

Advantages of heterogeneity of parameters in spiking neural network training

Nicolas Perez-Nieves (nicolas.perez14@imperial.ac.uk)

Department of Electrical and Electronic Engineering, Imperial College London, UK

Vincent C. H. Leung (chi.leung14@imperial.ac.uk)

Department of Electrical and Electronic Engineering, Imperial College London, UK

Pier Luigi Dragotti (p.dragotti@imperial.ac.uk)

Department of Electrical and Electronic Engineering, Imperial College London, UK

Dan F. M. Goodman (d.goodman@imperial.ac.uk)

Department of Electrical and Electronic Engineering, Imperial College London, UK

Abstract

It is very common in studies of the learning capabilities of spiking neural networks (SNNs) to use homogeneous neural and synaptic parameters (time constants, thresholds, etc.). Even in studies in which these parameters are distributed heterogeneously, the advantages or disadvantages of the heterogeneity have rarely been studied in depth. By contrast, in the brain, neurons and synapses are highly diverse, leading naturally to the hypothesis that this heterogeneity may be advantageous for learning. Starting from two state-of-the-art methods for training spiking neural networks (Nicola & Clopath, 2017; Shrestha & Orchard, 2018), we found that adding parameter heterogeneity reduced errors when the network had to learn more complex patterns, increased robustness to hyperparameter mistuning, and reduced the number of training iterations required. We propose that neural heterogeneity may be an important principle for brains to learn robustly in real world environments with highly complex structure, and where task-specific hyperparameter tuning may be impossible. Consequently, heterogeneity may also be a good candidate design principle for artificial neural networks, to reduce the need for expensive hyperparameter tuning as well as for reducing training time.

Keywords: Spiking neural networks; learning; heterogeneity

Introduction

In order to understand how a complex system such as the brain works, it can be very useful to start by studying a simplified version of it and then adding complexity step by step. The simplifying assumption of heterogeneity of neural and synaptic parameters makes theoretical analysis and simulation much easier, and from a statistical point of view heterogeneity has sometimes been thought to only add noise to the system. However, recent works have suggested that the heterogeneity present in the brain may play several important functional roles (Maass, Natschger, & Markram, 2002; Rotman & Klyachko, 2013; Kilpatrick, Ermentrout, & Doiron, 2013; Lengler, Jug, & Steger, 2013; Wu et al., 2018), notably heterogeneous tuning may be important for robust population coding of complex stimuli (Marsat & Maler, 2010; Goodman, Benichoux, &

Brette, 2013). Until recently, training spiking neural networks (SNNs) to carry out non-trivial tasks was very challenging, but recent methods have begun to change that (Nicola & Clopath, 2017; Shrestha & Orchard, 2018; Zenke & Ganguli, 2018). In this paper, we test the hypothesis that neural heterogeneity can contribute to learning in terms of accuracy, speed and robustness, by adding heterogeneity to two of these methods that previously only used homogeneous neurons.

Heterogeneity improves learning at multiple temporal scales

We first train a recurrent spiking neural network to reproduce a supervisor signal composed of a sum of two sinusoids at different frequencies f_1 (fixed at 20 Hz) and f_2 (varied between 1-19 Hz). We use leaky integrate and fire (LIF) neurons and double exponential synapses with decay time constant τ . The training is performed using the FORCE learning method as in Nicola and Clopath (2017). After training the network for 8s we test if the network has learned the supervisor by turning off the learning rule and the supervisor for 10s. We measure the learning error ϵ measured as the logarithm of the Euclidean norm between the supervisor signal and the network output.

We tested four different synaptic constant configurations: **Homogeneous Fast**, where all time constants of all synapses are set to 20ms, **Homogeneous Slow**, where all time constants of all synapses are set to 100ms, **Heterogeneous Double**, where we randomly allocate either a fast (20ms) or a slow (100ms) synaptic constant to each synapse with probability 0.5 and **Heterogeneous Gamma** where the value of each synaptic constant is drawn from a Gamma distribution. For numerical efficiency, all synaptic time constants are the same within a neuron receiving synapses, but may be different between different neurons.

We tested how these different configurations affected learning performance and we found that while the training error for the Fast synapse configuration is the lowest the test error is significantly worse (Figure 1). This is especially remarkable at $f_2 < 10\text{Hz}$ where the supervisor signal has structure at two very different time scales.

Two considerations may explain the poor results of using only a single fast time constant. First, the Homogeneous Fast





Figure 1: Training and test error averaged over 10 trials for a 1000 LIF network trained using the FORCE method to reproduce a sum of two sinusoids signal at frequencies $f_1 = 20\text{Hz}$ and f_2 . Synapses were either all fast or all slow (Single Fast, Single Slow), split randomly into a fast and slow group (Double), or fully randomly distributed (Gamma).

configuration results in a network that has a shorter memory than the rest. Secondly, during the training phase the weight update is performed every 2.5ms. Having a shorter memory means that the network stores more information about the last 2.5ms than the other configurations, thus allowing it to perform a more effective weight update at every learning step. However, it cannot remember what happened at longer time scales which means that when training stops it stays with a weight configuration that is too localised at a particular section of the signal to properly generalise at longer time scales.

By adding a second slower time constant (Heterogeneous Double configuration) or several time constants distributed following a Gamma distribution (Heterogeneous Gamma configuration) we introduce the memory required to properly generalise at longer time scales as shown in the testing error for these configurations at $f_2 < 10\text{Hz}$ (Figure 2).

We also note that the Gamma configuration performs slightly worse than the Heterogeneous Double configuration. This may be due to the fact that two time constants exactly matches the stimulus structure, which would not be the case for a more realistic stimulus with temporal structure at multiple timescales. In the next section, we therefore tested performance with this more realistic signal.

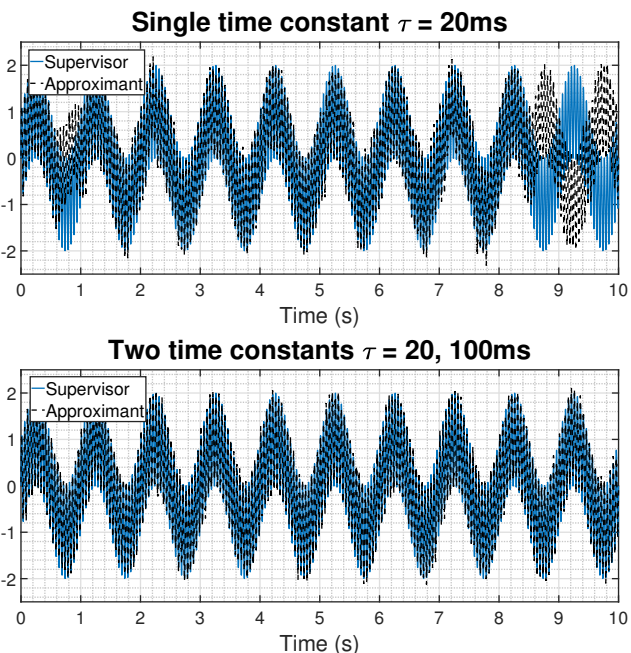


Figure 2: An example of learning the sum of a 1Hz and 20Hz sinusoid using two synaptic configurations: top, all synapses fast; bottom, synapses randomly assigned to a fast or slow group. In the case where all synapses are fast, the network struggles to retain the longer time scale variation of the signal.

Heterogeneity improves robustness

Nicola and Clopath (2017) show that the network performance is highly dependent on the hyperparameters G and Q chosen. Here, we investigate how robust the network is by varying the network size while keeping these hyperparameters constant.

We first tune G and Q for a network of 1000 LIF neurons trained to reproduce a spectrogram of a zebra finch bird song. This signal was chosen due to its high spatio-temporal complexity (Figure 3). Then, we vary the size of the network under the Homogeneous Fast, Heterogeneous Double and Gamma configurations. We do not use the Homogeneous slow configuration due to its poor training and testing error in the 1D signal.

We computed the test error for networks with sizes from 500 to 8000 neurons (Figure 4). As expected, the lowest test error was obtained with the 1000-neuron network, matching the size for which the hyperparameters G and Q were chosen. We see that beyond 3000 neurons the Homogeneous network fails to reproduce the signal and after beyond 6000 the same happens for the Heterogeneous Double configuration. The testing error of the Gamma configuration remains almost the same for all network sizes. This reveals how a fully heterogeneous distribution of synaptic constants can remarkably improve the robustness of learning to hyperparameter mistuning.

To demonstrate how well the heterogeneous versus homo-

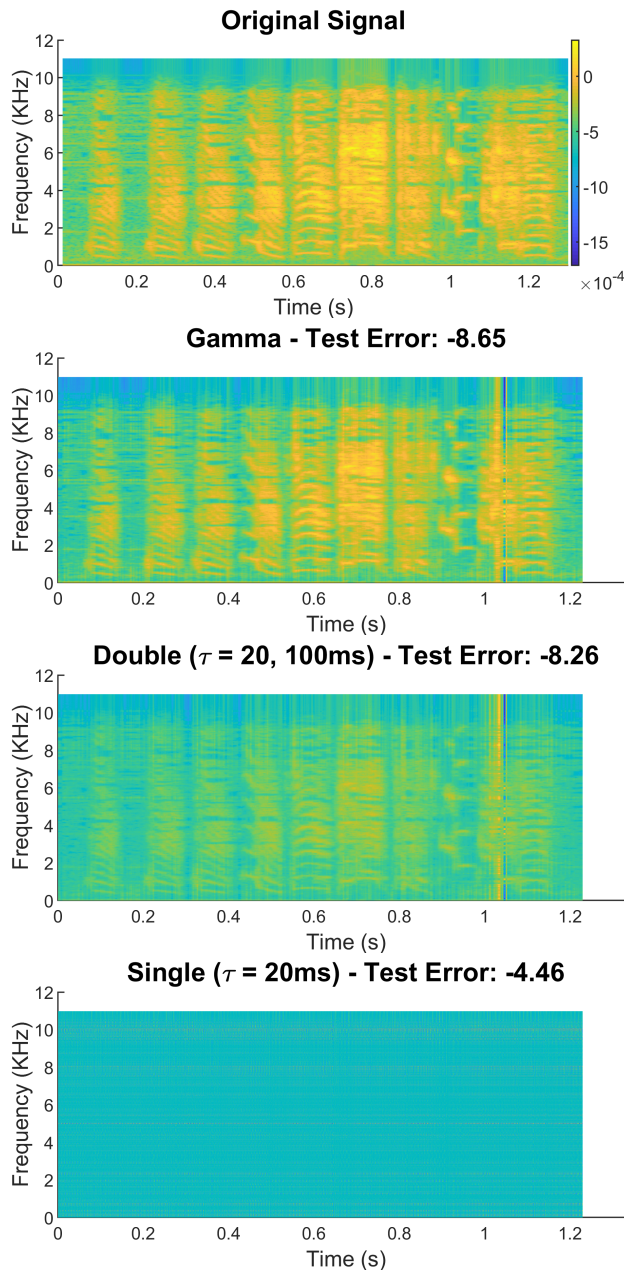


Figure 3: Original zebra finch spectrogram obtained from https://en.wikipedia.org/wiki/Zebra_finch and resulting spectrograms obtained on testing for the 4000-neuron network on three different synaptic configurations

genous networks are able to learn the spectrogram, we computed reconstructions with a 4000-neuron network (Figure 3). Notice how when using the single time constant the spectrogram is completely lost while the Gamma spectrogram is remarkably similar to the original.

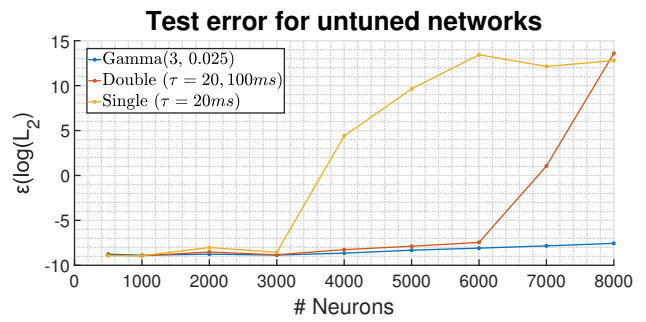


Figure 4: Test error of LIF networks of different sizes with hyperparameters tuned to reproduce a zebra finch spectrogram on a 1000-neuron network.

Heterogeneity reduces learning time

We also explored the advantages of heterogeneity on a state-of-the-art gradient descent learning method, Spike Layer Error Reassignment (SLAYER) (Shrestha & Orchard, 2018) for error back-propagation in spiking neural networks. We trained a spiking neural network to classify Neuromorphic-MNIST (N-MNIST) digits dataset (Orchard, Jayawant, Cohen, & Thakor, 2015), a spiking version of the original frame-based MNIST dataset. We follow the original implementation in choices of spike response kernels and network structure. The network consists of 4 layers with a $34 \times 34 \times 2$ input, followed by 3 dense layers of output size 512, 512 and 10 respectively.

To investigate the relationship between heterogeneity and learning time, the only parameter changed is the time constant of the spike response kernels. In the original implementation, the synaptic models used in the network were homogeneous with a single time constant $\tau=1$ ms. We experiment the heterogeneity by introducing three synaptic time constants $\tau=1$ ms, 2ms, 4ms, which was then evenly distributed to the neurons in the network. Results show that the time for training and testing accuracy to converge was greatly reduced in the heterogeneous network (Figure 5).

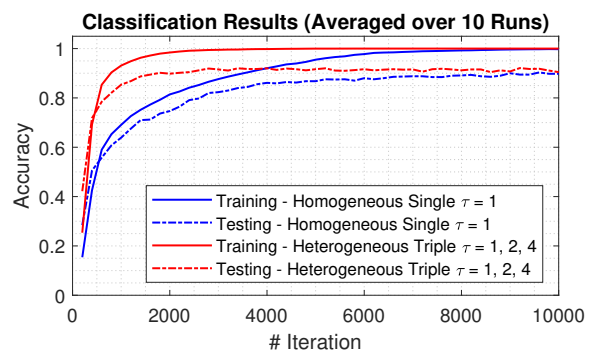


Figure 5: Mean Classification results of NMNIST dataset over 10 independent trials using SLAYER

Conclusion and Future Work

We found that adding time constant heterogeneity can dramatically improve the accuracy, robustness and convergence rate for training spiking neural networks across a range of tasks and training methods. This suggests that heterogeneity may be an important principle the brain uses to cope with difficult real world environments. It remains to be seen whether these results are specific to the training methods considered, and to heterogeneity of synaptic time constants in particular, or whether they generalise to other methods and parameters such as thresholds, refractory times, etc.

References

- Goodman, D. F., Benichoux, V., & Brette, R. (2013). Decoding neural responses to temporal cues for sound localization. *eLife*, *2*, 1–21. doi: 10.7554/eLife.01312
- Kilpatrick, Z. P., Ermentrout, B., & Doiron, B. (2013). Optimizing working memory with heterogeneity of recurrent cortical excitation. *Journal of Neuroscience*, *33*(48), 18999–19011. doi: 10.1523/JNEUROSCI.1641-13.2013
- Lengler, J., Jug, F., & Steger, A. (2013, 12). Reliable neuronal systems: The importance of heterogeneity. *PLOS ONE*, *8*(12), 1–10. doi: 10.1371/journal.pone.0080694
- Maass, W., Natschlger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*(11), 2531–2560. doi: 10.1162/089976602760407955
- Marsat, G., & Maler, L. (2010). Neural heterogeneity and efficient population codes for communication signals. *Journal of Neurophysiology*, *104*(5), 2543–2555. (PMID: 20631220) doi: 10.1152/jn.00256.2010
- Nicola, W., & Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nature Communications*, *8*(1), 1–15. doi: 10.1038/s41467-017-01827-3
- Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, *9*, 437. doi: 10.3389/fnins.2015.00437
- Rotman, Z., & Klyachko, V. A. (2013). Role of synaptic dynamics and heterogeneity in neuronal learning of temporal code. *Journal of Neurophysiology*, *110*(10), 2275–2286. (PMID: 23926043) doi: 10.1152/jn.00454.2013
- Shrestha, S. B., & Orchard, G. (2018). Slayer: Spike layer error reassignment in time. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems 31* (pp. 1412–1421).
- Wu, S., Zhang, Y., Cui, Y., Li, H., Wang, J., Guo, L., . . . Guo, D. (2018, Mar). Heterogeneity of Synaptic Input Connectivity Regulates Spike-based Neuronal Avalanches. *arXiv e-prints*, arXiv:1803.06180.
- Zenke, F., & Ganguli, S. (2018). SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*. doi: 10.1162/neco.a.01086