EFFICIENT LEARNING IN BIOLOGICAL NEURAL NETWORKS

PART IB - TECHNICAL MILESTONE REPORT

EDITED BY

KYLE MCMILLAN

SUPERVISOR: DR YASHAR AHMADIAN

UNIVERSITY OF CAMBRIDGE

Abstract

Artificial neural networks (ANNs) use simple synapses where weights are composed of only a single scalar value. However neural synapses contain contain many chemicals that act with a range of dynamic timescales. This range of timescales is believed to be important for increasing memory capacity and learning efficiency. This project aims to investigate the impact of using synapses composed of a mixture of dynamical states with a mixture of rigid and plastic components. To test this we propose a biological update rule called a recurrent chemical network (RCN) that contains a range of timescales. In order to keep with biological plausibility synapses in this network are only updated using local information. This report presents work undertaken so far and results that show case an improvement over an earlier model.

Introduction

Many systems that are designed to model the brain use traditional artificial neural networks (ANNs) composed of weights made up of scalar values. However unlike in ANNs, the strength of synapses in the brain are controlled by a range of chemicals that act on a range of timescales. This abstraction of synapses to a single scalar value is believed to be a limiting factor in the memory capacity and learning efficiency of ANNs.

This project aims to create neural networks where synapses more closely resemble those found in the brain. Specifically we incorporate a mixture of rigid and plastic components called chemicals in place of the singular scalar value of ANNs.

Background

Biological synapses are composed of many chemicals [1] that alter neural plasticity at different timescales. Phosphorylation of AMPA receptors has been shown to be a rapid short term mechanism that affect synaptic plasticity [2] on the other hand endocannabinoids can act over longer timescales to cause long term synaptic depression [3].

Theoretical models have shown that this range of timescales is important for increasing memory capacity. Work on simple synapses shows that when bounded the number of memories grows at best linearly with the square root of then number of synapses [4]. However more recent work has shown that complex synapses with plastic and rigid components can store memories almost linearly with the number of neurons [5]. By using a decay mechanism where successive components decay at different rates in accordance with τ^{-k} where k is the number of the component and τ is the time constant, the number of memories that can be stored greatly increases.

Learning in Biological Neural Networks

Traditional ANNs are typically trained using backpropagation:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{1}$$

where w_{ij} is the weight between neuron i and j, η is the learning rate and E is the error function. Backpropagation effectively solves the credit assignment problem (how responsible each weight is for the error of the network) by using the chain rule to propagate errors backwards through the network. However backpropagation biological implausibility is well documented [6] as it requires symmetric weights during it's forward and backwards pass. To combat this and explain learning in the brain a number of alternative learning rules have been proposed.

As [7] points out these rules typically follow the path of the gradient, however with a variance and bias. Some like simulated annealing [8] have an extremely high variance while others like feedback alignment (FA) [9] have a high bias. Feedback alignment is a learning rule where errors are propagated backwards through a network using fixed random matrices that cause weights to align themselves in this direction.

Specifically we will focus on a biological learning rule proposed by Shervani-tabar in [10]. Shervani-tabar proposes an improvement on FA, where they meta-learn a biological learning rule composed of a set of 10 biological plausible motifs to update weights of a neural network. These motifs are created from combining the activation before and after the weight, the FA error before and after the weight and the weight itself. Specifically they focus on 3 specific motifs as being responsible for the majority of the learning:

$$F^0 = e_l y_{l-1}^T \tag{2}$$

$$F^1 = e_l e_{l-1}^T \tag{3}$$

$$F^2 = y_l y_{l-1}^T - (y_l y_l^T) W_{l-1,l} \tag{4}$$

Where e_l is the FA error at layer l, y_l is the activation at layer l and $W_{l-1,l}$ is the weight matrix between layers $l-1$ and l. 2 is the FA gradient, 3 is a hebbian style error rule that reduces alignment angels between weights and a backpropagated basis and 4 is Oja's rule.

Weights are then updated using a linear combination of these rules:

$$W[t+1] = W[t] + \mathbf{P}^T \mathbf{F} \tag{5}$$

where $\mathbf{P}$ is the weighting vector that gets meta-learned and $\mathbf{F}$ is the biological motifs.

These adjustments allow the rules to more closely approximate the true gradient while still being biologically plausible and perform better when compared to pure FA. However these rules are still limited by the fact the weights are simple synapses. We aim to extend these rules to work with more complex synapses and will compare to Shervani-Tabar as a baseline.

Meta-Learning

Meta-learning [11] or learning to learning is the process of optimizing a learning algorithm. Specifically most meta-learning algorithms involves 2 loops. The inner loop is the learning loop where the model is updated by a learning rule (in our case the biological learning rule of the RCN) and the outer loop is the meta-learning loop where the learning rule's meta-parameters are updated. This is typically done calculating the gradient of the loss with respect to the meta-parameters.

Methods

Model and Task

We implement a 5 layer feedforward neural network with 784-170-130-100-70-47 dimensions where weights are recurrent chemical networks (RCNs). The network is trained on the EMNIST dataset [12].

In order to compare to the Shervani-tabar baseline we implement the same meta-learning algorithm. Specifically this algorithm involves selecting 5 random classes from the dataset and updating the weights from 50 examples using online learning and the biological learning rule. 10 test examples are then used to calculate the accuracy of the model and the meta-parameters are updated using the gradient of the loss with respect to the meta-parameters.

Recurrent Chemical Networks (RCNs)

In order to implement the RCNs we use the following equations:

$$\mathbf{h}[t+1] = \mathbf{y} \odot \mathbf{h}[t] + \mathbf{z} \odot f(K\mathbf{h}[t] + P\mathbf{F}[t] + \mathbf{b}) \tag{6}$$

$$w[t+1] = \mathbf{v} \cdot \mathbf{h}[t] \tag{7}$$

Where $\mathbf{h}$ is the chemical weights, w is the synapse weight, $\mathbf{y}$ is the decay rate, $\mathbf{z}$ is the gain, $\mathbf{F}$ is the Shervani-Tabar biological motifs, K is the recurrent weight matrix, P is the motif weight matrix, b is the bias, v is the weighting of the chemicals and f is the tanh activation function. They have sizes: $\mathbf{h}-(L,W), \mathbf{y}-L, \mathbf{z}-L, K-(L,L), P-(L,\text{Number of motifs}), \mathbf{v}-L, \mathbf{b}-L$. Where L is the number of chemicals and W is the size of the weight matrix - implementation details mean this can either be the actual 2D weight matrix or a flattened 1D weight matrix, both are equivalent. Meta Parameters P, K, v and b are meta-learned while y, z are fixed, all parameters and the number of chemicals are shared across all layers. f is the tanh activation function as experimenting has shown it outperforms other activation functions. $\mathbf{h}$ is randomly initialized to the same values using the Xavier initialization, $\mathbf{v}$ is initialized as $1/L$ and K and b are initialized to 0. Note the bias is layer specific.

In order to get the range of timescales $\mathbf{z}$ and $\mathbf{y}$ are initialized as follows:

$$\tau_i = \left(\frac{\tau_{\max}}{\tau_{\min}}\right)^{i/L}, \; z_i = \frac{1}{\tau_{min}\tau_i} \tag{8}$$

$$(y_i)_{i\neq 0} = 1 - z_i, \qquad y_0 = 1 \tag{9}$$

Where $\tau_{\max}$ and $\tau_{\min}$ are the maximum and minimum timescales respectively. We set $\tau_{\max} = 50$ (number of training images) and $\tau_{\min} = 1$.

These equations are a simplified version of the Gated Recurrent Unit (GRU) [13] where the reset gate is removed and the update gate is just a non-trainable bias.

Results

Baseline Comparison

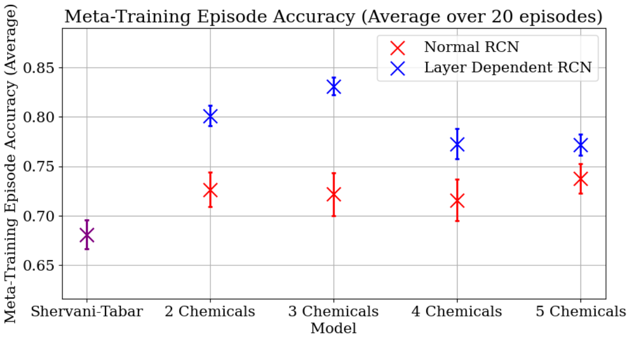

Figure 1: Comparison of the accuracy of the Shervani-Tabar baseline (purple), normal RCN (red) and Layer Dependent RCN (blue).

Fig1 shows the accuracy of the Shervani-Tabar baseline and the RCN model in red with different amount of chemicals per layer. We find our model outperforms the Shervani-Tabar baseline. However we also find that the number of chemicals has a fairly negligible impact on the performance of the model as performance of the model is fairly consistent across the number of chemicals. Our initial hypothesis would dictate that the model would perform better with more chemicals and so a larger range of timescales and while there is likely to be a saturation point this would probably not be reached with the number of chemicals we have used.

Per Layer Learning Rules

In the previous section the RCN equation 6 was the same for all layers. However we can hypothesis that neurons in different layers would require different learning rules. To test this we make the P, K, v meta parameters layer specific.

Fig1 shows the accuracy of the Shervani-Tabar baseline, the normal RCN and the Layer Dependent RCN. We find that the layer specific parameters outperform the shared layer parameters. Especially for the 2 and 3 chemical RCNs where the layer specific parameters outperform the shared layer parameters by a significant margin.

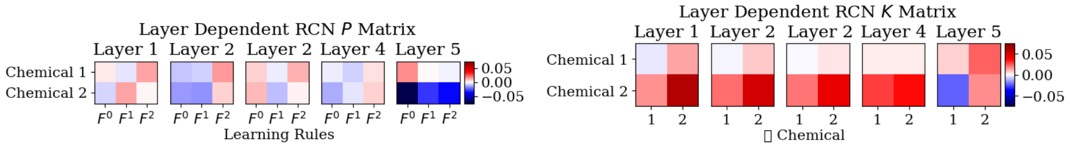

Figure 2: Left: P Matrix for the 2 chemical layer dependent RCN for learning rules F^0, F^1, F^2. Right: K Matrix for 2 chemical layer dependent RCN. Chemical 1 has fast changing timescales while chemical 2 has slow.

Fig 2 shows the P matrix for the 2 chemical RCN for learning rules F^0, F^1, F^2 - the 3 most important learning rules highlighted by Shervani Tabar. In the individual layer model case it is extremely interesting specifically with note to F^0. F^0 is the pseudo-gradient of the FA error and the activation before the weight. In the shared layer model this always trains positive - however in the individual layer model it trains negative for some layers and positive for others and seems to alternate. This could be causing the improvement in performance as it may somehow minimize bias in the propagated errors from the fixed FA weights.

Another very interesting point is that the strength of the P matrix in layer 5 is much larger than the other layers. This could be optimal however it could also be the result of the meta-learning backpropagation algorithm. Since there is no residual connections between layers we could be experiencing a vanishing gradient problem. Further investigation is needed to determine if this is the case.

The K matrix for the 2 chemical layer dependent RCN is also shown in Fig 2. It is less interesting than the P matrix however it is worth noting that the last layer is very different from the other layers. This could be due to the fact that the last layer is the output layer and so the weights need to be updated differently to the other layers.

Per Neuron Biases

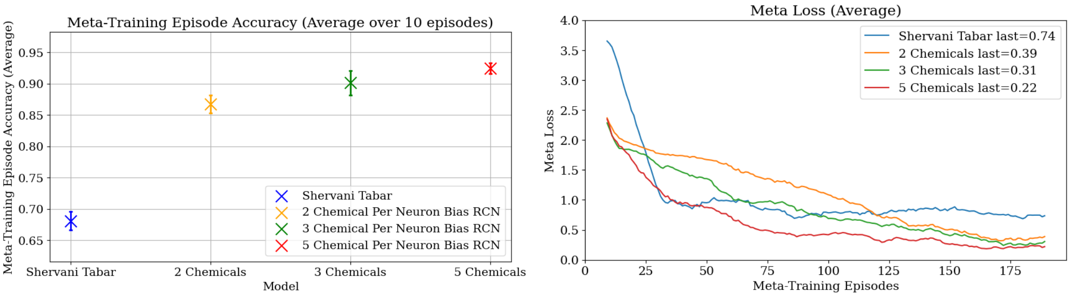

Figure 3: Left: Meta-Accuracy of a RCN with neuron specific biases. Right: Loss of a RCN with neuron specific biases.

It is extremely unlikely that individual neurons in a layer would have different learning rules - it would require a degree of complexity that is not biologically plausible. However it is possible that neurons in a layer could have different biases. To test this we make the b parameter neuron specific. This makes the bias to have a size of (L,W) where W is the weight matrix size.

Fig 3 present the meta-accuracy and loss of a RCN with neuron specific biases. These models perform significantly better than all models tested so far. The loss curve shows that the model is able to more accurately fit the data and the meta-accuracy curve is much improved.

However there is a chance the model is overfitting. The model could have learnt a specific bias for each neuron that is optimal for the training data but not generalizable. To test this we train the model with only neuron specific biases and no other update motifs. This model doesn't train at all - it is unable to learn the data. This suggests that the model is not overfitting the data in this way.

However a disadvantage is that this greatly increases the number of meta-parameters. The number of mta-parameters in the original RCN was $L^2 + L(\text{Number of update motifs}) + L + Ll$ - where L is the number of chemicals, l is the number of layers. This corresponds to the number of chemicals squared (K), the number of chemicals times the number of update motifs (P) and the number of chemicals (v) and the number of chemicals times the number of layer (b). The number of meta-parameters in the neuron specific bias model is $L^2 + L(\text{Number of update motifs}) + L + 2LWl$. This is a huge increase in the number of parameters and could be explain the increase in performance - more parameters = more capacity to learn the data - however whether or not this is biologically plausible is another question.

Further work is needed to determine if this is the case. Specifically the bias needs to be further investigated to see how it is acting (if it is acting as some sort of edge highlighter in early layers for example this is much more biological plausible). Secondly this trained RCN needs to be tested on another dataset to see if it is generalizable.

Adjusting the RCN Output

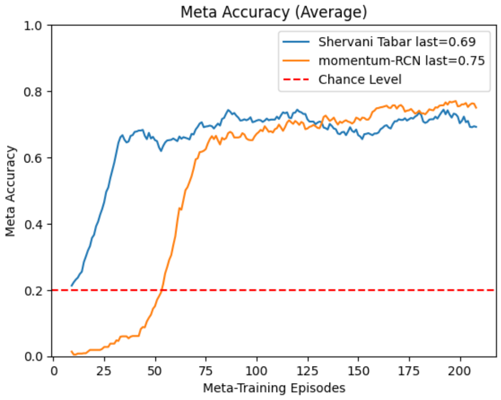

Figure 4: Meta accuracy of a 5 chemical RCN with momentum output

The RCN output is currently a linear combination of the chemical weights as defined in equation 7. However an alternative RCN called the momentum-RCN could be constructed where the output rule is instead:

$$w[t+1] = \mathbf{v} \cdot \mathbf{h}[t] + w[t] \tag{10}$$

A particularly interesting point here is that when expressed in this way the RCN would be approximately equivalent to a gradient descent with momentum. This provides a biological plausible explanation for the momentum term in gradient descent.

Fig4 shows the meta-accuracy of a 5 chemical RCN with momentum output. This model slightly outperforms Shervani-Tabar, however further work would need to be done on the optimal initialization of chemical weights and the optimal number of chemicals as well as what y and z should be.

Discussion and Future Work

Overall the work so far presents an introduction of a biological plausible learning rule in the RCN model. Results indicate that the model outperforms the Shervani-Tabar baseline and the further modifications such as layer specific parameters and neuron specific biases improve the model further. However we also find that the number of chemicals past 1 has a negligible impact on the performance of the model. Lastly we present a major modifications in the form of the momentum-RCN which could provide a biological plausible explanation for momentum in the brain.

Future work will go down two paths. The first is an adjustment to the meta-training task. Classification may not utilize the full potential of chemicals with different timescales so migrating to a regression task may be more appropriate. The second is to investigate more architectural changes to the RCN. Perhaps the first major change is to make the RCN y and z parameters depend on the chemicals themselves like in the GRU. This would allow the model to learn the optimal timescales for each chemical and is biological plausible as chemical timescales like the endocannabinoids are dependent on other chemicals for their timescales. A second architectural change to be investigated is to make the fixed FA weights adaptable and train a RCN rule to update them. This would allow the model to learn the optimal feedback weights for the task at hand and may offer a way to get closer to backpropagation's performance.

Bibliography

[1] "Principles of Neural Science, Fifth Edition". In: *Principles of Neural Science, Fifth Edition*. New York, NY: McGraw-Hill Education, 2014.

[2] John Q. Wang et al. "Phosphorylation of AMPA Receptors". In: *Molecular Neurobiology* 32.3 (Dec. 2005), pp. 237–249.

[3] Vivien Chevaleyre et al. "Endocannabinoid-Mediated Long-Term Plasticity Requires cAMP/PKA Signaling and RIM1alpha". In: *Neuron* 54.5 (June 2007), pp. 801–812.

[4] Stefano Fusi and L. F. Abbott. "Limits on the Memory Storage Capacity of Bounded Synapses". In: *Nature Neuroscience* 10.4 (Apr. 2007), pp. 485–493.

[5] Marcus K. Benna and Stefano Fusi. "Computational Principles of Synaptic Memory Consolidation". In: *Nature Neuroscience* 19.12 (Dec. 2016), pp. 1697–1706.

[6] Changze Lv et al. *Towards Biologically Plausible Computing: A Comprehensive Comparison*. June 2024.

[7] Blake A. Richards et al. "A Deep Learning Framework for Neuroscience". In: *Nature neuroscience* 22.11 (Nov. 2019), pp. 1761–1770.

[8] Harold Szu and Ralph Hartley. "Fast Simulated Annealing". In: *Physics Letters A* 122.3 (June 1987), pp. 157–162.

[9] Timothy P. Lillicrap et al. "Random Synaptic Feedback Weights Support Error Backpropagation for Deep Learning". In: *Nature Communications* 7.1 (Nov. 2016), p. 13276.

[10] Navid Shervani-Tabar and Robert Rosenbaum. "Meta-Learning Biologically Plausible Plasticity Rules with Random Feedback Pathways". In: *Nature Communications* 14.1 (Mar. 2023), p. 1805.

[11] Jurgen Schmidhuber. "Evolutionary Principles in Self-Referential Learning. On Learning now to Learn: The Meta-Meta-Meta...-Hook". Diploma Thesis. Technische Universitat Munchen, Germany, 14 May 1987.

[12] Gregory Cohen et al. *EMNIST: An Extension of MNIST to Handwritten Letters*. Feb. 2017.

[13] Kyunghyun Cho et al. *Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation*. Sept. 2014.