Sequential activation in biologically inspired RNNs

Alfred Ajay Aureate Rajakumar AAR653@MATH.NYU.EDU
Amartya Prasad AP5891@MATH.NYU.EDU
Department of Mathematics,

Weiyi Wang WW1420@NYU.EDU
Department of Computer Science, Courant Institute of Mathematical Sciences,
New York University, New York, NY 10012, USA

Editor: [Course project for Machine Learning course at NYU]

Abstract

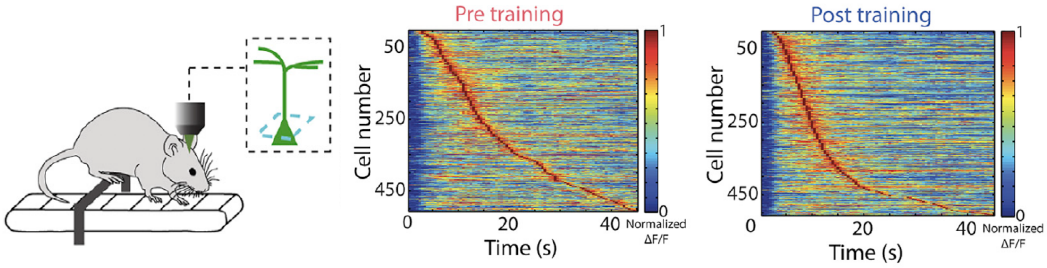
This paper uses a Biologically inspired RNN, that is, RNN constrained by the Dale's Principle, to simulate three different learning processes of an animal brain. We simulate the sequential activation pulses of neurons and perform different modifications to the weights to show experimentally observed shift in the activation due to inhibitory effects during motor cortex learning; characterize the structure of the (neuron) network to learn multiple tasks and attempt to solve the problem of Catastrophic Forgetting by combining two different techniques for mitigating - forgetting in basic NNs and MLPs and applying them in this Biological Framework.

1. Introduction

In neuroscience, it is widely known that during a variety of behaviors - including working memory and decision making, there is a sequence of activation or firing of neurons (or a collective of neurons) that happens in many parts of an animal brain - cortex, hippocampus, basal ganglia and cerebellum. This has been modeled by using artificial RNNs constrained by the Dale's Principle, that neurons are purely excitatory or inhibitory. We use the network from (Song et al., 2016) for our experiments, with the same architecture and structure.

1.1 Shift of sequential activation pulses of neurons

Recently, in (Adler et al., 2019), the authors have shown that after motor skill training, the sequential activation of neurons in the primary cortex of a mice brain could be affected due to directly and indirectly induced inhibitory effects. In this report, we show using simulation of the neuronal dynamics through continuous-time artificial RNNs, how such inhibitory effects could affect the sequential activation of neurons as observed in the experiments of the mice brain. For replicating the inducement of the inhibitory effects on RNNs, we simply modify the inhibitory parts (or connections) of the recurrent weight matrix of a trained network (fig. 1). Such effects of modifying recurrent weights are done in (Goudar and Buonomano, 2017) for different tasks.



(a) Training of mice (b) Pre-training (c) Post-training

Figure 1: Experiment with mice [Source: Ref. (Adler et al., 2019)]

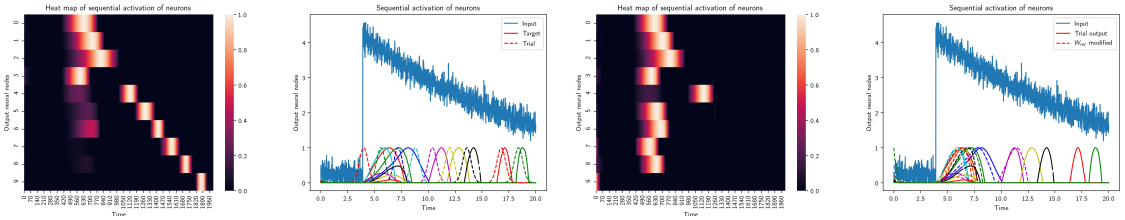
1.2 Multi-task network structure

Another topic in neuroscience is how the neuron network organizes multiple tasks in one network. In this report, we show possible neuron network structure characteristics by simulating with a RNN that learns multiple tasks and measuring the recurrent connection weight.

1.3 Catastrophic Forgetting

When a Neural Network is trained to learn some task and then retrained to learn another task, the performance of the network on the first task degrades considerably. This phenomenon is referred to as Catastrophic Forgetting. Biological RNNs also suffer from Catastrophic Forgettingadd reference to the paper that defines this. We combine dropout with Elastic Weights Consolidation(EWC) (Kirkpatrick et al., 2016) to mitigate forgetting in our constrained network. The network is first trained with dropout on the recurrent nodes. and then retrained after regularizing by adding an EWC penalty to the Cost Function. Implementing dropout can make it easier to isolate weights that were important for learning the first task while the EWC term penalises the network for straying too far from the learned parameters for task 1. This is explained in detail in the Methods section. There are both neuro-biological and machine learning motivations to attempt mitigate forgetting in such RNNs. From a Machine Learning Perspective, this has to do with computational costs and access to data. While it has been shown that training the network on data while replaying past data simultaneously and jointly optimising parameters can achieve performance as high as training separate networks independently, such an exercise would require massive working memory, besides computationally very expensive and impractical (Shin et al., 2017). Further, access to past data may not always be feasible.

Coming to the neuro-biological motivations, we know that humans are capable of continually acquiring new skills and knowledge while also retaining past knowledge. Being able to mitigate forgetting in Biologically Constrained ANNs(Artificial Neural Networks) could give significant insights into the mechanisms behind such memory retention in humans.



(a) Heatmap of trial output (b) Input, target and trial (c) (d)

Figure 2: Sequential activation trial outputs of the trained network as heatmaps ((a) and (c)) and as plots ((b) and (d)). (a) Heatmap of the sequential activation trial outputs; (b) Plots of input, target output and trial output (trial output same as (a)); (c) Heatmap of the sequential activation trial outputs after modifying W_{rec}; (d) Plots of input, target output and trial output after modifying W_{rec} (trial output same as (c)).

2. Methods

The input given is (varied for different experiments) a sudden pulse with exponentially decaying tail. The recurrent nodes network represent a part of an animal brain while satisfying the continuous-time neuronal dynamical equations (refer Appendix A). The output nodes are (depending on the task) either a subset of the recurrent nodes themselves (partially identity output layer W_{out}) or a separate layer. See Appendix D for the insignificant effect of W_{out}. The details of the dynamics governing this project are given in Appendix A.

2.1 Shift of the sequential activation pulses of neurons

The network was trained for two cases: i) $N = 100$ & $N_{out} = 10$ and ii) $N = 100$ & $N_{out} = 100$. It can be seen in fig. 2 that the output activation or pulses are shifted when some of the columns of I/E weights of W_{rec} matrix are modified. In order to study this effect in more detail, the "shift" was quantified using a measure given as follows:

$$shift_i = (t_{i,max,act} - t_{i,max,shifted}) \times \delta t\,,$$

where $t_{i,max}$ is the time instant where the i^{th} node output is maximum. δt is the step size (for this task, it is 0.02s). For the plots in fig. 3, these $shift_i$'s are averaged over all output nodes. This $shift$ measure is basically the difference between the maximum activation points of the actual trial output and the modified (shifted) output. The modification done for fig. 2 is reducing W_{rec} matrix's 87th column (1st 80 rows) weights by 0.04. For this task, the output nodes are just a subset of the recurrent nodes themselves, that is, W_{out} is a partial identity matrix.

2.2 Characterizing multi-task network

We investigate the neuron network structure and its changing when learning multiple tasks simultaneously. We set up the following training for multiple-task learning:

- The input is a pulse with exponential decay at a random time, similar to single-sequence training. However, the pulse can have N possible magnitude. The magnitude indicates different input type.

- There are $5 \cdot N$ output nodes. Upon receiving the input pulse, 5 nods out of all give sequential activation, while other nodes remain zero. Which group of 5 nodes is correspond to the input type (pulse magnitude).

We trained several network with $N = 1, 2, 4,$ and 8, respectively. We then investigated the structure of these networks, specifically the weight distribution and activation rate, to see how it changes for different amount of learned tasks.

2.3 Dropout and EWC for Catastrophic Forgetting

We first train the network to learn this first task, with dropout on the recurrent nodes which prevents the nodes from becoming co-dependent. Since the nodes then learn the task independent of each other, it should become possible to distinguish between nodes that are more important for learning from those less so. Thus, the nodes less important for learning the first task could then be used for learning the second task, while changing the other nodes as little as possible, thereby preserving previously learned knowledge. Following this, the already trained network is then retrained, to learn the second task with EWC regularisation. EWC involves modifying the cost function for the training of the second task by adding a regularisation term. This regularisation term penalises the network for moving away from the learned parameters with greater penalties for weights that were more important for learning the first task acting as a buffer that springs the new solution close to the previous learned solution. This outlines the theoretical argument for combining dropout with EWC.

The network is trained to learn the second task by minimising the cost function :

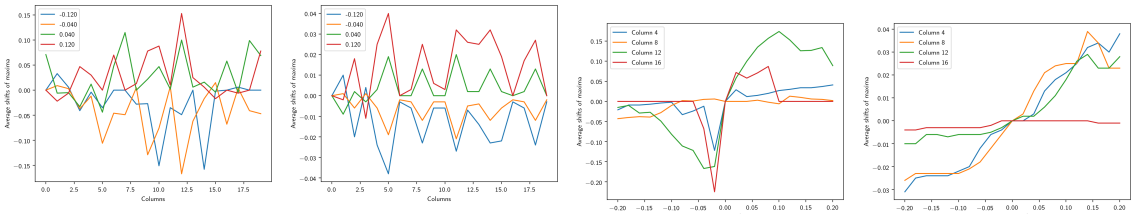
$$L(\theta) = L_2(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta_{1,i})^2$$

where, $L_2(\theta)$ is the squared loss for task 2, θ_1 is the set of parameters(weights) obtained after learning the first task, and F_i is the i-th diagonal element of the Fisher Information Matrix. λ is a hyper parameter that we use to tell the network how important the first task is, relative to the first task, that is, it determines the magnitude of the penalty. The Fisher information matrix, essentially tells us how important each parameter(weight) is to learning of the first task, relative to other parameters. The inputs, outputs and other details can be found in E.

3. Results

3.1 Shifts of sequential activation pulses for different parameters:

From fig. 2(b), we see that the trial output is close to the target output and from the heatmap view as in fig. 2(a), the pulses simulate very closely the similar pulses observed in the experiment with mice as shown in fig. 1(b). Now, after modifying the I/E column weights of the W_{rec} matrix, we find that the sequential activation pulses are shifted in time (fig. 2(c) and (d)) as observed in the experiment with mice (fig. 1(c)). Fig. 2(d) shows the shift in the pulses for each output node. These shifts were studied in more detail and the variation of these shifts were plotted for different choice of the I/E columns and different modification values are plotted in fig. 3. It could be observed that when I/E column weights of the W_{rec} matrix are decreased by a small value, the sequences shifts towards the left and vice versa (fig. 3(c) and (d)). Also, the shift magnitude seems to



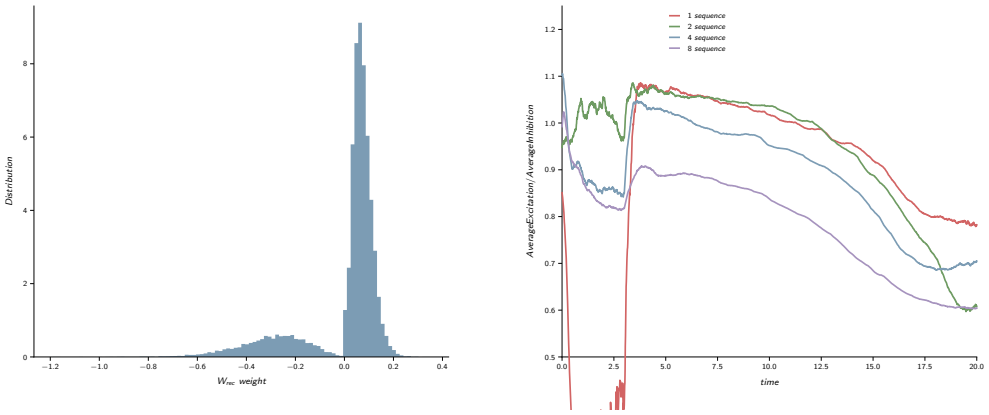
(a) $Shift$ vs (80+i)-th col. variation - case 1

(b) $Shift$ vs (80+i)-th col. variation - case 2

(c) $Shift$ vs values - case 1

(d) $Shift$ vs values - case 2

Figure 3: Effect of modifying I/E columns (columns 80 to 100) of W_{rec} matrix, with different values. (a) and (b): Variation of $Shift$ as different I/E columns are modified for cases-1 and 2 respectively. (Column labels are integers values of i, where $col(80+i)$ is the col. modified). This exercise is performed for different modification values, i.e. $W_{rec} = W_{rec} - value$; (c) and (d): Variation of $Shift$ as a particular I/E column weights are modified by a range of values, for cases - 1 and 2 respectively. This exercise is repeated by modifying weights of different columns.



(a) W_{rec} weight distribution of 4-sequence network

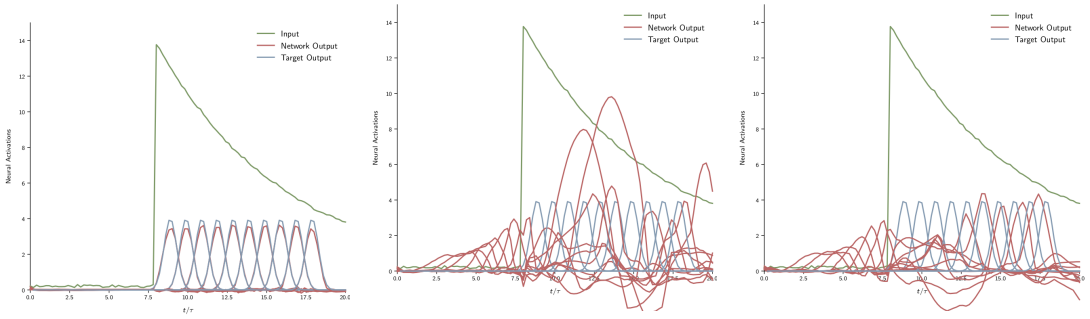
(b) Excitatory/Inhibitory average activation ratio

Figure 4

depend on the column that is being chosen for modification as shown in fig. 3(a) and (b). This is in accordance to the shifts observed in the sequential pulses due to the inhibitory effects induced in the hippocampus of the mice brain after motor learning (ref. (Adler et al., 2019)). In Appendix B, the mathematical explanation for observing such shift in the sequential pulses is explained.

3.2 Weight distribution and activation rate for multi-task network

We investigated the weight distribution of neuron networks that have learned multiple sequences. Figure 4a shows the weight distribution of networks that learn 4 sequences. It can be seen that the positive/excitatory weights are concentrated to small values, while negative/inhibitory weights have wide normal-like distribution. The wider distribution of inhibitory is likely related to the smaller portion of inhibitory neurons. The wide distribution of inhibitory neurons balance out with the large quantity of excitatory neurons. We will later show that the inhibitory neurons play an important role in controlling the correct output. A wide distribution of weights is important to keep this kind of dynamics going.



(a) Performance of the network on the first task

(b) Performance of the network on the first task after being re-trained for the second task with EWC

(c) Performance of the network on the first task after being re-trained for the second task without EWC

Figure 5: Comparison of Network Performance on Task 1 Before and After Training for Task 2

There is no significant difference between distributions for networks that learned different amount of sequences (See Appendix C for full plot). This indicates the network does not shift the overall weight distribution to learn more task, at least within the current test range. This can be explained by that the weight matrix must maintain an overall norm to avoid vanishing or explosion in the network.

We also investigate the activation rate of excitatory and inhibitory neurons in the network. Figure 4b is the plot of ratio between average activation of excitatory neurons and average activation of inhibitory neurons, for networks that have learned different amount of sequence. It can be seen that for networks that has learned more sequence, the E/I activation ratio is lower. This reveal a possible internal mechanism of the neural network, which might also apply to organic networks. The inhibitory neuron inhibits those neurons that would output the wrong sequence, and only allows the right one to activate. The more sequence it learns, the more inhibition happens to different outputs.

3.3 Dropout and EWC - Performance Outcomes

To assess the effectiveness of our strategy for mitigating forgetting we compare the performance of the network on the first task before training for the second task and then after training for the second task, to measure it's performance. We compare plots of our network output for the first task against the target output for both cases.

Looking at the plots in 5, one can see that the networks performs near perfectly on the first task, before training for the second task, with the network outputs almost overlapping with the true outputs. Looking at the second plot, it is clear that while the performance of the network does degrade significantly, the network still somewhat 'remembers' the previous task. Unfortunately, Retention is much lower than that achieved in (Kirkpatrick et al., 2016) and performance looks similar to that without EWC(the third plot), which could be a result of the choice of λ, or the vastly differing tasks, the biological constraints, the or that perhaps combining Dropout and EWC may not be a good strategy in general.

Supporting Information

The related code is available at https://github.com/wwylele/pycog. The branch specifically related to shift of sequential activation is at https://github.com/wwylele/pycog/tree/shift.

Acknowledgments

We would like to acknowledge Prof. Zhe Chen, from NYU Langone for the idea and discussions. Additionally, we would also like to acknowledge Prof. Rajesh Ranganath, Mark Goldstein, Mukund Sudarshan and Aahlad Manas Puli for the discussions.

Author Contributions

Shift in the sequential activation pulses was studied by AAAR. Multi-task network structure was studied by WW. Catastrophic Forgetting and Dropout/EWC Experiments were conducted by AP

References

A. Adler, R. Zhao, M.E. Shin, R. Yasuda, and W.-B. Gan. Somatostatin-expressing interneurons enable and maintain learning-dependent sequential activation of pyramidal neurons. *Neuron*, 102(Issue 1):202–216.e7, 2019. doi: https://doi.org/10.1016/j.neuron.2019.01.036.

Vishwa Goudar and Dean Buonomano. Encoding sensory and motor patterns as time-invariant trajectories in recurrent neural networks. 2017.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL `http://arxiv.org/abs/1612.00796`.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *CoRR*, abs/1705.08690, 2017. URL `http://arxiv.org/abs/1705.08690`.

HF Song, GR Yang, and XJ Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. *PLoS Comput Biol.*, 12(2)(e1004792): 1–30, 2016. doi: https://doi.org/10.1371/journal.pcbi.1004792.

Appendix A. Equations for simulating the neuronal dynamics:

From (Song et al., 2016), we see that the actual dynamical equation that is simulated using the continuous-time RNNs is given by:

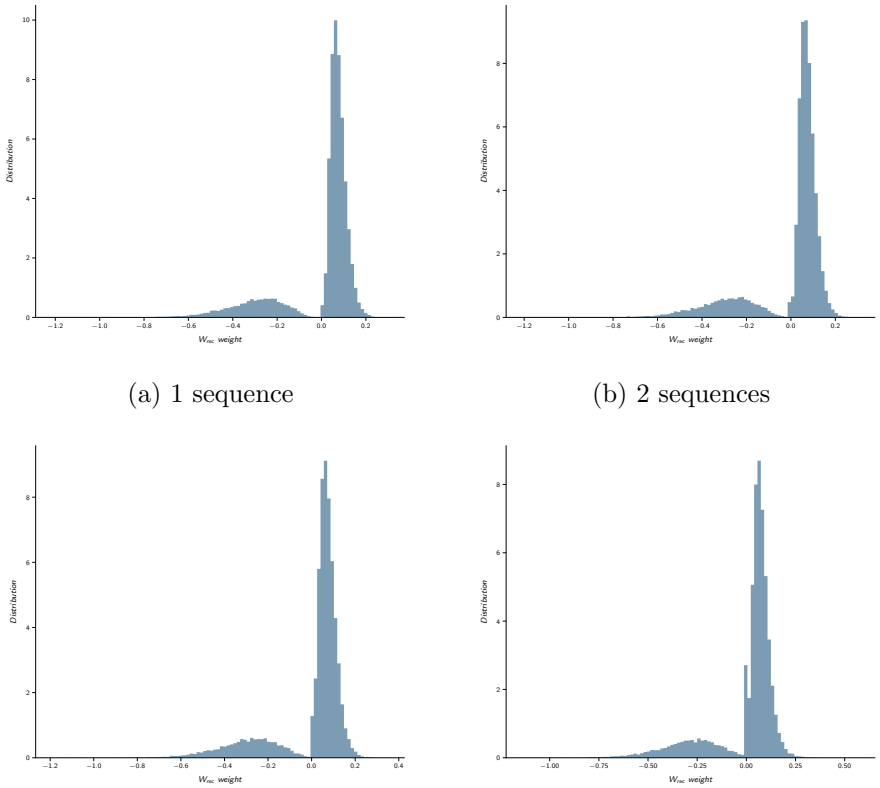
$$\tau\dot{\mathbf{x}} = -\mathbf{x} + W_{rec}\mathbf{r} + W_{in}\mathbf{u} + \sqrt{2\tau\sigma_{rec}^2}\xi \tag{1}$$

where $\mathbf{r} = [x]_+$, $\mathbf{z} = W_{out}\mathbf{r}$ and $\mathbf{u}(t)$ is the input given to the RNN.

- $N = N_{rec} = N_{Excitatory} + N_{Inhibitory}$: and Dale's principle is given by: $\frac{N_{Excitatory}}{N_{Inhibitory}} = 4$
- W_{in} has only positive elements
- W_{rec} has EE(+ve), EI(+ve), IE(-ve) and II(-ve) connections as follows: $\begin{bmatrix} EE & IE \\ EI & II \end{bmatrix}$
- W_{out} is either all positive or a partial identity matrix
- Output readouts are from few of the $N_{Excitatory}$ nodes

Appendix B. Shift of the sequential pulses of the neurons:

In this appendix we mathematically determine how the modifications in the W_{rec} weights could cause shifts in the sequential pulses.



(a) 1 sequence

(b) 2 sequences

(c) 4 sequences

(d) 8 sequences

Figure 6: W_{rec} weight distribution

Consider the equation mentioned in Appendix A, we could solve the 1st order differential equation for the noiseless case. For a simplified case, the $\mathbf{x}(t)$, which is the node activation or the outputs could be given by:

$$\mathbf{x}(t) = e^{At}\mathbf{x}_0 + e^{At}\int e^{-At}\mathbf{b}dt \tag{2}$$

where, $A(t) = \frac{1}{\tau}\big(W_{rec} - I\big)$, $\mathbf{b}(t) = \frac{1}{\tau}W_{in}\mathbf{u}$, τ is the time constant used in the simulation, and $\mathbf{x}_0$ is the initial value of $\mathbf{x}$. Changes to the I/E (or IE) column weights of the W_{rec} matrix changes the way e^{At} decays, and thereby causing the shift in the simulated and observed sequential pulses across the output nodes.

Appendix C. Weight distribution for different multi-sequence networks

See Figure 6

Appendix D. Effect of W_{out}

In (Song et al., 2016), it is mentioned that one can fixes W_{out} to identity matrix to train on the neurons directly, although they didn't provide detailed research on it. We tried a similar approach. While keeping the same amount of output dimensions, we fixed W_{out} to a rectangular matrix whose diagonal entries are all 1 and 0 elsewhere. This matrix will map the first few neurons to the output, effectively training directly on the neurons.

Using this configuration, the network is still able to learn the task to some degree. However, it is observed that some expected peaks are missing from the output. Their corresponding neuron simply remains zero during the whole sequence, even after excessive training.

As the output unit originally connects to all neurons via the original W_{out}, it should apparently equivalent to the same amount of neurons, which connect to all other neurons as well. It is theorized that the difference that caused the vanishing behavior by fixed W_{out} is that neurons also feed back to the network. To maintain the dynamics of the network, neurons in the network would have bounded activation rate. If a connection weight is randomly initialized to a large value, the neuron on the input side of the connection would try to maintain a low value to avoid the explosion on the other side, and eventually learn to maintain zero activation.

The original motivation to fix the W_{out} is to resemble organic neuron network better and to observe the behavior of such network. However, we would like to argue that the restriction on W_{out} is unnecessary and unhelpful. As discussed above, with the restriction on W_{out} the network is unable to fully learn the task. The existence of flexible W_{out} also has its organic analogy: there can be some neuron that acts purely for output and does not feed back to the network.

Appendix E. Inputs,Outputs and Network Structure Information for Catastrophic Forgetting Experiments

- We use 1 input node(N_{in}), 10 output nodes(N_{out}) and 100 (E/I : 80/20) recurrent nodes(N)
- For the first task, the sequence of inputs is the product of a decaying exponential function and the sinusoidal function and the outputs are 10 separate Gaussians with different means. The plot for the inputs and outputs for the first task is in 5
- For the second task the sequence of inputs is the product of an increasing exponential function(unlike a decreasing one as in the first task) with the sinusoidal function, but with different frequency and amplitude, while the outputs are the same as for the previous task but with their time order reversed. The plots for the inputs and outputs for this task are in 7.
- We use a dropout of 0.5 on the recurrent nodes.
- We used a $\lambda = 500$

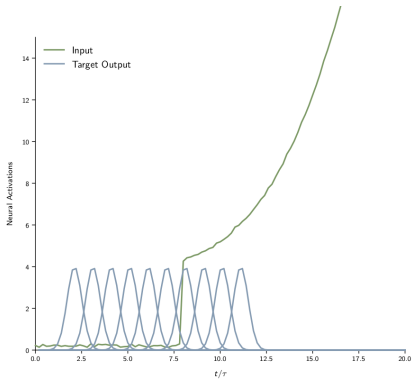


Figure 7: Task 2: Inputs with Target Outputs