aware of symbolic math representation of statistics and probability theory, they can still make use of its properties to inform their behaviour. In this work we have explored the Multi-Spike Tempotron (Gütig, 2016) (MST), a spiking neuron model of the neocortex, which can be trained by gradient-descent to produce precise number of output spikes in response to a certain stimulus. We showed, that this allows for basic numerical cognition to emerge, in the form of estimating numerosity by relating the number of spikes to the numerosity of features present in some input. While previous related work of gradient-based learning in spiking network models are mostly concerned with solving classical classification tasks, in this work we applied the single-neuron MST model to solve a regression problem. Specifically, we could use an improved learning rule and apply the MST to the problem of estimating numerosity of weakly labeled objects. For this we have used a visual counting task of handwritten even digits from the MNIST data-set. For successful learning, the model needed to solve the binding problem

using the single scalar label and count the number of even digits in each image. Finally we have asked the question, whether a single spiking neuron model is able to compete against more complex computer vision methods, rate-based artifical neural network models with approximately the same parameter space and training data. For this we have compared the MST against a conventional convolutional neural network of seven layers and assessed the performance using the *root mean-squared error* (RMSE) of wrongly counted digits. We have found, that the improved MST model can outperform the ConvNet in this setting, which needed 5-10x more training epochs to reach the MST performance. While in this work we specifically focused on the computational capabilities of the single-neuron model, the same model and learning rule can also be used to create more complex and layered networks. We leave the study of complex networks of multiple, interconnected Multi-Spike Tempotrons up for farther research. We also studied and quantified the learning and generalization performance of the model in the general context of task-related spiking activity in the neocortex. Since the exact spike-train statistic in the cortex is still unknown, we specifically studied several, different input statistics that deviate from homogeneous Poisson, which due to its mathematical convenience is the commonly used model of spiking statistics of the visual cortex and has also been used in the original work of the MST model (Gütig, 2016). We showed, that by choosing different and biologically more realistic input statistics, the MST model exhibits large variance with regard to training error and more importantly with regard to generalization on unseen inputs. In order to overcome this issue, we have successfully proposed a modified learning rule that uses adaptive learning rates per synapses and smoothing over past gradients instead of the original Momentum-based learning. We evaluated both methods on data-sets with different input statistics that resemble task-related spiking activity in the cortex and under different levels of background noise complexity. We were able to show, that the adaptive learning rate method performs consistently better as compared to Momentum in terms of variability of training error and generalization. The modified learning rule has a regularizing effect and prevents the model from overfitting to the training

data, without modifying the model's equation and gradient derivation. We conclude that using only the basic quantity accessible within the brain, spikes of neurons, we showed that it is possible for a single neuron to perform basic numerical recognition tasks despite complex and noisy input. We suggest that using spikes to represent numerosity with a single neuron can be a beneficial strategy especially for small-brained organisms which economize on their number of neurons. To support further research we make our code publicly available (Rapp and Stern, 2018).

4. Limitations of the study

(1) In this work we specifically focus on the computational capabilities of a single neuron and what kind of cognitive tasks it is able to solve. Hence we did not explore complex networks of multiple Tempotron neurons which in general would certainly be able to perform much better.

Acknowledgments

A significant portion of this work has been conducted during the OIST Computational Neuroscience Course and has been supported by accommodations and travel grants from the Okinawa Institute of Science and Technology (OIST). HR is supported by the German Research Foundation (grant no. 403329959 to MN) within the Research Unit 'Structure, Plasticity and Behavioral Function of the Drosophila mushroom body'(FOR 2705)".

Author Contributions

Conceptualization, H.R., MP.N. and M.S.; Methodology, H.R and M.S.; Writing – Original Draft, H.R., Writing – Review and Editing, H.R., MP.N. and M.S.

Declaration of Interests

The authors declare no competing interests.

References

Avarguès-Weber, A., Deisig, N., Giurfa, M., 2011. Visual cognition in social insects. Annual review of entomology 56, 423–443.

Avarguès-Weber, A., Dyer, A.G., Combe, M., Giurfa, M., 2012. Simultaneous mastering of two abstract concepts by the miniature brain of bees. Proceedings of the National Academy of Sciences 109, 7481–7486.

Avarguès-Weber, A., Giurfa, M., 2013. Conceptual learning by miniature brains. Proceedings of the Royal Society B: Biological Sciences 280, 20131907.

Bengio, Y., Léonard, N., Courville, A., 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. `arXiv:1308.3432`.

Bhattacharya, B.S., Furber, S.B., 2010. Biologically inspired means for rank-order encoding images: A quantitative analysis. IEEE Transactions on Neural Networks 21, 1087–1099. doi:`10.1109/TNN.2010.2048339`.

Bi, G.q., ming Poo, M., 2001. Synaptic modification by correlated activity: Hebb's postulate revisited. Annual review of neuroscience 1, 139–166.

Bohte, S.M., Kok, J.N., Poutré, H.L., 2000. Spikeprop: backpropagation for networks of spiking neurons, in: ESANN.

Caporale, N., Dan, Y., 2008. Spike timing–dependent plasticity: a hebbian learning rule. Annual review of neuroscience 31, 25–46.

Chittka, L., Geiger, K., 1995. Can honey bees count landmarks? Animal Behaviour 49, 159 – 164. URL: `http://www.sciencedirect.com/science/article/pii/0003347295801634`, doi:`https://doi.org/10.1016/0003-3472(95)80163-4`.

Chittka, L., Niven, J., 2009. Are bigger brains better? Current biology 19, R995–R1008.

Dacke, M., Srinivasan, M.V., 2008. Evidence for counting in insects. Animal Cognition 11, 683–689. URL: http://dx.doi.org/10.1007/s10071-008-0159-y, doi:10.1007/s10071-008-0159-y.

Farkhooi, F., Muller, E., Nawrot, M.P., 2011. Adaptation reduces variability of the neuronal population code. Physical Review E 83, 050905.

Fomoro, . http://github.com/fomorians/counting-mnist. URL: http://github.com/fomorians/counting-mnist.

Gütig, R., 2016. Spiking neurons can discover predictive features by aggregate-label learning. Science 351.

Gütig, R., Sompolinsky, H., 2006. The tempotron: a neuron that learns spike timing-based decisions. Nature Neuroscience 9, 420 EP –. URL: http://dx.doi.org/10.1038/nn1643.

Howard, S., Avarguès-Weber, A., Garcia, J., Greentree, A., Dyer, A., 2018. Numerical ordering of zero in honey bees. Science 360, 1124–1126.

Huerta, R., Nowotny, T., García-Sanchez, M., Abarbanel, H.D.I., Rabinovich, M.I., 2004. Learning classification in the olfactory system of insects. Neural Computation 16, 1601–1640. URL: https://doi.org/10.1162/089976604774201613, doi:10.1162/089976604774201613, arXiv:https://doi.org/10.1162/089976604774201613.

Huh, D., Sejnowski, T.J., 2017. Gradient descent for spiking neural networks URL: https://arxiv.org/abs/1706.04698, arXiv:1706.04698.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105. URL: http://papers.nips.cc/paper/

4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LeCun, Y., Cortes, C., 2010. MNIST handwritten digit database URL: http://yann.lecun.com/exdb/mnist/.

Lempitsky, V., Zisserman, A., 2010. Learning to count objects in images, in: Advances in neural information processing systems, pp. 1324–1332.

Memmesheimer, R.M., Rubin, R., Ölveczky, B.P., Sompolinsky, H., 2014. Learning precisely timed spikes. Neuron 82, 925 – 938. URL: http://www.sciencedirect.com/science/article/pii/S0896627314002566, doi:https://doi.org/10.1016/j.neuron.2014.03.026.

Menzel, R., Fuchs, J., Nadler, L., Weiss, B., Kumbischinski, N., Adebiyi, D., Hartfil, S., Greggers, U., 2010. Dominance of the odometer over serial landmark learning in honeybee navigation. Naturwissenschaften 97, 763–767. URL: http://dx.doi.org/10.1007/s00114-010-0690-y, doi:10.1007/s00114-010-0690-y.

Mochizuki, Y., Onaga, T., Shimazaki, H., Shimokawa, T., Tsubo, Y., Kimura, R., Saiki, A., Sakai, Y., Isomura, Y., Fujisawa, S., et al., 2016. Similarity in neuronal firing regimes across mammalian species. Journal of Neuroscience 36, 5736–5747.

Nawrot, M.P., 2010. Analysis and interpretation of interval and count variability in neural spike trains, in: Analysis of parallel spike trains. Springer, pp. 37–58.

Nicola, W., Clopath, C., 2017. Supervised learning in spiking neural networks with force training. Nature Communications 8, 2208. URL: https://doi.org/10.1038/s41467-017-01827-3, doi:10.1038/s41467-017-01827-3.

O'Connor, P., Gavves, E., Welling, M., 2017. Temporally efficient deep learning with spikes URL: https://arxiv.org/abs/1706.04159, arXiv:1706.04159.

OpenAI, . https://blog.openai.com/ai-and-compute/. URL: https://blog.openai.com/ai-and-compute/.

Pahl, M., Si, A., Zhang, S., 2013. Numerical cognition in bees and other insects. Frontiers in psychology 4, 162.

Polyak, B., 1964. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics 4, 1 – 17. URL: http://www.sciencedirect.com/science/article/pii/0041555364901375, doi:https://doi.org/10.1016/0041-5553(64)90137-5.

Rapp, H., Stern, M., 2018. Matlab implementation of multispike tempotron with adaptive learning. (doi:10.5281/zenodo.1247442). URL: https://doi.org/10.5281/zenodo.1247442.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks URL: https://arxiv.org/pdf/1506.01497, arXiv:1506.01497.

Rose, G.J., 2018. The numerical abilities of anurans and their neural correlates: insights from neuroethological studies of acoustic communication. Philosophical Transactions of the Royal Society B: Biological Sciences 373, 20160512.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117.

Seguí, S., Pujol, O., Vitrià, J., 2015. Learning to count with deep object features URL: https://arxiv.org/abs/1505.08082, arXiv:1505.08082.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition URL: https://arxiv.org/abs/1409.1556, arXiv:1409.1556.

Skorupski, P., MaBouDi, H., Galpayage Dona, H.S., Chittka, L., 2018. Counting insects. Philosophical Transactions of the Royal Society B: Biological Sciences 373, 20160513.

Song, S., M.K.D., Abbott, L.F., 2000. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nature neuroscience 3, 919.

Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4, 26–31.

Vasas, V., Chittka, L., 2019. Insect-inspired sequential inspection strategy enables an artificial network of four neurons to estimate numerosity. iScience 11, 85 – 92. URL: http://www.sciencedirect.com/science/article/pii/S2589004218302384, doi:https://doi.org/10.1016/j.isci.2018.12.009.

Yu, F., Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122.

Zenke, F., Ganguli, S., 2018. Superspike: Supervised learning in multilayer spiking neural networks. Neural Computation 30, 1514–1541. URL: https://doi.org/10.1162/neco{_}a{_}01086, doi:10.1162/neco{_}a{_}01086, arXiv:https://doi.org/10.1162/neco_a_01086. pMID: 29652587.

main figure titles and legends

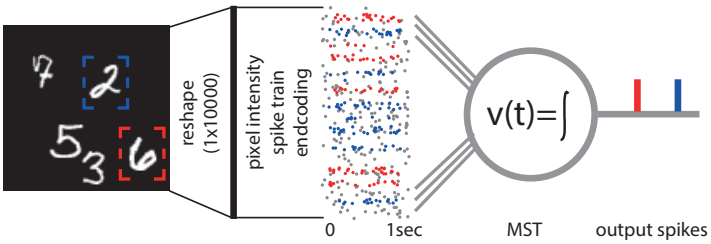

Figure 1: **Sketch showing how to count even digits with spikes:** Example image (100x100 px) with multiple digits from the couting MNIST data set. The image is reshaped to a single vector (1x10.000 px). Each pixel's intensity is encoded as a 1sec long spike train by a spike encoding frontend (see methods). This spike train is fed into a single Multi-Spike Tempotron (MST) with 10.000 synapses. The MST is trained to elicit exactly p output spikes where p is equal to the number of even digits present in the original image.

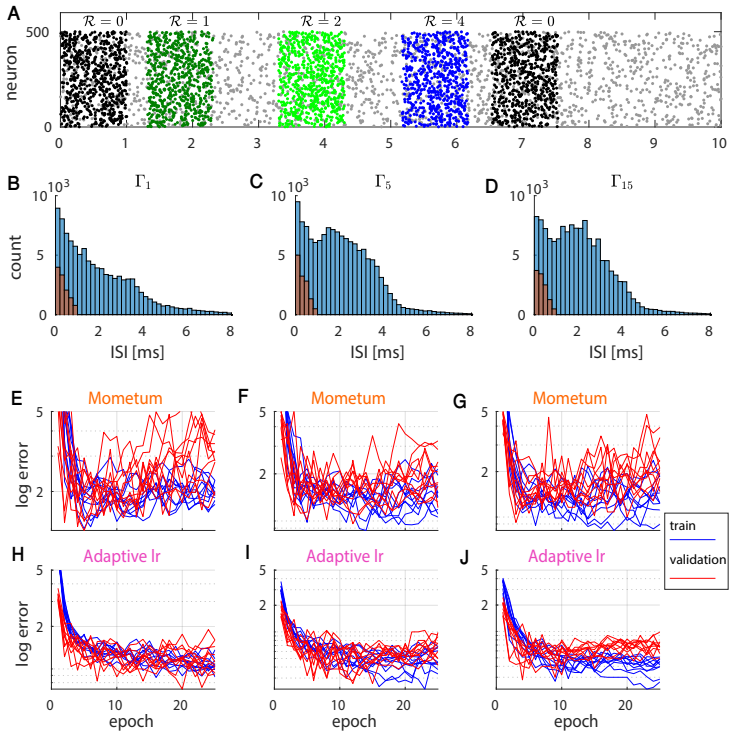

Figure 2: Training convergence for momentum and adaptive learning in multiple background noise distributions. **A.** A 10sec duration spike train input example. The spike train is composed of three patterns, each with a distinct reward (dark green, green, blue), background activity (gray) and two distracting patterns (black). The MST is supposed to fire $\Sigma_i \mathcal{R}_i = 7$ spikes over the whole sequence, whereby $\mathcal{R} = 0$ spikes for the distracting patterns and $\mathcal{R} \in \{1, 2, 4\}$ for the colored patterns. **B-D.** Inter-spike interval (ISI) histograms for the three different renewal processes. The histograms are shown for the complete input sequence (blue) and patterns only (brown). Patterns are simulated with gamma processes with different order parameter: **B.** Γ_1 (Poisson), **C.** Γ_5, and **D.** Γ_{15}. The patterns are embedded in 10sec inhomogeneous Poisson background activity. **E-J.** Learning curves (blue) and validation curves (red) for 10 independet simulation of the **E,H.** (Γ_1 (Poisson), **F,I.** Γ_5, and **G,J.** Γ_{15}) patterns. **E-G.** MST with momentum-based learning implementation (Gütig, 2016). **H-J.** MST with adaptive learning implementation. Learning convergence shows significantly more variance when using momentum compared to adaptive learning. The same is true for the validation error. This indicates that adaptive learning is capable of finding wider optima as compared to momentum.

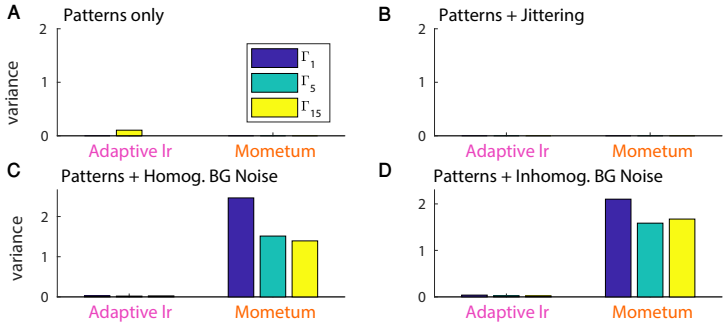

Figure 3: Generalization error variances for different data-sets with increasing noise complexity. For each data set the variance of the generalization error is shown for Momentum and adaptive learning (lr). Variance is measured after 10 training epochs where convergence is considered to be reached by both methods.

main tables and legends

Table 1: Results for the Counting MNIST experiment where the model should learn to count the number of even handwritten digits present in a given 100x100 pixel image. We evaluate each model with regard to it's complexity (number of parameters or number of synapses), and RMSE of wrongly counted digits (lower is better). The ConvNet, regular multilayer perceptron (MLP) and MST models have all been trained for 30 epochs on the same training set of 800 samples. Evaluation was done on an independent set of 800 validation samples. For reference we also report performance for naive models which *always-predict zero* and do *random-guessing*.

Counting MNIST Results		
Model	#Parameters	RMSE
ConvNet	26471	1.49
3-layer MLP (FoCal enc.)	960038	1.65
$MST_{adaptive}$ (naive enc.)	10000	2.34
$\mathbf{MST_{adaptive}}$ (FoCal enc.)	10000	**1.22**
$MST_{Momentum}$ (naive enc.)	10000	1.87
$MST_{Momentum}$ (FoCal enc.)	10000	1.28
always-zero	n/a	1.65
random guessing	n/a	2.50

Supplemental Information

4.1. Transparent Methods

4.1.1. Multispike Tempotron Model

The Multi-Spike Tempotron (MST) is a current-based leaky integrate-and-fire neuron model (Gütig, 2016). Its membrane potential, $V(t)$, follows the dynamical equation:

$$V(t) = V_{rest} + \sum_{i=1}^{N} \omega_i \sum_{t_i^j < t} K(t - t_i^j) - \vartheta \sum_{t_{spike}^j < t} e^{-\frac{t - t_{spike}^j}{\tau_m}} \quad (1)$$

where t_i^j denotes the time of spike number j from the input source (presynaptic) number i, and t_{spike}^j denotes the time of spike number j of the Tempotron neuron model. For mathematical convenience the resting potential is chosen to be $V_{rest} = 0$ and thus is being dropped from now on. Every input spike at t_i^j contributes to the potential by the kernel:

$$K(t - t_i^j) = V_{norm}(e^{-\frac{t - t_i^j}{\tau_m}} - e^{-\frac{t - t_i^j}{\tau_s}}) \quad (2)$$

multiplied with the synaptic weight ω_i of input source i. These synaptic input weights are learned via the gradient decent algorithm. The kernel is normalized to have its peak value at 1 with $V_{norm} = \eta^{\eta/(\eta-1)}/(\eta - 1)$ and $\eta = \tau_m/\tau_s$ where τ_m and τ_s are the membrane time constant and the synaptic decay time constant. The kernel is causal, it vanishes for $t < t_i^j$. When $V(t)$ crosses the threshold ϑ the neuron emits a spike and is reset to $V_{rest} = 0$ by the second term in equation 1.

In order to have the neuron emit the required number of p spikes in response to a specific concept (implemented as synaptic input spike pattern) the weights ω_i are modified. Since the required spike numbers are non-differentiable discrete numbers the gradient for the weights is derived from the spiking threshold. The weights need to be changed such that the neuron's voltage would reach a critical threshold ϑ_p^* that would coincide with its threshold ϑ, that would be crossed exactly p times to generate the p desired spikes. This loss function is called

Spike-Threshold Surface (STS). Hence the appropriate gradient can be described by:

$$\Delta\omega = \eta\lambda\vec{\nabla}_{\vec{\omega}}\vartheta_p^* \tag{3}$$

Where $\eta \in \{-1, 1\}$ controls whether to increase or decrease the number of output spikes towards the k required, λ is the learning rate parameter that controls the size of the gradient step and $\vec{\nabla}_{\vec{\omega}}\vartheta_p^*$ is the gradient of the critical voltage threshold with respect to the synaptic weights. In practice, multiple concepts are presented and the learning signal is the sum of spikes that are required by the collection of concepts to generate. To evaluate the expression in eq. 3 the properties of the p-th spike time t_p^* in which the potential reaches the critical threshold ϑ_p^* are used. Hence

$$\vartheta_p^* = V(t_p^*) = V(t_s^j) \quad \text{where } t_s^j \text{ are all spike times before } t_p^* \tag{4}$$

In what follows a recursive expression is derived for the gradient in equation 3 using equations 1 and 2.

For simplify the recursive equation for the gradient is derived for a single component ω_i of the vector $\vec{\omega}$. The generalization to $\vec{\omega}$ is immediate.

Here m is denoted as the number of spikes the neuron fires before t_p^*. So that for each synapse i the following relationships hold for the gradient:

$$\vartheta_{i,p}^{*\prime} = \frac{\mathrm{d}}{\mathrm{d}\omega_i}\vartheta_p^* = \frac{\mathrm{d}}{\mathrm{d}\omega_i}V(t_p^*) = \frac{\mathrm{d}}{\mathrm{d}\omega_i}V(t_s^j) \tag{5}$$

$$= \frac{\partial}{\partial\omega_i}V(t_p^*) + \sum_{j=1}^{p}\frac{\partial}{\partial t_s^j}V(t_p^*)\frac{\mathrm{d}}{\mathrm{d}\omega_i}t_s^j \tag{6}$$

In the last equation, the local maximum at $V(t_p^*)$ was used for the vanishing $\frac{\partial V(t_p^*)}{\partial t_p^*} = 0$. All terms that do not depend on ω_i were dropped as well. For all spike times $k \in 1, .., m$ that the neuron fires before t_p^* the following relationships

hold:

$$\frac{\mathrm{d}}{\mathrm{d}\omega_i} V(t_s^k) = \frac{\partial}{\partial \omega_i} V(t_s^k) + \sum_{j=1}^{k} \frac{\partial}{\partial t_s^j} V(t_s^k) \frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^j \tag{7}$$

$$\frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^k = \frac{1}{\dot{V}(t_s^k)} \left[\vartheta_i^{*\prime} - \frac{\partial}{\partial \omega_i} V(t_s^k) - \sum_{j=1}^{k-1} \frac{\partial}{\partial t_s^j} V(t_s^k) \frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^j \right] \tag{8}$$

$$\dot{V}(t_s^k) = \frac{\partial}{\partial t} V(t) \bigg|_{t=t_s^{k-}} \text{evaluated from the left before spike reset} \tag{9}$$

To solve equation 5 for $\vartheta_{i,p}^{*\prime}$, the right hand side of eq 8 is being refactored into:

$$\frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^k = \frac{1}{\dot{V}(t_s^k)} \left[\vartheta_{i,p}^{*\prime} A_k + B_k \right] \tag{10}$$

$$A_k = 1 - \sum_{j=1}^{k-1} \frac{A_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_s^k) \tag{11}$$

$$B_k = -\frac{\partial}{\partial \omega_i} V(t_s^k) - \sum_{j=1}^{k-1} \frac{B_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_s^k) \tag{12}$$

Similarly the same recursion formula holds for t_p^*. By plugging 10 into 5 the gradient $\vartheta_{i,p}^{*\prime}$ for each vector component i of ω can be expressed as:

$$A_* = 1 - \sum_{j=1}^{m} \frac{A_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_p^*) \tag{13}$$

$$B_* = -\frac{\partial}{\partial \omega_i} V(t_p^*) - \sum_{j=1}^{m} \frac{B_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_p^*) \tag{14}$$

$$\vartheta_{i,p}^{*\prime} = -\frac{B_*}{A_*} \tag{15}$$

The learning rate λ is global for all synaptic weights. Hence, the gradient descent takes an equal size step along all directions. If this parameter is too small the training process will take very long, but if it's too big the algorithm might miss an optimum within the error surface and never converge to a desired solution. Hence, tuning this learning rate is important to achieve decent training speed. A possible approach (Gütig, 2016) to avoid these problems is to update

the weights according to the accumulating error, using the *Momentum* heuristic:

$$\begin{aligned}\Delta\omega^{Momentum} &= \alpha\Delta\omega(t-1) + \Delta\omega(t) \\ &= \alpha\Delta\omega(t-1) + \eta\lambda\vec{\nabla}_{\vec{\omega}}\vartheta_k^*, \end{aligned} \tag{16}$$

where α is the *Momentum* parameter.

4.1.2. Adaptive input weight learning and gradient smoothing

We propose here to use an adaptive learning approach for the weight updates instead of the *Momentum* heuristic. The proposed algorithm fits each input synapse with its own update rate and by doing so it takes into account that each synapse contributes to the overall update with a different level of importance. For example, updates should be larger for directions that provide more consistent information across examples. The RMSprop (Root Mean Square (back-)propagation) (Tieleman and Hinton, 2012) is a possible approach to achieve this. It was successfully used in deep learning for training mini-batches. It computes an adaptive learning rate per synapse weight ω_i as a function of its previous gradient steps :

$$\begin{aligned} v_i(t) &= \gamma v_i(t-1) + (1-\gamma)(\Delta\omega_i(t))^2 \\ \Delta\omega_i^{Adaptive}(t) &= \frac{\eta\lambda}{\sqrt{v_i(t)}}\nabla_{\omega_i}\vartheta_k^* \end{aligned} \tag{17}$$

4.1.3. Numerosity of even handwritten digits

In this section 2.1 we have considered the problem of estimating numerosity, namely the problem of counting even digits using the MNIST (LeCun and Cortes, 2010) data-set of handwritten digits $0-9$. Following Seguí et al. (2015) and Fomoro we generated new images of size 100x100 pixels which contain a random set of up to 7 MNIST digits (that can include between 0 to 6 digits that represent even numbers). These digits are randomly positioned within the image (fig 1). Rejection sampling is used to ensure digits are well separated. Each such image is weakly labeled with the total number of even digits present in that image. The data set is imbalanced and contains significantly more samples showing

zero even digits, rendering this a difficult task. Thus a naive model that always predicts zero is already able to achieve a better performance than chance level (random-guessing). The model is supposed to learn to count the number of even digits given a single scalar label in order to solve this task correctly. All models are trained on a training set for several epochs. After each training epoch the parameters learned at this point are fixed and a separate validation data set is fed into the model to assess the model's generalization capability to unseen data. For the Multi-Spike Tempotron the images have to be encoded as spiketrains. We first consider a naive spike-encoding, which encodes each individual pixel as 3s long spiketrain generated by a Γ_5 process with the rate proportional to the pixel's intensity (grey value). This type of encoding is naive in the sense, that it considers each pixel to be independent and thus does not exploit local spatial correlations of images. Next we consider a more sophisticated spike-encoding frontend, the *Filter-Overlap Correction Algorithm* (FoCal), a model of the fovea centralis (Bhattacharya and Furber, 2010). This encoding algorithm makes use of spatial correlations in order to reduce the amount of redundant information. This is similar to the convolutional filters embedded in current deep neural networks (Simonyan and Zisserman, 2014; Krizhevsky et al., 2012). For reference, we train a conventional ConvNet architecture that has been shown to successfully accomplish this task when trained on 100000 samples Fomoro. The architecture uses several layers (conv1 - MaxPool - conv2 - conv3 - conv4 - fc - softmax) and includes recently discovered advances like strided and dilated convolutions (Yu and Koltun, 2015). To train the ConvNet we use the ADAM (Kingma and Ba, 2014) optimizer which has been found to be an effective optimizer for training ConvNets. For the MST model we use our adaptive learning rate method and the originally proposed Momentum method. Since we want to evaluate the models with regard to computational and sample efficiency all models are trained for 30 epochs on the same training set of 800 images and are evaluated on an independent test set of 800 unseen images.

4.1.4. Generic task-related inhomogenous activity

In this task we study the more general case of counting arbitrary, task dependent features. To this end we use 1sec long spike trains generated from point processes as a model of complex spatio-temporal patterns that represent task dependent features. An input to the MST model consists of a sequence of such features, each of which labeled with a specific category (reward). The patterns are superimposed onto a background spike train and serve as input to the MST model. Similar to the task in (Gütig, 2016) the MST model is trained to respond with spikes for each feature occurrence where the number of spikes per feature depends on its category. It has been shown that in-vivo cortical spiking activity is typically more regular then Poisson (Mochizuki et al., 2016; Nawrot, 2010). In general any correlated stimuli input is expected to deviate from Poisson (Farkhooi et al., 2011). Moreover, input is generally non-homogenous, i.e. time-varying. However in (Gütig, 2016) only homogeneous Poisson statistic of input patterns and background were considered. In this task we study learning capabilities of the Multi-Spike Tempotron when the statistic of input patterns deviates from the Poisson statistic and when the background statistic is non-homogeneous, i.e. time-varying as under realistic noisy conditions. We construct three data-sets, each including 9 generated patterns. Out of this 9 patterns, 5 patterns are considered to be *task-related* and are associated with some positive reward $\mathcal{R}$. The remaining 4 patterns are considered to be distractor patterns with reward 0. All patterns are generated as 1sec long spike trains by drawing instantaneous firing rates from three different point processes (renewal processes): Γ_1 representing the homogeneous Poisson process, Γ_5, and Γ_{15} represent Gamma-Processes (simulated by thinning a Poisson process with 5x and 15x of the Gamma process' intensity) with a fixed intensity (or rate) of $\lambda = 0.89$ spike events per second (fig. 2 B-D.). Each pattern is associated with a fixed, positive integer reward $\mathcal{R}_i \in [0, 9]$. Input spike trains of 10 sec and 500 pre-synaptic inputs are generated by simulating a background activity spike train and superimposing patterns. They are assembled by drawing a

random number of patterns from a Poisson distribution of mean 5 patterns (with replacement). These patterns are randomly positioned within those 10 sec but are not allowed to overlap (an example of an input spike-train is shown in fig. 2 A). The training target for each of such input spike train is determined as the sum over all individual rewards $\Sigma_i \mathcal{R}_i$ of each occurring pattern. We evaluate learning under different noise conditions (fig. 3 bottom): *patterns only*, *patterns + spike jittering*, *patterns + homogenous Poisson background activity* and finally *patterns + inhomogenous Poisson background activity*. The homogenous background activity is drawn from a stationary Poisson process ($\lambda = 0.3$ spikes/sec) while for the inhomogenous case the instantaneous firing rates are modulated by $\lambda(t) = sin(\frac{10\pi}{10000}t) + (\frac{4\pi}{10}\xi(t))$ where $\xi(t)$ is noise drawn from a standard normal distribution. To test the robustness against noise within the patterns themselves, spike jittering has been used. This is performed by additive noise to the pattern's spike times drawn from $\mathcal{N}(0, 4dt)$ where $dt = \frac{1}{1000}$ is the time resolution of the simulations.