Biologically-inspired Deep-learning Workshop

Laura Bernáez Timón[1], Maximilian F. Eggl[2], Will Greedy[3]
Joseph Pemberton[3], Rui Ponte Costa[3]

Abstract

A specialized workshop, the first in a series of workshops, for young researchers took place in the spring of 2023 in Guntersblum. This workshop was funded by the "Begegnungszonen" program of the Joachim Herz Stiftung and was designed to provide a novel workshop at the intersection of machine learning and neuroscience. The spirit of the funding program supported this workshop's aim: "supporting the design and implementation of interdisciplinary events for young scientists". To this end, 16 doctoral students and postdoctoral researchers were brought together to expose them to relevant and interesting issues in the field of "Biologically-inspired Deep Learning". The speaker for the 2023 workshop on the topic of the credit assignment problem was Prof. Ponte Costa from the Department of Computer Science of the University of Bristol. Two of his doctoral students, Will Greedy and Joseph Pemberton, supported him. This report briefly introduces the workshop series' underlying idea, a motivation for the study and the importance of credit assignment in bio-inspired deep learning, and a representative exposé of the students' work during the workshop.

1 The Workshop format

The exchange with established researchers and early-career scientists is invaluable to scientific progress. It allows new ideas to circulate through the community while providing a network for new and upcoming talent. To foster this exchange and enable interdisciplinary networking among early career scientists and renowned experts, a workshop was held in Guntersblum, Germany, and funded by the "Begegnungszonen" program of the Joachim Herz Foundation.

A novel workshop format not common in the neuroscientific community was utilized to accomplish this exchange while providing an avenue for participants to learn about bio-logically inspired deep learning. The current standard format of workshops in neuroscience consists of longer time formats, several invited speakers to guide participants and a more lecture-based approach to learning. This format has proven effective, but the format's passive learning and larger number of experts means that direct interactions between students and established researchers may be lost. To augment the current set of workshops, we introduce our novel format, which takes place over a shorter time and encourages direct learning akin to on-the-job training. Hands-on demonstration by working through small-scale yet meaningful examples leads to pedagogical exposure to advanced concepts that may be difficult to understand if solely covered in lectures.

Additionally, the format is designed to be fundamentally interdisciplinary. Interdisciplinarity is becoming increasingly important in an interconnected science world, and thus, by bringing together scientists from a variety of different fields, the workshop provides a robust exchange of ideas and technologies from different scientific disciplines that enables the participants to become better scientists.

The format of this workshop is directly inspired by the Young ERCOFTAC workshop series, an initiative of Peter Schmid, Shervin Bagheri and Taraneh Sayadi, and which M. Eggl had the pleasure of attending during his PhD. This ongoing workshop series provides a platform for early-career researchers to be exposed to state-of-the-art research within the fluid dynamics community.

[1]Polytechnic University of Catalonia
[2]University of Mainz
[3]University of Bristol

We mirrored their approach by inviting 15-20 PhD students and Postdoctoral Fellows to the Weingut Domhof, a small remote location 30 km south of Mainz. The Domhof, a historical winery, has been converted into a hotel while still retaining its wine-making tradition. It features a quiet and rural surrounding ideal for work in a secluded yet pleasant environment. The remote nature of the accommodation requires every student's active participation in daily activities, such as communal meals. These activities further contribute to the quick and lasting bonding of the participants that often lasts beyond the duration of the workshop.

The scientific program format was then devised to introduce the students (who had very different backgrounds and were at various stages of their scientific careers) to the workshop topic and prepare them for the group work in the days ahead. After an initial set of introductory talks on machine learning, neuroscience and the particular workshop topic by the invited speaker, the organizing committee (L. Bernaez Timon and M. Eggl), the invited speaker (R. Ponte Costa) and the invited speaker's graduate students (W. Greedy and J. Pemberton) divided the workshop participants into groups of three to four students according to their skill level, experience and scientific maturity. Each group was then presented their research topic, which covered a particular aspect of the expertise of the invited speaker. The scope of the projects was limited to typical laptop problems that required expertise from a set of different fields. While working in their groups, they were closely supervised by the invited speaker. The direct involvement in a time-limited project and the immediate access to guidance and expertise yielded a more long-lasting result than a passive lecture series. Although each student group concentrated on one aspect of the technique, they all benefited from exposure to the other projects during the final presentation of the results; in this manner, the students were not only proficient in their own project but knowledgeable in other related details.

Thus, with this format in mind, the goals of our workshop are as follows: *i)* provide students with an exposure to the ever-growing field of Biologically inspired deep learning, *ii)* foster networking between the students by arranging them in groups and giving them state-of-the-art research projects to complete and *iii)* presenting the participants an opportunity to meet and discuss with experts in the field.

After receiving feedback from our students, we are confident we have achieved all these goals. In future iterations, this workshop can establish itself as a significant opportunity for young scientists to study problems at the intersection of neuroscience and machine learning. By providing specialized training opportunities and by encouraging the formation of collegial networks and connections, we hope to encourage, educate and foster the next generation of scientists in these interesting and important fields.

2 Topic of the workshop: Credit assignment

Deep learning (DL) and artificial neural networks (ANNs) dominate many facets of human society. From image recognition (Pak and Kim, 2017), natural language processing (Otter et al., 2020; Brown et al., 2020) to competitive games (Silver et al., 2017) and protein folding (Ruff and Pappu, 2021), DL has become an increasingly more important feature of the algorithmic toolbox. However, their power is often based on adding more layers or nodes to the network, which can lead to energy requirements and training times that are only available to big tech companies and far exceed the resources of academic research groups (Strubell et al., 2019). For further breakthroughs, fundamental changes need to be made that change the structure of the ANNs themselves. Inspirations for these changes are the archetypal network itself: the brain. Despite relying on noisy and slow chemical processes, the brain

can outclass most ANNs when faced with multi-task environments, out-of-distribution problems or energy consumption (Grace et al., 2018). Transferring the insights from this inherently biological system to the numerical nature of deep learning requires a fundamentally interdisciplinary approach(Richards et al., 2019) with biological knowledge to provide the brain structures, mathematical techniques to transfer this to a numerically implementable framework, and computer science and data science tools to realize this framework on the computer.

One such insight, which is an area of intense research, relates to the mechanisms of biological learning and how they can be implemented in ANNs, i.e., how to perform efficient credit assignment for cortical circuits. The majority of artificial neural networks are trained using the backpropagation algorithm. However, its biological plausibility has been questioned because (among other things) it relies on using symmetric weight matrices for the backwards weight update step, which is not found in biological networks (Lillicrap et al., 2020). Furthermore, the training of ANNs relies on some predetermined cost function, while it is unclear what this cost function would be for the brain (Hebb, 2005). Understanding how the brain learns could fundamentally impact the performance and energy costs of training and running ANNs. As these algorithms become evermore important, harnessing the brain's efficiency for the artificial counterpart can only have benefits. Vice versa, using ANNs as models for the biological brain, provided they rely on biologically plausible algorithms, can be invaluable to neuroscience.

Both of these aspects are exemplified in the work of the invited speaker, both in terms of biological alternatives to backpropagation and how hierarchical structures can help the brain learn more effectively (Greedy et al., 2022; Boven et al., 2023). Regarding the credit assignment problem, Dr Ponte Costa and his PhD student Will Greedy have introduced the concept of the Bursting Cortico-Cortical Networks (BurstCCN) (Greedy et al., 2022). They showed that the BurstCNN model could effectively backpropagate errors using a single-phase learning process, which contrasts the two-phase process of backpropagation and is more akin to how biological systems learn. Next, they showed that learning this way approximated backdrop-derived gradients. Two projects of this workshop revolved around this model: *i)* a study of how robust BurstCNN is to noise during training and *ii)* a study on how Dalean principles (i.e., that neurons must stay either inhibitory or excitatory) could be implemented in the BurstCNN framework.

Dr Ponte Costa has also published significant work with his PhD student Joseph Pemberton on learning in the cerebral cortex (Boven et al., 2023). The information the brain often gets is sparse, unlike how ANN are commonly trained. Thus, the mechanisms underlying how the cerebral cortex learns efficiently despite the sparse feedback remain unclear. Thus, they introduced a systems-level computational model of cerebro-cerebellar interactions where a cerebral recurrent network received feedback predictions from a cerebellar network. Thus, they achieved decoupling learning in cerebral networks from future feedback. They found that using this model achieved faster learning and reduced dysmetria-like (lack of coordination of movement) behaviours. Using this framework, two further projects were posed to the students: *i)* a numerical study on the potential role of the thalamus in learning and information processing in the brain. Using autoencoders as a thalamus-like feedback loop, they tested the performance against a baseline model trained using standard backpropagation. *ii)* A study on incorporating sparsity in the connections cortex to the cerebellum, mirroring the biological structure of mossy fibres. The results of these studies follow this editorial.

While all the projects were ambitious in scope and topic, the results achieved by the students are nonetheless impressive and point to interesting future endeavours. We are grateful for all their hard work and openness to work on novel topics.

References

Ellen Boven, Joseph Pemberton, Paul Chadderton, Richard Apps, and Rui Ponte Costa. Cerebro-cerebellar networks facilitate learning through feedback decoupling. *Nature Communications*, 14 (1):51, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Katja Grace, John Salvatier, Allan Dafoe, Baobao Zhang, and Owain Evans. When will ai exceed human performance? evidence from ai experts. *Journal of Artificial Intelligence Research*, 62: 729–754, 2018.

Will Greedy, Heng Wei Zhu, Joseph Pemberton, Jack Mellor, and Rui Ponte Costa. Single-phase deep learning in cortico-cortical networks. *Advances in Neural Information Processing Systems*, 35: 24213–24225, 2022.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2): 604–624, 2020.

Myeongsuk Pak and Sanghoon Kim. A review of deep learning in image recognition. In *2017 4th international conference on computer applications and information processing technology (CAIPT)*, pages 1–3. IEEE, 2017.

Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

Kiersten M Ruff and Rohit V Pappu. Alphafold and implications for intrinsically disordered proteins. *Journal of Molecular Biology*, 433(20):167208, 2021.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Breaking BurstCNN

Oliver Braganza[1], Rieke Fruengel[2], Alejandro Tlaie Boria[3]

Abstract

Artificial neural networks and deep learning typically depend on the error-backpropagation (backprop) algorithm. However, this mechanism seems implausible in the neuroscientific context. The Bursting Cortico-Cortical Network (BurstCCN) model represents a recent attempt to bridge this gap by modeling backprop-like error signals as distal dendritic top-down inputs based on burst-dependent short term plasticity. It approximates backprop gradients and enables efficient deep learning on benchmark data sets (MNIST, CIFAR-10). However, biological neurons and real-world processes are inherently noisy, which was not accounted for in the standard training setup for the BurstCCN. In this workshop project, we experimented with introducing noise into the model during training to test its robustness. We found that BurstCCN is often more robust to noise than conventional backprop.

1 Introduction

Artificial neural networks and deep learning typically depend on the error-backpropagation (backprop) algorithm. However, this mechanism seems implausible in the neuroscientific context. The Bursting Cortico-Cortical Network (BurstCCN) model represents a recent attempt to bridge this gap. BurstCCN models backprop-like error signals as distal dendritic top-down inputs based on burst-dependent short term plasticity (Fig. 1). It allows backpropagation of errors through multiple layers using a single-phase learning process, approximating backprop-derived gradients, and enabling efficient deep learning on benchmark data sets (MNIST, CIFAR-10). However, biological neurons are inherently noisy. Presumably, throughout learning, neurons in the brain need to become robust to background noise. However, this was not accounted for in the standard training setup for the BurstCCN.

In this workshop project, we experimented with introducing noise into the model during training to test its robustness. Specifically, we added various degrees and forms of noise at different stages of processing and investigated how this affected classification performance and network sparsity.

2 Methods

All network training sessions were run using PyTorch in Google Colab based on the BurstCCN code available in the workshop GitHub page: https://github.com/meggl23/BioDLWorkshop.

[1]Institute for Experimental Epileptology and Cognition Research, University of Bonn, Bonn
[2]Max Planck Institute for Neurobiology of Behavior – caesar, Bonn
[3]Ernst Strüngmann Institute for Neuroscience in cooperation with Max Planck Society, Frankfurt

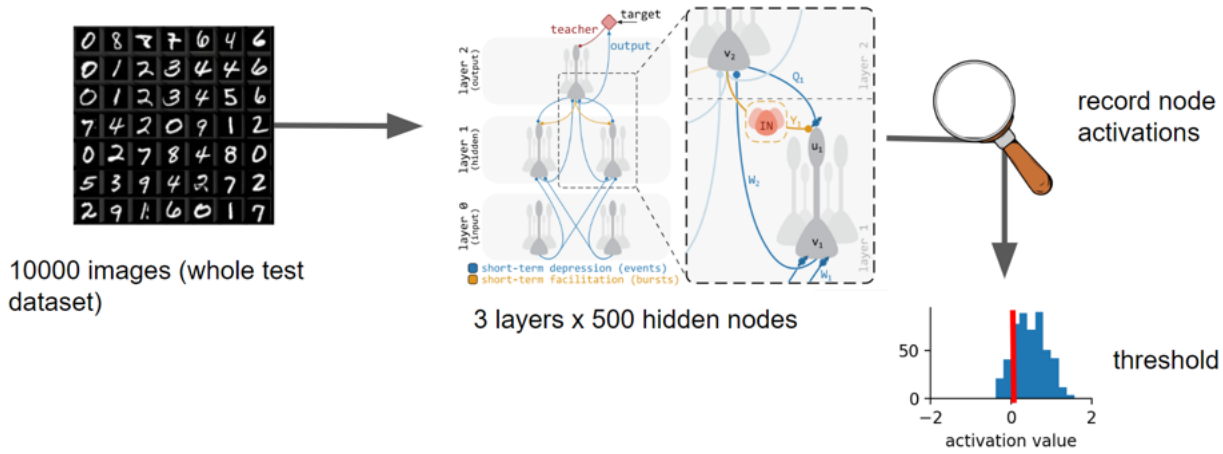


Figure 1: Method for measuring sparsity of activity

Where possible, training was performed on GPU. Visualizations were created using standard Python libraries, including matplotlib and seaborn.

To investigate the effect of noise in the input images, we added Gaussian noise to the value of each pixel in the image of the MNIST dataset (both train and test datasets). We systematically varied the standard deviation of the noise in order to investigate the effect of different noise levels.

To add uncorrelated noise to node activations, we added Gaussian noise to the activations of all nodes in all hidden layers after the activation function was applied. Noise was added during both the training and testing phases. As with noise in the input images, we varied the standard deviation of the added noise.

We compared the representations formed by different networks by measuring the sparsity of node activity (Fig. 1). To this end, we presented each image from the testing dataset to the network (10000 images), and recorded the activations of all nodes in each hidden layer. If activations were below zero (e.g. due to added noise), we thresholded the value to zero.

After we characterized the algorithm performance and robustness to uncorrelated noise, we turned our attention into the slightly more convolved case of correlated sequences of noise. Particularly, we made use of the Cholesky decomposition. The Cholesky decomposition is a linear algebra technique used to decompose a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. It is particularly useful for generating correlated random sequences, as it provides an efficient method for transforming independent random variables into correlated ones. Given a covariance matrix C, the Cholesky decomposition can be expressed as $C = LL^T$, where L is a lower triangular matrix with real and positive diagonal entries, and L^T is the transpose of L. The Cholesky decomposition is unique when the matrix is positive-definite, ensuring the stability and consistency of the generated correlated sequences.

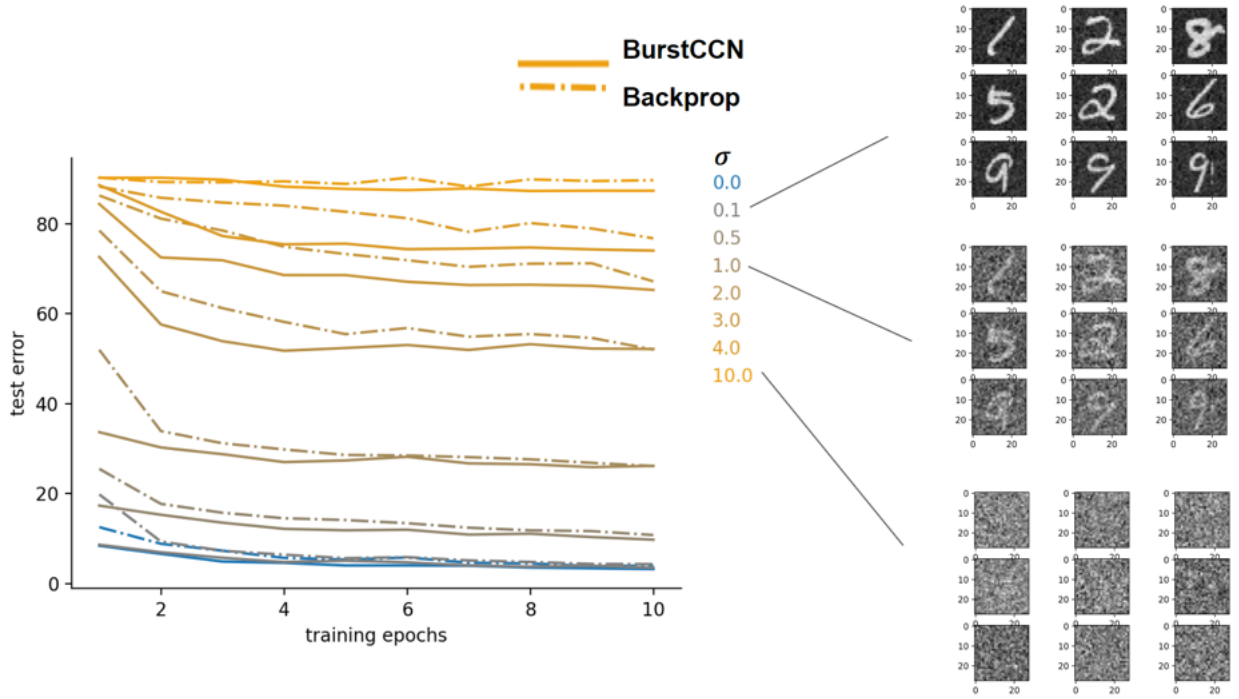


Figure 2: Robustness to noisy input images

3 Results

In the real world, biological sensors and environmental factors may be a source of noisy input to brain areas. We therefore first experimented with adding noise to the input images (Fig. 2). We found that at all tested noise levels, BurstCCN outperforms conventional backprop with slightly lower test errors. The difference in performance between the two training methods is most notable during the early epochs of training. It is possible that with further training, the two methods would become indistinguishable.

Neurons do not always reliably respond to inputs. We therefore next investigated the effect of adding noise to node activations (Fig. 3). We found that at low noise levels, the BurstCCN method results in a lower test error than conventional backprop. Interestingly, however, at very high noise levels backprop greatly outperforms BurstCCN.

To compare the representations formed by different networks, we measured how sparse their activity was. Overall, we found that BurstCCN forms more sparse representations than backprop (Fig. 4A). We also noticed that, in networks with node noise added, activity became increasingly sparse in later layers. To investigate why, we recorded the unthresholded activation values of the nodes (Fig. 4B). As we added noise after the ReLU activation function, which normally bounds the node outputs to between 0 and 1, it is possible for nodes to pass negative activation values to the subsequent layer. This, in turn, increases the probability of a negative weighted sum of inputs in downstream layers, which then gets rectified to zero by the activation function. Once noise is then added, about half of these activations are likely to become negative

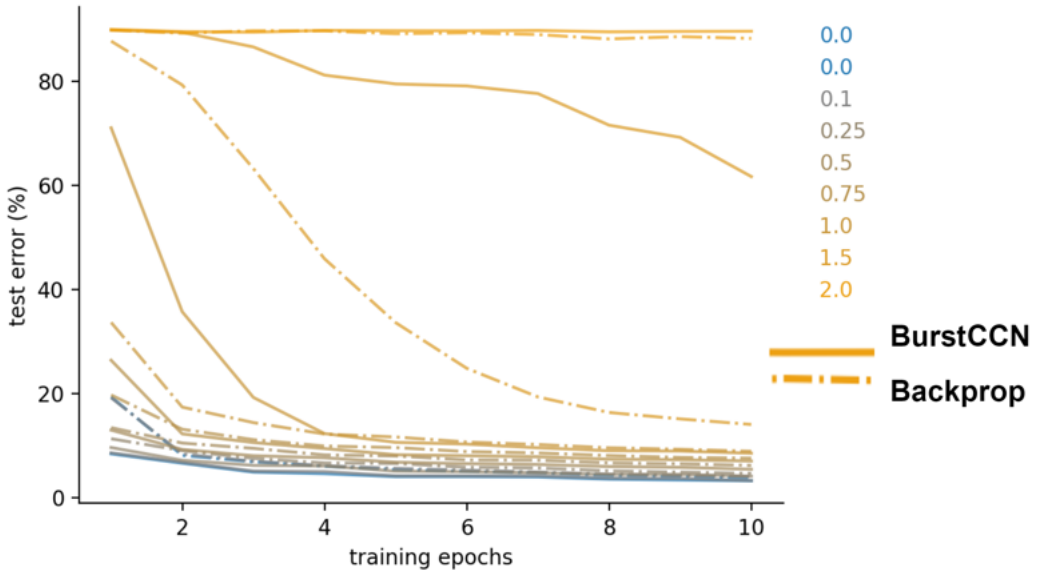


Figure 3: Robustness to uncorrelated noise added to node activations

again, which further increases the chance of low or negative activations in the next layer. Further work should be done to investigate the effect of adding noise to node activations before the activation function, however we speculate that the result would be similar, if perhaps slightly weaker.

To generate correlated random sequences, we first obtained the Cholesky decomposition L of the desired covariance matrix C. Next, we generated a set of independent, identically distributed random variables (e.g., Gaussian) and arranged them in a vector, say X. We multiplied L by X to obtain a new vector $Y = LX$, which now represented the correlated random variables with the specified covariance structure. To generate correlated noise that can be parametrically explored, we simply adjusted the elements of the covariance matrix C, in our case sampled from $\mathcal{N}(0, \sigma)$, with σ being the parameter we modify. Changing these elements in C directly influences the correlations between the generated random sequences. We chose to use the Cholesky decomposition because it provides an efficient and stable method for transforming the covariance matrix into the lower triangular matrix L, enabling the generation of various correlated noise patterns by simply adjusting the covariance matrix, thus offering a powerful and simple tool for parametric exploration of correlated noise.

4 Conclusions

We found that BurstCCN is often more robust to noise than conventional backprop. This could be relevant for real-world applications, for example where data are imperfect. Furthermore, the sparse representations formed by networks trained with BurstCCN could, if implemented correctly on the hardware and software level, potentially allow ANNs to be more energy- and memory-efficient.

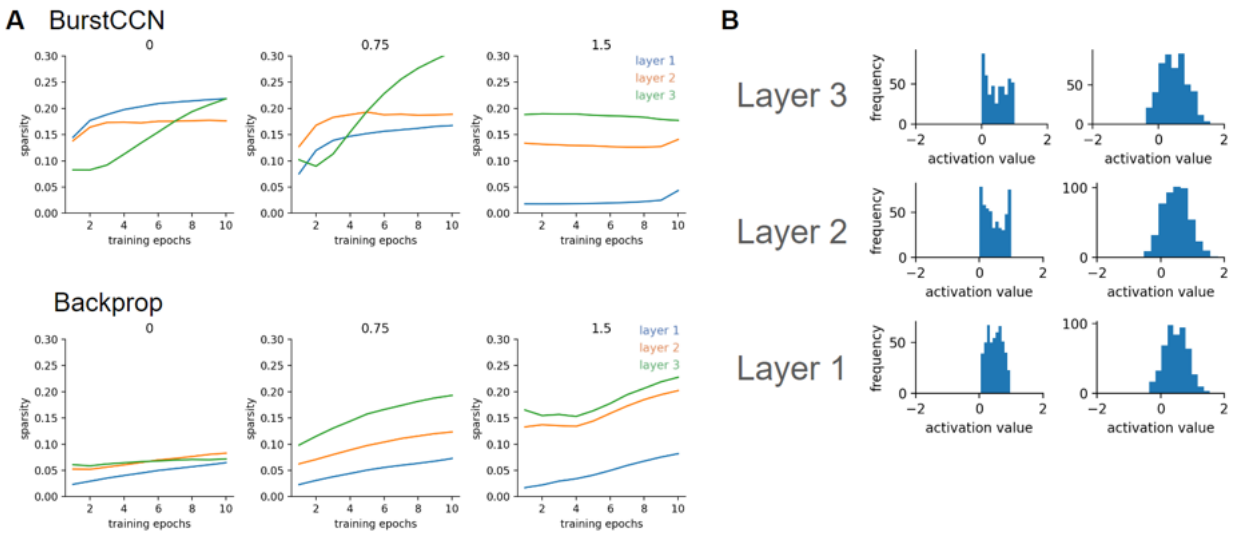


Figure 4: A: Sparsity of activity at different node noise levels, B: Activation values of nodes in hidden layers 1-3 before adding noise (left), and at noise level 0.75 (right)

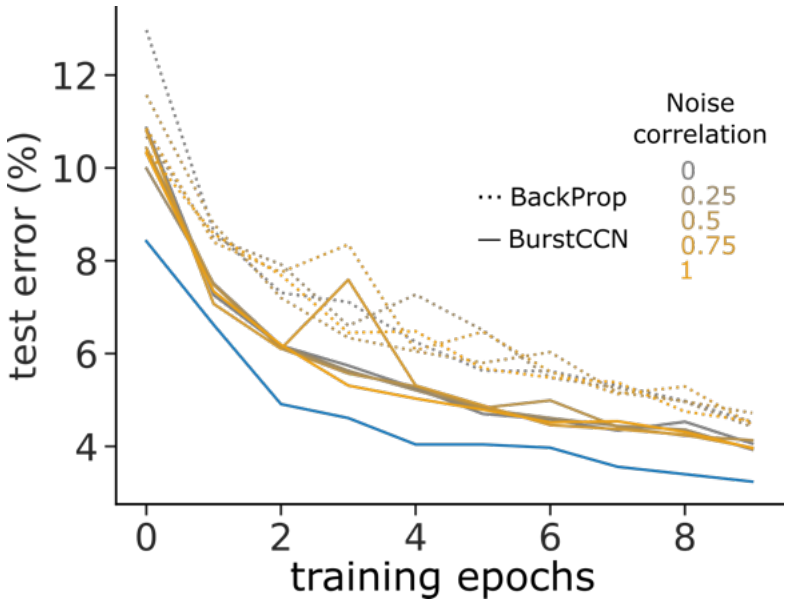


Figure 5: Different noise correlation levels, measured by the amplitude of the generating sequences in the covariance matrix, given by samples of $\mathcal{N}(0, \sigma)$. For all correlation levels, BurstCCN outperformed Backprop.

5 Acknowledgements

We would like to thank Max and Laura for organising the workshop.

Introduce Dalean weight constraints into the BurstCCN model

Carlos Wert Carvajal[1], Khaldoon Alnosairy[2], Maria Royo Cano[3], Tom Burkart[4]

Abstract

The BurstCCN model does not respect Dales Law in the sense that any of the weighted connections in the network can have either positive or negative weights which can even change sign during training. Neurons in the brain typically have only an excitatory or inhibitory effect so this is a substantial implausibility in the model. The goal of this project is to modify the existing BurstCCN implementation to restrict the weights to satisfy Dales' principle. Firstly, consider how the connectivity might have to change as a result - if you introduce inhibitory interneuron populations how do these connect to the pyramidal cells? Does the network performance suffer as a result of these additions? Using this implementation or just the base BurstCCN implementation, can you design an experiment to relate the model to results from (Khan et al.) showing that a higher level of coupling between SST and Pyramidal cells predicts an increase in plasticity that the pyramidal cells experience? Aim 1: Extend BurstCCN with Dales Law. Hint: Keep in mind that this will require separate pathways on all the weights (W, Q and Y). You might find this paper [2] useful. Aim 2: Using simple pair-wise correlations can you relate the "SST" and "PC" cells in the model to the experimental observations of Khan et al (https://www.nature.com/articles/s41593-018-0143-z)

1 Introduction

The BurstCCN model introduces a single-phase learning system for biologically plausible back-propagation in the brain [4], which previous dendritic and bursting models were not able to produce [8, 7]. Previous work by William Greedy and colleagues particularly introduced three pathways and a short-term plasticity (STP) component in the inhibitory feedback that multiplexes information routes without requiring more than one learning phase (Figure 1, [4]). Nonetheless, the model does not follow Dales' law in constraining the connectivity to strictly positive or negative components and maintaining sign-consistent changes in plasticity [3, 6]. We aim to evaluate how the model performs under different Dalean implementations. Namely,

(i) Having weights Q, Y, W under their respective E-I weights, per the input layer, with a hard-boundary condition.

(ii) Physically-motivated soft-boundary condition for sign-consistent updates.

(iii) Separation of weights under sign-consistent updated with random and guided perturbations.

(iv) Addition of an inhibitory layer, following [2], to divide weights into positive and negative contributions.

These four approaches are meant to influence the evolution of weights W, Q, Y, shown in Figure 1.

[1]University of Bonn, Bonn, Germany
[2]Faculty of Medicine, Otto-von-Guericke University, Magdeburg, Germany
[3]Max Planck Institute for Neurobiology of Behavior, Bonn, Germany
[4]Ludwig-Maximilians University of Munich, Munich, Germany

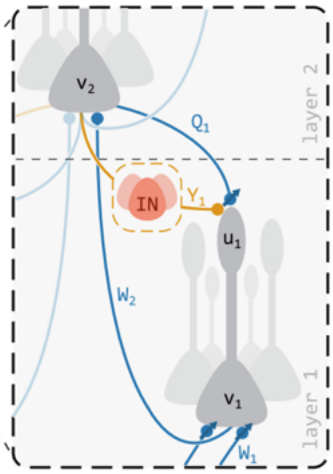


Figure 1: Original model BurstCCN model from [4]. There are three hidden-layer information routes: an excitatory feedforward (W_2), an excitatory feedback (Q_1) and one inhibitory-to-excitatory connection with STP (Y_1).

2 Methods

2.1 Restricted weights sign

The first approach was one of defining W and Q as strictly positive weights and Y as negative ones, given their excitatory and inhibitory nature, respectively. We can impose such conditions by imposing hard boundaries or clipping in the weights after each N update round or epoch. Hence, we have

$$W^{N+1} = \max\left(W_N + \Delta W^N, 0\right), \tag{1}$$

$$Q^{N+1} = \max\left(Q_N + \Delta Q^N, 0\right), \tag{2}$$

$$Y^{N+1} = \min\left(Y_N + \Delta Y^N, 0\right). \tag{3}$$

We note that this implementation allows for sparsity, given that values can be zero. Additionally, by not affecting the loss function of each individual weight, we do not assume or constrain the descent direction.

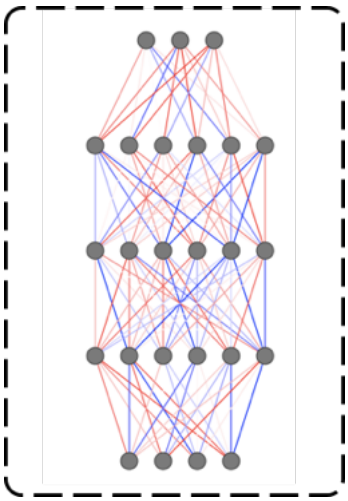


Figure 2: Restricted weights sign. The weights are strictly positive or negative and cannot change their sign during learning. The units in the network represent excitatory neurons, but in the case the weight between two neurons is negative, an inhibitory neuron is supposed to lie in the pathway between them.

2.2 Soft boundaries on weights

As an alternative, we aimed to adopt methods from statistical physics to ensure sign consistency of the weights. This ansatz is based on the strong similarity of the loss function $\mathcal{L}$ in generic neural networks to the free energy functional $\mathcal{F}$ in statistical physics. Specifically, in out-of-equilibrium physics the time evolution of a system's dynamic variables (equivalent to the time evolution of all weights and biases in a neural network) is governed by thermodynamics [5]: over time, the total entropy can only increase, which implies that the total free energy can only decrease when the system is in contact to a heat bath. Systems without conservation laws (commonly referred to as Model A) thus obey the following differential equation:

$$\frac{\partial \phi_i}{\partial t} = -\Gamma \frac{\delta \mathcal{F}}{\delta \phi_i} \,, \tag{4}$$

where ϕ_i are the dynamic quantities of the system and Γ is a characteristic time scale. Heuristically, this means that all quantities change such that they decrease the free energy, and is therefore formally identical to the general idea of backpropagation where weights w_{ij} are updated according to

$$\Delta w_{ij} = -\eta \frac{\partial \mathcal{L}}{\partial w_{ij}} \tag{5}$$

with η being the learning rate.

In statistical physics, any physical considerations and limitations—such as symmetry arguments or non-negativity—need to be encoded into the free energy functional. This can either be done by expansion around the free energy minimum including all terms that are allowed by symmetry (which would also be possible in a neural network close to a minimum of the loss landscape), or by directly imposing potentials for the dynamic variables to force them to take specific values as done for liquid crystals or in phase field methods [1]. Here, we use the latter method and add potential barriers to the loss function to prevent sign changes of the weights. The simplest approach is to modify the loss function such that weights are pushed away from the $w_{ij} = 0$ subspace with a strength that increases as the weights approach this subspace:

$$\Delta w_{ij} = -\eta \left(\frac{\partial \mathcal{L}}{\partial w_{ij}} - g(w_{ij}) \right) \tag{6}$$

where

$$\begin{aligned} g(w_{ij})/w_{ij} &> 0 \,, \\ \lim_{w_{ij} \to 0} \quad \frac{1}{g(w_{ij})} &= 0 \,, \\ \lim_{w_{ij} \to \pm\infty} \quad g(w_{ij}) &= 0 \,. \end{aligned}$$

These requirements are fulfilled by a large number of functions, and choosing the optimal function is a separate task.

This approach is appealing since it does not require any changes to the machine learning methodology. Since the loss function $\mathcal{L}$ is in general a linear operator the addition of this repulsive force g can be easily absorbed into the loss function. Some interesting examples for the updated loss function

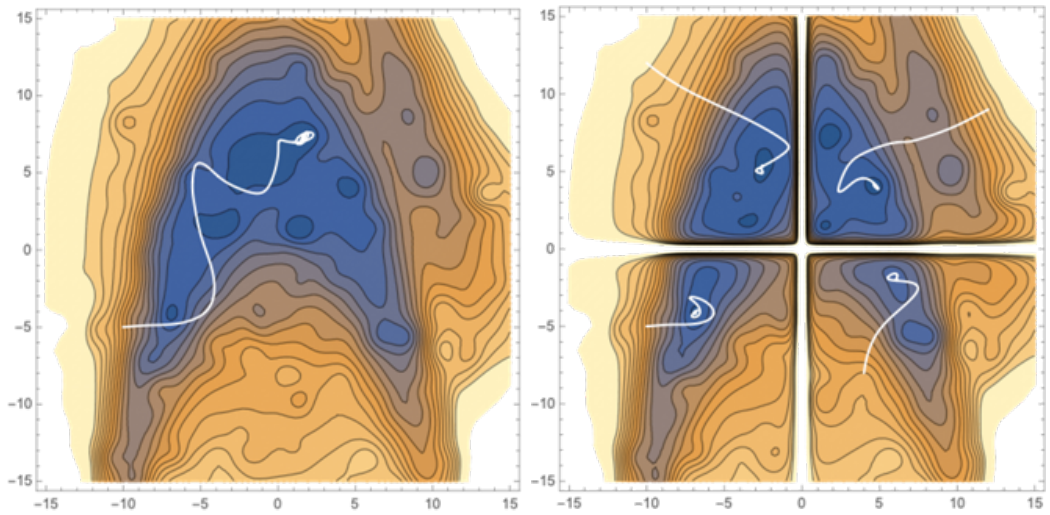


Figure 3: Schematic illustration of a two-dimensional loss landscape without (left) and with (right) soft boundaries enforced by a potential $\sim w_{ij}^{-2}$. White lines indicate a trajectory determined by gradient descent with momentum. With boundaries, each trajectory arrives at the global minimum within its own subspace (i.e., without sign changes).

$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_g$ include

$$\mathcal{L} = \mathcal{L}_0 - \sum_{ij} \frac{\zeta}{2\eta} \ln(w_{ij}^2) \qquad \rightarrow \qquad \Delta w_{ij} = -\eta \frac{\partial \mathcal{L}}{\partial w_{ij}} + \zeta \frac{1}{w_{ij}}\,, \tag{7}$$

$$\mathcal{L} = \mathcal{L}_0 + \sum_{ij} \frac{\zeta}{2\eta} \frac{1}{w_{ij}^2} \qquad \rightarrow \qquad \Delta w_{ij} = -\eta \frac{\partial \mathcal{L}}{\partial w_{ij}} + \zeta \frac{1}{w_{ij}^3}\,. \tag{8}$$

The latter case is shown schematically in Fig. 3. Alternatively, one might consider more complex potentials such as a Lennard-Jones potential $\mathcal{L}_g = 4\zeta/\eta \left((\bar{w}/w_{ij})^{12} - (\bar{w}/w_{ij})^6 \right)$ which—on top of enforcing sign consistencies—also confines weights to take the value $\bar{w}$ once they approach the $w_{ij} = 0$ subspace, as a method of eliminating weights from the network if they become too small.

2.3 Weight separation and perturbations

We also followed a previous approach in separating weights into positive and negative contributions [9]. As shown in Figure 4, the weight matrix can have masks corresponding to each excitatory or inhibitory component. We can make use of the clipping factor (as described in Eqs 1-3) to ensure updates are sign-consistent. Nonetheless, one key aspect is that such condition is not enough to guarantee convergence in training, since hard-boundaries can act as local minima in the case that the update direction is pushing weights to change sign. Therefore, one solution is to include some additional feature to allow weights to fluctuate and increase the value diversity. One way is to use the Kullback–Leibler (KL) divergence between weight distributions as an additional constrain in the loss function [9]. In particular, we can use use a weighted contribution of this metric between the weight distribution between the current parameters and the previous ones. Thus for a given parameter θ,

$$\mathcal{L}(\theta) = \mathcal{L}_{BurstCCN}(\theta) + A_{KL} D_{KL}(P \,||\, P_{old}), \tag{9}$$

where P_{old} represents the distribution of values within a previous version of the matrix θ. A_{KL} is the weighting given to the divergence in the compound loss.

Finally, another solution to avoid local minima is to use random perturbations around weights [10]. Hence, for a weight θ we have

$$\theta^{N+1} = clip(\Delta\theta^N + \xi), \tag{10}$$

where ξ is a Gaussian-process sampled from zero-mean normal distribution of deviation σ.

$$W^{\text{rec},+} = \underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}}_{M^{\text{rec}}} \odot \underbrace{\begin{pmatrix} \cdot & + & + & + & \cdot \\ + & \cdot & + & \cdot & + \\ + & + & \cdot & + & + \\ \cdot & + & + & \cdot & \cdot \\ + & + & + & + & \cdot \end{pmatrix}}_{W^{\text{rec,plastic},+}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & w_1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{W^{\text{rec,fixed},+}},$$

Figure 4: Separation of recurrent weights through a mask to impose excitatory conditions taken from [9].

2.4 Adding a inhibitory layer

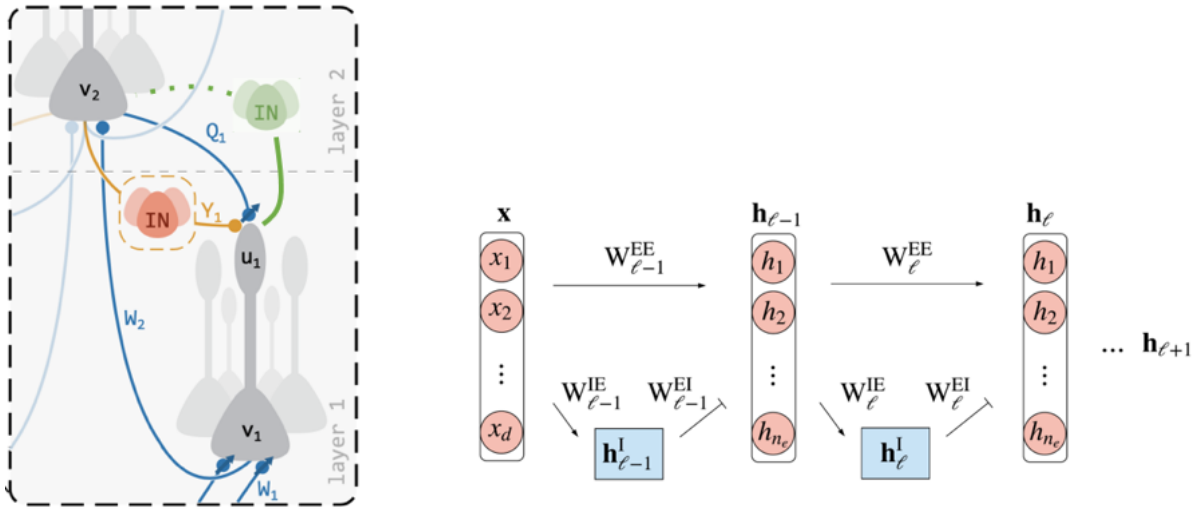
Finally, another approach we used to fulfill Dale's law was to keep the weights in the original model positive and introduce inhibitory layers. Firstly, inspired by ref [2],we introduced an inhibitory layer between every layer in the original network as shown in figure 5b, while keeping the direct connections between the excitatory layers (W_{EE}). In the scheme, x is the input the network receives from another excitatory population, h_l every excitatory hidden layer and h_l^I every inhibitory hidden layer. The inhibitory layers had the same number of units as the excitatory ones and the connections from every excitatory to inhibitory layer (W_{IE}) were kept fixed while learning occurred in the inhibitory to excitatory synapses (W_{EI}). Initially, the inputs to every excitatory layer were modelled as

$$h_l = f[W_{l-1}^{EE} h_{l-1} + b_{l-1} - (W_l^{EI} h_l^I + b_I)] \tag{11}$$

Then, the inputs to every excitatory layer were modelled following the following equation, in which inhibition impacts the excitatory units both in a subtractive and in a divise manner:

$$h_l = f[\frac{g_l}{W_l^{EI} h_l (e^{\alpha l} \odot h_l^I)} \odot (W_{l-1}^{EE} h_{l-1} - W_l^{EI} h_l^I) + \beta_l] \tag{12}$$

where for each layer l, β_l is a bias, g_l modulates the gain and αl determines the strength of the divisive inhibition; $\odot$ denotes the elementwise multiplication; and f is the rectified linear function (ReLU).



(a) (b)

Figure 5: Insertion of inhibitory layers. (a) Inhibitory layers were introduced between the excitatory layers, but keeping direct connections from excitatory to excitatory layers. Adapted from [4] (b) ANN architecture. Pink units are excitatory neurons, while blue layers are inhibitory. Figure from [2].

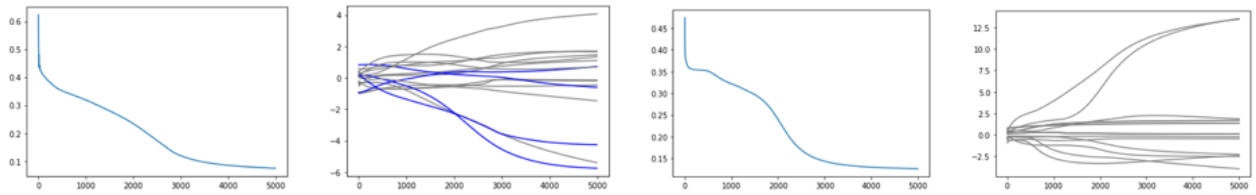
3 Results

3.1 Restricted weights sign

Hard boundaries were unable to reproduce the accuracy of BurstCCN under normal conditions, only obtaining around 9% in comparison with the 90% of the original for the same number of epochs with the MNIST dataset. One key feature is that weight matrices become increasingly sparse under this restriction and are not able to change the update direction.

3.2 Soft boundaries on weights

As a proof of concept we implemented the soft boundary method first for a simple regression task using a neural network with a single hidden layer consisting of five nodes. As shown in Fig. 6 the network minimizes the loss reasonably well when the soft boundaries are used, even though more epochs are needed for training. Note that a subset of the weights shown in Fig. 6d quickly approach zero and do not change anymore afterwards, indicating that these weights cannot improve the accuracy of the network without sign changes. Instead, the effective elimination of these weights has to be compensated by other weights, which naturally decreases the accuracy of the network. An important insight from this proof of principle is that the soft boundary method is sufficient to ensure sign consistency, and that the elimination of a subset of weights can in parts be compensated by other weights.



(a) Losses, no boundaries (b) Weights, no boundaries (c) Losses, boundaries (d) Weights, boundaries

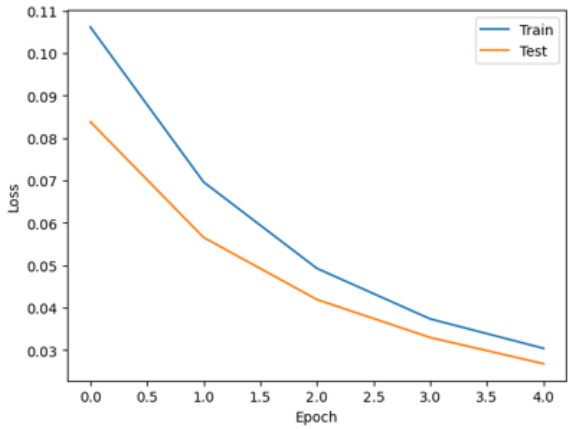
Figure 6: Losses (a,c) and values of the weights (b,d) over 5000 epochs for a network without soft boundaries (a,b) and with soft boundaries (c,d) as defined in Eq. 8. Gray lines in (b) and (d) indicate weights that preserve the sign, while blue lines indicate weights that change sign.

The weights remained sign-preserving when applying this method to the BurstCCN network, however in our implementation the network was not able to learn. While this may well be an implementation issue, there is also an important practical issue that needs to be considered when using the soft boundary method: weight initialization often uses a normal distribution around zero. Weights that are initialized close to zero experience a strong repulsion at the very beginning of the learning, and the soft boundary effectively changes their initial value to $\sim -\zeta \partial_{w_{ij}} \mathcal{L}_g$. Since weight initialization is crucial for successful learning in neural networks, this can be a critical problem when applying the soft boundary method. One way to circumvent this issue is to use a bimodal distribution for the weight intialization that does not allow for any weights with small absolute magnitude.

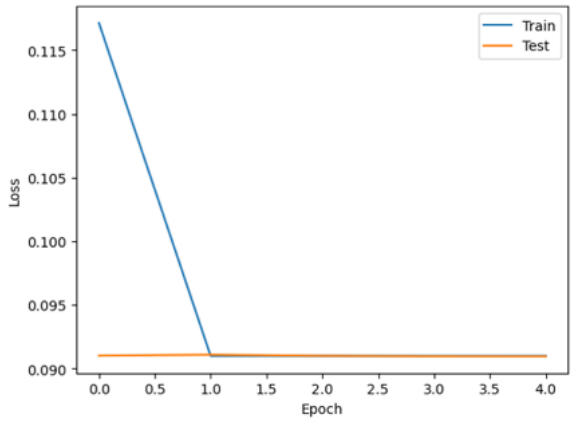
3.3 Weight separation and perturbations

The separation of weights into strictly positive and negative components through masking and clipping was unsuccessful in allowing BurstCCN training. In fact, for a 80% excitatory and 20% inhibitory connectivity, the loss function quickly decays to a stable minima for both train and test sets (cross-validation average). In fact, the addition of a KL divergence penalization retained the same loss along epochs compare to the decay in the BurstCCN, as seen in Fig 7.

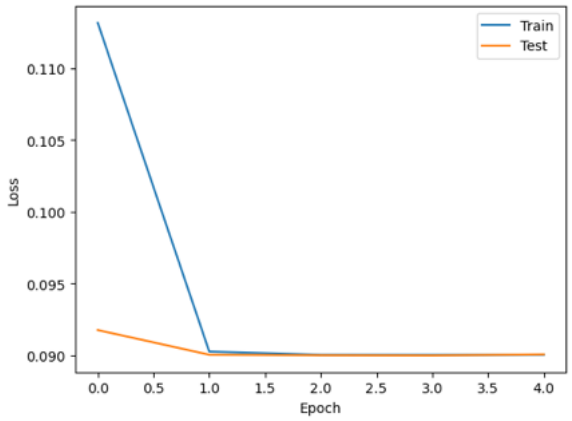
The addition of random perturbations along synaptic connections in training also did not change the landscape, as observed in Fig 7c. We can find an answer to this in the level of sparsity along the connectivity matrix W, which has a singular value decomposition (SVD) that is very low dimensional (Fig 7d). This suggests most of the power is mostly concentrated along a single dimension and that value (which are non-zero due to the random perturbation) stay close to a similar minima (or maxima). Again, this relates back to the problem of adjusting weights to a minima or maxima. Further work should use such definitions for gradients alone and find better soft-boundary approaches to the Dalean problem.



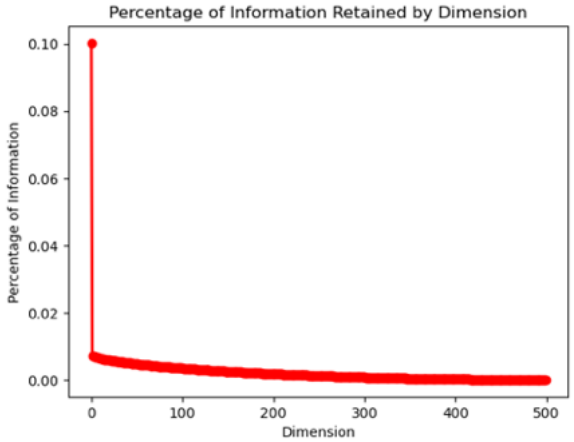
(a) Loss evolution over epochs of the normal BurstCCN for the MNIST dataset



(b) Loss evolution for BurstCCN with separated strictly positive and negative weights with a KL divergence condition in the loss.



(c) Loss evolution for a KL divergence-weighted loss and random edge perturbations.



(d) For the case of random perturbations. spectral analysis (SVD) of the W parameter showing a high degree of sparsity.

Figure 7: Weight separation and clipping is not able to reproduce the loss landscape of BurstCCN under a similarity penalization and random wight perturbations.

3.4 Adding a inhibitory layer

Unfortunately through this approach we did not reach results higher that those obtained by chance, an accuracy of 9%. Two days were not enough to fine tune the network under the changes made to the original model.

4 Conclusions

The BurstCCN model introduces a biologically plausible back-propagation learning system for the brain, but it does not follow Dales' law in constraining the connectivity to positive or negative com-

Sparsity and compression of Cerebellar-driven cortical dynamics

Lisa Bast[1], Yuyang Huang[2], Johannes Striebel[3], Pietro Verzelli[4]

Abstract

Anatomically, the connection from the cortex to cerebellum is sparse, with each granule cell having an average of four incoming mossy fibers. This structural feature is evolutionarily conserved and can even be observed in fish [1]. Another observation from nature is that for some species a compression layer (or Pons) is included between cortex and cerebellum. These findings prompt investigations into incorporating sparsity and compression into learning models to better reflect biological reality. Here, we build upon the work of Pemberton *et al.* [3] by introducing sparsity in two ways: a sparse input to the cerebellum and a sparse recurrent neural network (RNN) in the cortex. While sparsity did not improve performance, the stable test mean squared error (MSE) observed at 50% and 70% sparsity suggests that many connections in the network can be removed without degrading performance. This finding could lead to more economical and efficient computation of sparse architectures. However, longer training and more detailed hyper parameter exploration are recommended to further investigate sparsity of the system.

1 Introduction

1.1 The fully connected, non-compressed cortical model with cerebellar input

Cerebellar-driven cortical dynamics can be described with a network model consisting of a cortical Recurrent Neural Network (RNN) and a cerebellar feedforward (FF) network of one hidden layer, as introduced in [3]. In the cerebellar FF network, the cortical input of the RNN is received by the granule cells via mossy fibers and is then further passed on to the Purkinje cells via parallel fibers.

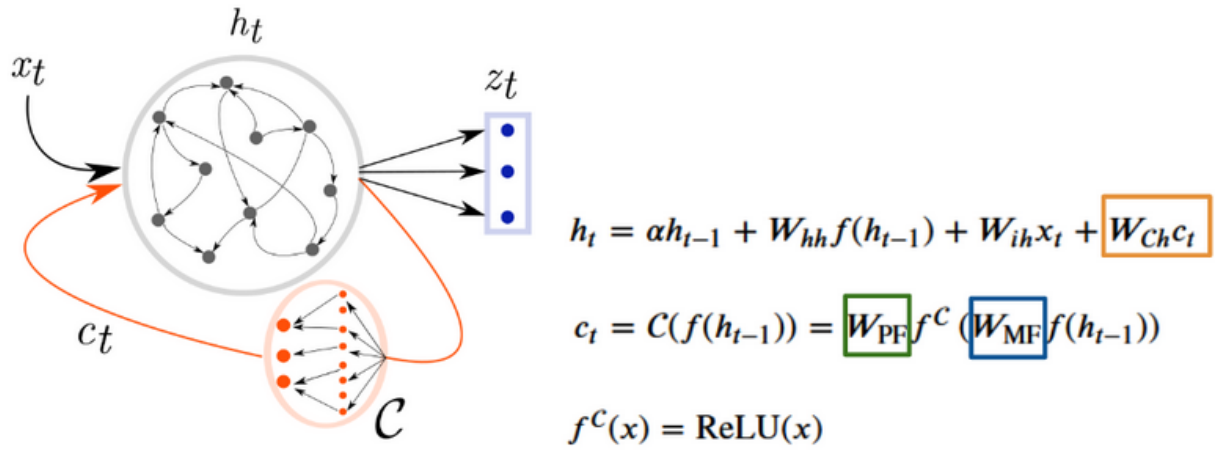


Figure 1: Scheme and mathematical formulation of cortical model with cerebellar input [3].

[1]Department of Medical Biochemistry and Biophysics, Karolinska Institutet, Solna (SE)
[2]Division of Neurobiology, Faculty of Biology, Ludwig-Maximilians-Universität, Munich (DE)
[3]Department of Ophthalmology, Medical Faculty, University of Bonn, Bonn (DE)
[4]Institute of Experimental Epileptology and Cognitive Research, Bonn Medical Center, Bonn (DE)

1.2 The task and objective function/ plasticity rules

Figure 2: Line drawing task [3].

While learning the line drawing task (Fig. 2), the objective is to minimize the task error across the entire task sequence for the weight parameters in matrices W_{rdt} and W_{PF}. Therefore two error functions $E_t = \mathcal{E}(y_t, z_t)$ and $E_t^C = \mathcal{E}(y_t, c_{t-\tau})$ for cortical and cerebellar output respectively are minimized.

2 Results

2.1 Case 1: Sparse cerebellar input

To imitate a reduced number of mossy fibres in the cerebellar FF model, we changed the average number of connections from the RNN to the first layer of the FF network. This was done by chaning the "nfibres" parameter in the model definition. We did a parameter scan by running the optimization for a range of values (nfibres=4,5,6, 10, 15, 20, . . . , 50) to see if the performance of the model changes in these cases. For highly connected models, i.e. large values of nfibres the MSE training error plateaus already after 10-15 epochs (Fig.3), for sparse cerebellar input at least 20 epochs are required to train network parameters.

Increasing sparsity in the cerebellar input does not improve the test MSE, but using only 50% of the mossy fibers leads to similar performance as compared to a fully connected cerebellar input (Fig. 5).

2.2 Case 2: Sparse cortical RNN

Another exploration we tried to carry out is the sparsification of the Cerebrum, i.e., the recurrent layer of the model. To do this, we defined a sparsity coefficient α which controls the sparsity of $W_h h$: when $\alpha = 1$, the matrix is fully connected, while for $\alpha = 1$, no connection is present. We then ran the model for different values of α spanning from 0.1 to 0.9 and computed the test error. The results are reported in Fig.5.

Performance appears to be relatively stable for a sparsity value below 0.7, i.e., when 70% of the total connections are removed. Even if this does not lead to any improvement, it shows that the recurrent layer can work properly even when a high number of connections is removed.

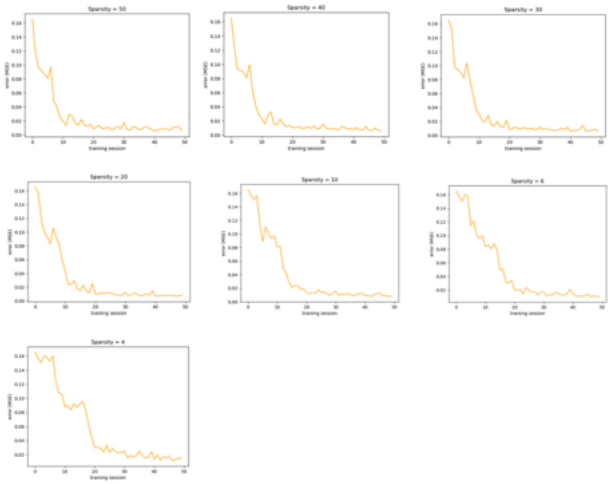


Figure 3: Mean squared error (MSE) of training set after 0 to 50 training sessions for various numbers of mossy fibers (sparsity).

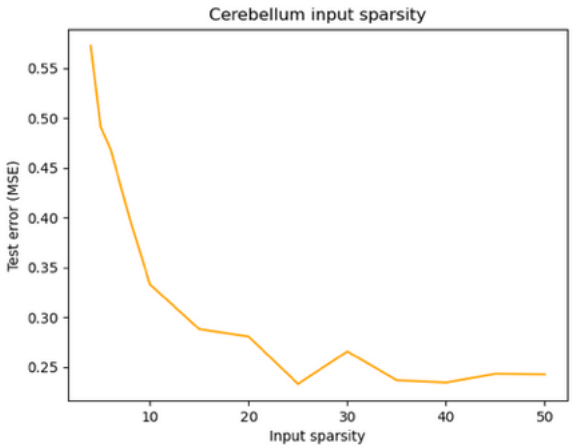


Figure 4: MSE of test set after 50 training sessions for a range of numbers of mossy fibres (sparsity).

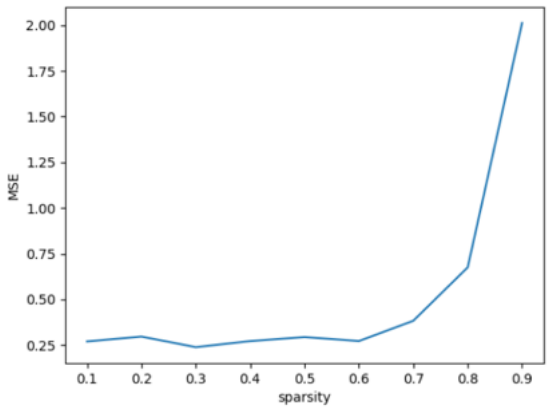


Figure 5: MSE of the test set after 50 training sessions for different values of the Cerebrum sparsity.

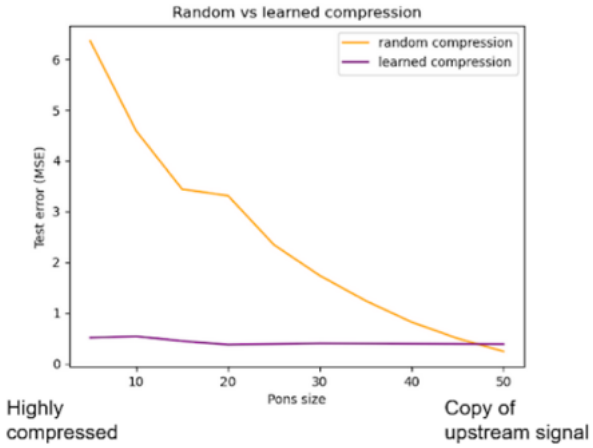


Figure 6: Test MSE after 50 epochs with a varying size of the compressing layer (pons) and fixed vs. learned compression paradigms.

2.3 Case 3: Compress prefrontal cortex output

It was hypothesized that a compression layer helps reduce the information that is transferred to the cerebellum to a task-relevant subspace thus making cerebellar predictions more accurate [2]. A single compression layer can be included between the cortical RNN and the cerebellar FF network (see Fig.1) mimicking observations of such compression layers in various species. Several architectures are possible: a random compression in which information is only fed through the layer and a learned compression, where the weights connecting the compression layer to the cerebellum are subjected to backpropagation. We hypothesized that the inclusion of a compression layer between the cortex and cerebellum could improve the performance of our system. We included an intermediate hidden layer (Pons) to test this hypothesis. As described in the introduction, random compression and learned compression are viable implementations. For both cases, we varied the size of the pons in the architecture. Test errors for various pons sizes are shown in Fig. 6.

2.4 Case 4: Combine compression and sparsity

In the next step, we wanted to bring the architecture closer to biological reality by combining both the cerebellar input and the compression layer sparsity. Therefore we fixed the sparsity to 4, which is the average number of inputs per granule cell as observed in vivo. Fig. 7 shows a plot of the test error over a varying size of the pons. We observe a decrease in test error with the increase in pons size, consistent with previous observations.

3 Conclusion and discussion

Based on the work of Pemberton *et al.* [3], which explores the interplay of the cortex and cerebellum in learning, we strove to make the model more closely resembling biological reality by introducing sparsity and compression. Anatomical findings show that each granule cell has, on average, four incoming mossy fibers. Thus, the connection from the cortex to the cerebellum is sparse and not fully connected. Another sparsity parameter that we included in the model which is also found in nature

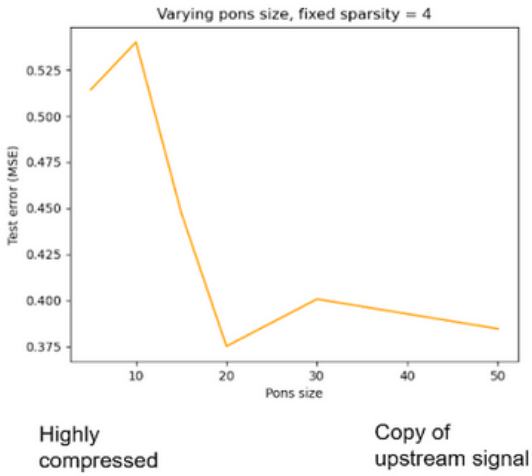


Figure 7: Combination of sparse cerebellar input (case 1) and a compression layer (case 3). In this case, the sparsity was fixed to 4, the average number of mossy fibers connecting to one granule cell. Varying the size of the pons reveals a sharp decrease in the test error when the pons is larger than 20.

is a sparse cortex. This region is not fully connected as well. Referring to the model we use, this corresponds to introducing some sparsity in the RNN. Both ideas were implemented, the sparse input to the cerebellum (case 1) and the sparse RNN (case 2). In both cases, sparsity did not improve the performance measured in the MSE of the test. On the other hand, we observed a stable test MSE down to 50% and 70% of sparsity, respectively. Also, concerning the compression of prefrontal cortex output (case 3), we observed that compression generally degrades performance, which is partially mitigated by training. Finally, sparsifying the different layers simultaneously to compression (case 4) also leads to a decrease in performance. For a more detailed investigation of sparsity in the system, we would recommend more extended training exceeding the 50 epochs we used in our case study. Additionally, a more detailed hyper parameter space exploration could be beneficial. Importantly, our results show that many connections in the network can be taken away without degrading the performance. This would point to a more economical and efficient computation of sparse architectures.

References

[1] Guy Billings et al. “Network Structure within the Cerebellar Input Layer Enables Lossless Sparse Encoding”. In: *Neuron* 83 (2014), pp. 960–974.

[2] Samuel Muscinelli, Mark Wagner, and Ashok Litwin-Kumar. “Optimal routing to cerebellum-like structures”. In: *bioRxiv* (2022), pp. 2022–02.

[3] Joseph Pemberton et al. “Cortico-cerebellar networks as decoupling neural interfaces”. In: *Advances in neural information processing systems* 34 (2021), pp. 7745–7759.

Efficient Feedback Encoding via Thalamo-Cortical Connections

Matthew Keaton[1], Manuel Mittag[2], Janko Petkovic[3], Katharine Shapcott[4]

Abstract

Artificial Neural Networks (ANNs) architectures are increasingly inspired by actual neural networks and circuitry found in the brain. In this report, we describe an experimental approach to investigating the potential role of the thalamus in learning and information processing in the brain. We propose that the thalamus may act as an efficient bridge between the cerebellum and the cortex, providing feedback signals that can enhance learning and improve the robustness of neural networks. To test this hypothesis, we designed a neural network model with a thalamus-like feedback loop, using autoencoders to process and compress the feedback signals. We compared the performance of this model to a baseline model trained using standard backpropagation. The results suggest that the thalamus-like feedback loop can indeed enhance learning and may improve the robustness of the model to adversarial samples. We suggest that further research is needed to explore the features of the model weights and to test the approach on other datasets.

1 Introduction

In order to make sense of the multitude of stimuli and to actively navigate its environment, every animal has to create an internal model of the world. This model is required to be highly flexible, as the brain constantly encounters novel inputs and has to learn to respond with appropriate behavioral outputs. In other words, learning means updating the internal model based on external stimuli. In this context, the brain crucially depends on environmental feedback, which can be compared to the internally produced behavioral patterns. From motor control, for example, it is known that afferent sensory signals inevitably have a certain degree of temporal delay in reaching the central nervous system. Contributing factors include synaptic delays and delays imposed by nerve conductance. The delay due to axonal transmission can thereby range from 10 ms for the shrew to more than 100 ms for an elephant [5, 7]. Overall, in terms of feedback-integration and updating of the internal model, this causes a critical problem: the feedback very often is incomplete, sparse, or simply too late. The brain must have developed tools or means to circumvent this major problem, otherwise learning would be hard to achieve. One mechanism, which has been proposed to cope with the delay in sensory feedback, is the computation of a future state of the body based on the current estimate and the efferent signal of the motor system. This predictive computation internally emulates or models an actual movement of the body by essentially solving an equation of motion of the body forward in time [7]. This model is known as the internal forward model.

Due to its unique structure, internal connectivity, computational power and due to the fact that it is placed as a parallel computational unit besides the sensory-motor system, it is assumed that the cerebellum is the origin of the brain's internal forward model that predicts motor feedback. The cerebellum is characterized by a highly organized and layered structure. The outer layer, called the cerebellar cortex, is made up of several types of neurons arranged in a repeating pattern of folds called folia. The cerebellar cortex has access to the necessary sensory information to create the internal

[1]In Silico Brain Sciences Group, Max Planck Institute for Neurobiology of Behavior - caesar
[2]DZNE, Bonn
[3]Mainz University Medicine, JGU Mainz
[4]Ernst Strüngmann Institute, Frankfurt am Main

model as it receives input from the brain stem, spinal cord, and cerebral cortex. However, the output of the cerebellum is not dense enough to be sent back to the cortical structures in order to inform the learning process directly. Rather, most of the output is first directed to the thalamus. The thalamus is situated in the diencephalon and is separated into different nuclei that are distinguished according to their individual inputs and the output regions they connect to. Neurons in the thalamus are mainly excitatory and show no local recurrent connections [6, 4]. Based on its architecture, connectivity and physiological features, it is assumed that the thalamus serves as a relay for cerebro-cerebellar information processing.
Recently, an Artificial Neural Network (ANN) architecture has been introduced, that mimics the computations of the cerebro-cerebellar system and which is capable of facilitating learning in typical sensorimotor tasks [1]. However, this architecture is biologically implausible due to the assumption of large numbers of direct feedback projections from the cerebellum to the cerebral cortex. Here, we propose the addition of a biologically plausible thalamic module to this architecture, which could both compress and relay the cerebellar feedback.

2 Methods

In this model, the cerebral area was designed as a simple feed-forward network where the input layer receives the input data and passes it on to the first hidden layer. The size of the input layer is 784. The first hidden layer receives the input data from the input layer and applies a linear transformation to it. The output of this layer is the passed through a ReLu activation function, which applies a non-linear transformation to the data. The number of neurons in this layer in this model are 32. The output of the first hidden layer is then passed to a second hidden layer with the same structure as the first hidden layer. Finally, an output layer applies a linear transformation to the data received from the second hidden layer. The output of the output layer is then passed through a log softmax function, which transforms the output into a probability distribution over the possible classes in the logartihmic space. The number of neurons in this layer equals the number of classes in the dataset. Training was performed using the Adam optimizer and Negative Log-Likelihood loss function. The thalamic area was designed as an individual auto-encoder attached to each layer (output, hidden2, hidden1). The auto-encoder takes in as input the gradients of each backward-pass. The architecture of the model is depicted in Fig. 1.
To investigate the dimensionality of the manifold that the gradients of different layers live on, three types of training are carried out on the FashionMNIST[8] dataset:

1. a classic back-propagation driven training is performed in order to obtain the baseline accuracy for the cortical model

2. a second training is conducted on a newly instantiated model, using gradients that have been processed by layer-specific autoencoders. Instead of feeding directly each set of gradients to the respective weights, they are first reconstructed via 3 layer-specific "thalamic" autoencoders. The autoencoders (one for each linear cortical layer) contain 2 encoding layers, one ten-dimensional bottleneck layer, and two decoding layers. Importantly, each autoencoder receives the *true* gradients as input, i.e. weight-dependent portion of the gradients is computed as usual, and is not affected by the introduction of the autoencoders.

3. the third training mirrors the second training, utilizing, however, one-dimensional bottleneck autoencoders.

FashioinMNIST was chosen to make the task suitably challenging. The trainings were carried out over 10 epochs using an ADAM optimizer, with a learning rate of 10^{-3} and a batch size equal to 64.
To evaluate the effect of the "thalamic" autoencoders introduction, we compare the test accuracy, the loss curve and the final weight distribution obtained with the three different trainings.

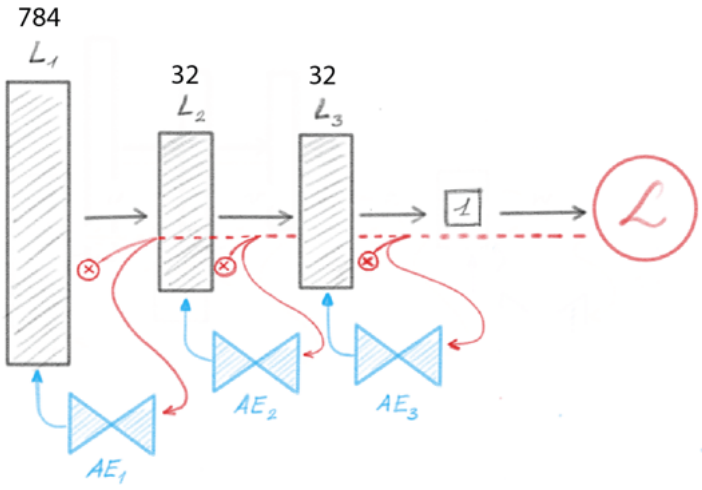


Figure 1: Schematic representation of the proposed model for thalamo-cortical processing. In training 1 the gradients are backpropagated through the cortical model in the usual fashion. In trainings 2 and 3, the gradients (red arrows) are fed into the layer-specific "thalamic" autoencoder and the output of the latter is used to update the cortical weights (blue lines).

3 Results

The cerebral cortical model was able to achieve 87% accuracy with the classical training, showing a performance compatible with the expected behaviour for this class of "naive" approaches (Fig. 2, gray line). Training with reconstructed gradients (Fig. 2, red and blue lines) achieved lower but surprisingly comparable results. Very intriguingly, training with one-dimensional bottleneck autoencoders was more effective than using 10-dimensional ones, both in terms of final accuracy (85.5% versus 84.5%. The same effectiveness increase could be observed in the loss vs epoch curves, where the 1-dimensional bottleneck training lead to a faster convergence compared to the 10-dimensional training. To explore a possible cause of these observations, we investigated the layer weight distributions arising from the three trainings (Fig. 3). Despite the overall lack of statistically significant differences, an interesting fact can be noted in the first layer of the model, where the 1-dimensional bottleneck training was able to achieve better sparsification (more peaked distribution) than the 10-dimensional bottleneck one.

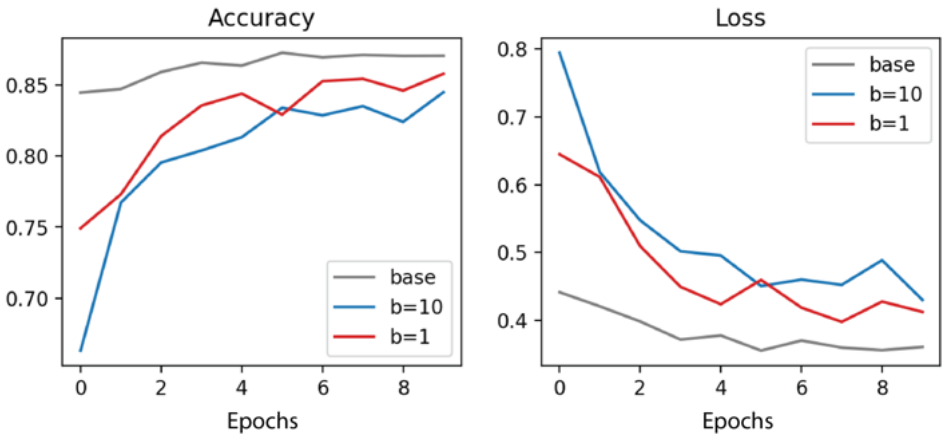


Figure 2: Training accuracy and loss for FashionMNIST dataset. Training accuracy and loss on FashionMNIST dataset was compared between model with auto-encoder with latent space dimension of 1 ($b = 1$, red), model with auto-encoder with latent space dimension 10 ($b = 10$, blue) and baseline model (base, grey) without auto-encoder.

4 Conclusions

Generally, the efforts shared in this work represent initial experimentation with the idea that, among other functions, the thalamus could operate as an efficient information bridge between the cerebellum and the cortex, with the former operating as a feedback signal to the latter. We believe our work would fit well within the framework of [1], which utilizes a cerebellum-like network to approximate the true error signal. This affords them the capability to provide feedback to the primary cortical model prior to the end of the task, producing more efficient learning, among other things. Additionally, they implement the cortical model as a one-layer recurrent neural network, which would allow for the thalamus to be represented by a singular autoencoder, enabling a greater alignment with the underlying biology.

One further group of hypotheses to test are whether the compressed feedback from the thalamus generates valuable features within the cortical network. For instance, it may be the case that the learned low-rank error signal primarily contains only the most important information for carrying out the given task, which might in turn produce a model that is more invariant to changes in the input. Similarly, it is conceivable that this type of learning could enable the cortex model to be more robust to so-called adversarial samples which are easily discernible by humans, an issue that current neural networks face. Digging into the features of the model could in part be facilitated by using datasets such as SmallNORB [3] or DAmageNet [2], in addition to more in-depth analysis of particular features of the model weights including rank and sparsity.

5 Acknowledgements

We would like to thank Maximilian Eggl and Laura Bernáez Timón for putting together this awesome workshop and Joseph Pemberton for supervising this project.

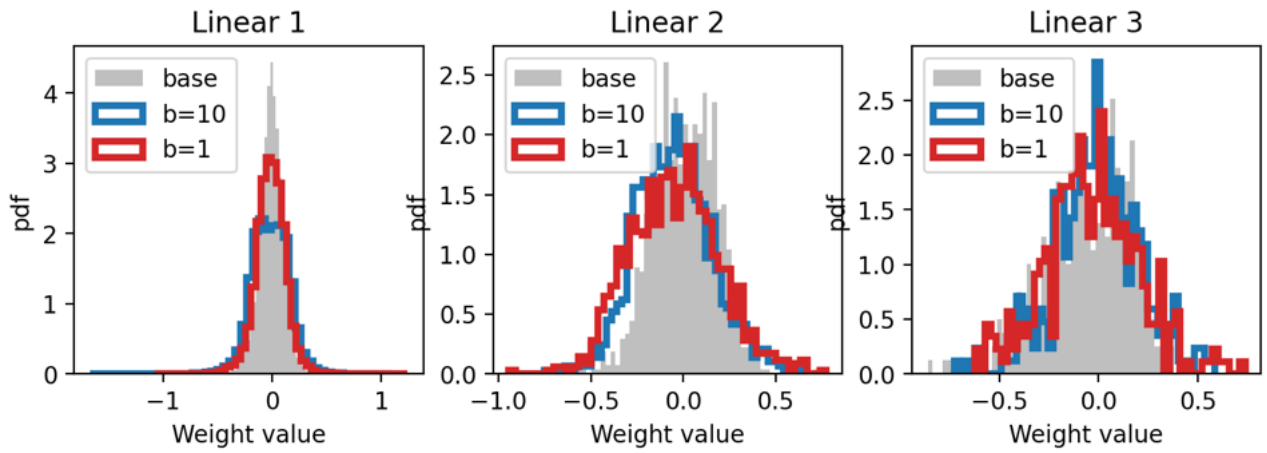


Figure 3: Cortical linear layers' weight distributions after training was carried out using classical gradient barckprop (gray), a 10-dimensional (blue), and a 1-dimensional bottleneck "thalamic" autoencoder (red). The addition of the thalamic unit does not induce any visible sparsification of the weights, achieving, surprisingly, the opposite effect in the first linear layer (left image).

References

[1] Ellen Boven et al. "Cerebro-Cerebellar Networks Facilitate Learning through Feedback Decoupling". In: *Nature Communications* 14.1 (Jan. 4, 2023), p. 51. DOI: 10.1038/s41467-022-35658-8.

[2] Sizhe Chen et al. *DAmageNet: A Universal Adversarial Dataset*. Dec. 15, 2019. DOI: 10.48550/arXiv.1912.07160. arXiv: 1912.07160 [cs, eess, stat]. preprint.

[3] Yann LeCun, Fu Jie Huang, and Léon Bottou. "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting". In: IEEE Computer Society, June 1, 2004, pp. 97–104. DOI: 10.1109/CVPR.2004.144.

[4] Anna S. Mitchell et al. "Advances in Understanding Mechanisms of Thalamic Relays in Cognition and Behavior". In: *The Journal of Neuroscience* 34 (2014).

[5] Heather L More and J. Maxwell Donelan. "Scaling of sensorimotor delays in terrestrial mammals". In: *Proceedings of the Royal Society B: Biological Sciences* 285 (2018).

[6] Rajeev V. Rikhye, Ralf D. Wimmer, and Michael M. Halassa. "Towards an Integrative Theory of Thalamic Function". In: *Annual Review of Neuroscience* 41 (2018).

[7] Hirokazu Tanaka et al. "The Cerebro-Cerebellum as a Locus of Forward Model: A Review". In: *Frontiers in System Neuroscience* 14 (2020).

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].

ponents. The goal of this project was to modify the BurstCCN model to satisfy Dales' principle. Four approaches were evaluated: restricted weights sign, weight separation and perturbations, and the addition of an inhibitory layer. The results showed that the hard boundary approach was unable to reproduce the accuracy of the BurstCCN model and resulted in increasingly sparse weight matrices. The weight separation and perturbation approach also failed to allow BurstCCN training and the random perturbations along synaptic connections did not change the landscape of the loss function.

Potential improvements of the inhibitory layers insertion approach would be to reduce the amount of inhibitory neurons, to resemble more the cortex organization; initialize differently the fixed W_{IE} weights, which in our solution were simply an identity matrix; and introduce inhibitory layers compensating the Q and Y positive weights matrices. Additionally, a physics-based approach into adapting the loss function for soft-boundaries was also quite promising. Further work is needed to find a better soft-boundary approach to satisfy Dales' law.

5 Acknowledgements

We thank William Greedy, the author of the original model we have worked on [4], for his support and ideas during the development of this project. We are also very grateful to the Bio-inspired DL Workshop organizers for allowing us to learn about this area, meet experts on it, and work closely with these types of models.

References

[1] Thierry Biben, Klaus Kassner, and Chaouqi Misbah. "Phase-field approach to three-dimensional vesicle dynamics". In: *Physical Review E* 72.4 (2005), p. 041921.

[2] Jonathan Cornford et al. "Learning to live with Dale's principle: ANNs with separate excitatory and inhibitory units". In: *bioRxiv* (2020), pp. 2020–11.

[3] Henry Dale. *Pharmacology and nerve-endings*. 1935.

[4] Will Greedy et al. "Single-phase deep learning in cortico-cortical networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24213–24225. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/99088dffd5eab0babebcda4bc58bbcea-Paper-Conference.pdf.

[5] Pierre C Hohenberg and Bertrand I Halperin. "Theory of dynamic critical phenomena". In: *Reviews of Modern Physics* 49.3 (1977), p. 435.

[6] Neville N Osborne. "Is Dale's principle valid?" In: *Trends in Neurosciences* 2 (1979), pp. 73–75.

[7] Alexandre Payeur et al. "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits". In: *Nature neuroscience* 24.7 (2021), pp. 1010–1019.

[8] Blake A Richards and Timothy P Lillicrap. "Dendritic solutions to the credit assignment problem". In: *Current opinion in neurobiology* 54 (2019), pp. 28–36.

[9] H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. "Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework". In: *PLoS computational biology* 12.2 (2016), e1004792.

[10] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.