Plasticity and Firing Rate Dynamics in Leaky Integrate-and-Fire Models of Cortical Circuits

A dissertation presented

by

Joseph Olson

to

The Department of Physics

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Physics

Harvard University

Cambridge, Massachusetts

July 2019

Dissertation Advisor: Gabriel Kreiman Joseph Olson

Plasticity and Firing Rate Dynamics in Leaky Integrate-and-Fire Models of Cortical Circuits

Abstract

A very large part of computational neuroscience is to understand neuronal firing rates and how each neuron's activity effects its neighbors. Detangling the complex patterns observed in neuronal activity is a challenging subject. Understanding how these patterns emerge from a single cell is even more daunting. In this body of work, we aim to shed some light on firing rate dynamics as well as on how plasticity may play a role in developing cortical circuits. In chapter 2, we describe how just a couple plasticity rules can together generate both feedforward and feedback connections in a model resembling cortical neurons. We use simulations of leaky integrate-and-fire neurons to explore the different outcomes. We find that a specific pattern of plasticity rules gives rise to synaptic connectivity patterns observed in cortex. In chapter 3, we develop a method for understanding higher order terms to the firing rate equation derived from leaky integrate-and-fire neurons. We equate the dynamics to those of electrical circuits and find the firing rate is equivalent to current. The analysis shows that oscillations in firing rate will necessarily exist when an inhibitory network is connected to a network of excitatory LIF neurons. Furthermore, the framework may provide new tools for analyzing weight changes. In chapter 4, we investigate learning in artificial neural networks. Specifically, we aimed to understand catastrophic forgetting as well as how to build spiking artificial neural networks. Results in chapter 4 are inconclusive.

Table of Contents

Acknowledgements

"It takes a village …" – An African proverb

I could not have made it to where I am today without the help of countless people. My success is their success. I would like to thank Gabriel Kreiman for his valuable mentorship, both personally and academically. I have learned more than I thought I could from him. I would also like to thank, Aravi Samuel, Haim Sompolinsky, Mara Prentiss, Lakshminarayanan Mahadevan, and Mark Andermann for their support along the way.

The physics department has been a true source of support and encouragement the past few years. I cannot express enough how much the faculty has supported my growth. In particular Lisa Cacciabaudo, Carol Davis, and Jacob Barandes. I would also like to thank my fellow comrades Ellen Klein, Andrew Chael, James Mitchell, Henry Wilkin, Thomas Plumb-Reyes, Victor Buza, Maryrose Barrios, Michael Rowan, Hanrong Chen, Liujun Zao, and Katie Huang.

I would like to thank everyone from the Kreiman Lab. They have made this journey enjoyable and possible. Countless times I have relied on their support. In particular, Hanlin Tang, William Lotter, Leyla Isik, Jerry Wang, Emma Krause, Jiye Kim, Pranav Misra, Frederico

Azevedo, Martin Schrimpf, Mengmi Zhang, Kristofor Payer, Will Xiao, Yuchen Xiao, Jie Zheng, Sarit Szpiro, and Megan Bendsen-Jensen.

I would like to thank my roommates and friends who have been there when times were rough. Enough said. Thank you Emily Mackevicius (and family), Isa Garbutt, Toby Kaiser, Andrew Day, Murat Uzun, Rasit Mete Esrefoglu, Haydar Emin Evren, Marinna Madrid, Franco Doyle, Ellen Klein, Jerry Wang, Rachael Rosales, Xenia Leviyah, Connor Vance, John Mayer, Zach Wunderly, Iman Tamimi, Kim Everett, Naomi Levine, Lee and Gecia Bravo Hermsdorff, Sadhvi Batra. The work presented in chapter 4 was done with Murat Uzun and that section is dedicated to him. Also, the dear people of Conant 3rd floor will forever be dear to me – Mélissa Vrolixs, Allison White, Kevin Gurley, Xiaoxuan Li. The Dudley House also played a big part of my time at Harvard. I would like to thank, in particular, Susan Zawalich, Jeffrey Shenette, and the Dudley Fellows. Also special thanks to the World Music Ensemble led by Daniel Ang for sharing their talents with me.

I would also like to thank members of the Center for Brains, Minds, and Machines at MIT and the Center for Brain Science at Harvard for greatly contributing to my understanding of computational neuroscience. Additionally, I would like to thank Allen Institute for Brain Science, in particular Christof Koch, Michael Buice, and Saskia de Vries. As well as Marie Tolkiehn and Jan-Matthis Lueckmann for their friendship.

Last and most importantly, I want to thank my family. My parents Joseph and Katherine have been a never-ending source of love and support. Without them, I would never have come this far. My siblings are all amazing and I cherish them dearly. Anna, Kristin, Nick, Liz – thank you.

Introduction

Learning is itself one of life's biggest lesson. We need to learn how to learn, what to learn, why to learn. How do we learn anything at all? What does it mean to learn? If I learn something today and forget it tomorrow, did I really learn it? These are very fundamental questions. Currently, the blooming field of computational neuroscience is being faced with very basic questions such as these as well as many other interesting questions regarding intelligence.

For example, the question regarding forgetting what we have previously learned is not only a problem in humans, it is a problem for machine learning. Artificial networks often lack the ability to remember an old behavior if they are asked to learn a new behavior. This is known as catastrophic forgetting and solutions to this problem are not currently well understood. We explore catastrophic forgetting in chapter 4 although we do not offer major contributions or insights into understanding this problem.

Generally speaking, if you don't want to lose something you need to store it somewhere. That is why some of us store all of our photos on Facebook. Neurons have the ability to store something too – electric charge. They act as capacitors which can charge and discharge in the form of action potentials. They also need to learn something. They need to learn how much charge to store and when to store it. We explore the concept of charge storage in chapter 3.

This analogy puts neurons into a unique position. They need to learn what kind of capacitor to be in order to store charge in a way which is optimal for them. They also act as the storage

compartment itself, a place for the larger network to store charge. The network needs to learn, as a whole, where to store charge and when. The network needs to learn it collectively and each neuron needs to learn it individually. This, abstractly, sounds a bit like how society works. When presidential election year comes around, everyone needs to do their individual research to determine information about the candidates and also what information is good information. We then store that information inside of ourselves. As a group, we store these opinions online and in each other. We learn as a group which information to store by debating our opinions against each other.

Bees also demonstrate a similar process. They will vote on a new location for their hive by democracy. Each bee flies to possible locations and they learn their favorite one. They then dance accordingly as a means of voting for their choice. In the groupd dance, individual bees may change their daces or pressure other bees to switch dances. In this way, they learn collectively what dance to do and each individual bee needs to learn what dance it wants to do. The dynamics of this process has been likened to the dynamics of a recurrent neural network.

If nature produces such similar patterns at varies scales, there may be a small set of fundamental principles which systemically produce such phenomenon. In chapter 2, we investigate a possible set of simple rules for weight changes which can lead to network adaptation to input. We do not explore voting behavior but instead explore a different pattern. We are interested in generating connectivity patterns in a neural network which resemble those observed in neocortex. We choose a pattern which is believed to be a repeated pattern across cortical areas and to some extent across species. We use two variations of spike timing dependent plasticity (STDP) and show that if we initialize the synapses with a specific pattern of STDP type across the synapses, we can robustly generate the target model circuit.

Simple Learning Rules Generate Complex Canonical Circuits

Cortical circuits are characterized by exquisitely complex connectivity patterns that emerge during development from undifferentiated networks. The development of these circuits is governed by a combination of precise molecular cues that dictate neuronal identity and location along with activity dependent mechanisms that help establish, refine, and maintain neuronal connectivity. Here we ask whether simple plasticity mechanisms can lead to assembling a cortical microcircuit with canonical inter-laminar connectivity, starting from a network with all-to-all connectivity. The target canonical microcircuit is based on the pattern of connections between cortical layers typically found in multiple cortical areas in rodents, cats and monkeys. We use a computational model as a proof-of-principle to demonstrate that classical and reverse spike-timing dependent plasticity rules lead to a formation of networks that resemble canonical microcircuits. The model converges to biologically reasonable solutions provided that there is a balance between potentiation and depression and enhanced inputs to layer 4, only for a small combination of plasticity rules. The model makes specific testable predictions about the learning computations operant across cortical layers and their dynamic deployment during development.

Neocortical circuits constitute the fundamental building blocks for cognitive computations and are characterized by a bewildering complexity in connectivity patterns. How such intricate and precise connectivity arises through development and learning constitutes a fundamental challenge

for neuroscience. In part, the answer relies on a web of molecular cues that guide neuronal precursors to specific brain areas (e.g., specifying which neurons will end up in primary visual cortex versus olfactory cortex), and to specific layers within those areas (e.g., specifying which neurons will reside in layer 4 versus layers 2/3) (Bolz et al. 1996; Castellani and Bolz 1997; Callaway 1998b; Larsen and Callaway 2006; Lui et al. 2011; Silbereis et al. 2016). In addition to molecular cues, activity- dependent mechanisms play a central role in shaping and/or refining neural circuits, both during development and subsequent learning (Feldman and Brecht 2005; Fox and Wong 2005; Karmarkar and Dan 2006; Butts et al. 2007; Espinosa and Stryker 2012; Bennett and Bair 2015; Lim et al. 2015).

Here we investigate how simple activity-dependent mechanisms can give rise to complex circuit structures by adequately modifying the strength of neuronal connections. An important activity-dependent mechanism governing the connection strength between neurons is spike-timing dependent plasticity (STDP) (Markram et al. 1997; Bi and Poo 1998). Different forms of STDP have been observed throughout biological circuits (for reviews, see Abbott and Nelson 2000; Caporale and Dan 2008; Froemke et al. 2010). We consider two specific forms of STDP that have been widely observed in cortex: classical STDP (cSTDP, Fig. 1a-b top) and reverse STDP (rSTDP, Fig. 1a-b bottom). In cSTDP, long-term potentiation (LTP) strengthens connections when a pre-synaptic action potential precedes a post-synaptic action potential while long-term depression (LTD) weakens connections when the post-synaptic action potential precedes the pre-synaptic action potential (Markram et al. 1997; Bi and Poo 1998; Debanne et al. 1998; Feldman 2000; Sjöström et al. 2001; Froemke et al. 2005). cSTDP can be thought of as a mechanism that promotes causally linked feedforward connections. In rSTDP, connection strengths change in the opposite direction: LTD weakens connections when a pre-synaptic action potential precedes a post-synaptic action potential while LTP strengthens connections when the post-synaptic action potential precedes the pre-synaptic action potential (Letzkus et al. 2006; Sjöström and Häusser 2006; Burbank and Kreiman

2012). rSTDP can be thought of as a mechanism that promotes feedback connections. Fig. 1b schematically illustrates connection becoming stronger or weaker depending on the relative timing of the pre/post-synaptic spikes and the STDP rule. The assignment of learning rules across connections can have a major impact on the resulting structure of a neural circuit. For instance, computational simulations show that cSTDP leads to the elimination of loops in fully connected networks (Kozloski and Cecchi 2010) and rSTDP enhances feedback connections in a multiple-layer network (Burbank and Kreiman 2012). We extend these ideas by investigating whether it is possible to generate complex connectivity patterns such as those observed in neocortical circuits purely from activity-dependent mechanisms based on STDP and starting from all-to-all connectivity.

We focus on the approximately canonical inter-laminar connectivity observed in neocortical circuits. Such connectivity has been observed in macaque V1 (Callaway 1998a, Fig. 2) and other visual cortical areas (Felleman and Van Essen 1991), in cat V1 (Douglas and Martin 2004, Fig. 1) and in mice (Larsen and Callaway 2006). The target canonical circuitry of inter-laminar connections is simplified to the structure in Fig. 1c. This simplified circuitry ignores significant aspects of neocortical circuits including sub-laminar structure such as horizontal connections within a layer (Binzegger et al. 2004), sub-divisions of layer 4, distinctions between layers 5 and 6, different neuronal types within each layer, and real-valued connection strengths that are not 0 or 1 (see Discussion). To a reasonable first-order simplification, the inter-laminar connectivity pattern is conserved across multiple cortical regions and even across species. We start with a spiking network that contains 3 layers, labeled layer 4, layer 2/3 and layer 5/6. These layers are initially connected all-to-all and connections undergo either cSTDP or rSTDP (Fig. 1d). We investigate which combinations of STDP-based learning rules give rise to connections that match the target circuitry. We demonstrate that it is possible to rapidly develop a good approximation to the target canonical circuit in Fig. 1c from the initial random circuit in Fig. 1d using a small cluster of configurations of simple activity dependent STDP learning rules.

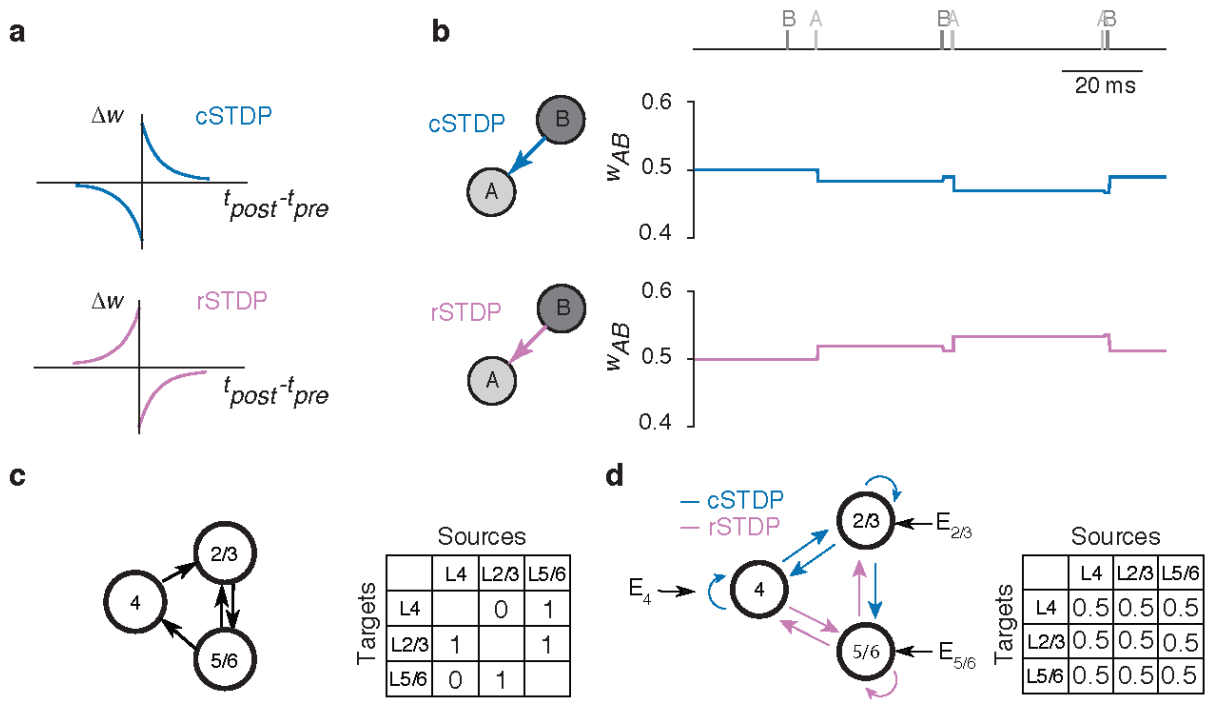


Figure 1

Model description. a, Schematic illustration of how the change in synaptic weights depends on the relative timing of pre- and post-synaptic spikes for classical STDP (top) and reverse STDP (bottom). **b,** Sample spike trains from two neurons, A and B (top), and how the synaptic weight from B to A (w_{AB}) evolves with the occurrence of each spike under cSTDP (middle) or rSTDP (bottom). **c,** Schematic of target connectivity in the canonical circuit, simplifying the inter-laminar connectivity patterns found in cortical circuits in rodents, cats and monkeys. There are 3 layers (L4, L2/3 and L5/6); the direction of the arrows denotes the desired connectivity. The connections are idealized in the connectivity weight matrix shown on the right where row i, column j is 1 iff there is a connection from column j onto row i (see Methods). **d,** Example initial conditions where all weights start at 0.5. Each layer receives external excitatory inputs ($E_4, E_{2/3}, E_{5/6}$) in addition to recurrent inputs within the same layer and inputs from other layers. A specific plasticity rule was assigned to each of the 9 possible connections between or within layers (see Methods). The combination of learning rules depicted here is only one of the 512 possible combinations examined throughout this study.

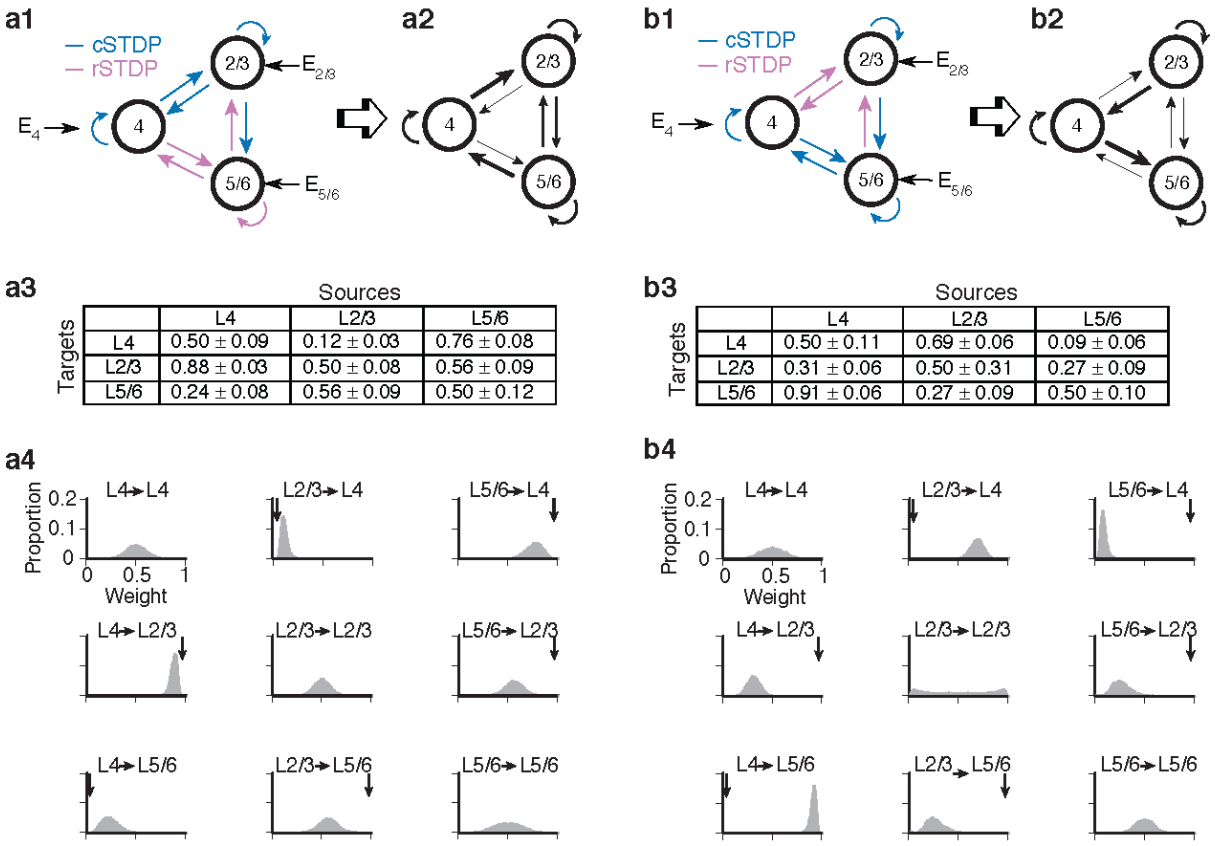


Figure 2

Two example simulations, one successful (a), one not (b). a1/b1, Initial configuration. **a2/b2,** Network at the end of the simulation. Line widths are proportional to the corresponding weights. **a3/b3,** Weight matrices at the end of simulation, repeated 5 times (mean ± SD across neurons, $n = 33 \times 33 \times 5 = 5{,}445$, averaged over the last 5 seconds of simulations, see Methods). **a4/b4,** Histograms showing the distribution of weights for each pair of layers.

Results

We asked whether it is possible to develop complex architectures with connectivity similar to that of neocortical circuits starting from fully connected neurons distributed into three layers and following simple STDP rules: classical and reverse STDP. We consider as a target the idealized version of a canonical microcircuit schematically illustrated in Fig. 1c. This circuit is an abstraction of the inter-laminar connectivity in cortical areas reported in macaque, cats, and mice (Callaway 1998a, Douglas and Martin 2004, Larsen and Callaway 2006). In the simplified version of biological connectivity considered here, connections are either maximally strong (strength of 1) or absent (strength of 0) and only the main connections are represented (see Discussion). In the initial conditions for the developmental simulations, all neurons in one layer are connected to all neurons in another layer and all weights are initialized to 0.5. Each layer contains 33 integrate-and-fire neurons (see Supplementary Table S1 for simulation parameters). In each simulation and for each pair of layers and connectivity direction (e.g. neurons from layer 4 projecting to layer 2/3), we consider a specific learning rule (cSTDP or rSTDP) governing how the weights evolve for all the corresponding synapses. Because there are 9 different types of connections (3 types of within-layer connections plus 6 types of between-layer connections), there is a total of $2^9 = 512$ different configurations (we refer to a configuration as a particular combination of cSTDP or rSTDP for each connection type). Fig. 1d (expanded in Supplementary Fig. S1) shows one of those possible configurations. Each neuron receives excitatory input from independent homogenous Poisson neurons (E_4, $E_{2/3}$, and $E_{5/6}$). Layer 4 is assumed to receive more excitatory input than layers 2/3 and 5/6 (i.e. $E_4 > E_{2/3}, E_{5/6}$) because it is typically the layer receiving input from the thalamus or from earlier cortical areas (Felleman and Van Essen 1991). Additionally, each neuron receives inhibitory input from independent Poisson neurons whose firing rates change as a function of the fraction of

active integrate-and-fire neurons. The STDP curves are modeled as two exponential functions with amplitudes A_+ and A_-, and time constants τ_+ and τ_- (see Methods for details and Supplementary Table S1 for parameter values). Each configuration was simulated n=5 times for 60 seconds. After stable equilibrium was reached, usually well before 60 seconds, weight fluctuations remained small compared to the weight values (Supplementary Fig. S2). At the end of each simulation, we averaged the weights into a 3×3 weight matrix W.

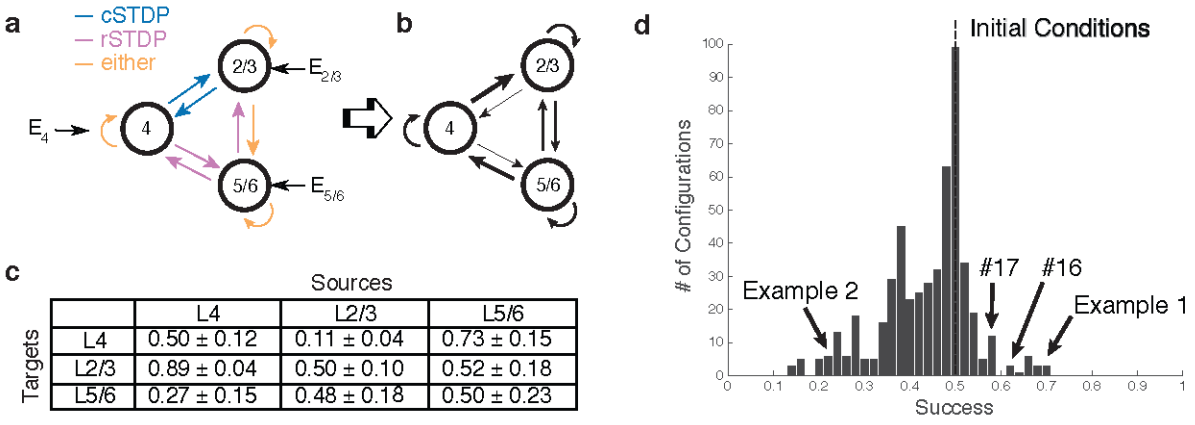
Some configurations develop into networks that resemble the target microcircuit

Fig. 2 shows one STDP configuration that leads to a network resembling the target microcircuit and one that does not. The final weights for the circuit in Fig. 2a approximate the target matrix for the idealized network in Fig. 1c. We compared the final weight matrix W with the target matrix T by defining the degree of success of each configuration as $s = 1 - 6^{-\frac{1}{2}}\|W - T\|_F$ where $\|\cdot\|_F$ is the Frobenius matrix norm. The diagonal elements, corresponding to the within-layer weights, do not contribute to the success metric (see Discussion). Since weights are bounded between 0 and 1, s is bounded between 0 and 1 with $s = 1$ if and only if $W = T$. The configuration in Fig. 2a has a success of $s = 0.70 \pm 0.01$. In contrast, the configuration in Fig. 2b has a success of $s = 0.22 \pm 0.01$. The initial condition has a success $s = 0.5$, hence the configuration in Fig. 2a develops into a circuit that becomes more similar to the target whereas the configuration in Fig. 2a develops into a circuit that is even less similar to the target than the initial conditions.

The best configurations share a specific combination of learning rules

We computed the degree of success for each of the 512 possible learning rule configurations (Supplementary Fig. S3). The degree of success ranges from $s = 0.14 \pm 0.01$ (worst) to $s = 0.70 \pm 0.01$ (best) (Fig. 3d). The weights and success of the best 16, middle 16, and worst 16 configurations are shown in Supplementary Table S2. For most configurations, the degree of success is lower than

that of the initial conditions (Fig. 3d), i.e., most combinations of learning rules do not lead to the formation of circuits resembling the target one. Interestingly, in order for the model to arrive at an architecture that resembles the target canonical circuit, the plasticity rules between layers need to be within a certain configuration of cSTDP/rSTDP rules (Fig. 3a). Other combinations of cSTDP and rSTDP led to different architectures (e.g. Fig. 2b, 3d, 4a, Supplementary Fig. S3). Specifically, the model predicts that connections $L4 \rightarrow L2/3$ and $L2/3 \rightarrow L4$ both follow cSTDP; connections $L4 \rightarrow L5/6$ and $L5/6 \rightarrow L4$ both follow rSTDP; and connections $L5/6 \rightarrow L2/3$ follow rSTDP. The connection $L2/3 \rightarrow L5/6$ formed equally well with either cSTDP and rSTDP (Fig. 4b). Altogether there are 4 unspecified connections among the best $2^4 = 16$ configurations. A configuration is in the best 16 if and only if it shares the combination of rules specified above and illustrated in Fig. 3a. Furthermore, the best 16 configurations are separated from the rest by a gap in the success curve (Fig. 3d, Supplementary Fig. S3a). A similar gap separates the worst 8 which also display a common configuration of STDP learning rules (Supplementary Fig. S3a, Supplementary Table S2). The combinations of learning rules shown in Fig. 3a for the best 16 configurations, lead to the average weights shown in Fig. 3c and the circuit depicted in Fig. 3b, which resembles the target canonical circuit in Fig. 1c.



c

Targets \ Sources	L4	L2/3	L5/6
L4	0.50 ± 0.12	0.11 ± 0.04	0.73 ± 0.15
L2/3	0.89 ± 0.04	0.50 ± 0.10	0.52 ± 0.18
L5/6	0.27 ± 0.15	0.48 ± 0.18	0.50 ± 0.23

Figure 3

Configuration for the best 16 models. a, Learning rules for each connection for the best 16 models. **b,** Final circuit at the end of the simulations, averaged across the best 16 models. **c,** Final weights for the best 16 models ($n = 5{,}445 \times 16 = 87{,}120$). **d,** Average success of each of 512 configurations ($n = 5$). Example 1 is the configuration shown in Fig. 2a and Example 2 is the configuration shown in Fig. 2b. Also shown is the success, 0.5, of the initial conditions. Note the gap between configuration number 16 and configuration number 17, as well as the gap before the worst 8 simulations.

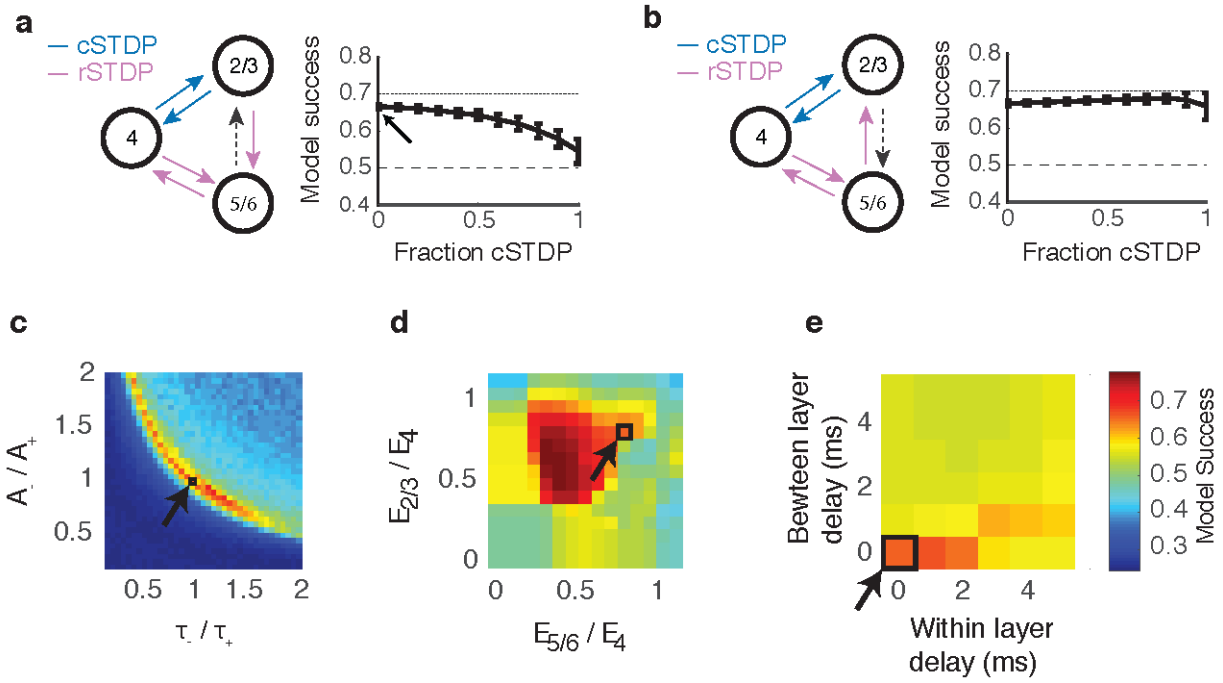


Figure 4

Robustness of the best configurations. a-b, We vary the fraction of cSTDP connections from 0 to 1 (all rSTDP to all cSTDP) from layer 5/6 to layer 2/3 (**a**) or from layer 2/3 to layer 5/6 (**a**). The connection shown as a dashed arrow is the one that is subject to different fractions of cSTDP. The model success curve is averaged across 5 simulations and across within-layer connections (8 possible configurations) for a total of $n = 40$. Error bars represent standard deviations. The dashed line shows the model success for the initial conditions. The dotted line shows the model success for the overall best configuration, which is depicted in Fig. 2a. The arrow in (**a**) points to the default condition corresponding to best 16 configurations. In (**b**), where both extremes correspond to best 16 configurations, cSTDP and rSTDP lead to equivalent model success. **c,** Model success (color scale shown on right) for different combinations of STDP amplitude and time constant ratios. The arrow points to the default condition. Success is averaged across 5 simulations and across the best 16 configurations for a total $n = 80$. **d,** Model success for different ratios of excitatory inputs ($n = 80$). **e,** Model success for different combinations of within and between layer delays ($n = 80$).

Models with only one type of learning rule between layers outperform models with mixed learning rules

The previous results assume that all the connections from one layer to another follow the same learning rule. In order to evaluate the impact of this assumption on the results, we systematically consider each pair of layers and vary the fraction of connections following cSTDP from none to all (Fig. 4a-b, Supplementary Fig. S4). For example, in Fig. 4a, we vary the fraction of cSTDP connections $L5/6 \rightarrow L2/3$, such that 0% cSTDP (100% rSTDP) corresponds to one of the 16 best configurations (arrow in Fig. 4a, right). The success value decreases monotonically as more cSTDP connections are added, departing from the best configuration. At 100% cSTDP, success drops to almost the initial condition value. In contrast, success is essentially unperturbed while varying the fraction of cSTDP connections $L2/3 \rightarrow L5/6$ (Fig. 4b), further confirming that either learning rule is adequate for the connections between these two layers.

We vary the fraction of cSTDP connections between each pair of layers in the best 16 configurations. In each case, success peaks when models have either 100% cSTDP or 100% rSTDP, matching one of the configurations in the best 16 group (Supplementary Fig. S4). The right column in Supplementary Fig. S4 shows large error bars because the configurations considered in these averages, having cSTDP connections $L2/3 \rightarrow L5/6$, come from both the higher and lower ends of the best 16 ranking (Supplementary Table S2).

The formation of the target microcircuit depends on the balance between potentiation and depression

Next, we examine the robustness of the conclusions to several of the critical parameters and assumptions in the simulations. In Fig. 4c, we vary the STDP exponential parameters A− and τ− away from their default values $A_- = A_+$ and $\tau_- = \tau_+$. There is a sharp decrease in success away from the curve defined by $A_+\tau_+ = A_-\tau_-$. The quantities $A_+\tau_+$ and $A_-\tau_-$ correspond to the area under the

positive and negative parts of the cSTDP curve in the best part of Fig. 1a, and conversely, the area under the negative and positive parts of the rSTDP curve. The decrease in success is due to weights strengthening and weakening as a result of a bias towards potentiation or depression. Setting $A_+\tau_+ > A_-\tau_-$ leads to enhanced strengthening/weakening of connections following the cSTDP/rSTDP rules respectively. Conversely, setting $A_+\tau_+ < A_-\tau_-$ leads to enhanced weakening/strengthening of connections following the cSTDP/rSTDP rules respectively. As an example of a failure mode, increasing $A_-\tau_-$ results in strong connections that follow rSTDP from layer 4 to layer $5/6$ whereas the target circuit has none of those connections.

The formation of the target microcircuit depends on increased inputs to layer 4

In the models described so far, the external inputs to layer 4 (E_4) are stronger than the external inputs to the other two layers ($E_{2/3} = E_{5/6} = 275, E_4 = 350$). We examined the impact of the relative external input strengths on the degree of success of a model by varying $E_{2/3}$ and $E_{5/6}$ (Fig. 4d). Consistent with the assumption that layer 4 is the main input layer, there is a sharp decrease in success for models with $E_{2/3} > E_4$ or $E_{5/6} > E_4$. In contrast, as the amount of input to layer 4 increases in comparison to layers $2/3$ and $5/6$, the degree of success also increases. Supplementary Fig. S3 depicts the degree of success for all 512 configurations under two such conditions with different levels of E_4 inputs. Some configurations in these models with smaller $E_{2/3}/E_4$ and $E_{5/6}/E_4$ ratios show large degrees of success close to 1 (e.g. best configurations in Supplementary Fig. S3. Additionally, these models with enhanced E_4 inputs also show increased separation for the best models from the rest (Supplementary Fig. S3). However, as E_4 increases, there is also a decrease in the average equilibrium firing rates in layers $2/3$ and layers $5/6$ (Supplementary Fig. S3).

Conversely, when $E_{2/3}$ and $E_{5/6}$ are enhanced, there is a decrease in success. This is because as $E_{2/3}$ and $E_{5/6}$ get close (or even surpass) E_4, there is no longer a driving force into layer 4. We investigated further the case where after circuit development, the enhanced driving force into layer

4 is taken away and all inputs are equal (Supplementary Fig. S5). In this case, the structure of the circuit vanishes and the circuit adapts to reflect the symmetry in the inputs with the weights converging towards 0.5.

Note that the strength of the external inputs into a layer depends the number of connections as well the weights which undergo cSTDP. However, the variability of the external excitatory neuron's spike statistics leads to the same weight values from the external populations into each layer. The average final weights into layer 4, 5/6, and 2/3 from their respective external inputs are 0.53±13, 0.52±15, and 0.53±14. Thus, the number of external input connections determines the strength of the input.

Long delays between layers disrupt the development of the target microcircuit

In the simulations reported so far, synaptic transmission was considered to be instantaneous, i.e., a spike in one neuron exerted an immediate effect on its post-synaptic target. We evaluate the consequences of introducing delays between layers (Fig. 4e). The degree of success remains high for short synaptic delays of up to 2 ms between neurons within the same layer. Outside of this regime, introduction of delays disrupted the success of the simulations.

Early development of L5/6 to L4 connections disrupts the development of the target microcircuit

In the simulations presented thus far, the architecture and STDP rules were established from the onset and all the connections started to change at the same time. The ensuing dynamics for the different inter-laminar connections were similar, and they achieved their final values approximately at the same time (Fig. S2b). We next considered scenarios in which one of the six inter-laminar connections arose before the others to evaluate whether the development of the target circuit was influenced by the order in which connections solidified. We ran the simulations while fixing each of the 6 inter-laminar connections separately to the final weight value obtained in the default

simulations (Fig. 3c) while all the other connections changed according to the corresponding STDP rules. When the weights from L5/6 to L4 were fixed to 0.73 from the beginning, the network was unable to converge the target circuit (Supplementary Fig. S6). However, in all other cases when one of the connections was pre-determined, the network was able to converge to the target circuit (Supplementary Fig. S6).

Discussion

We asked whether simple plasticity rules can give rise to the rich connectivity patterns of canonical circuits in neocortex. Starting from a fully connected 3-layered network, we demonstrate that a simple combination of spike-timing dependent plasticity (STDP) rules can rapidly lead to a complex architecture which captures some of the essential connectivity patterns of cortical circuits. The proposed model follows the essential ingredients of previous work with spiking networks undergoing plasticity including integrate-and-fire neurons, STDP, 'tabula rasa' initial conditions, and biologically plausible parameters (Abbott and Nelson 2000; Kozloski and Cecchi 2010; Burbank and Kreiman 2012). The model leads to a stable (Supplementary Fig. S2) and robust solution (Fig. 3) that resembles a simplified version of the canonical circuit (Fig. 1c), provided that the connections respect a specific combination of cSTDP and rSTDP rules (Fig. 3, Supplementary Fig. S3), provided that there is a balance between potentiation and depression ($A_+\tau_+ \approx A_-\tau_-$, Fig. 4c), and provided that there are stronger external inputs to layer 4 (Fig. 4d).

We compared the resemblance of the final states of our model to the target canonical circuit with a success metric. The success of our simulations does not reach 1.0, but this is to be expected for several reasons. First, the target canonical circuit is idealized to have connection strengths of 0 or 1 whereas real connections follow a distribution of synaptic strengths. Second, noise is continuously introduced into the circuit from the external Poisson spiking neurons so that the weights cannot reach a stable value of 0.0 or 1.0. Furthermore, the soft bounds imposed on the weights (see Methods) push the weight values away from 0.0 and 1.0, making it highly unlikely that weights would settle on those values. Third, although it is possible to fine tune parameters such that the models have a higher degree of success, e.g. Fig. 4e, our aim is not to reach success $= 1$, but rather to show as a proof-of-principle, that activity dependent mechanisms can build circuits qualitatively similar to those found

in biological systems.

The success metric did not include the within-layer connections, because the relative strength of within layer connections compared to the between layer connections remains unclear. The within-layer connections do not contribute to the success metric because the average within-layer weight is consistently 0.5 during the entire simulation. This is because the within-layer weights all undergo the same type of STDP, they are initialized at 0.5, and potentiation of weight w_{ij} is exactly the opposite of depression of w_{ji}. Although within-layer connections did not directly contribute to the degree of success of a configuration, they indirectly affected the weights of the between-layer connections. In the most concrete example, when STDP rules are configured as in the best 16 configurations with the additional constraints that both $L2/3 \rightarrow L5/6$ and $L5/6 \rightarrow L5/6$ follow cSTDP, multimodal weight distributions were observed. The weight distributions, averaged across these 4 configurations, are compared to those averaged across the other (unimodal) 12 configurations in Supplementary Fig. S7.

In order for the model to arrive at an architecture that resembles the target canonical circuit, the plasticity rules between layers need to be within a certain configuration of cSTDP/rSTDP rules (Fig. 3a). Interestingly, this configuration is consistent with experimental studies. Plasticity governed by cSTDP at proximal synapses and rSTDP at distal synapses of $L2/3 \rightarrow L5/6$ pyramidal neurons has been observed in rat primary somatosensory cortex (S1) (Letzkus et al. 2006; Sjöström and Häusser 2006). Furthermore, our results are consistent with a study which reports cSTDP from $L4 \rightarrow L2/3$ in rat S1 (Feldman 2000). Although our simulations do not make any strong predictions about STDP rules within layers, experimental studies have observed that connections within L2/3 and within L5/6 follow cSTDP (Markram et al. 1997; Egger et al. 1999), and connections within L4 follow rSTDP (Egger et al. 1999). Our model predicts that rSTDP may also be observed between connections $L5/6 \rightarrow L4$ and $L5/6 \rightarrow L2/3$.

The requirement for an approximate balance between potentiation and depression has also

been proposed in previous studies of plasticity in spiking networks (Burbank and Kreiman 2012; Babadi and Abbott 2013). Consistent with these studies we see that unbalanced potentiation and depression can lead to unchecked strengthening or weakening of connections. While precise measurements of A_+, A_-, τ_+, τ_- are difficult to come by, we estimated these quantities from different empirical STDP studies. Supplementary Fig. S8 shows that these estimates are approximately consistent with a balance between total potentiation and depression (the area under the curve above and below the y-axis in the STDP curves in Fig. 1a).

The second requirement is that the external inputs to layer 4 need to be stronger than those to other layers. This requirement is consistent with a large body of literature which indicates that cortical areas mostly receive input via layer 4. For primary sensory areas, this input comes from thalamus, and for higher sensory cortical areas this input comes from layer 2/3 of other cortical areas (Felleman and Van Essen 1991; Callaway 1998a; Miller 2003). It has recently been reported that layer 5/6 also receives direct input from the thalamus (Constantinople and Bruno 2013). As each layer in our model receives external input, this does not contradict our assumptions as long as the input to layer 4 is stronger. Our model is *not* specific to the thalamocortical system, though. As long as the external input to layer 4 is stronger, this model may also capture the formation of between layer connections in other cortical areas.

More is known about the development of primary cortical areas deriving inputs from the thalamus (e.g., primary visual cortex) than about other cortical areas (e.g. visual areas V2, V4, etc.). Early stages of primary cortical circuit development occur *before* thalamic afferents reach cortical layer 4. This observation has led many investigators to conclude that the development of between layer connectivity is primarily driven by molecular cues with the role of activity-dependent mechanisms confined to circuit refinement (Lund and Mustari 1977; Rakic 1977; Callaway 1998b; Pasko Rakic 2009). However, it is conceivable that the type of rapid restructuring of between layer connectivity proposed by this model might rely on inputs from a transient structure called the

subplate, rather than on direct inputs from the thalamus. Positioned directly beneath developing cortical cells, the subplate is the target of early thalamic afferents where they wait for days (in rats) or weeks (in cats) before entering the cortical plate (Lund and Mustari 1977; Shatz and Luskin 1986). During this time, subplate neurons project to a developing layer 4 and are capable of firing action potentials (Allendoerfer and Shatz 1994) and are the first cortical neurons to respond to sensory stimuli (Wess et al. 2017). Taken together, it is possible that early spontaneous activity in the subplate, rather than thalamus, may drive developing cortical circuits by providing enhanced input to layer 4. Consistent with this notion, disruption of thalamocortical afferents results in largely intact laminar structure (Miyashita-Lin et al. 1999; Li et al. 2013), perhaps because in this preparation the thalamic projections to the subplate remained undisturbed.

The type of activity-dependent plasticity mechanism proposed here does not necessarily rely on actual sensory experience. For example, in the context of vision, the model does not require post-natal visual inputs and could well take place during the embryonic stage. The type of activity used in the current study contains no structure (beyond the enhanced inputs to layer 4). We speculate that richer and structured activity patterns, in combination with molecular cues, might lead to even more complex circuits. Indeed, the target canonical microcircuit considered here clearly constitutes a major oversimplification abstracting away much of the exquisite and enigmatic architecture of cortex, including the differentiation between six neocortical layers, the vast array of different types of excitatory and inhibitory neurons, the distance dependence in connectivity patterns, and the non-uniform distribution of synaptic inputs along dendrites, among many others. The current model clearly does *not* claim that every aspect of the cortical connectivity pattern can be purely generated by STDP. The model demonstrates that adequately combining very simple activity-dependent learning rules can rapidly lead to the emergence of complex circuits that capture essential principles of the cortical connectome.

Methods

Model description

All the models have the same overall structure, consisting of 99 integrate-and-fire neurons split evenly into 3 layers, 33 neurons per layer (Supplementary Fig. S1). We refer to those layers as 'layer 2/3' (L2/3), 'layer 4' (L4), and 'layer 5/6' (L5/6). The network is initially connected all-to-all (no self-connections) with weights set to 0.5, half the maximum value of $w_{max} = 1$. The weights are constrained to be non-negative and the bounds are imposed using a soft-max mechanism within the STDP update rule described in the section **Weight Changes**.

In addition to the input from the internal network described above, each neuron receives input from external excitatory Poisson neurons of firing rate 20 Hz. Each neuron in layer 4, layer 2/3, and layer 5/6 receive input from $E_4 = 350$, $E_{2/3} = 275$, $E_{5/6} = 275$ external excitatory Poisson neurons, respectively. Each layer has a separate pool of 2500 external excitatory neurons supplying input. Connections from the external population to each network neuron are drawn randomly. All neurons also receive external inhibition from 250 randomly selected neurons chosen from a pool of 1250 Poisson neurons. The inhibitory neurons had firing rates which track average network activity to provide excitatory/inhibitory balance for the network. The firing rate of these external inhibitory neurons, $r_{inh}(t)$, depends on the fraction of firing neurons in the network at time t, denoted by $\gamma(t)$. At each time step, dt $= 0.1$ ms, the rate is updated by $r_{inh}(t+1) = r_{inh}(t) + \gamma(t)(rmax - rmin)$ where $r_{inh}(0) = 20$ Hz, rmax $= 1000$ Hz, and rmin $= 5$ Hz. Also, $r_{inh}(t)$ decays exponentially every time step with a time constant of $\tau_I = 2$ ms, obeying $\tau_I \frac{dr_{inh}}{dt} = -r_{inh}$. See Supplementary Table S1 for a full list of parameters used in the simulations.

Individual neuron dynamics

The simulations are based on networks proposed by (Song et al. 2000; Kozloski and Cecchi

2010). All simulations were run in MATLAB 2013b (Mathworks, Natick, MA) and all the code is available at http://klab.tch.harvard.edu. Each neuron's membrane potential is governed by

$$\tau_m \frac{dV_i}{dt} = V_{rest} - V_i + \sum_{j\in\{exc\to i\}} g_{exc}^{ij}(t)(E_{exc} - V_i) + \sum_{j\in\{inh\to i\}} g_{inh}^{ij}(t)(E_{inh} - V_i)$$

where j and i refer to pre-synaptic and post-synaptic neurons respectively, $\{exc \to i\}$ denotes the set of excitatory inputs to neuron i, $\{inh \to i\}$ denotes the set of inhibitory inputs to neuron i, $g_{exc}^{ij}(t)$ is the excitatory synaptic conductivity from j onto i at time t, $g_{inh}^{ij}(t)$ is the inhibitory synaptic conductivity from j onto i at time t, $\tau_m = 20$ ms, $V_{rest} = 60$ mV, $E_{exc} = 0$ mV, and $E_{inh} = 70$ mV. The set of excitatory inputs includes those from the external Poisson neurons as well as those from the internal network. The set of inhibitory inputs include only those from the external Poisson neurons. After the voltage reaches a threshold, $V_{thresh} = -54$ mV, the neuron spikes and the voltage is reset to $V_{reset} = -60$ mV.

Weight changes

When a presynaptic spike occurs, the synaptic conductance is increased by an amount proportional to the synaptic weights: $g_{exc}^{ij}(t) = g_{exc}^{ij}(t-1) + \alpha w_{ij}(t)$ and $g_{inh}^{ij}(t) = g_{inh}^{ij}(t-1) + \alpha w_{inh}$ with $\alpha = 0.01$ and $w_{inh} = 1.5$. Otherwise, $g_{exc}^{ij}(t)$ and $g_{inh}^{ij}(t)$ decay exponentially with time constants $\tau_{exc} = \tau_{inh} = 5$ ms. All excitatory synaptic weights in the model are subject to plasticity (including those from the external excitatory inputs which are initialized at $w_{exc} = w_{max}$); all the inhibitory synaptic weights are fixed. Excitatory weights are updated by $w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t)$ where $\Delta w_{ij}(t)$ is determined by either classical STDP (cSTDP) or reverse STDP (rSTDP) rules. As depicted in Fig. 1a, the equations governing cSTDP and rSTDP are given by:

$$\text{cSTDP: } \Delta w_{ij}(t) = \begin{cases} A_+(1-w_{ij})^{\mu} e^{-\Delta t/\tau_+} & if\ \Delta t > 0 \\ -A_- w_{ij}{}^{\mu} e^{\Delta t/\tau_-} & if\ \Delta t < 0 \end{cases}$$

$$\text{rSTDP: } \Delta w_{ij}(t) = \begin{cases} -A_+ w_{ij}{}^{\mu} e^{-\Delta t/\tau_+} & if\ \Delta t > 0 \\ A_-(1-w_{ij})^{\mu} e^{\Delta t/\tau_-} & if\ \Delta t < 0 \end{cases}$$

for $\Delta t = t_i^{spike} - t_j^{spike} = t_{post} - t_{pre}$ which is positive if j fires before i, $A_+ = 0.035$, $A_- = 0.035$ (unless otherwise stated), $\tau_+ = 20$ ms, $\tau_- = 20$ ms (unless otherwise stated), and $\mu = 0.1$. The parameter μ modulates the update rule between additive ($\mu = 0$) and multiplicative STDP ($\mu = 1$) (Gütig et al. 2003). Additive STDP has the advantage of allowing the weights to explore more of the allowed range of values (Babadi and Abbott 2013). However, it has a couple drawbacks. First, it can generate bi-modal weight distributions of extreme values which are sensitive to changes in the firing rates of pre- and post-synaptic neurons (Rubin et al. 2001). Second, it requires the use of a hard boundary condition ($w_{ij} \to w_{max}$ if $w_{ij} > w_{max}$). The soft boundary conditions of multiplicative STDP does not suffer from these disadvantages but it limits the dynamics of the weights. Here we use $\mu = 0.1$ which blends the advantages of the two (Gilson and Fukai 2011).

We assume that the change in weight w_{ij} from pre-synaptic neuron j to post-synaptic neuron i sums linearly if j fires multiple times shortly before i fires. Thus, in the simulation, the cSTDP learning rule is implemented algorithmically as follows.

$$\text{cSTDP: } \Delta w_{ij}(t) = \begin{cases} \left(1 - w_{ij}(t-1)\right)^{\mu} P(j,t) & if\ i\ fires \\ w_{ij}(t-1)^{\mu} M(i,t) & if\ j\ fires \end{cases}$$

$$\text{rSTDP: } \Delta w_{ij}(t) = \begin{cases} -w_{ij}(t-1)^{\mu}\, P(j,t) & if\ i\ fires \\ -\left(1 - w_{ij}(t-1)\right)^{\mu} M(i,t) & if\ j\ fires \end{cases}$$

where $P(j,t)$ and $M(i,t)$ are an exponentially decaying functions with time constants τ_+ and τ_- respectively. $P(j,t)$ is increased by A_+ when j fires, and $M(i,t)$ is decreased by A_- when i fires. $P(j,t)$ and $M(i,t)$, being functions of neurons, not connections, are independent of STDP type.

These equations implement the STDP exponentials. To illustrate this, consider the case when pre-synaptic neuron j fires (possibly multiple times) before post-synaptic neuron i. Each time j fires, $P(j,t)$ is increased by A+ and decays exponentially. Thus, at time t, P (j, t) is the sum of exponential residues of the STDP potentiation curve due to all the spikes pre- synaptic neuron j fired before time t. In other words, $P(j,t)$ is the convolution of the positive half of the STDP curve with pre-synaptic neuron j's spike train up until time t. Therefore, when post-synaptic neuron i fires at time t, the weight w_{ij} updated by $P(j,t)$ reflects the sun total change of STDP due to the interaction of i's action potential with all of the pre-synaptic neuron j's prior action potentials.

We model the weights within and between layers as obeying either cSTDP or rSTDP. In most simulations, all the projections between two layers follow the same learning rule. For example, all the connections from layer 4 to layer 2/3 follow cSTDP or all of those connections follow rSTDP. There are 9 different types of connections: $L4 \rightarrow L4$, $L2/3 \rightarrow L4$, $L5/6 \rightarrow L4$, $L4 \rightarrow L2/3$, $L2/3 \rightarrow L2/3$, $L5/6 \rightarrow L2/3$, $L4 \rightarrow L5/6$, $L2/3 \rightarrow L5/6$, $L5/6 \rightarrow L5/6$. This gives a total of $2^9 = 512$ possible STDP configurations. In Fig. 4a-b and Supplementary Fig. S4, we examine scenarios where x% of the connections between two layers follow one rule and $(100 - x)$% follow the other rule. Simulations were run for 60s of simulation time to allow the matrix of average weights to converge (Supplementary Fig. S2). We ran each simulation 5 times with identical parameters except for the noisy input through external Poisson neurons.

Statistics and analysis

While weights changed dynamically throughout the simulations, they largely hovered around mean values towards the end of the simulations. Examples of the dynamic changes in individual

weights throughout the whole simulation are provided in Supplementary Fig. S2a. Additionally, Supplementary Fig. S2b,c shows the dynamic changes in the weights averaged across all pairs of neurons within each specific pair of layers. To evaluate the degree of convergence in the simulations, we computed the final weight variation defined as the standard deviation of individual weights over the last 5 seconds of the simulation. Histograms of final weight variation for the example STDP configuration used in Fig. 2 are shown in Supplementary Fig. S2d. Simulations showed that, on average, the final weight variation remained small (Supplementary Fig. S2e).

In the analyses of the results, we averaged each individual weight w over the last 5 seconds of the simulation. We show the distribution of all individual weights for each pair of layers for two example configurations in Fig. 2a4, b4. Next, we compute the average across all neuron pairs to build a weight matrix W that has 9 entries (e.g., Fig. 2a3, b3). Averaging is justified by unimodal weight distributions (e.g. Fig. 2a4). Note W denotes average weight matrices while w denotes individual weights.

To evaluate the output of each model, we compared the resulting weight matrices with an idealized target matrix T, defined in Fig. 1c, which is a simplification of a canonical inter-laminar connectivity observed in neocortical circuits of macaques and cats (Callaway 1998a, Douglas and Martin 2004). The average weight matrix for each configuration was scored against the binary target weight matrix T using a scaled version of the Frobenius norm while ignoring the diagonal elements. Model success is defined as

$$s = 1 - \sqrt{\frac{1}{6}\sum_{i \neq j}\left(T_{ij} - W_{ij}\right)^2}$$

where diagonal elements were ignored as not to make any assumptions about the distributions of weights between neurons within the same layer in the target circuit. Note that model success is

bounded between 0 and 1 with $s = 1$ if and only if $= T$. We averaged the model success across simulations and ranked the different STDP models according to success.

The best 16 models as ranked by success shared the same STDP configurations at many of the connections. We therefore focused on these configurations and investigated how the success of the best 16 configurations changed with modifications to key model parameters. All of the parameters used in the simulations are shown in Supplementary Table S1 with their default and varied values for testing robustness. Specifically, we varied the ratio of the amounts of excitatory input into each layer, the ratio between STDP parameters A_-/A_+ and τ_-/τ_+, synaptic transmission delays (default = 0 ms), and the percentage of rSTDP and cSTDP in connections between layers.

Analytical Model of Leaky Integrate-and-Fire Network with Fast Inhibition

Consider N leaky integrate-and-fire (LIF) neurons in the configuration described in the previous chapter. We will index the LIF neurons by i. Let neuron i receive input from N_i^{exc} excitatory Poisson neurons of firing rate $r_{exc} = 20\ Hz$. Let it also receive input from N_i^{inh} inhibitory neurons of firing rate r_{inh} which will have firing rates which depend on the global firing rate of the LIF neurons. Let the excitatory and inhibitory neurons have weights w_{exc} and w_{inh} respectively. The subthreshold voltage equation for a LIF neuron i is:

Equation (1)

$$\tau_m \frac{dV_i}{dt} = V_{rest} - V_i + (E_{exc} - V_i)\left(\sum_{j=1}^{N_i^{exc}} g_{ij}^{exc}(t) + \sum_{j\neq i}^{N} g_{ij}(t)\right) + (E_{inh} - V_i)\sum_{j=1}^{N_i^{inh}} g_{ij}^{inh}(t)$$

where g_{ij} is the conductance measured at the synapse from neuron j to neuron i. If excitatory Poisson neuron j fires then $g_{ij}^{exc} \rightarrow g_{ij}^{exc} + w_{ij}^{exc}$ else it decays $\tau_{exc}\frac{dg_{ij}^{exc}}{dt} = -g_{ij}^{exc}$. A similar equation governs the inhibitory neurons. If inhibitory neuron j fires then $g_{ij}^{inh} \rightarrow g_{ij}^{inh} + w_{ij}^{inh}$ else it decays $\tau_{inh}\frac{dg_{ij}^{inh}}{dt} = -g_{ij}^{inh}$. Dropping the exc and inh labels, the dynamics of the conductance can be summarized either in terms of a specified spike train $\delta_j^{spike}(t)$ or in terms of the probability of a spike $p_j^{spike}(t)$:

$$\frac{d}{dt}g_{ij} = -\frac{1}{\tau}g_{ij} + w_{ij}\,\delta_j^{spike}(t) \rightarrow \frac{1}{\tau}\left(-g_{ij} + w_{ij}\,p_j^{spike}(t)\right)$$

$$g_{ij}(t) = w_{ij}\int_{-\infty}^{t} e^{-(t-s)/\tau}\delta_j^{spike}(s)\,ds \rightarrow w_{ij}\int_{-\infty}^{t}\frac{1}{\tau}e^{-(t-s)/\tau}p_j^{spike}(s)\,ds$$

The above equations show that the conductance $g_{ij}(t)$ is simply the weighted (w_{ij}) convolution of the spike train for input neuron j with an exponential (whose time-constant is the exc/inh synaptic time-constant of 5ms). Note that when you take the derivative of the convolution, you should pull $e^{-t/\tau}$ out of the integral and use the product rule for differentiation along with the fundamental theorem of calculus. Also note that $\delta(t)$ has units of inverse time. Let $\Theta(t)$ denote the Heaviside step function, $\circledast$ denote a convolution, and $\Theta\phi_\tau = \frac{1}{\tau}e^{-t/\tau}$.

$$g_{ij}(t) = w_{ij}\left(\Theta(t)e^{-t/\tau} \circledast \delta_j^{spike}\right) \rightarrow w_{ij}\left(\Theta(t)\Theta\phi_\tau \circledast p_j^{spike}\right)$$

Note that if the input neuron has firing rate r_j, then the expected time between input spikes is $t_{ISI} = 1/r_j$ (which, for example, is t_{ISI} =50ms for r_j=20Hz, t_{ISI} =10ms for 100Hz, etc.). For $t_{ISI} > \tau$ (which for these purposes is 5ms), we can approximate $g_{ij}(t)$ as if there is "only one spike per synaptic convolution". If there is a spike at $t = 0$, then g will decay $g_{ij}(t) = g_{ij}(0)e^{-t/\tau} = w_{ij}e^{-t/\tau}$. It is fair to approximate the average conductance $\bar{g}$ to be the integral of g over the timescale of 1/r. This average conductance is approximately:

$$\bar{g} = \frac{1}{1/r}\int_0^{1/r} w * e^{-1/\tau}dt \approx w * r * \tau$$

We can equate $r * \tau$ as the expected number of spikes during interval τ. We can also view it as the probability of a spike since $r\tau < 1$ most of the time. Now we will define sums of conductance to be:

$$G_i^{exc}(t) = \sum_j g_{ij}^{exc}(t)$$

$$G_i^{inh}(t) = \sum_j g_{ij}^{inh}(t)$$

If we let all excitatory Poisson input neurons and the inhibitory neurons to have equal strength weights into our network of LIF neurons ($w_{ij}^{exc} = w_{exc}$ and $w_{ij}^{inh} = w_{inh}$), and assume they have equal firing rates ($r_j^{exc} = r_{exc}$ and $r_j^{inh}(t) = r_{inh}(t)$), then we get an averages:

$$G^{exc} = G_i^{exc} = N_{exc} w_{exc} \tau_{exc} r_{exc}$$

$$G^{inh}(t) = G_i^{inh}(t) = N_{inh} w_{inh} \tau_{inh} r_{inh}(t)$$

Note that r_{inh} is bounded by $r_{inh}^{min} \leq r_{inh} \leq r_{inh}^{max}$ where $r_{inh}^{min} = 0.005\ kHz, r_{inh}^{max} = 1\ kHz$. So that G^{inh} is bounded by $G_{min}^{inh} \leq G^{inh} \leq G_{max}^{inh}$ where $G_{min/max}^{inh} = N_{inh} w_{inh} \tau_{inh} r_{inh}^{min/max}$. For the parameters $N = 100$, $N^{exc} = 300$, $N^{inh} = 250$, $w_{max} = \frac{1}{N}$, $w_{exc} = \frac{w_{max}}{2}$, $w_{inh} = \frac{3w_{max}}{2}$, we have $G^{exc} = 0.15, G_0^{inh} = 18.65\ ms$. The term

$$g_{ij}(t) = w_{ij} \tau_{exc} \left(\Theta \phi_{\tau_{exc}} \circledast r_j \right)$$

$$G_i(t) = \sum_j g_{ij}(t) = \mathbf{W}_i \tau_{exc} \left(\Theta \phi_{\tau_{exc}} \circledast r_j \right)$$

describes the conductance at the recurrent synapses within the LIF network, where $\mathbf{W}_i$ is the i^{th} row of the weight matrix $\mathbf{W}$. We will make approximations to simplify the inhibitory term. The firing rate of these external inhibitory neurons, $r_{inh}(t)$, decays exponentially with a time constant of $\tau_I = 2$ ms. It is increased by an amount $\gamma(t)R$ where $\gamma(t)$ is the fraction of active LIF neurons at time t, and $R =$

$rmax - rmin = 1\ kHz - 0.005\ Hz = 0.995kHz$. Equivalently, $\gamma(t)$ is the probability of a unit being active at time t, averaged across all LIF neurons. Thus, we can write

$$\frac{d}{dt} r_{inh} = -\frac{1}{\tau_I} r_{inh} + \gamma(t) R$$

which has a solution

$$r_{inh}(t) = R \int_{-\infty}^{t} \frac{1}{\tau_I} e^{-(t-s)/\tau_I} \gamma(s)\, ds = R\left(\Theta\phi_{\tau_I} \circledast \gamma\right)$$

where $\gamma(t)$ is the fraction of active units at time t and is unitless. If $\gamma(t)$ is approximately constant across time scales of τ_I, then $r_{inh}(t) \approx R\gamma(t)$. What is a convenient expression for $\gamma(t)$? Considering $\gamma(t)dt$ is the expected number of active units / N during an interval dt where N is the total number of LIF neurons in our network, we give $\gamma(t)$ the form.

$$\gamma(t)dt = \frac{1}{N}\sum_{i=1}^{N} p_i^{spike}(t)\, dt = \frac{1}{N}\sum_{i=1}^{N} r_i(t)\, dt = \frac{|\mathrm{r}(t)|_1}{N} dt$$

$$\gamma(t) = \frac{|\mathrm{r}(t)|_1}{N}$$

We have defined $\gamma(t)$ to be the L1-norm of the firing rate of the LIF network normalized by the size of the network. Thus, the assumption that $\gamma(t)$ is constant across time scales of τ_I is that same as assuming $|\mathrm{r}(t)|_1$ is constant across time scales of τ_I. For $G_0^{inh} = N_{inh} w_{inh} \tau_{inh} R$, it follows that:

$$g^{inh}(t) = w_{inh}\tau_{inh}\left(\Theta\phi_{\tau_{inh}} \circledast r_{inh}\right)(t) = w_{inh}\tau_{inh} R \left(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \frac{|\mathrm{r}|_1}{N}\right)(t)$$

$$G^{inh}(t) = G_0^{inh} \left(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \frac{|\mathrm{r}|_1}{N}\right)(t)$$

Note that in order to avoid stacking units carelessly, we need to allow only one of R or $|\mathrm{r}|_1$ to carry units. Generally, we will prefer $|\mathrm{r}|_1$ has units of rate while R is unitless. We will now solve the voltage equation (1), which has the form

Equation (2)

$$\tau_m \frac{dV_i}{dt} = V_{rest} - V_i + (E_{exc} - V_i)\left(G^{exc} + \tau_{exc}\mathbf{W}_i\left(\Theta\phi_{\tau_{exc}} \circledast \mathbf{r}\right)\right) + (E_{inh} - V_i)G_0^{inh}\left(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \frac{|\mathbf{r}|_1}{N}\right)$$

by assuming that $\mathbf{r}(t)$ is approximately constant over timescales of $\tau_m = 20ms$. Then we can use $(\Theta\phi_\tau \circledast 1) = 1$ and $|\mathbf{r}|_1 = \boldsymbol{j}^T\mathbf{r}$ for $\boldsymbol{j} = (1,1,\dots,1) \in \mathbb{R}^N$ (since $r_i \geq 0$) to get

$$\tau_m \frac{dV_i}{dt} = V_{rest} - V_i + (E_{exc} - V_i)(G^{exc} + \tau_{exc}\mathbf{W}_i\mathbf{r}) + (E_{inh} - V_i)G_0^{inh}\frac{\boldsymbol{j}^T\boldsymbol{r}}{N}$$

Since the G_0^{inh} term is so large, we see that a large global firing rate can elicit large inhibitory response quickly. This effect is diminished by the fact thatthe $E_{exc} - V_i$ is larger than $E_{inh} - V_i$. In fact, we will later use the approximations:

$$E_{exc} - V_i \approx \Delta E_+ - \frac{1}{2}\Delta V = 57mV$$

$$E_{inh} - V_i \approx \Delta E_- - \frac{1}{2}\Delta V = -13mV$$

where

$$\Delta V = V_{thresh} - V_{reset} = -54mV - (-60mV) = 6mV$$

$$\Delta E_+ = E_{exc} - V_{reset} = 0 - (-60mV) = 60mV$$

$$\Delta E_- = E_{inh} - V_{reset} = -70mV - (-60mV) = -10mV$$

In other words, the driving potential for inhibition is weaker than that for excitation but a very large inhibitory signal can still be elicited. We can easily integrate by assuming constant **r** to get

Equations (3)

$$V_i(t) = E_i(\mathbf{r}) + \left(V_i(0) - E_i(\mathbf{r})\right) * e^{-t/\tau_i(\mathbf{r})}$$

for

$$E_i(\mathbf{r}) = \frac{V_{rest} + E_{exc}(G^{exc} + \tau_{exc}\mathbf{W}_i\mathbf{r}) + E_{inh}G_0^{inh}\frac{\boldsymbol{j}^T\boldsymbol{r}}{N}}{1 + G^{exc} + \tau_{exc}W_i\mathbf{r} + G_0^{inh}\frac{\boldsymbol{jr}}{N}}$$

$$\tau_i(\mathbf{r}) = \frac{\tau_m}{1 + G^{exc} + \tau_{exc}W_i\mathbf{r} + G_0^{inh}\frac{\boldsymbol{jr}}{N}} < \tau_m$$

Define the expected inter-spike interval for neuron i to be $\tau_i^{ISI} = 1/r_i(t)$. If neuron i spikes at time 0, then $V_i(0) = V_{reset}$ and it is expected that $V_i\left(\tau_i^{ISI}\right) = V_{thresh}$.

$$V_{thresh} = E_i(\mathbf{r}) + \left(V_{reset} - E_i(\mathbf{r})\right) * e^{-\tau_i^{ISI}/\tau_i(\mathbf{r})}$$

$$\tau_i^{ISI} = \tau_i(\mathbf{r})\log\left(\frac{E_i(\mathbf{r}) - V_{reset}}{E_i(\mathbf{r}) - V_{thresh}}\right)$$

$$r_i(t) = {}^{1}\!/_{\tau_i^{ISI}} = \left[\tau_i(\mathbf{r})\log\left(\frac{E_i(\mathbf{r}) - V_{reset}}{E_i(\mathbf{r}) - V_{thresh}}\right)\right]^{-1}$$

Let us define

$$x(r) = G^{exc} + \tau_{exc}\mathbf{W}_i\mathbf{r}$$

$$y(r) = G_0^{inh}\frac{\boldsymbol{j}^T\boldsymbol{r}}{N}$$

$$z(r) = \frac{\Delta E_+ x(r) + \Delta E_- y(|r|)}{1 + x(r) + y(|r|)}$$

We can write:

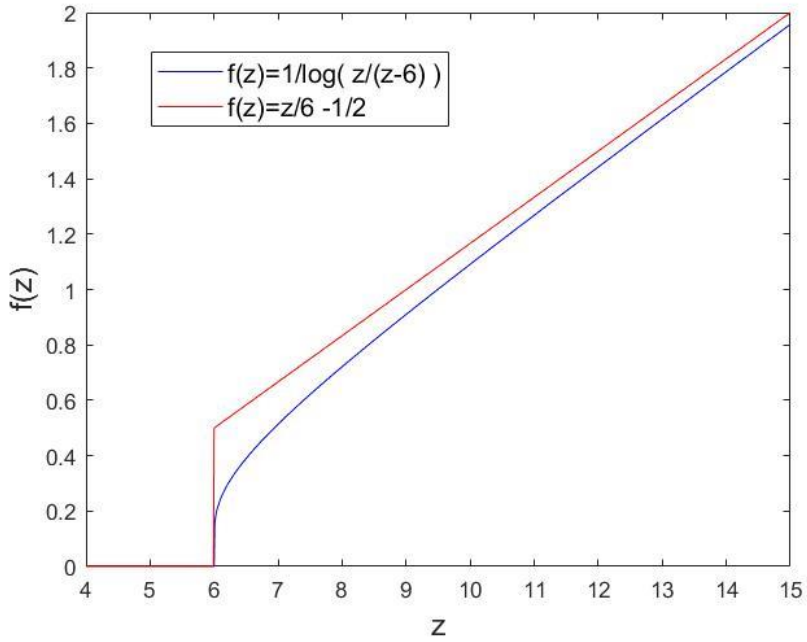


Figure 5

Demonstrating the approximation $\left(\log\left(\frac{z(r)}{z(r)-\Delta V}\right)\right)^{-1} = \frac{z(r)}{\Delta V} - \frac{1}{2}$ for $z(r) \geq \Delta V$. Here $\Delta V = 6$. Notice the approximation is a rectified linear unit (ReLU) plus a bias, a common function used in artificial neural networks.

$$\log\left(\frac{E_i(\mathbf{r})-V_{reset}}{E_i(\mathbf{r})-V_{thresh}}\right) = \log\left(1+\frac{\Delta V}{E_i(\mathbf{r})-V_{thresh}}\right) = \log\left(\frac{z(r)}{z(r)-\Delta V}\right)$$

and under the condition $z(r) > \Delta V$,

$$r_i(t) = \frac{1}{\tau_i(\mathbf{r})}\left(\log\left(\frac{z(r)}{z(r)-\Delta V}\right)\right)^{-1} = \frac{1}{\tau_i(\mathbf{r})}\frac{z(r)}{\Delta V} - \frac{1}{2}$$

This yields an approximation for the firing rate:

$$r_i(t) \approx \max\left(\frac{1}{\tau_i(\mathbf{r})}\left(\frac{z(r)}{\Delta V} - \frac{1}{2}\right), 0\right)$$

Figure M.1 shows the approximation made for the firing rate. The condition $z(r) > \Delta V$ is equivalent to $\left(\frac{\Delta E_+}{\Delta V} - 1\right)x(r) + \left(\frac{\Delta E_-}{\Delta V} - 1\right)y(|r|) > 1$, or with our parameters, $9x(r) > 1 + \frac{16}{6}y(|r|)$, where $x(r)$ and $y(|r|)$ are the excitatory and inhibitory input respectively. Then this is a condition that the L1-norm of that recurrent activity does not exceed some threshold determined by the excitatory input. When $w, r = 0$, we have $9G^{exc} = 1.35 > 1 + \frac{16}{6}y(|r|) = 1.25$. Let us define some more terms so we can write the steady state firing rate as

$$\mathbf{r}^*(t) = C_0\mathbf{j} + C_1\,\mathbf{W}(t)\,\mathbf{r}^*(t) - C_R\,\frac{|\mathbf{r}^*(t)|_1}{N}\,\mathbf{j}$$

for

$$C_0 = \frac{1}{\tau_m}\left(\left(\frac{\Delta E_+}{\Delta V} - \frac{1}{2}\right)G^{exc} - \frac{1}{2}\right) = \frac{1}{20ms}\left(\left(10 - \frac{1}{2}\right)0.05\frac{N^{exc}}{N} - \frac{1}{2}\right) = 0.046\ kHz$$

$$C_1 = \frac{\tau_{exc}}{\tau_m}\left(\frac{\Delta E_+}{\Delta V} - \frac{1}{2}\right) = \frac{5ms}{20ms}\left(\frac{60\text{mV}}{6\text{mV}} - \frac{1}{2}\right) = \frac{1}{4}\left(10 - \frac{1}{2}\right) = \frac{19}{8} = 2.375$$

$$C_R = -\frac{G_0^{inh}}{\tau_m}\left(\frac{\Delta E_-}{\Delta V} - \frac{1}{2}\right) = -\frac{7.46ms}{20ms}\frac{N^{inh}}{N}\left(\frac{-10\text{mV}}{6\text{mV}} - \frac{1}{2}\right) \approx 0.81\frac{N^{inh}}{N} = 2.02$$

$$C_R^{min} \le C_R\frac{|\mathbf{r}(t)|_1}{N} \le C_R^{max}$$

$$C_R^{min} = \frac{1}{\tau_m}\left(\frac{\Delta E_-}{\Delta V} - \frac{1}{2}\right) G_{min}^{inh} = 0.0082\ kHz$$

$$C_R^{max} = \frac{1}{\tau_m}\left(\frac{\Delta E_-}{\Delta V} - \frac{1}{2}\right) G_{max}^{inh} = 1.6\ kHz$$

The firing rate of the network should not realistically exceed 1kHz. Thus, we can consider $r_i(t) \in [0, 1]\ kHz$. Then we can have reasonable intuition for the relative magnitude of C_1 and C_R. Let $\mathbf{J} = \mathbf{j}\mathbf{j}^{\boldsymbol{T}}$ be the all-ones matrix and define $\mathbf{H}^* = C_1\mathbf{W} - C_R\frac{\mathbf{J}}{N}$. Then we can write

Equations (4)

$$\mathbf{r}^* = \boldsymbol{C_0} + \mathbf{H}^*\mathbf{r}^* = (1 - \mathbf{H}^*)^{-1}\boldsymbol{C_0} = \sum_{\boldsymbol{k=0}}^{\infty} \mathbf{H}^{*k}\boldsymbol{C_0}$$

$$\mathbf{H}^* = C_1\mathbf{W} - C_R\frac{\mathbf{J}}{N}$$

We can check stability by perturbing the firing rate and seeing if the firing rate increases or decreases. Let $\mathbf{r} = \boldsymbol{r}^* + \boldsymbol{v}$. Then

$$\boldsymbol{r} \to C_0\mathbf{j} + \mathbf{H}^*(\boldsymbol{r}^* + \boldsymbol{v}) = \boldsymbol{r}^* + \mathbf{H}^*\boldsymbol{v}$$

If $\|\mathbf{H}^*\| = \left\|C_1\mathbf{W} - C_R\frac{\mathbb{1}}{N}\right\| < 1$, then the firing rate is decreasing (for a positive perturbation) which means the equilibrium is stable. This term is interesting in light of the **Gershgorin circle theorem** (GC). Let $A = [a_{ij}]$ be a $n \times n$ matrix and $R_i = \sum_{j \neq i}|a_{ij}|$ be the sum of the off-diagonal absolute values (L1-norm minus the diagonal). Define the Gershgorin disc $D(a_{ii}, R_i^D)$ to be the closed disc in the complex plane, centered at a_{ii} and with radius R_i^D. Then the Gershgorin circle theorem states that every eigenvalue of A lies within at least one of the Gershgorin discs $D(a_{ij}, R_i^D)$.

Let $\mathbf{W} = [w_{ij}]$ where $0 \leq w_{ij} \leq w_{max} = \frac{1}{N}$. Let $\mathbf{W}_i$ be a row of $\mathbf{W}$. Note that $\mathbf{W}_i$ is the set of *input* weights into neuron i. Take $A = \mathbf{H}^* = C_1\mathbf{W} - C_R\frac{\mathbf{J}}{N}$, and $a_{ii} = -\frac{C_R}{N}$ because $w_{ii} = 0$. In the limit $C_R \to 0$, we see that the Gershgorin discs become centered at 0 and the eigenvalues are bounded by the largest set of input weights. Since each weight is bounded by $w_{max} = \frac{1}{N}$, the set of input weights is bounded by $\frac{N-1}{N} < 1$. A mathematical side note, $R_{max}^D = \max_i(C_1|\mathbf{W}_i|_1) = C_1\|\mathbf{W}\|_\infty$ where $|\cdot|_1$ is the vector L1-norm and $\|\cdot\|_\infty$ is the matrix infinity-norm. The input $\boldsymbol{C_0}$ into the steady-state rate firing $\mathbf{r}^*$, can be represented as $\boldsymbol{C_0} = \sum C_{0_i}\mathbf{e_i}$ where $\mathbf{e_i}$ are unit eigenvectors of W, with eigenvalues λ_i. The term $C_{0_i} = \langle \boldsymbol{C_0}, \mathbf{e_i} \rangle$ is the overlap of the input with the i^{th} eigenvector of W, sometimes called a pattern. Then we can write

$$\mathbf{r}^* = C_1\sum_{\boldsymbol{k=0}}^{\infty}\mathbf{W}^k\boldsymbol{C_0} = C_1\sum_{\boldsymbol{i}}\sum_{\boldsymbol{k=0}}^{\infty}{\lambda_i}^k C_{0_i}\mathbf{e_i} = \sum_{\boldsymbol{i}}\frac{C_1}{1-\lambda_i}C_{0_i}\mathbf{e_i}$$

which is a filtered version of the input $\boldsymbol{C_0}$**.** The new coefficients are scaled by $\frac{C_1}{1-\lambda_i}$ which become very large and diverges as $\lambda_i \to 1$. For $C_R > 0$, we have that the centers of the Gershgorin discs move by an amount $-\frac{C_R}{N}$ and the radii are $R_i^D = \sum_{j\neq i}\frac{C_1}{N}\left|\frac{w_{ij}}{w_{max}} - \frac{C_R}{C_1}\right|$. For our parameter choice, we have $\frac{C_R}{C_1} = 0.85$. Thus, the contribution from large weights to the radii is now small – and conversely, small weights can contribute a lot.

Higher Order Corrections to the Steady State Solution

We have derived a steady state equation for $\mathbf{r}^*$ but we needed to assume that $\mathbf{r}$ was constant on timescales of τ_m. This assumption is equivalent to assuming that the derivative of the voltage, $\frac{dV}{dt}$, is constant. We can add perturbation by setting $\mathbf{r}(t) = \mathbf{r}^* + \boldsymbol{v_0} t$ where $\boldsymbol{v_0}$ is constant, making $\mathbf{r}$ and $\frac{dV}{dt}$ not constant. Since the firing rate must be positive, we restrict $-\mathbf{r}^* < \boldsymbol{v_0} t$. But we still have the useful relation $|\mathbf{r}|_\mathbf{1} = \mathbf{j}^T\mathbf{r} = \mathbf{j}^T\mathbf{r}^* + \mathbf{j}^T\boldsymbol{v_0} t$. Let $\tau_m \frac{dV_i{}^*}{dt}$ denote the solution to the voltage equation (5) when we have constant firing rate $\mathbf{r}^*$. Then the perturbation leads to

$$\begin{aligned}
\frac{dV_i}{dt} &= \frac{dV_i{}^*}{dt} + \frac{\tau_{exc}}{\tau_m}(E_{exc} - V_i)\mathbf{W}_i\left(\Theta\phi_{\tau_{exc}} \circledast t\right)\boldsymbol{v_0} + \frac{1}{\tau_m}(E_{inh} - V_i)G_0^{inh}\left(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast t\right)\frac{\mathbf{j}^T\boldsymbol{v_0}}{N} \\
&= \frac{dV_i{}^*}{dt} + \frac{\tau_{exc}}{\tau_m}(E_{exc} - V_i)(t - \tau_{exc})\mathbf{W}_i\boldsymbol{v_0} + \frac{1}{\tau_m}(E_{inh} - V_i)G_0^{inh}(t - \tau_{inh} - \tau_I)\frac{\mathbf{j}^T\boldsymbol{v_0}}{N} \\
&\approx \frac{dV_i{}^*}{dt} + \frac{\tau_{exc}}{\tau_m}\left(\Delta E_+ - \frac{1}{2}\Delta V\right)(t - \tau_{exc})\mathbf{W}_i\boldsymbol{v_0} + \frac{1}{\tau_m}\left(\Delta E_- - \frac{1}{2}\Delta V\right)G_0^{inh}(t - \tau_{inh} - \tau_I)\frac{\mathbf{j}^T\boldsymbol{v_0}}{N} \\
&= \frac{dV_i{}^*}{dt} + \Delta V(t - \tau_{exc})C_1\mathbf{W}_i\boldsymbol{v_0} - \Delta V(t - \tau_{inh} - \tau_I)C_R\frac{\mathbf{j}^T\boldsymbol{v_0}}{N}
\end{aligned}$$

Let us define a more general operator

$$\mathbf{H}\boldsymbol{x} = C_1\mathbf{W}\left(\Theta\phi_{\tau_{exc}} \circledast \boldsymbol{x}\right) - C_R\frac{\mathbf{J}}{N}\left(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \boldsymbol{x}\right)$$

For $\boldsymbol{x} = \boldsymbol{constant}$, we have $\mathbf{H} = \mathbf{H}^*$. For $\boldsymbol{x} = \boldsymbol{v_0} t$, we have

$$\mathbf{H}\boldsymbol{v_0}t = C_1\mathbf{W}\boldsymbol{v_0}(t-\tau_{exc}) - C_R\frac{\mathbf{J}\boldsymbol{v_0}}{N}(t-\tau_{inh}-\tau_I)$$

The right-hand side includes two different time delays we saw present in the perturbation to the voltage equation. The excitatory term enters the network a little bit ahead of the inhibitory term. Here, that time constant in very fast, being $\tau_I = 2ms$. The synaptic time constants are also fast, being $\tau_{exc} = \tau_{inh} = 5ms$. Still, the inhibitory signal will always lag the recurrent signal because it requires time to integrate the activity of the recurrent network. For $\boldsymbol{x} = \boldsymbol{v_0}t$, the voltage derivative reduces to

$$\frac{dV}{dt} = \frac{dV^*}{dt} + \Delta V\mathbf{H}\boldsymbol{v_0}t.$$

We will later justify the general form for the firing rate to be

Equation (5)

$$\mathbf{r} = \boldsymbol{C_0} + \mathbf{H}\mathbf{r} = (1-\mathbf{H})^{-1}\boldsymbol{C_0} = \sum_{\boldsymbol{k=0}}^{\infty}\mathbf{H}^k\boldsymbol{C_0}$$

The time dependence of $\mathbf{r}$ is incorporated into the convolutions within $\mathbf{H}$. We can do the same perturbation $\mathbf{r} \to \mathbf{r}^* + \boldsymbol{v_0}t$. Then we get the relation

$$\mathbf{r} \to \mathbf{r}^* + \mathbf{H}\boldsymbol{v_0}t = \mathbf{r}^* + (t-\tau_{exc})C_1\mathbf{W}\boldsymbol{v_0} - (t-\tau_{inh}-\tau_I)C_R\frac{\mathbf{J}\boldsymbol{v_0}}{N}$$

which is the same perturbation we obtained from the voltage equation but scaled by ΔV^{-1} in order to achieve proper units. This make sense since under the conditions that $\mathbf{r} = \mathbf{r}^*$ and $\frac{dV}{dt} = \frac{dV_i^*}{dt}$ are

constant, we have the relation $\frac{dV_i}{dt}^* = \Delta V \mathbf{r}^*$ by the definition of slope and firing rate. What are solutions to equation (5)? Using the properties

$$\frac{d}{dt}(\Theta\phi_\tau \circledast v) = \frac{1}{\tau}\big(v(t) - (\Theta\phi_\tau \circledast v)\big)$$

$$\frac{d}{dt}(\Theta\phi_\tau \circledast \Theta\phi_{\lambda\neq\tau} \circledast v) = \frac{(\Theta\phi_\lambda(t) \circledast v) - (\Theta\phi_\tau(t) \circledast v)}{\tau - \lambda}$$

we can first find it's derivative.

$$\mathbf{r} = \boldsymbol{C_0} + \mathbf{Hr} = \boldsymbol{C_0} + C_1 W\big(\Theta\phi_{\tau_{exc}} \circledast \mathbf{r}\big) - C_R \frac{\mathbf{J}}{N}\big(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \mathbf{r}\big)$$

$$\mathbf{r}' = (\mathbf{Hr})' = C_1 \mathbf{W} \frac{1}{\tau_{exc}}\Big(\mathbf{r} - \big(\Theta\phi_{\tau_{exc}} \circledast \mathbf{r}\big)\Big) - C_R \frac{\mathbf{J}}{N} \frac{\big(\Theta\phi_{\tau_I}(t) \circledast \mathbf{r}\big) - \big(\Theta\phi_{\tau_{inh}}(t) \circledast \mathbf{r}\big)}{\tau_{inh} - \tau_I}$$

As a sanity check, it is indeed zero for constant $\mathbf{r}$. Note that the whole derivative $\mathbf{r}'$ is a function of $\mathbf{r}$. Thus, we can loosely think of the solution as an exponential function. However, the time constants would have very complicated behaviors, changing from negative (exponential decay) to positive (exponential growth) and depending on the history of the firing rate itself (convolutions).

We can take the Fourier transform of equation (5) to find more properties. We denote the Fourier transform as $\mathcal{F}\{f\}(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt$ and denote the Fourier transform of $\mathbf{r}$ as $\mathcal{F}\{\mathbf{r}(t)\}$. We will use the convolution theorem $\mathcal{F}\{f \circledast g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$ and the property $\mathcal{F}\{\Theta\phi_\tau(t)\} = \frac{1}{1+i\omega\tau}$.

$$\mathcal{F}\{\mathbf{r}\} = \mathcal{F}\{\boldsymbol{C_0}\} + \left(\frac{1}{1 + i\omega\tau_{exc}} C_1 W - \frac{1}{1 + i\omega\tau_I} \frac{1}{1 + i\omega\tau_{inh}} C_R \frac{\mathbf{J}}{N}\right) \mathcal{F}\{\mathbf{r}\}$$

To simplify, we will take $\tau_{syn} = \tau_{exc} = \tau_{inh}$ which is the case for our parameters. Multiplying both sides by $\left(1 + i\omega\tau_{syn}\right)$ and $(1 + i\omega\tau_I)$ yields

$$\left(1 + i\omega\left(\tau_I + \tau_{syn}\right) + (i\omega)^2\tau_I\tau_{syn}\right)(\mathcal{F}\{\mathbf{r}\} - \mathcal{F}\{\boldsymbol{C_0}\}) = \left(C_1(1 + i\omega\tau_I)\mathbf{W} - C_R\frac{\mathbf{J}}{N}\right)\mathcal{F}\{\mathbf{r}\}$$

We can take the inverse Fourier transform, $\mathcal{F}^{-1}\{f\}(t) = (2\pi)^{-1}\int_{-\infty}^{\infty} e^{i\omega t}f(\omega)\,d\omega$, and use the property $\mathcal{F}\{f'\} = i\omega\mathcal{F}\{f\}$. After rearranging terms, we get

$$\left(\mathbf{1} - C_1\mathbf{W} + C_R\frac{\mathbf{J}}{N}\right)\mathbf{r} + \left(\tau_I + \tau_{syn} - \tau_I C_1\mathbf{W}\right)\mathbf{r}' + \tau_I\tau_{syn}\mathbf{r}'' = \boldsymbol{C_0}$$

Note that if we took $\tau_{exc} \neq \tau_{inh}$ during the Fourier transform, then we would have ended up with terms which were third derivatives of the firing rate. We recognize the first term as $(\mathbf{1} - \mathbf{H}^*)\mathbf{r} = \left(\mathbf{1} - C_1\mathbf{W} + C_R\frac{\mathbf{J}}{N}\right)\mathbf{r}$, which is equal to $\boldsymbol{C_0}$ when $\mathbf{r} = \mathbf{r}^*$. Thus, the equation correctly reduces to the steady state solution. We will denote the first order coefficient as $\mathbf{H_1}$ and the second order coefficient as $\mathbf{L_1}$.

Equations (6)

$$(\mathbf{1} - \mathbf{H}^*)\mathbf{r} + \mathbf{H_1}\mathbf{r}' + \mathbf{L_1}\mathbf{r}'' = \boldsymbol{C_0}$$

$$\mathbf{H_1} = \tau_{syn}\mathbf{1} + \tau_I(1 - C_1\mathbf{W})$$

$$\mathbf{L_1} = \tau_I\tau_{syn}\mathbf{1}$$

This is simply a second order linear differential equation and will most likely behave like a driven damped harmonic oscillator. Let's write this equation as a system of first order linear differential equations. Consider the substitutions $\boldsymbol{y_1} = \mathbf{r}$ and $\boldsymbol{y_2} = \mathbf{r}'$. We have the system

$$\boldsymbol{y_1}' = \boldsymbol{y_2}$$

$$\boldsymbol{y_2}' = \boldsymbol{C_0} - \frac{\mathbf{1}}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*)\boldsymbol{y_1} - \frac{\mathbf{1}}{\tau_I \tau_{syn}}\mathbf{H_1}\boldsymbol{y_2}$$

We will now, for convenience, do a change of variables letting $\boldsymbol{\rho} = \mathbf{r} - \mathbf{r}^*$. Again, define the state variables to be $\boldsymbol{\psi_1} = \boldsymbol{\rho}$ and $\boldsymbol{\psi_2} = \boldsymbol{\rho}' = \mathbf{r}'$. Then we can rewrite equation (6) in terms of its deviations from the steady state solution

$$\boldsymbol{\rho}'' + \frac{1}{\tau_I \tau_{syn}}\mathbf{H_1}\boldsymbol{\rho}' + \frac{1}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*)\boldsymbol{\rho} = \mathbf{0}$$

which gives us the set of first order differential equations

$$\boldsymbol{\psi_1}' = \boldsymbol{\psi_2}$$

$$\boldsymbol{\psi_2}' = \frac{\mathbf{1}}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*)\boldsymbol{\psi_1} - \frac{\mathbf{1}}{\tau_I \tau_{syn}}\mathbf{H_1}\boldsymbol{\psi_2}$$

We can generalize the system to a matrix equation. Let the vector $\boldsymbol{\psi}(t) = [\boldsymbol{\psi_1}(t), \boldsymbol{\psi}(t)]^{\boldsymbol{T}} = [\boldsymbol{\rho}(t), \boldsymbol{v}(t)]^{\boldsymbol{T}}$ consist of the steady-state subtracted firing rate $\boldsymbol{\rho}(t)$ and the derivative (velocity) of the firing rate $\mathbf{r}'(t) = \boldsymbol{\rho}'(t) = \boldsymbol{v}(t)$ at time t.

$$\boldsymbol{\psi}' = \mathbf{A}\boldsymbol{\psi}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ -\frac{1}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*) & -\frac{1}{\tau_I \tau_{syn}}\mathbf{H_1} \end{bmatrix}$$

We see that the steady-state solution $\boldsymbol{\psi}^* = [\,\boldsymbol{\rho}^* = \mathbf{r}^* - \mathbf{r}^*, \mathbf{0}]^{\boldsymbol{T}} = \mathbf{0}$ is the zero vector. Our goal is to solve for $\boldsymbol{\psi}$ by finding eigenvalues and eigenvectors of $\mathbf{A}$. Note the trace of $\mathbf{H}^*$ is $\text{trace}(\mathbf{H}^*) = -C_R$ because $\text{trace}(\mathbf{J}) = N$ and $\text{trace}(\mathbf{W}) = 0$. Then the trace and determinant of $\mathbf{A}$ are

$$\text{trace}(\mathbf{A}) = -\frac{1}{\tau_I \tau_{syn}} \text{trace}(\mathbf{H_1}) = -\frac{\tau_I + \tau_{syn}}{\tau_I \tau_{syn}} N$$

$$\det(\mathbf{A}) = \left(\frac{1}{\tau_I \tau_{syn}}\right)^N \det(\mathbf{1} - \mathbf{H}^*)$$

Eigenvectors of $\mathbf{A}$ are of the form

$$\mathbf{A}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ -\frac{1}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*) & -\frac{1}{\tau_I \tau_{syn}} \mathbf{H_1} \end{bmatrix} \begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} = \lambda \begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}$$

Which satisfy the equations

$$\boldsymbol{y} = \lambda \boldsymbol{x}$$

$$-\frac{1}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*)\boldsymbol{x} - \frac{1}{\tau_I \tau_{syn}} \mathbf{H_1}\boldsymbol{y} = \lambda \boldsymbol{y}.$$

We aim to solve for eigenvectors and eigenvalues of $\mathbf{A}$ by solving the following equation:

$$\left(\lambda^2 \mathbf{1} + \frac{\lambda}{\tau_I \tau_{syn}} \mathbf{H_1} + \frac{1}{\tau_I \tau_{syn}}(\mathbf{1} - \mathbf{H}^*)\right)\boldsymbol{x} = \mathbf{0}$$

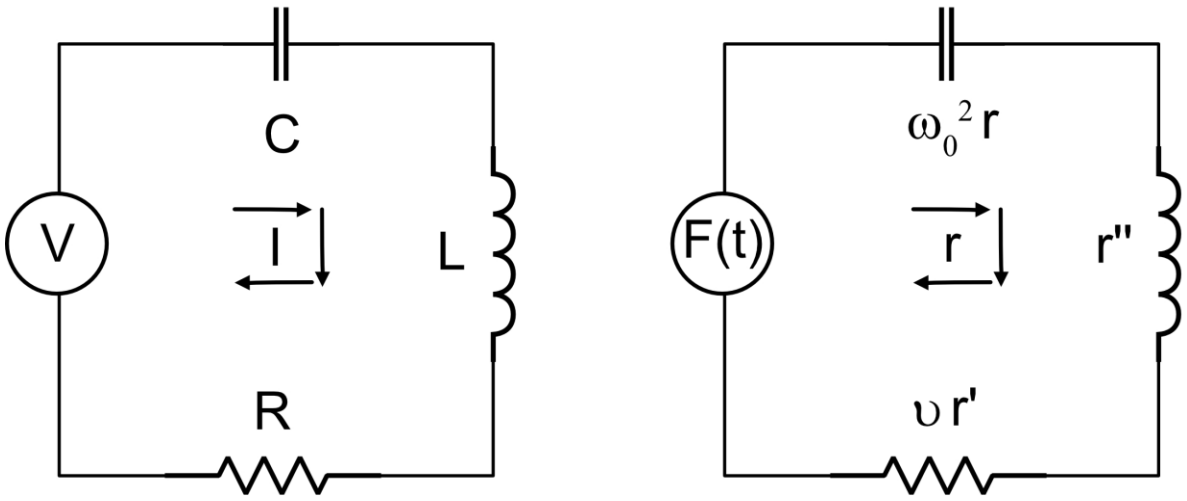


Figure 6

Equivalence of an LRC circuit with a driven damped harmonic oscillator. (Left) A driven LRC circuit diagram. Positive current is shown to travel clockwise. (Right) corresponding terms in driven damped harmonic oscillator.

This is a relatively simple equation; however, it is tedious to solve in more detail than this for the general case. Any particular solution is going to depend on the exact form of the weight matrix **W.**. We can gain further insight into the solution by mapping the problem onto an equivalent electrical circuit. It is well known that the driven damped harmonic oscillator is equivalent to a driven LRC circuit as demonstrated in Figure 6. The form of a driven damped harmonic oscillator equation is

$$\boldsymbol{F}(t) = \boldsymbol{r}'' + \upsilon \boldsymbol{r}' + \omega_0^2 \boldsymbol{r}.$$

The term ω_0^2 is the undamped angular frequency of the oscillator. The linear term υ is related to the bandwidth which will be discussed soon. We immediately see that the weight matrix effects both the frequency and the bandwidth in almost the same way. That is, as a $1 - C_1\mathbf{W}$ term.

$$\omega_0^2 = \frac{(\mathbf{1} - \mathbf{H}^*)}{\tau_I \tau_{syn}} = \frac{\left(\mathbf{1} - C_1\mathbf{W} + C_R \frac{\mathbf{J}}{N}\right)}{\tau_I \tau_{syn}}$$

$$\upsilon = \frac{\mathbf{H_1}}{\tau_I \tau_{syn}} = \frac{\tau_{syn}\mathbf{1} + \tau_I(1 - C_1\mathbf{W})}{\tau_I \tau_{syn}}$$

We can set up the solution to the RLC circuit by solving Kirkhoff's laws. Specifically, the second law states that the sum of the voltages around any closed loop is zero. In this circuit, we have only one closed loop, which give us

$$V = \frac{Q}{C} + L\frac{dI}{dt} + IR$$

$$V' = \frac{1}{C}I + RI' + LI''$$

We recover equation (6) if we equate $I = \mathbf{r}, V' = \boldsymbol{C_0}, C^{-1} = (\mathbf{1} - \mathbf{H}^*), R = \mathbf{H_1}, L = \mathbf{L_1}$. Thus, our LIF and inhibitory neurons form a network which behaves equivalently to this circuit, where we equate firing rates as currents. It actually makes physical sense to model firing rate as current. This is because we realized while dealing with the membrane voltage equation that the firing rate was proportional to the derivative of the voltage. In circuits, the derivative of the voltage is generally proportional to current.

However, we need to make a very important clarification. The firing rate is equivalent to the current, which is the derivative of charge, not voltage. In a RC circuit, the voltage is given by $Q = VC$ so the firing rate is proportional to the derivative of the voltage for constant capacitance. However, when the other partial derivatives of voltage are not 0, then the firing rate is no longer simply the derivative of the voltage. It is still the derivative of charge. When there are synapses with different time constants, it seems that we need to take into its partial derivative with respect to charge. This is likely the physics underlying the fact that we see these oscillations.

A capacitor resembles the membrane potential of a neuron while the current across its membrane is the firing rate. In fact, this relation to circuit design should not be that surprising because we may recall that a LIF neuron is equivalent to a RC circuit. What is surprising is that we have introduce an inductor. These will produce oscillations which is something absent from an RC circuit. What else is surprising is that the weight matrix shows up as a capacitance term. The common intuition for weights is to be analogous to inverse resistance (conductance). However, there presence in the resistor term is most likely due to the effective resistance of the network and current leaking from the capacitor across this effective resistance.

Thus, it should be possible to think of the weights as capacitance. For example, as water flows through a tree, when it reaches a branch the water bifurcates. How much water flows through each branch is related to the diameter of the branch. One may this of the branch diameter as inverse resistance or as a weight. However, if there is flow, the water has to be flowing somewhere. It can

either become stored somewhere in the tree such as in a leaf or in chemical reactions, or it is dissipated into the air. It either gets stored on some capacitor or it leaks out through a resistor. Thus, the weight matrix should be relatable to capacitance as it seems to also be a measure of how much charge a neuron is capable of storing. It may prove fruitful mathematically to establish a stronger connection between weights and capacitance. This is because the weight matrix is generally not symmetric while the Maxwell capacitance matrix is always symmetric by construction. Therefore, if we can reframe the system in terms of voltage, charge, and capacitance, we may have new mathematical tools to work with.

A very common use for LCR circuits is to tune AM radios to a particular frequency ω_0, and with a certain bandwidth υ. In practice, the resistor and inductor are held constant while the capacitor changes by the twist of a knob which moves the parallel plates further or closer to each other. In computational neuroscience, the "knob" would consist of other factors modifying a neuron's membrane potential. Some example may include feedback connections, other network correlation effects, neuromodulators like dopamine, and in general, the brain's state taken as a whole. If we add synaptic plasticity to our models, then we are also able to "tune" our LRC circuit somewhat more permanently by changing the weights. However, this would change both the resistance and the capacitance simultaneously. How does this effect things? For a given capacitance, the circuit will resonate with an input frequency ω if ω is in the range

$$\omega_0 \pm \upsilon = \frac{1}{\sqrt{LC}} \pm \frac{R}{L} = \left(\frac{\mathbf{1} - \mathbf{H}^*}{\mathbf{L_1}}\right)^{\mathbf{1/2}} \pm \frac{\mathbf{H_1}}{\mathbf{L_1}}.$$

The circuit will resonate at frequencies proportional to the eigenvalues of $\mathbf{1} - \mathbf{H}^*$. If the eigenvalues of $\mathbf{W}$ are close to 1, then the resistance is small, and the bandwidth is narrow. This property is called a good "quality" factor in signal processing.

Furthermore, we can see limitation of the steady state approximation. If $\mathbf{r}'$ and $\mathbf{r}''$ are both 0, then the derivative of voltage is constant. Since this is the same as the condition that current across the capacitor is constant. So, we learn *why* the steady-state approximation is insufficient. This is because an LRC circuit will *always* have oscillating current, even for constant DC current source. Thus, while the constant firing rate approach to solving the dynamics may sometimes be convenient, the assumption $\mathbf{r} = constant$ is biologically and physically false. Generally, when making the steady state approximation, one assumes that the second derivative is small and only the first derivative is considered. The system does behave like a damped oscillator so the first derivative will be negative and give the appearance of a steady state solution. However, if there is an inhibitory network present, it appears that the second derivative may actually be large enough to call into question the accuracy of the steady state approximation.

More importantly, we err by assuming that firing rate is the derivative of voltage. However, the correct way to think about firing rate is as the current which is the derivative of charge. And generally, we need to consider partial derivatives

$$r = I = \frac{dQ}{dt} = \frac{\partial Q}{\partial t} + \frac{\partial Q}{\partial V}\frac{dV}{dt} + \frac{\partial Q}{\partial \Phi}\frac{d\Phi}{dt} = \frac{\partial Q}{\partial t} + C\frac{dV}{dt} + \frac{1}{M}\frac{d\Phi}{dt}$$

where Φ is the magnetic flux, C is the capacitance, and M is the memristance. When we have $\frac{\partial Q}{\partial t} = 0$ and $\frac{d\Phi}{dt} = 0$ then we are tempted to equate firing rate with the derivative of the voltage. In which case, we are dealing with an RC circuit and we can take advantage of $Q = VC$. Then the firing rate is still the derivative of charge, but we accidentally think of it as the derivative of voltage. When we introduced the inhibitory network with a time delay, then most likely we introduced $\frac{\partial Q}{\partial t} \neq 0$ terms explicitly – and maybe $\frac{d\Phi}{dt} \neq 0$ terms implicitly.

The terms $\mathbf{L_1} = \tau_{exc}\tau_{inh}$, and $\mathbf{H_1} = \left(\tau_{syn} + \tau_I(1 - C_1\mathbf{W})\right)$ appear to be direct consequences of the convolutions used while integrating input at each synapse. The convolutions with an exponential decay were used to model the leaky aspect of a synapse. Notice the relations

$$\int_0^t (\Theta\phi_\tau \circledast \Theta)(s)ds = \Theta(\mathrm{t})\big(t - \tau(\Theta\phi_\tau \circledast \Theta)(t)\big) \approx \frac{1}{2}\frac{t^2}{\tau}\Theta(\mathrm{t})$$

$$\int_0^t \big(\Theta\phi_{\tau_1} \circledast \Theta\phi_{\tau_2} \circledast \Theta\big)(s)ds \approx \frac{1}{6}\frac{t^3}{\tau_1\tau_2}\Theta(\mathrm{t})$$

$$\big(\Theta\phi_{\tau_{exc}} \circledast \Theta t\big) \approx (\mathrm{t} - \tau_{exc})\Theta(\mathrm{t} - \tau_{exc})$$

$$\big(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \Theta t\big) \approx (\mathrm{t} - \tau_{exc} - \tau_I)\,\Theta(\mathrm{t} - \tau_{exc} - \tau_I)$$

The integrals are in units of voltage which is an integral of current. So, firing rate terms will look like derivatives of the integrals. Thus $\int_0^t \big(\Theta\phi_{\tau_1} \circledast \Theta\phi_{\tau_2} \circledast \Theta\big)(s)ds$ gives the second order terms for the firing rate. It includes multiplication of time constants just like the inductor term. Thus, we see that we get oscillations in firing rate when we introduce the inhibitory network with a time delay. The resistor $\mathbf{H_1}$ can also be understood by considering the leaky integrate-and-fire neurons as being leaky capacitors. It is common practice in circuit design to approximate a non-ideal capacitor as an ideal capacitor in series with a resistor.

Separation of Inhibitory and Excitatory Populations

Starting with equation (5), we can choose instead to keep the variables $\mathbf{r}$ and $\mathbf{r}_{inh}$ as separate. Then we have the system

Equations (7)

$$\mathbf{r}_{inh} = R\left(\Theta\phi_{\tau_I} \circledast \frac{\mathbf{J}}{N}\mathbf{r}\right)$$

$$\mathbf{r} = \boldsymbol{C_0} + C_1\mathbf{W}\left(\Theta\phi_{\tau_{exc}} \circledast \mathbf{r}\right) - \frac{C_R}{R}\left(\Theta\phi_{\tau_{inh}} \circledast \mathbf{r}_{inh}\right)$$

Let $R = 0.995 \approx 1$ for convenience. We can follow a similar procedure as the last section by applying Fourier transform,

$$\mathcal{F}\{\mathbf{r}_{inh}\} = \frac{1}{1 + i\omega\tau_I}\frac{\mathbf{J}}{N}\mathcal{F}\{\mathbf{r}\}$$

$$\mathcal{F}\{\mathbf{r}\} = \mathcal{F}\{\boldsymbol{C_0}\} + \frac{1}{1 + i\omega\tau_{exc}}C_1\mathbf{W}\mathcal{F}\{\mathbf{r}\} - \frac{1}{1 + i\omega\tau_{inh}}C_R\mathcal{F}\{\mathbf{r}_{inh}\}$$

simplifying,

$$(1 + i\omega\tau_I)\mathcal{F}\{\mathbf{r}_{inh}\} = \frac{\mathbf{J}}{N}\mathcal{F}\{\mathbf{r}\}$$

$$(1 + i\omega(\tau_{exc} + \tau_{inh}) + (\mathrm{i}\omega)^2\tau_{exc}\tau_{inh})(\mathcal{F}\{\mathbf{r}\} - \mathcal{F}\{\boldsymbol{C_0}\}) =$$

$$(1 + i\omega\tau_{inh})C_1\mathbf{W}\mathcal{F}\{\mathbf{r}\} - (1 + i\omega\tau_{exc})C_R\mathcal{F}\{\mathbf{r}_{inh}\}$$

and taking inverse Fourier transform while noting the derivatives we obtain,

$$\tau_I \mathbf{r'}_{inh} = \frac{\mathbf{J}}{N}\mathbf{r} - \mathbf{r}_{inh}$$

$$\tau_{exc}\tau_{inh}\mathbf{r''} = \boldsymbol{C_0} + (C_1\mathbf{W} - 1)\mathbf{r} - C_R\mathbf{r}_{inh} - \tau_{exc}C_R\mathbf{r'}_{inh} + (\tau_{inh}(C_1\mathbf{W} - 1) - \tau_{exc})\mathbf{r'}$$

We can rewrite the second equation as

$$\tau_{exc}\tau_{inh}\mathbf{r''} + \big(\tau_{exc} + \tau_{inh}(1 - C_1\mathbf{W})\big)\mathbf{r'} + \left(1 - C_1\mathbf{W} + \frac{\tau_{exc}}{\tau_I}C_R\frac{\mathbf{J}}{N}\right)\mathbf{r} = \boldsymbol{C_0} + \left(\frac{\tau_{exc} - \tau_I}{\tau_I}\right)C_R\mathbf{r}_{inh}$$

Note that $\mathbf{K}\frac{\mathbf{J}}{N} = \mathbf{K}$. Then we can summarize the system with the following set of equations:

Equations (8)

$$\mathbf{L_2}\mathbf{r''}(t) + \mathbf{H_2}\mathbf{r'} + (\mathbf{1} - \mathbf{H}^*)\mathbf{r} + \mathbf{K}(\mathbf{r} - \mathbf{r}_{inh}) = \boldsymbol{C_0}$$

$$\tau_I\mathbf{K}\mathbf{r'}_{inh} = \mathbf{K}(\mathbf{r} - \mathbf{r}_{inh})$$

for

$$\mathbf{L_2} = \tau_{exc}\tau_{inh}$$

$$\mathbf{H_2} = \tau_{exc} + \tau_{inh}(1 - C_1\mathbf{W})$$

$$\mathbf{K} = \left(\frac{\tau_{exc} - \tau_I}{\tau_I}\right)C_R\frac{\mathbf{J}}{N}$$

This solution resembles closely a driven damped harmonic oscillator as in the LRC circuit**.** It is complicated by the fact that the driving force is a function of the "position" as well, i.e. the firing rate. The capacitance is still unchanged. The inductance has changed to generally slower timescales as $\mathbf{L_2}^{-1} < \mathbf{L_1}^{-1}$**.** To gain more insight, we can also map these sets of equations onto an electrical circuit.

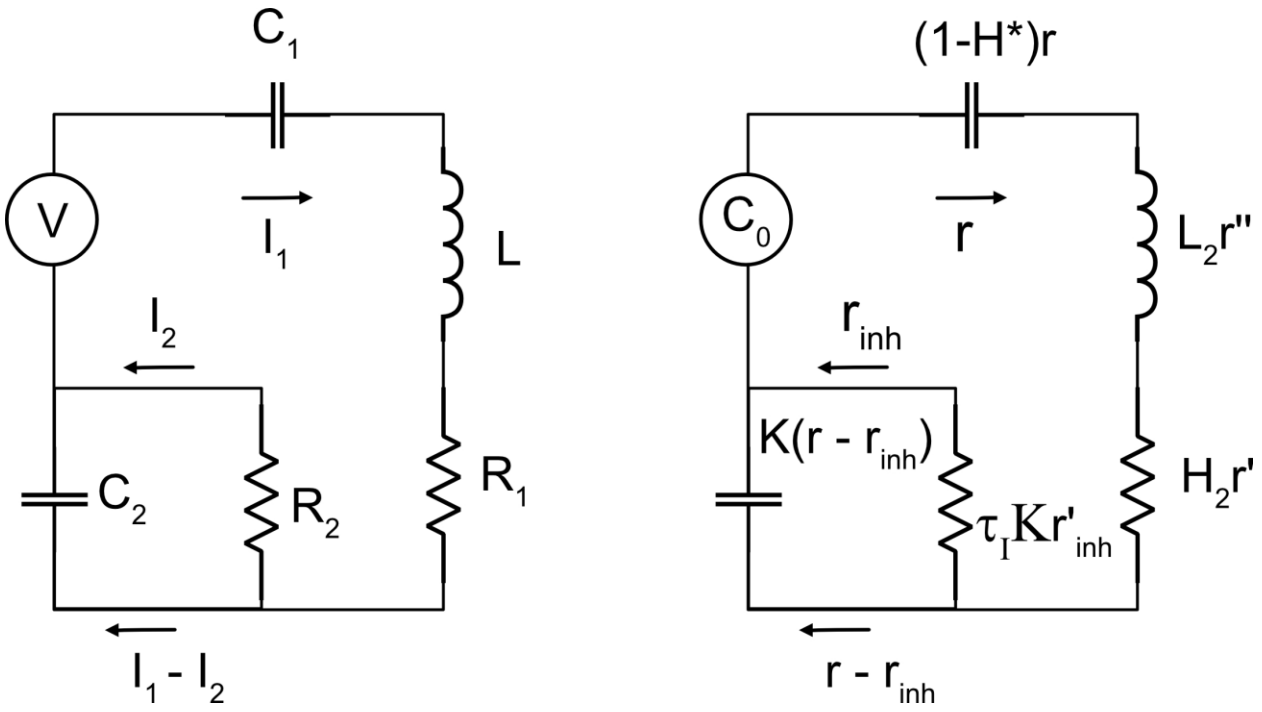


Figure 7

Equivalence circuit for equations (8). (Left) A driven LRC circuit with an embedded RC circuit. (Right) Current through the outer loop is equivalent to the firing rate of the LIF network of neurons. Current through the outer loop is equivalent to the firing rate of the inhibitory neurons.

Consider the circuit shown in Figure 7. We can sum the voltages around the larger and smaller loop respectively to get

$$V = \frac{Q_1}{C_1} + L\frac{dI_1}{dt} + I_1 R_1 + \frac{Q_2}{C_2}$$

$$\frac{Q_2}{C_2} = I_2 R_2$$

Taking the derivative with respect to time, we get

$$V' = \frac{1}{C_1} I_1 + R_1 I_1' + L I_1'' + \frac{1}{C_2}(I_1 - I_2)$$

$$\frac{1}{C_2}(I_1 - I_2) = R_2 I_2'$$

We recover equations equation (8) if we equate $I_1 = \mathbf{r}$, $I_2 = \mathbf{r}_{inh}$, $V' = \boldsymbol{C_0}$, ${C_1}^{-1} = (\mathbf{1} - \mathbf{H}^*)$, $R_1 = \mathbf{H_2}$, $L = \mathbf{L_2}$, ${C_2}^{-1} = \mathbf{K}$, $R_2 = \tau_I \mathbf{K}$. Thus, our LIF and inhibitory neurons form a network which behave equivalently as this circuit, where we equate firing rates as currents through each loop. The current of the outer loop is the firing rate of the LIF network neurons and behaves like a driven LCR circuit just as before. The current of the inner wire is $\mathbf{r}_{inh}$ and acts as a negative feedback to high currents. If the inner (inhibitory) capacitor is initially uncharged, then current will pass through the capacitor as it starts charging exponentially with a time constant τ_I. The current starts decaying exponentially with time constant τ_I from its initial max value. The capacitor momentarily acts like an open circuit on the timescale of τ_I. Very little current passes through the resistor R_2. As the capacitor becomes fully charged, it stops allowing current to flow through. Then a lot of the current must be passing through the resistor R_2.

Transfer Learning in Spiking Neural Networks

Neural networks trained by traditional backpropagation are well known to undergo *catastrophic forgetting* when applied to transfer learning problems. That is, when a neural network trained on some Task A is then trained sequentially on another Task B, it tends to completely forget how to perform Task A. While there has been some work in overcoming this catastrophe, it remains poorly understood. We aim to take a principled approach to solve this problem based upon the biological observation that biological organisms are seemingly able to perform transfer learning seamlessly. As such, we explore the use of the biological learning rule Spike-Timing Dependent Plasticity, instead of back propagation, to check if it also suffers from catastrophic forgetting, and if so, what additional biological mechanisms allow organisms to perform transfer learning. Here we show preliminary results.

Introduction

Catastrophic forgetting, or catastrophic interference, is the effect in which artificial neural networks are completely un-robust to new types of data. This is framed in the context of stability and plasticity. It is most desirable that some plasticity remain present in the network so that learning can continually occur while at the same time remaining stable so that it retains memory of what has already been learned. This ability to remember tasks while learning additional tasks can also be framed as *generalization*.

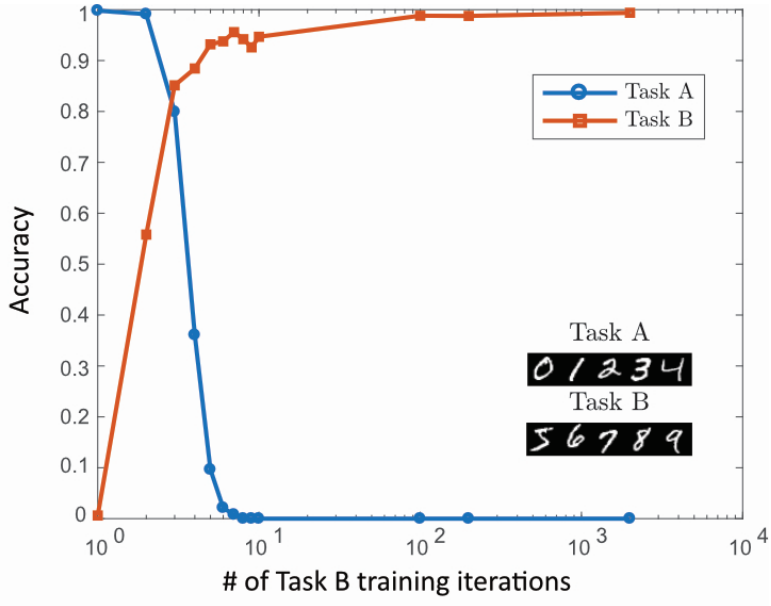


Figure 8

Catastrophic forgetting. A network forgets Task A while training for Task B, i.e., catastrophic forgetting.

Transfer learning in artificial neural networks was first formalized by McCloskey and Cohen (1989) and Ratcliff (1990) where it was demonstrated that back propagation, while able to achieve very high performance on tasks on which the network is trained, fails completely on transfer learning tasks. For example, Figure 8 shows a convolutional neural network first trained to classify MNSIT digits 0-4 (Task A) then trained sequentially on digits 5-9 (Task B) completely forgets how to classify 0-4. This CNN had 2 convolutional layers, each followed by a pooling layer, then 2 fully connected dense layers. In general, it is not uncommon for performance of artificial neural networks to drop to 0% on Task A after training on Task B. Some efforts have been made to overcome this catastrophic forgetting in back-propagation. Most notably using node sharpening (French 1991), novelty (Kortge 1990), pre-training (McRae and Hetherington 1993, French 1999), pseudo-recurrent networks (French 1997, Robins 1995), self-refreshing memory (Ans and Rousset 1997, Ans and Rousset 2000, Musca et al. 2009, Ans 2004), latent learning (Gutstein and Stump 2015), and elastic weight consolidation (Kirkpatrick et al. 2017). However, while these techniques have achieved some success, it is still not fully understood how the brain is able to achieve transfer learning so easily and what basic principles underlie this problem.

Specifically, no one has yet tried to solve the problem of catastrophic forgetting in spiking neural network which learn with a biological learning rule known as spike-timing dependent plasticity (STDP) shown in Figure 1. In fact, it is not even known if networks trained with STDP suffer catastrophic forgetting. Therefore, we are interested in exploring the problem of transfer learning in STDP networks.

To begin, we first need a network trained with STDP which is capable of performing object recognition. This is a difficult problem in-and-of itself but interest in developing such networks has increased in recent years [Masquelier and Thorpe 2007, Diehl and Cook 2015, Kheradpisheh et al. 2017, Sengupta et al. 2018, Bellec et al. 2019]. We chose to base our model off the work by Kheradpisheh et al. because the authors reported good performance and, more importantly, because

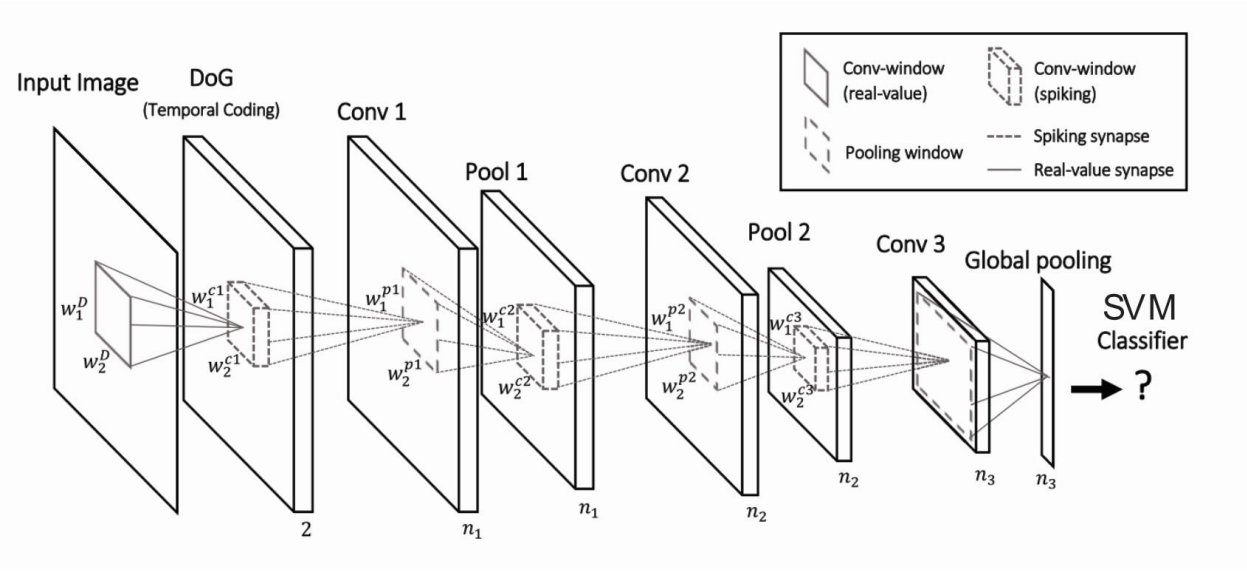


Figure 9

Proposed Spiking Artificial Neural Network. A sample architecture that was proposed by Kheradpisheh et al.

their network uses multiple layers (see Figure 9). We were attracted to the multiple layers because it makes sense to investigate the properties of STDP between two layers of spiking neurons. We unfortunately had a difficult time reproducing the results and ultimately decided to switch to a spiking neuron we build from scratch.

Approach

Integrate and Fire neurons and STDP

Spike-timing dependent plasticity is a Hebbian learning rule which can roughly be summarized as "neurons that fire together wire together" [Hebb 1949 , Lowel and Singer 1992]. In summary, if neuron A fires before neuron B, the weight from A to B should increase in accordance with the notion that A caused B to fire. Neuron A is called a pre-synaptic neuron and neuron B is called a post-synaptic neuron. Conversely, if A fires after B, the weight should decrease in accordance with the notion that A certainly did not contribute to B's activation. By "fire" we mean the neuron became activated (fired an action potential also called a spike) as a result of summing its inputs. This model of neuron is called an integrate-and-fire (IF) neuron. ReLu neurons in artificial neural networks are analogous to IF neurons. The equations governing an IF neuron is:

Equation (9)

$$V_B(t) = V_B(t-1) + \sum_A w_{BA} S_A(t-1)$$

where $S_A(t)$ is the spike train of neuron A indicating if A fired at time t or not. An example spike train is shown in Fig. 1. w_{BA} is the weight from neuron A to neuron B. $V_B(t)$ is the voltage of neuron B at time t. When the voltage crosses some threshold V_{th} at time t, then $V_B(t)$ is reset to $V_B(t) = 0$ and a spike is recorded as $S_B(t) = 1$.

As the name indicates, spike-timing dependent plasticity means the weight update depends on the timing of the spikes of neurons A and B. STDP is commonly modeled as a separable function of time but also of the current weight:

Equation (10)

$$\Delta w_{BA}(\Delta t; w_{BA}) = \begin{cases} a_+ f_+(w_{BA}) e^{-|\Delta t|/\tau_+}, & \Delta t > 0 \\ a_- f_-(w_{BA}) e^{-|\Delta t|/\tau_-}, & \Delta t < 0 \end{cases}$$

where $\Delta t = t_{post} - t_{pre} = t_B - t_A$. Note that a_- and a_+ specify the learning rates. The purpose of $f_+(w_{BA})$ and $f_-(w_{BA})$ is to implicitly set the bounds of the weights by changing the amplitude of Δw_{BA} as a function of w_{BA}. For example, a commonly used function is $f_+(w_{BA}) = f_-(w_{BA}) = w_{BA}(1 - w_{BA})$. When $w_{BA} = 0$ or 1, then $\Delta w_{BA} = 0$ thus eliminating the need to set hard boundary conditions like weight clipping. The exponential term introduces time dependence into the weight update rule. The magnitude of Δw_{BA} is exponential in the absolute value of the time difference between spikes. Thus, spikes that fire closely together contribute more to the weight update than spikes which fire further apart in time. For example, Fig. 1 shows a weight being updating in accordance with spikes and the weight changes more when the spikes are close together.

Kheradpisheh et al. Network

Different from regular fully connected or deep convolutional neural nets, in which the neurons transmit their activation levels within each other, in a spiking neural net (SNN) that was proposed by Kheradpisheh et al. neurons communicate by their spiking times. Each spiking neuron affects the surrounding connected neurons (post-synaptic neurons) according to the inter-synaptic weight (Equation 9). A post-synaptic neuron in a convolutional layer spikes only if the internal potential value reaches a specific threshold (i.e., integrate and fire).

The initial layer transforms the input image (e.g. MNIST images) into On and Off channels through the use of a difference of Gaussians (DoG) filter. This filter models the neural responses in retina which response to positive and negative contrast in the input images. As shown in Figure 10(a), filtering extracts information from the salient pixels of the input image. Note that, a higher output value of this operation (positive or negative) indicates a region of high spatial contrast. The input image is then converted to spikes times by analyzing the intensity of the On/Off channels. This is done by first discretely sorting ($t = 1, 2, 3, \dots, N$) the pixel intensities from greatest to least then converting high intensities to faster spikes times. For example, if neuron A and neuron B are the most and the second most activated neurons respectively, their spike times are set to $S_A(t) = 1$ and $S_B(t) = 2$. This is called rank-order coding. Thus, Kheradpisheh et al. throw out information about specific timing of spikes and only keep the relative (sorted) times. Consequentially, they use a simplified version of the STDP rule which throws of the exponential term (Equation 11). The following layers of the proposed architecture (Figure 9) consists of consecutive convolutional and pooling layers until the last global pooling layer. Similar to the regular convolutional nets, proposed deep SNN architecture contains multiple convolutional filters (neuronal maps), e.g., 32 filters of size 5x5x2, that share the same weights. Following Equation 9, each convolutional neuron's internal potential increases according to pre-synaptic neurons' synaptic weights. When the pre-specified threshold potential is exceeded, convolutional neuron fires. This spiking time is registered and then fed into the following pooling layer.

Notably, the authors propose a lateral inhibition mechanism which requires that only one "filter" neuron per location is activated (winner-take-all mechanism). This mechanism is believed to cause each neuron to be sensitive to different visual feature and to prevent duplicate convolutional filters (synaptic weights). This is biologically plausible as local inhibition is observed in the brain which serves a similar purpose. STDP is determined by the following method. Within each filter map (fix the filter and consider all neurons designated for that filter), the neuron that fires first updates

its pre-synaptic weights and all neurons in the filter map use these same set of weights. Note that this deviates from biological plausibility in that the STDP rule is no longer strictly a local function. The weights are updated according to the following simple STDP rule [15]

Equation (11)

$$\Delta w_{BA}(\Delta t; w_{BA}) = \begin{cases} a_+ w_{BA}(1 - w_{BA}), & \Delta t > 0 \\ a_- w_{BA}(1 - w_{BA}), & \Delta t < 0 \end{cases}$$

where A and B respectively represents the pre- and post-synaptic neurons. Unlike in Equation 10, in Equation 11 the weight adjustment does not depend on the explicit spike-time difference. Learning is done only on the convolutional layers and each layer is trained after the training for the previous layers are completed. Pooling layers are utilized to compress the information and achieve translational invariance of the learned features. Pooling neurons pass along the fastest spike time within the pooling window. No learning/synaptic weight updating occurs for these neurons. The last convolutional layer, instead of consisting of integrate and fire neurons which spike, consists of simply integrate neurons which accumulate voltages (no "fire" because there are no thresholds). These can be thought of as integrate and fire neurons with infinite thresholds. It is unclear why the authors chose to do this. Therefore, the last layer of the SNN performs global max pooling, passing forward the voltage of the neuron with the largest voltage. This yields one value for each learned feature in the last convolutional layer. In the training phase, for each labeled input image the output of global pooling layer is used to train an SVM classifier. Similarly, in the testing phase, the input image is classified by feeding the final voltage values of the global pooling layer into the trained SVM.

Kheradpisheh et al. uses two different models for different tasks. They use a 2 layer (2 convolutional layers) model for the smaller MNIST dataset and a larger 3 layer model for Caltech 101 and ETH-80 datasets. We chose to base our model off of the 2 layer version in order to use less computational power. Plus, for trying to analyze the effect of transfer learning, it is presumably easier

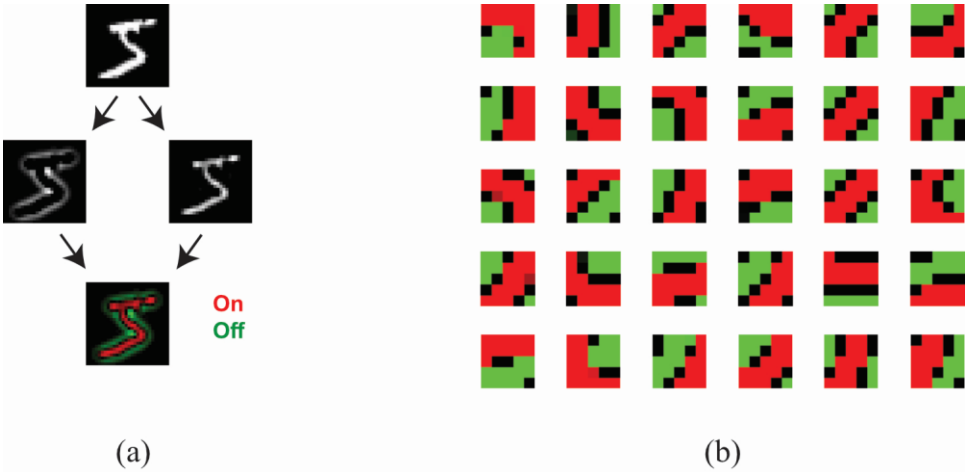


Figure 10

Example filters learned in the proposed SNN. (a) Example output after DoG filtering. On and off channels are colored red and blue respectively. (b) All 30 weights learned in convolutional layer 1.

to figure out what is going on with a 2 layer network than a 3 layer network. The authors provided code only for the 3 layer network thus we built the 2 layer network from scratch using the parameters given in the paper. We achieved very low performance using these parameters. We noticed that parameters in the code we had for the 3 layer network differed from what was reported in the paper. Based upon this realization, we changed some of the parameters in our 2 layers model and achieved reasonable performances.

Results

To test the transfer (sequential) learning capabilities of SNN's, we train three networks, namely, SNN_{FULL}, SNN_A, and SNN_B. As the training requires heavy computation time and fine tuning of the hyper-parameters (e.g. initial weights, learning rates, etc.) we reduce the number of classification categories within each task. For our first case study, Task A and Task B respectively contains MNIST digits of 1,2,3 and 4,5,6. The network called SNN_A was trained on Task A only. SNN_B is trained on Task A and sequentially on Task B (transfer learning). Finally, SNN_{FULL} is trained on both Task A and Task B in parallel (normal learning).

Figure 11 illustrates the performance of each network on Task A and Task B. Note that we are able to test the performance of a network on data which it was not trained with by training an SVM classifier on that data (using that data's output from the network). For example, weights in the SNN_A model are not trained on images from Task B. But we are able to feed Task B through the network, obtain outputs, and train an SVM classifier on these outputs. This will give us an idea how the SNN is actually transforming the data in a way which encodes the task. Figure 11 shows performance of these networks on MNIST dataset. Task A AND Task B refers to training the SVM classifier on outputs from the model for Task A and Task B. In this case, it is equivalent to training the SVM on MNIST digits 1-6 (regardless of how the weights within the network were trained), and simply separately the test performance into Task A and Task B categories. Therefore, averaging the

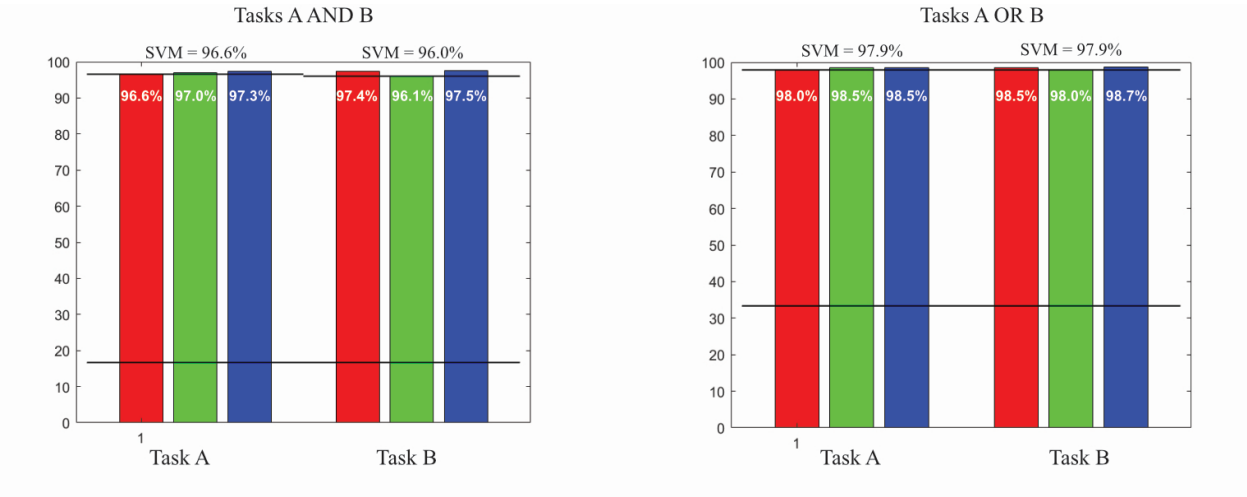


Figure 11

MNIST performance. Proposed network's performance on the MNIST data categories. Left: SVM classifier is trained on both Task A and Task B. Right: SVM classifier is trained on either Task A or Task B. Top lines and bottom lines indicate SVM's accuracy on raw pixels and the chance performance for the corresponding tasks. Red, green, and blue bars respectively depict the recognition accuracy of SNN_{FULL}, SNN_A, and SNN_B.

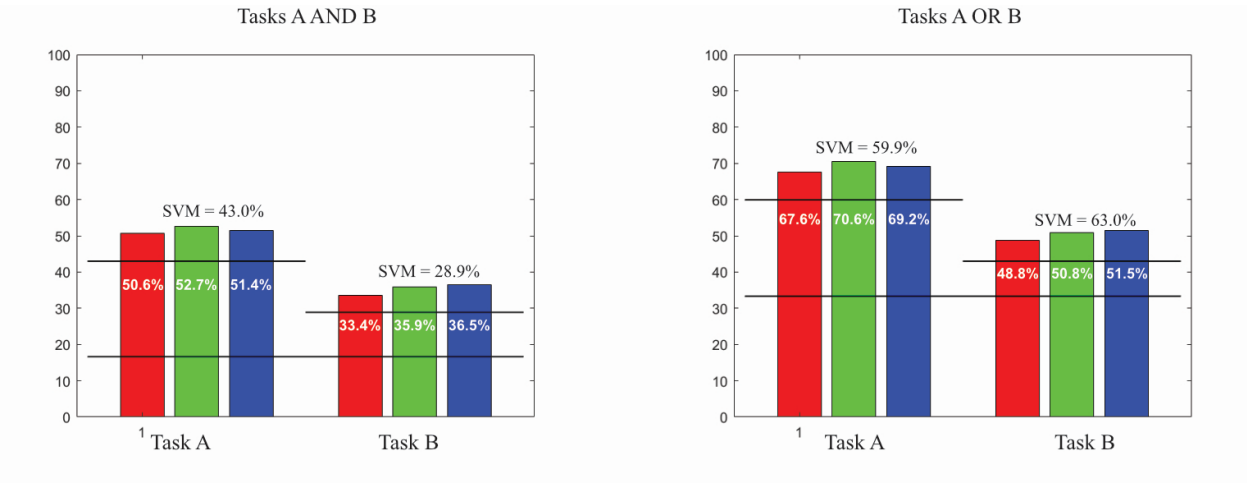


Figure 12

CIFAR performance. Proposed network's performance on the sub-sampled CIFAR data.

Task A and Task B performances is equivalent to determining the performance of the network on MNIST digits 1-6. Task A OR Task B refers to training the SVM classifier on the outputs from the model on either Task A or Task B. In this case, averaging the performances reported across Task A and Task B would not make sense as the SVMs used are different. The black solid lines at the bottom represent chance performance (guessing randomly). Note that this is 1/6 for Task A AND Task B and 1/3 for Task A OR Task B. The black solid lines on the top represents SVM accuracies of the tasks on raw pixel data and exact values are reported above the bars. For the MNIST case, the performance is consistently high and comparable to the values reported from Kheradpisheh et al. However, the performance is always slightly above that of the SVM's performance on the raw pixels. This is proof that MNIST is simply a very easy dataset. For example, an SVM classifier on the full MNIST dataset (digits 0-9) achieves 94.1% but is not shown here.

Figure 12 shows an analogous plot for our 2 layer SNN trained on CIFAR-10 dataset. Again, we sub-sampled the dataset to use 3 classes in Task A and Task B each. Task A = {airplane, automobile, bird} and Task B = {cat, deer, dog}. One interesting observation is that Task B seems objectively harder than Task A as indicted by the lower performance of both the SNN and the SVM on Task B compared to Task A. Maybe this is because Task B consists of all animals which share common features while Task A consists of both animals and man-made objects. The second interesting observation is that the SNN performs well even on tasks for which it was not trained. In particular, the performance of SNN_A on Task B is essentially the same as the performance of SNN_B on Task B. Both achieve performance slightly above that of the SVMs on raw pixels. This seems to imply that the weights of the SNN learned via STDP do not encode meaningful information about the data. It seems to imply that the network somehow transforms the data into a feature space which is more linearly separable than the raw pixel space. And that most of the network's performance is driven by the SVM classifier itself rather than STDP. Therefore, for future work, we will scratch this model and attempt to build another spiking neural network from scratch.

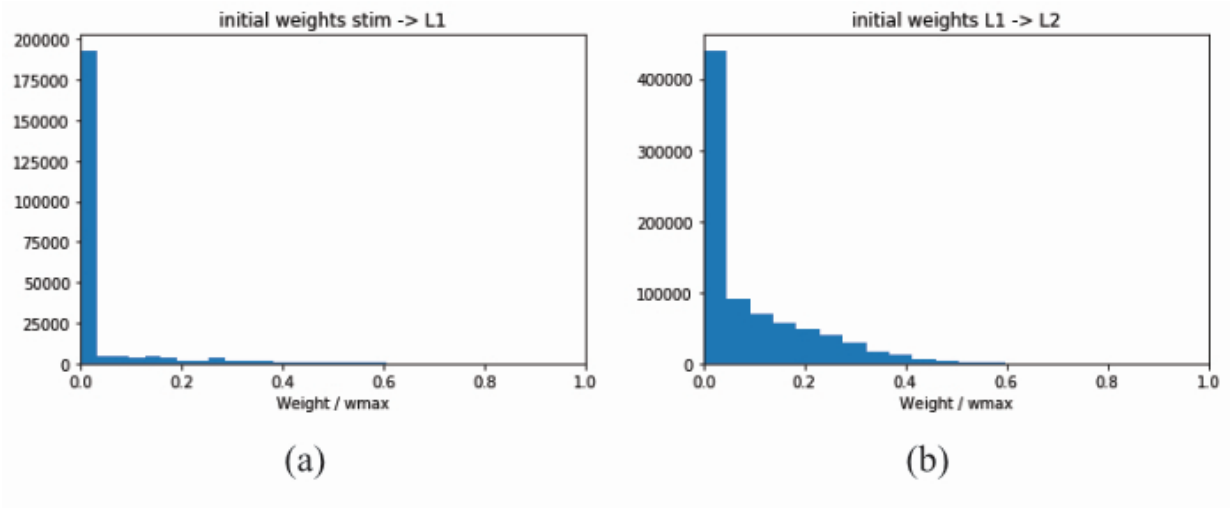


Figure 13

Distribution of weights. (a-b) Histogram that respectively represents the synaptic weights (neuronal maps) of the first and the second convolutional layer

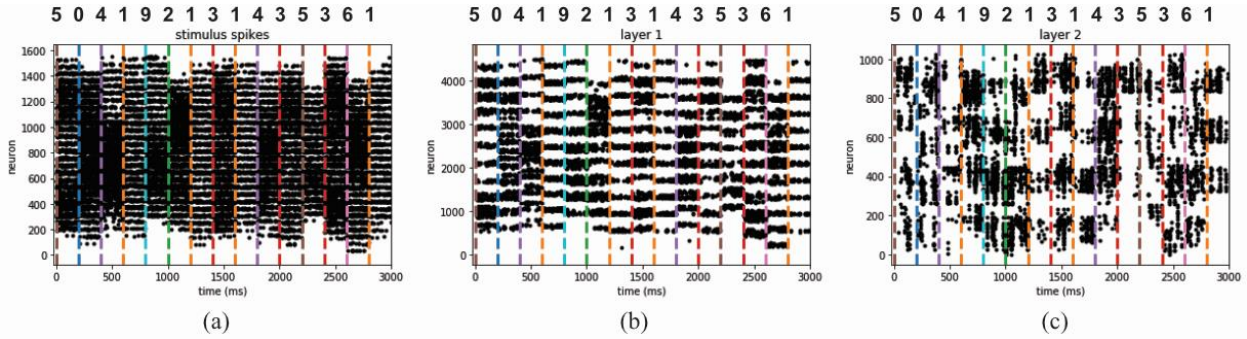


Figure 14

Spike trains in a biological inspired spiking neural network. (a) Input spike trains of MNIST digits. (b-c) Spike trains after convolutional layer 1-2. Each input MNIST digit is color coded by dashed lines, e.g., digit 1 is fed into the network at times marked by orange dashed lines.

Discussion

As a first attempt to build a spiking neural network, we learned how to use TensorFlow and also Brian, a python library for spiking networks. We thought it would be interesting to initialize the weights of our SNN with weights imported for a CNN build in TensorFlow. We built a 2 layer CNN without any pooling layers (instead using stride of 2 in the convolutional layers) and a single dense layer at the end for classification. We didn't include pooling because it would add slight complication for transferring the architecture to the SNN. Also we used Keras kernel constraint to make the weights for the convolutional layers non-negative, inspired by biological plausibility of excitatory neurons. The resulting weight histograms are shown in Figure 13. This CNN achieved a performance on the full MNIST dataset (digits 0-9) of 90.4%.

We imported these weights into Brian. Here we used a 2 layer "convolutional" spiking network with an SVM on top. But by convolutional layer, we mean that we initialize the weight matrix to be a "block" matrix in the sense that the topology is that of a convolution. However, each weight will be updated by the *local* STDP rule. Thus, after STDP learning, the weights will no longer be the same across the layer. So it is like a convolution with a changing kernel. We also use leaky integrate and fire neurons which are like the integrate and fire neurons but whose voltages also decay exponentially, returning to the reset potential in the absence of any inputs. We also tried using spike trains (multiple spikes per image) compared to one spike per image as used in the Kheradpisheh et al. model. The spike trains are shows in Figure 14 along with the resulting spike trains in convolutional layers 1 and 2. Each dotted line corresponded to when a different MNIST image is presented and the color of the line correspond the which digit was shown. The digits are labeled above the figures. As we can see, spikes indeed to propagate through the network. Using the TensorFlow weights we achieve a performance of 84.1%. MNIST dataset was converted into spike trains by

However, we did not have time to determine how much of the performance is due to the SVM layer (or dense layer in the TensorFlow case). This will be determined in future work. Furthermore, the Brian model did not undergo STDP as we simply used the TensorFlow weights. However, preliminary results show that STDP of the form in Eq. 10 with $f_+(w_{BA}) = f_-(w_{BA}) = w_{BA}(1 - w_{BA})$ will probably not be consistent with the weight extracted from TensorFlow. It is known that the weight distributions typical of these STDP equations do not resemble those observed from TensorFlow. We have observed that when STDP is implemented, the weight histograms begin to change in significant ways. However, we have not yet stabilized the dynamics of the network undergoing STDP. There are potentially other forms of STDP which use different functions $f_+(w_{BA})$ and $f_-(w_{BA})$ which may result in weight histograms which are more similar to those observed from TensorFlow (i.e. back propagation). These are all interesting questions to pursue moving forward.

Conclusion

In chapter 2, we demonstrated how different plasticity rules such as classical and reverse spike-timing dependent plasticity (cSTDP and rSTDP) can robustly give rise to complex circuits. In the simplified cortical model considered during chapter 2 (Figure 1), there are three nodes representing populations of neurons. The connectivity pattern between these three nodes is already fairly complicated, including both a "feed-forward loop" from L4 to L2/3 to L5/6 and a strong "feed-back loop" between L2/3 and L5/6. The term "feed-forward loop" is somewhat paradoxical but we simply mean a loop with a clear directionality and hierarchy starting at L4 and ending at L5/6. The strong feed-back loop may be cause of concern to generate runaway excitation. In our model, this is prevented by strong and quick inhibition. The fact that the L2/3 $\leftrightarrow$ L5/6 loop is nested inside of the larger L4 $\rightarrow$ L2/3 $\rightarrow$ L5/6 $\rightarrow$ L4 loop causes complicated circuitry which make analytical analysis difficult. Thus, although simulations show that simple plasticity rule can robustly generate such interesting and complicated, understanding these circuits is still a big challenge. However, the mathematical analysis presented in chapter 3 may help guide new ways of understanding these circuits. In particular, it suggests that it may be useful to shift from think from weights and firing rates as our primary variables to capacitance, charge, and voltage.

For example, the main insight in the analysis was to apply Kirchhoff's law which is a consequence of the conservation of charge. Since firing rate is the derivative of charge, having a conservation law for the integral of the firing rate may prove quite useful. It almost seems as if we

may want to view neuroscience as the study of conservation of charge in the brain and how simply moving charge back and forth creates the vastly diverse set of attributes we assign to the brain. After all, net charge is essentially constant in the brain.

Supplementary Material

This section contains supplementary figures and tables for chapter 2

.

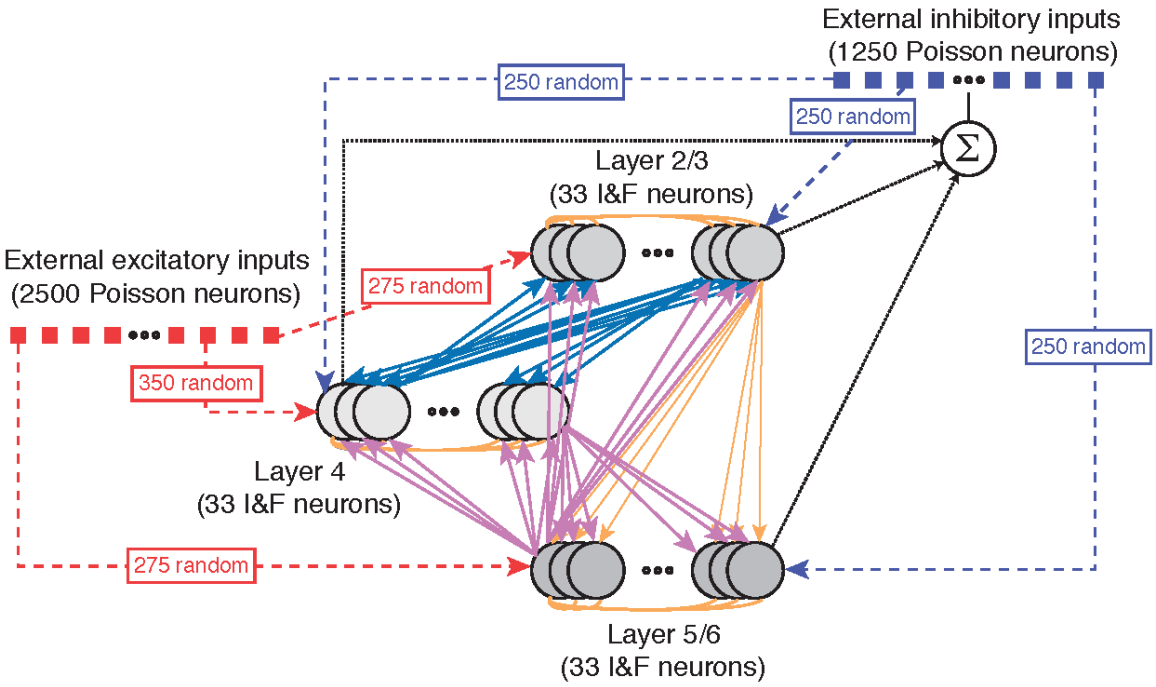


Figure S1

Schematic illustration of model architecture. The model consists of 3 layers, each one with 33 neurons, plus external excitatory inputs (red squares) and external inhibitory inputs (blue squares). Neurons are initially connected in an all-to-all fashion, only some of the representative connections are rendered here for pictorial clarity. The color of the connections corresponds to the colors and proposed learning rules in Fig. 3.

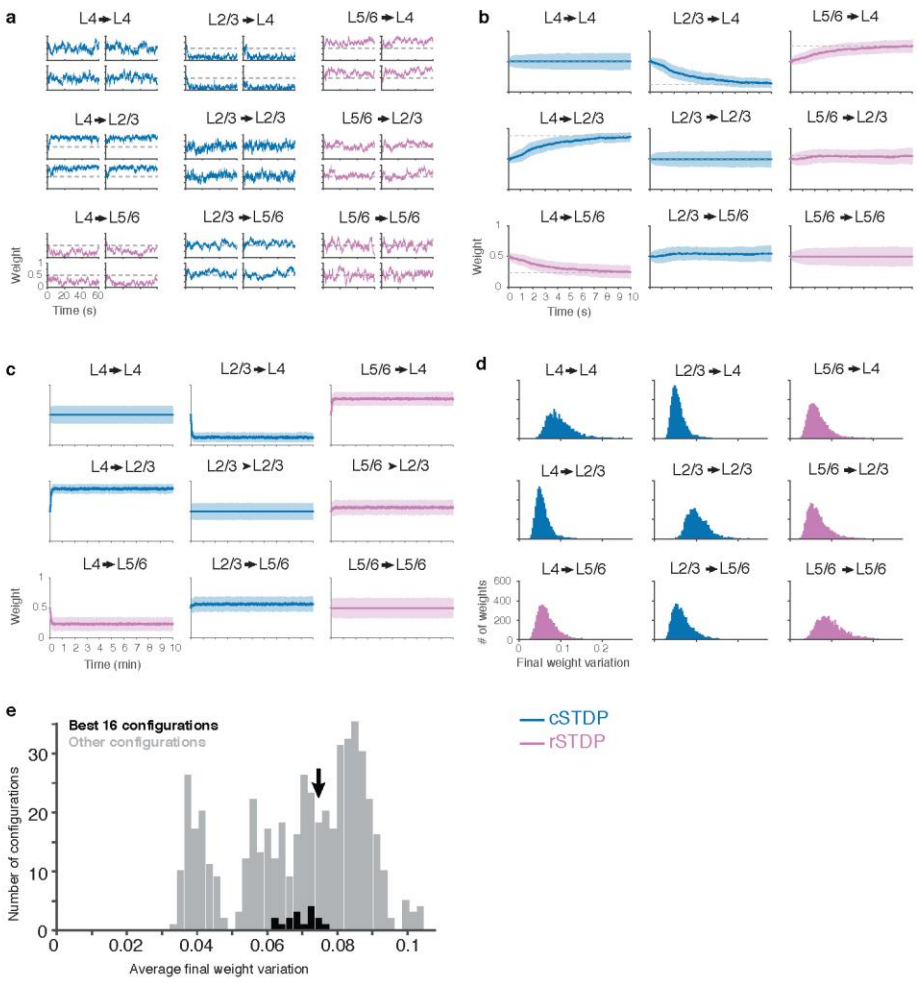


Figure S2

Convergence of simulations. a, Example dynamics of individual weights from the configuration in Fig. 2a. For each pair of layers, the plots follow 4 random example weights over the 60 seconds of simulation. The dashed lines indicate the initial conditions. **b,** Dynamics during the first 10 seconds, showing the average of all weights for each pair of layers from a single simulation and for the same configuration as in (**a**). The shaded areas denote 1 SD and $n = 5{,}445$. **c,** Dynamics during 10 minutes, showing the average of all weights for each pair of layers from a single simulation and for the same configuration as in (**a**). The shaded areas denote 1 SD and $n = 5{,}445$. **d,** Histograms showing distribution of final weight variation (standard deviation of the weights over the last 5 seconds of the simulation) for the same configuration in (**a**) and across 5 simulations ($n = 5{,}445$). **e,** Average of final weight variation for each of the 512 configurations. The best 16 configurations are highlighted in black and the example from (**a-c**) is labeled by an arrow.

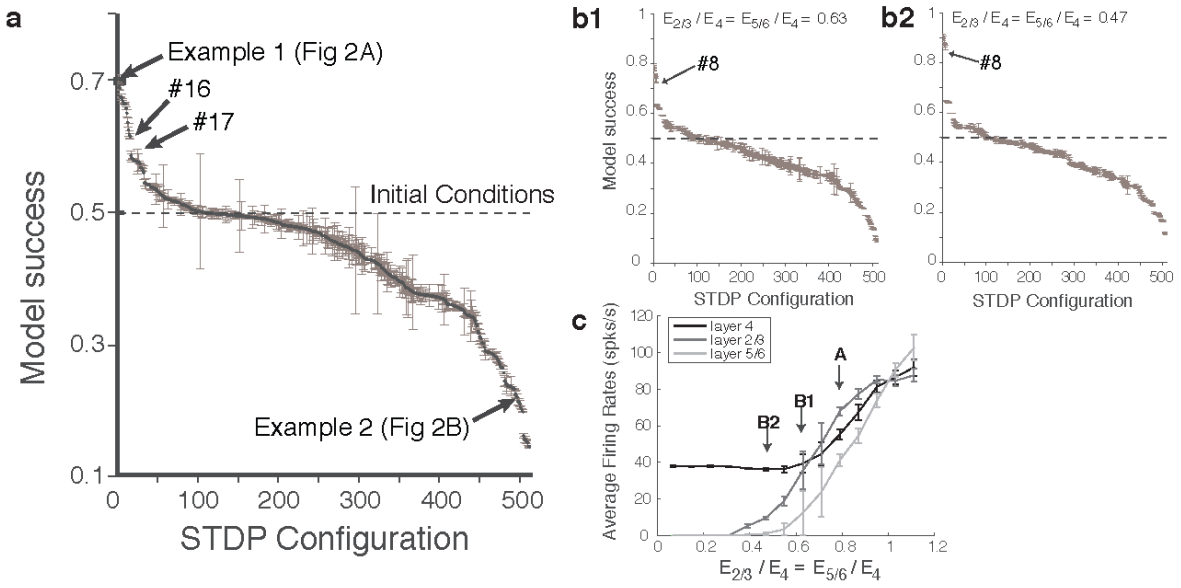


Figure S3

Model success for all possible configurations. a, The y-axis shows the model success (defined in the text), 0.5 is the success of the initial conditions (horizontal dashed line). Model success is averaged over 5 simulations. Error bars denote 1 SD. Example 1 is the configuration shown in Fig. 2a and Example 2 is the configuration shown in Fig. 2b. Note the gap between configuration number 16 and configuration number 17, as well as the gap before the bottom 8 simulations. **b,** Model success (mean $\pm$ 1 SD with $n = 5$), for all possible configurations with $E_{2/3}/E_4 = E_{5/6}/E_4 = 0.63$ (left), 0.47 (right). In (**a**), $E_{2/3}/E_4 = E_{5/6}/E_4 = 0.79$. Note that as the excitatory input ratio decreases, a large gap emerges between configuration 8 and 9 and the gap between 16 and 17 grows. **c,** The average firing of the top 16 (averaged over 5 simulations as well as the 16 configurations) separated by layer as a function of $E_{2/3}/E_4 = E_{5/6}/E_4$. Firing rates are averaged over the last 10 seconds of simulation time. Note that although $E_{2/3}/E_4 = E_{5/6}/E_4 = 0.47$ shows a more defined "top 16", the simulations result in a network with unrealistically low firing rates in layer 5/6.

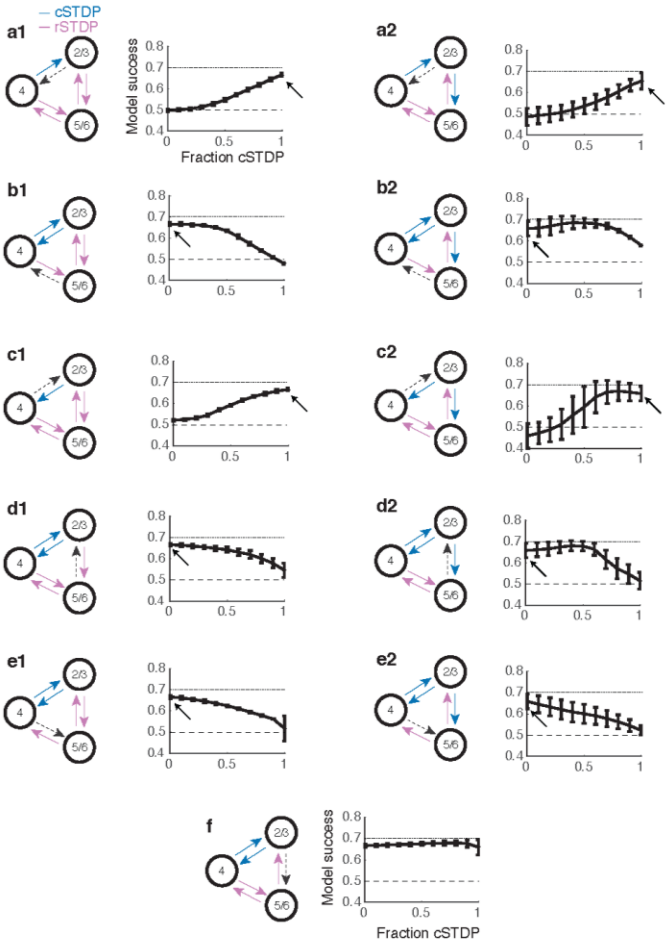


Figure S4

Performance of hybrid models combining cSTDP and rSTDP. Following the procedure illustrated in Fig. 4a-b, one of the connections is allowed to have a mixture of cSTDP and rSTDP (dashed arrow in model scheme) while all the other connections keep the configuration in Fig. 3 (fraction of cSTDP = 0 indicates all weights follow rSTDP and fraction = 1 indicates that all weights follow cSTDP). The y axis shows the model success, averaged over 5 simulations and across within-layer connections (8 possible configurations) for a total of $n = 40$; error bars denote 1 SD. The horizontal dashed line shows the initial conditions (success = 0.5) and the dotted lines shows the success of the best configuration. The arrow indicates the configuration in Fig. 3. The left column shows models where the connection from L2/3 to L5/6 has rSTDP and the right column shows models with cSTDP for that connection. Part (**d1**) is identical to Fig. 4a and part (**f**) is identical to Fig. 4b, and they are reproduced here for completeness.

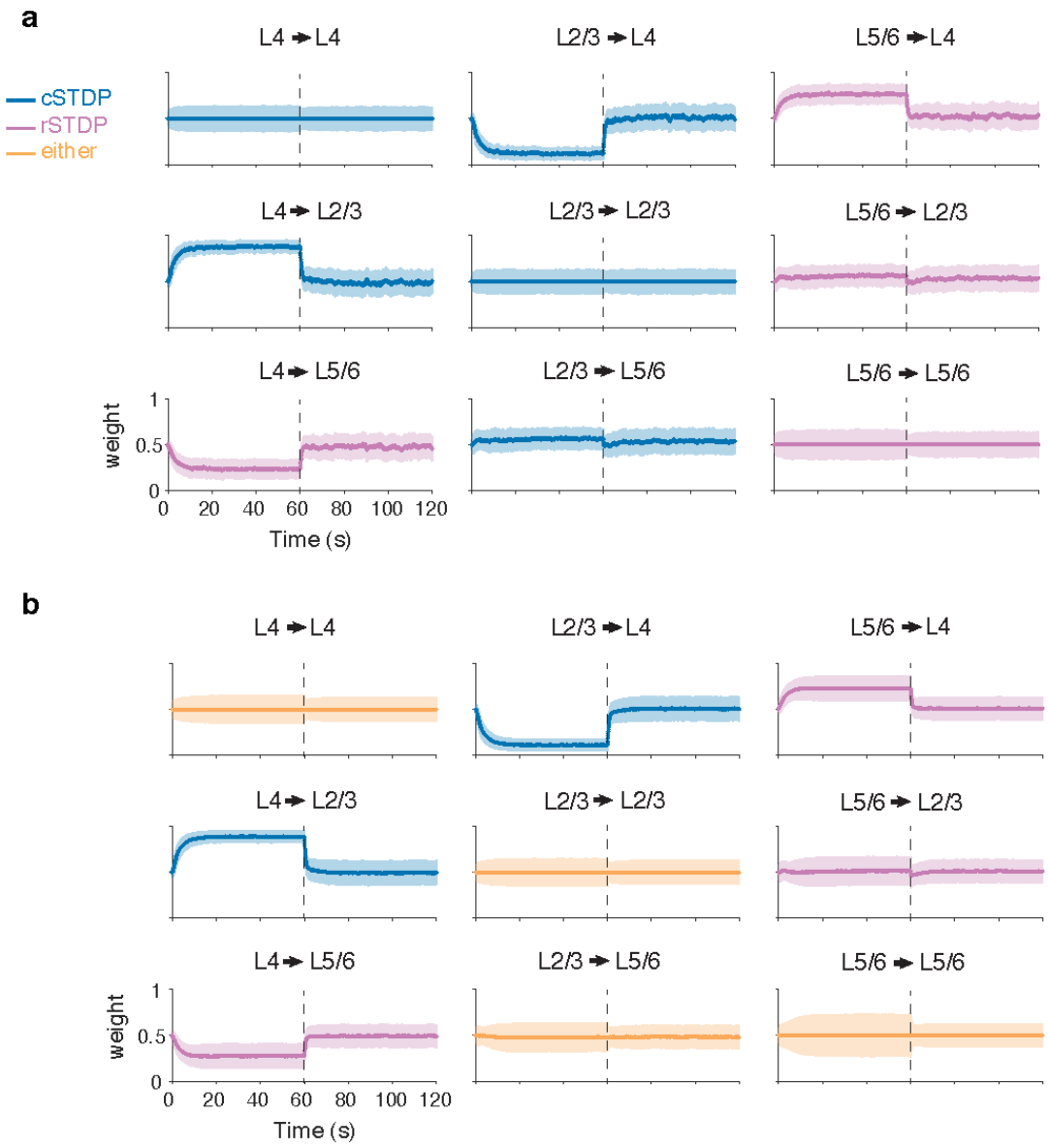


Figure S5

Weight dynamics when external input is switched to being equal for all layers. After 60 seconds of simulation (dashed line), the amount of external input is changed from the default values to $E_4 = E_{2/3} = E_{5/6} = 350$. Shown are the average of all weights for each pair of layers. . Error bars denote 1 SD. **a,** Example dynamics of weights from the configuration in Fig. 2a ($n = 5{,}445$). **b,** Example dynamics of weights for the best 16 configurations ($n = 87{,}120$).

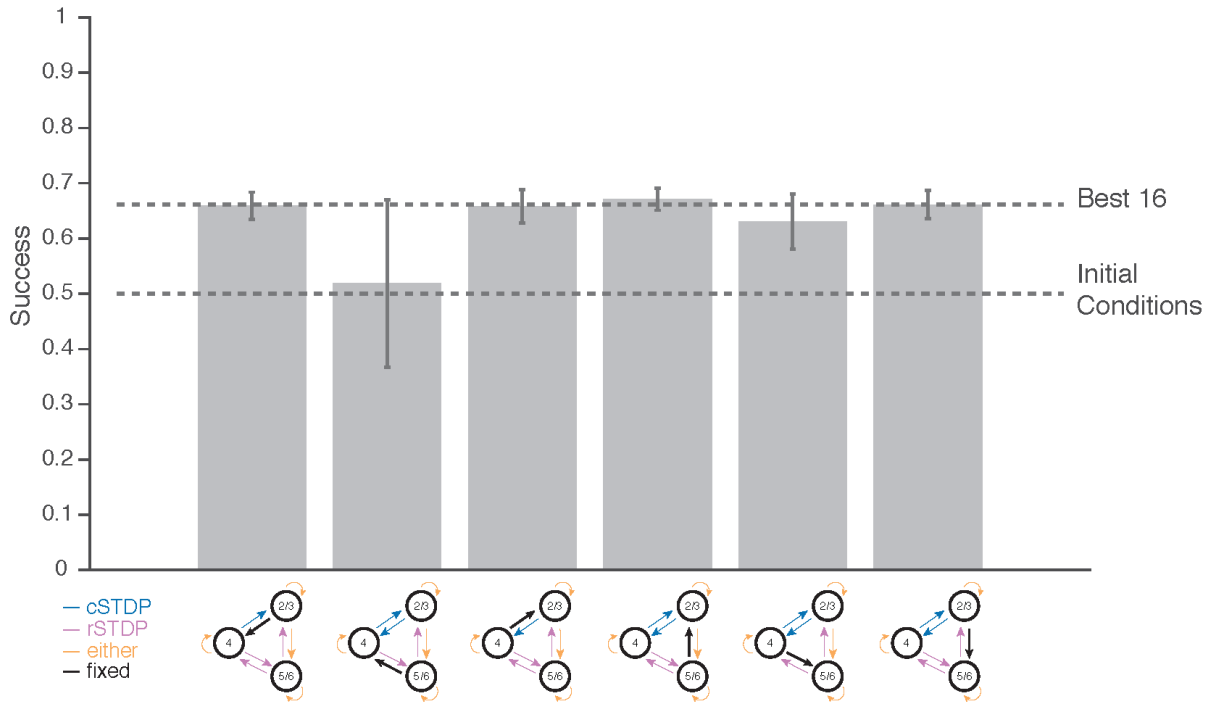


Figure S6

Success of best 16 configurations when one inter-laminar connection develops first. In these simulations, one of the inter-laminar connections (shown in black) is fixed from the beginning to the weight values corresponding to the value reported in Fig. 3c (final averages for the 16 best configurations). All the remaining connections are initialized and undergo STDP as in the default simulations. Error bars denote 1 SD ($n = 80$).

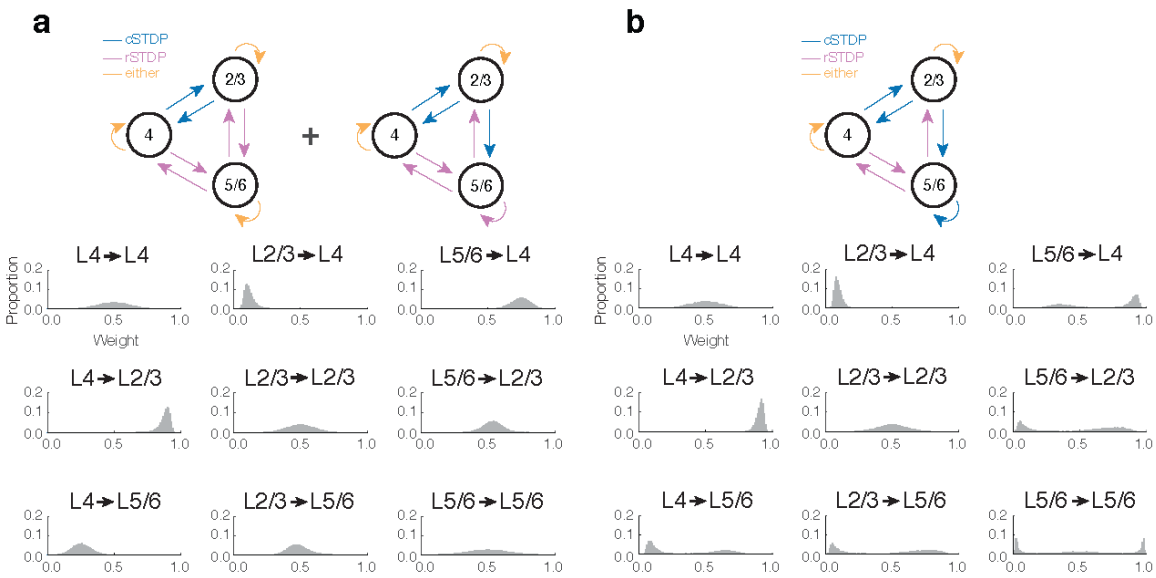


Figure S7

Multimodality within the best 16 configurations. a, Weight histograms from the 12 best configurations that display unimodal weight distributions, pooled across 5 simulations for a total $n = 60$. **b,** Weight histograms from the 4 best configurations that display multi-modal weight distributions in connections into and out of layer 5/6 ($n = 20$).

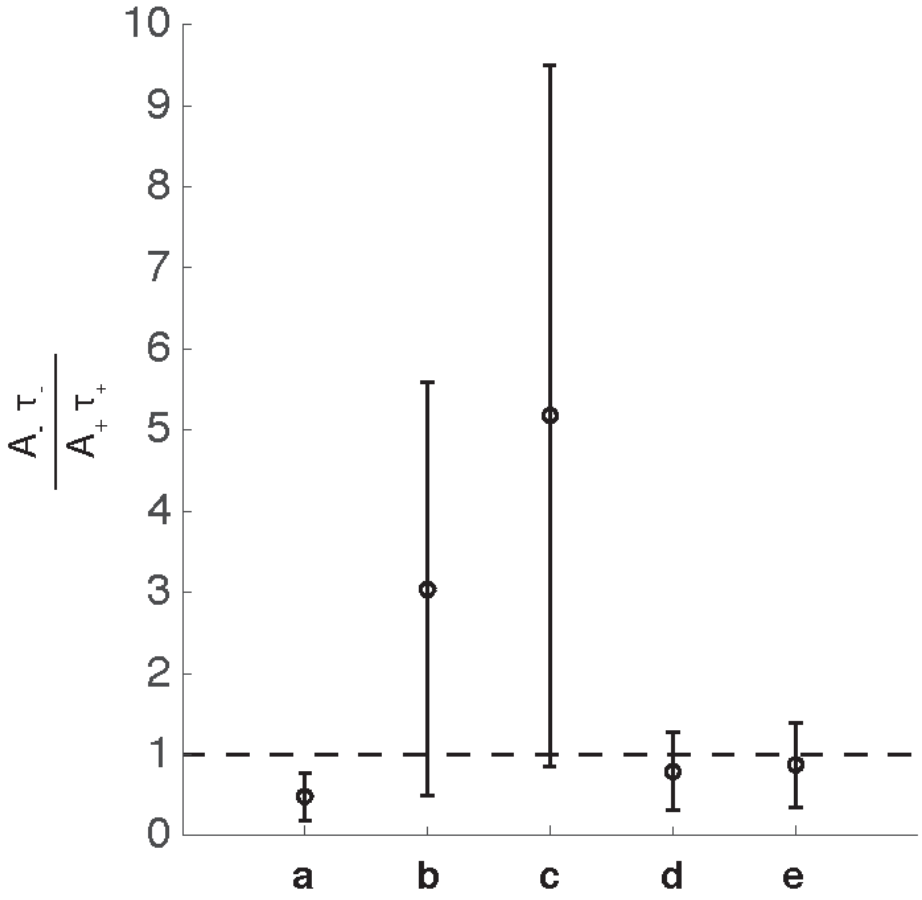


Figure S8

Experimentally estimated balance between potentiation and depression. **Experimentally estimated balance between potentiation and depression. a,** Data from proximal synapses along the apical dendrite of layer 2/3 pyramidal neurons in rat visual cortex (Froemke et al. 2005). **b,** Data from distal synapses along the apical dendrite of layer 2/3 (Froemke et al. 2005). **c,** Data from vertical inputs to layer 2/3 pyramidal neurons of rat S1 (Feldman 2000). **d,** Data from glutamatergic synapses from dissociated rat hippocampal neurons (Bi and Poo 1998). **e,** Data of retinal neuron's synapses onto optic tectum neurons in *Xenopus* tadpoles (Zhang et al. 1998). Note about the calculations. We extracted the change in synaptic strength values as a function of time between spikes from the corresponding figures. We estimated A and τ by fitting the curves with an exponential function (Matlab's "fit" function) and used the fitted values. The figure shows the average values and the error bars were calculated by error propagation.

Simulation Parameters	Description	Default value	Values explored
dt	Simulation time step	0.1 ms	
T	Length of simulation	60 sec	
STDP parameters	**Description**	**Default value**	**Values explored**
A_-	STDP amplitude for $\Delta t < 0$	$3.5 * 10^{-2}$	$[0.7, 7]*10^{-2}$
A_+	STDP amplitude for $\Delta t > 0$	$3.5 * 10^{-2}$	
τ_-	Time-constant for $\Delta t < 0$	20 ms	[4, 40] ms
τ_+	Time-constant for $\Delta t > 0$	20 ms	
μ	Governs additive vs multiplicative STDP	0.1	
w_{max}	Maximum weight value for inter-network	1	
w_{min}	Minimum weight value for inter-network	0	
w_{inh}	Fixed weight of inhibitory connections	1.5	
STDP_{mod}	Percent of connections for a given layer that are cSTDP	Either 0 or 1	[0, 1]
Neuron Parameters	**Description**	**Default value**	**Values explored**
τ_m	Decay time-constant for membrane potential	20 ms	
τ_{exc}	Decay time-constant for voltage potential at excitatory synapses	5 ms	
τ_{inh}	Decay time-constant for voltage potential at inhibitory synapses	5 ms	
V_{rest}	Resting membrane potential	-60 mV	
V_{thresh}	Threshold membrane potential	-54 mV	
V_{reset}	Membrane potential reset value following a action potential	-60 mV	
E_{exc}	Excitatory reversal potential	0 mV	
E_{inh}	Inhibitory reversal potential	-70 mV	
Network Parameters	**Description**	**Default value**	**Values explored**
$N_{neurons}$	Number of neurons in the network	99	
N_{Exc}	Number of extra-network excitatory homogenous Poisson neurons	2500	
E_4	Number of extra-network excitatory Poisson neurons projecting to layer 4	350	
$E_{2/3}$	Number of extra-network excitatory Poisson neurons projecting to layer 2/3	275	[0, 389]
$E_{5/6}$	Number of extra-network excitatory Poisson neurons projecting to layer 5/6	275	[0, 389]
r_{exc}	Firing rate for excitatory Poisson neurons	20 Hz	
N_{Inh}	Number of fast inhibition non-homogenous Poisson neurons	1250	
I_{layer}	Number of fast inhibition Poisson neurons projecting to each layer	250	
τ_I	Firing rate time constant for inhibitory Poisson neurons	2 ms	
r_{inh}^{max}	Maximum firing rate for inhibitory Poisson neurons	1000 Hz	
r_{inh}^{min}	Minimum firing rate for inhibitory Poisson neurons	5 Hz	
D_{intra}	Synaptic delay between neurons within the same layer	0 ms	[0, 5] ms
D_{inter}	Synaptic delay between neurons in different layers	0 ms	[0, 5] ms

Table S1

Parameters used in the simulations. This table lists all the parameters used in the simulations, the corresponding default values and the range of values explored for some of them when evaluating robustness to parameter changes (see text for further details). The interval step used for varying parameters are: 0.175×10^{-2} for A_-; 1 ms for τ_-; 0.1 for STDP_{mod}; 28 for $E_{2/3}$ (for values 25 through 389); 28 for $E_{5/6}$ (for values 25 through 389); 1 ms for D_{intra} and D_{inter}.

	L2/3→L4	L5/6→L4	L4→L2/3	L5/6→L2/3	L4→L5/6	L2/3→L5/6	Success
T	0.00	1.00	1.00	1.00	0.00	1.00	
1	0.12 ± 0.03	0.76 ± 0.08	0.88 ± 0.03	0.56 ± 0.09	0.24 ± 0.08	0.56 ± 0.09	0.70 ± 0.01
2	0.15 ± 0.05	0.74 ± 0.08	0.85 ± 0.05	0.56 ± 0.09	0.26 ± 0.08	0.56 ± 0.09	0.69 ± 0.00
3	0.10 ± 0.02	0.77 ± 0.08	0.90 ± 0.02	0.53 ± 0.10	0.23 ± 0.08	0.53 ± 0.10	0.69 ± 0.00
4	0.10 ± 0.03	0.77 ± 0.05	0.90 ± 0.03	0.53 ± 0.06	0.23 ± 0.05	0.47 ± 0.06	0.68 ± 0.00
5	0.12 ± 0.04	0.74 ± 0.09	0.88 ± 0.04	0.52 ± 0.10	0.26 ± 0.09	0.52 ± 0.10	0.68 ± 0.01
6	0.10 ± 0.03	0.75 ± 0.05	0.90 ± 0.03	0.52 ± 0.07	0.25 ± 0.05	0.48 ± 0.07	0.67 ± 0.00
7	0.11 ± 0.03	0.75 ± 0.06	0.89 ± 0.03	0.56 ± 0.05	0.25 ± 0.06	0.44 ± 0.05	0.67 ± 0.00
8	0.12 ± 0.04	0.74 ± 0.06	0.88 ± 0.04	0.55 ± 0.06	0.26 ± 0.06	0.45 ± 0.06	0.67 ± 0.00
9	0.13 ± 0.04	0.74 ± 0.06	0.87 ± 0.04	0.53 ± 0.06	0.26 ± 0.06	0.47 ± 0.06	0.67 ± 0.00
10	0.12 ± 0.04	0.73 ± 0.06	0.88 ± 0.04	0.51 ± 0.07	0.27 ± 0.06	0.49 ± 0.07	0.67 ± 0.00
11	0.13 ± 0.05	0.72 ± 0.06	0.87 ± 0.05	0.55 ± 0.05	0.28 ± 0.06	0.45 ± 0.05	0.66 ± 0.00
12	0.15 ± 0.07	0.70 ± 0.06	0.85 ± 0.07	0.55 ± 0.07	0.30 ± 0.06	0.45 ± 0.07	0.65 ± 0.01
13	0.09 ± 0.03	0.68 ± 0.27	0.91 ± 0.03	0.47 ± 0.34	0.32 ± 0.27	0.47 ± 0.34	0.64 ± 0.01
14	0.10 ± 0.03	0.66 ± 0.27	0.90 ± 0.03	0.46 ± 0.34	0.34 ± 0.27	0.46 ± 0.34	0.63 ± 0.01
15	0.09 ± 0.03	0.67 ± 0.26	0.91 ± 0.03	0.44 ± 0.31	0.33 ± 0.26	0.44 ± 0.31	0.62 ± 0.01
16	0.08 ± 0.02	0.67 ± 0.28	0.92 ± 0.02	0.43 ± 0.34	0.33 ± 0.28	0.43 ± 0.34	0.62 ± 0.00
…							
241	0.57 ± 0.08	0.42 ± 0.08	0.43 ± 0.08	0.44 ± 0.13	0.42 ± 0.08	0.56 ± 0.13	0.47 ± 0.00
242	0.49 ± 0.18	0.51 ± 0.28	0.49 ± 0.18	0.43 ± 0.28	0.51 ± 0.28	0.43 ± 0.28	0.47 ± 0.03
243	0.48 ± 0.11	0.53 ± 0.07	0.48 ± 0.11	0.40 ± 0.18	0.47 ± 0.07	0.40 ± 0.18	0.47 ± 0.02
244	0.42 ± 0.07	0.55 ± 0.06	0.42 ± 0.07	0.38 ± 0.15	0.45 ± 0.06	0.38 ± 0.15	0.47 ± 0.01
245	0.56 ± 0.06	0.44 ± 0.06	0.44 ± 0.06	0.47 ± 0.16	0.44 ± 0.06	0.47 ± 0.16	0.47 ± 0.01
246	0.46 ± 0.07	0.57 ± 0.13	0.46 ± 0.07	0.42 ± 0.12	0.57 ± 0.13	0.42 ± 0.12	0.47 ± 0.01
247	0.54 ± 0.06	0.51 ± 0.05	0.46 ± 0.06	0.44 ± 0.13	0.49 ± 0.05	0.44 ± 0.13	0.47 ± 0.01
248	0.50 ± 0.09	0.52 ± 0.05	0.50 ± 0.09	0.40 ± 0.17	0.48 ± 0.05	0.40 ± 0.17	0.47 ± 0.01
249	0.58 ± 0.07	0.42 ± 0.07	0.42 ± 0.07	0.50 ± 0.10	0.42 ± 0.07	0.50 ± 0.10	0.47 ± 0.00
250	0.58 ± 0.06	0.42 ± 0.07	0.42 ± 0.06	0.56 ± 0.12	0.42 ± 0.07	0.44 ± 0.12	0.47 ± 0.00
251	0.46 ± 0.08	0.52 ± 0.11	0.46 ± 0.08	0.40 ± 0.18	0.48 ± 0.11	0.40 ± 0.18	0.47 ± 0.01
252	0.57 ± 0.13	0.47 ± 0.07	0.57 ± 0.13	0.42 ± 0.13	0.47 ± 0.07	0.42 ± 0.13	0.47 ± 0.00
253	0.57 ± 0.05	0.52 ± 0.12	0.43 ± 0.05	0.64 ± 0.20	0.52 ± 0.12	0.36 ± 0.20	0.47 ± 0.00
254	0.54 ± 0.05	0.46 ± 0.11	0.46 ± 0.05	0.62 ± 0.19	0.54 ± 0.11	0.38 ± 0.19	0.47 ± 0.01
255	0.59 ± 0.15	0.46 ± 0.06	0.59 ± 0.15	0.41 ± 0.12	0.46 ± 0.06	0.41 ± 0.12	0.47 ± 0.01
256	0.54 ± 0.06	0.35 ± 0.25	0.46 ± 0.06	0.70 ± 0.23	0.35 ± 0.25	0.30 ± 0.23	0.47 ± 0.01
…							
497	0.69 ± 0.06	0.09 ± 0.06	0.31 ± 0.06	0.27 ± 0.09	0.91 ± 0.06	0.27 ± 0.09	0.22 ± 0.01
498	0.01 ± 0.01	0.06 ± 0.08	0.01 ± 0.01	0.99 ± 0.01	0.94 ± 0.08	0.01 ± 0.01	0.21 ± 0.01
499	0.71 ± 0.06	0.10 ± 0.08	0.29 ± 0.06	0.26 ± 0.08	0.90 ± 0.08	0.26 ± 0.08	0.21 ± 0.01
500	0.01 ± 0.00	0.05 ± 0.09	0.01 ± 0.00	0.99 ± 0.01	0.95 ± 0.09	0.01 ± 0.01	0.21 ± 0.01
501	0.01 ± 0.00	0.04 ± 0.02	0.01 ± 0.00	0.99 ± 0.00	0.96 ± 0.02	0.01 ± 0.00	0.20 ± 0.00
502	0.01 ± 0.00	0.03 ± 0.02	0.01 ± 0.00	0.98 ± 0.01	0.97 ± 0.02	0.02 ± 0.01	0.20 ± 0.00
503	0.01 ± 0.00	0.03 ± 0.01	0.01 ± 0.00	0.99 ± 0.00	0.97 ± 0.01	0.01 ± 0.00	0.20 ± 0.00
504	0.01 ± 0.00	0.02 ± 0.01	0.01 ± 0.00	0.99 ± 0.01	0.98 ± 0.01	0.01 ± 0.01	0.20 ± 0.00
505	0.07 ± 0.04	0.09 ± 0.11	0.07 ± 0.04	0.08 ± 0.05	0.91 ± 0.11	0.08 ± 0.05	0.16 ± 0.00
506	0.05 ± 0.03	0.05 ± 0.06	0.05 ± 0.03	0.11 ± 0.06	0.95 ± 0.06	0.11 ± 0.06	0.15 ± 0.00
507	0.07 ± 0.04	0.08 ± 0.08	0.07 ± 0.04	0.08 ± 0.04	0.92 ± 0.08	0.08 ± 0.04	0.15 ± 0.00
508	0.05 ± 0.03	0.05 ± 0.05	0.05 ± 0.03	0.10 ± 0.06	0.95 ± 0.05	0.10 ± 0.06	0.15 ± 0.00
509	0.04 ± 0.02	0.04 ± 0.02	0.04 ± 0.02	0.12 ± 0.06	0.96 ± 0.02	0.12 ± 0.06	0.15 ± 0.00
510	0.04 ± 0.02	0.04 ± 0.02	0.04 ± 0.02	0.11 ± 0.07	0.96 ± 0.02	0.11 ± 0.07	0.15 ± 0.00
511	0.06 ± 0.03	0.04 ± 0.02	0.06 ± 0.03	0.08 ± 0.04	0.96 ± 0.02	0.08 ± 0.04	0.14 ± 0.00
512	0.06 ± 0.03	0.05 ± 0.02	0.06 ± 0.03	0.08 ± 0.04	0.95 ± 0.02	0.08 ± 0.04	0.14 ± 0.00

Table S2

Weights and success metric for best, middle, and worst 16 configurations. The first column indicates the configuration number from 1 through 512, ranked based on the success metric (the first row labeled T depicts the target values). Each successive column indicates one of the 6 between layer connection types (described at the top). Average weights +/- 1 SD are shown for each configuration and connection type (averaged across all neurons between the pair of layers and across 5 simulations, $n = 33 \times 33 \times 5 = 5{,}445$). Colors correspond to cSTDP (blue) or rSTDP (pink). Note the high degree of consistency in the learning rules for 5 of the 6 between-layer connections for the best 16 and worst 16 configurations.

List of Useful Mathematical Formulas

This section contains mathematical formulas that were either derived or used for chapter 3

Define the Heaviside step function

$$\Theta(t) = \begin{cases} 1 & if\ t > 0 \\ 0 & if\ t < 0 \end{cases}$$

Define the (right-sided) exponential decay function

$$\phi_\tau(t) = \frac{1}{\tau} e^{-t/\tau}$$

Define the convolution $\circledast$

$$(\Theta f \circledast g)(t) = \int_{-\infty}^{\infty} \Theta(t-s) f(t-s) g(s) ds = \int_{-\infty}^{\infty} \Theta(u) f(u) g(t-u) \Theta(u) du$$

Define the Fourier Transform

$$\mathcal{F}\{f\}(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt$$

$$\mathcal{F}^{-1}\{f\}(t) = (2\pi)^{-1} \int_{-\infty}^{\infty} e^{i\omega t} f(\omega)\, d\omega$$

Properties of the (right-sided) exponential decay function

$$(\Theta\phi_\tau \circledast 1) = 1$$

$$(\Theta\phi_\tau \circledast \Theta)(t) = \Theta(\mathrm{t})\left(1 - e^{-t/\tau}\right)$$

$$\left(\Theta\phi_{\tau_1} \circledast \Theta\phi_{\tau_2} \circledast \Theta\right) = \Theta(\mathrm{t})\left(1 - \frac{\tau_1 e^{-\frac{t}{\tau_1}} - \tau_2 e^{-\frac{t}{\tau_2}}}{\tau_1 - \tau_2}\right)$$

$$(\Theta\phi_\tau \circledast \Theta\phi_\lambda) = \Theta(\mathrm{t})\frac{e^{-t/\tau} - e^{-t/\lambda}}{\tau - \lambda} = \Theta(\mathrm{t})\frac{\tau\phi_\tau(t) - \lambda\phi_\lambda(t)}{\tau - \lambda}$$

$$\frac{d}{dt}(\Theta\phi_\tau \circledast v) = \frac{d}{dt}\int_{-\infty}^{t}\frac{1}{\tau}e^{-(t-s)/\tau}v(s)ds = \frac{1}{\tau}\left(v(t) - (\Theta\phi_\tau \circledast v)\right)$$

$$\frac{d}{dt}(\Theta\phi_\tau \circledast \Theta\phi_{\lambda\neq\tau} \circledast v) = \frac{(\Theta\phi_\lambda(t) \circledast v) - (\Theta\phi_\tau(t) \circledast v)}{\tau - \lambda}$$

$$\int_0^t (\Theta\phi_\tau \circledast \Theta)(s)ds = \Theta(\mathrm{t})\left(t - \tau\left(1 - e^{-\frac{t}{\tau}}\right)\right) = \Theta(\mathrm{t})\left(t - \tau(\Theta\phi_\tau \circledast \Theta)(t)\right) \approx \frac{1}{2}\frac{t^2}{\tau}\Theta(\mathrm{t})$$

$$\int_0^t \left(\Theta\phi_{\tau_1} \circledast \Theta\phi_{\tau_2} \circledast \Theta\right)(s)ds \approx \frac{1}{6}\frac{t^3}{\tau_1\tau_2}\Theta(\mathrm{t})$$

$$\left(\Theta\phi_{\tau_{exc}} \circledast \Theta t\right) = \Theta(\mathrm{t})\left(t + \tau_{exc}\left(e^{-\frac{t}{\tau_{exc}}} - 1\right)\right) \approx (\mathrm{t} - \tau_{exc})\Theta(\mathrm{t} - \tau_{exc})$$

$$\left(\Theta\phi_{\tau_{inh}} \circledast \Theta\phi_{\tau_I} \circledast \Theta t\right) = \Theta(\mathrm{t})\frac{e^{-\frac{t}{\tau_{inh}}} - e^{-\frac{t}{\tau_I}}}{\tau_{inh} - \tau_I} \circledast \Theta t$$

$$= \frac{1}{\tau_{inh} - \tau_I}\left[\tau_{inh}\Theta\phi_{\tau_{inh}} - \tau_I\Theta\phi_{\tau_I}\right] \circledast \Theta t$$

$$= \Theta t + \frac{\Theta(t)}{\tau_{inh} - \tau_I}\left(\tau_{inh}^2\left(e^{-\frac{t}{\tau_{inh}}} - 1\right) - \tau_I^2\left(e^{-\frac{t}{\tau_I}} - 1\right)\right)$$

$$\approx (\mathrm{t} - \tau_{exc} - \tau_I)\,\Theta(\mathrm{t} - \tau_{exc} - \tau_I)$$

Properties of the Fourier transform

$$\mathcal{F}\{f \circledast g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

$$\mathcal{F}\{f'\} = i\omega\mathcal{F}\{f\}$$

$$\mathcal{F}\{tf\} = i\mathcal{F}\{f'\}$$

$$\mathcal{F}\left\{\int_{-\infty}^{t} f(s)ds\right\} = \frac{\mathcal{F}\{f\}}{i\omega}$$

$$\mathcal{F}\{f(t-s)\} = e^{-is\omega}\mathcal{F}\{f\}$$

$$\mathcal{F}\left\{e^{is\omega}f\right\} = \mathcal{F}\{f\}(\omega - s)$$

$$\mathcal{F}\{\Theta\phi_{\tau}(t)\} = \frac{1}{1 + i\omega\tau}$$

References

Abbott LF, Nelson SB. 2000. Synaptic plasticity: Taming the beast. Nat Neurosci. 3:1178–1183.

Allendoerfer KL, Shatz CJ. 1994. The subplate, a transient neocortical structure: Its role in the development of connections between thalamus and cortex. Annu Rev Neurosci. 17:185–218.

Ans, B., Rousset, S. (1997) Avoiding catastrophic forgetting by coupling two reverberating neural networks. *CR Academie Science Paris, Life Sciences*, 320, 89-997.

Ans, B., Rousset, S. (2000) Neural networks with a self-refreshing memory: Knowledge transfer in sequential Learning tasks without catastrophic forgetting. *Connection Science*, 12, 1-19.

Ans, B. (2004) Sequential learning in distributed neural networks without catastrophic forgetting: A single and realistic self-refreshing memory can do it. *Neural Information Processing-Letters and Reviews*, 4, 27-32.

Babadi B, Abbott LF. 2013. Pairwise analysis can account for network structures arising from spike-timing dependent plasticity. PLoS Comput Biol. 9:e1002906.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., Maass, W. (2018) Long short-term memory and Learning-to-learn in networks of spiking neurons. arXiv *preprint arXiv*: 1803.09574.

Bennett JE, Bair W. 2015. Refinement and pattern formation in neural circuits by the interaction of traveling waves with spike-timing dependent plasticity. PLoS Comput Biol, 11:e1004422.

Binzegger T, Douglas RJ, Martin KA. 2004. A quantitative map of the circuit of cat primary visual cortex. Journal of Neuroscience, 24(39), 8441-8453.

Bi G, Poo M. 1998. Synaptic modifications in cultured hippocampal neurons: Dependence on spike

timing, synaptic strength, and postsynaptic cell type. J Neurosci. 18:10464–10472.

Bolz J, Castellani V, Mann F, Henke-Fahle S. 1996. Specification of layer-specific connections in the developing cortex. Prog Brain Res. 108:41–54.

Burbank KS, Kreiman G. 2012. Depression-biased reverse plasticity rule is required for stable learning at top-down connections. PLoS Comput Biol. 8:e1002393.

Butts DA, Kanold PO, Shatz CJ. 2007. A burst-based "hebbian" learning rule at retinogeniculate synapses links retinal waves to activity-dependent refinement. PLoS Biol. 5:e61.

Callaway EM. 1998a. Local circuits in primary visual cortex of the macaque monkey. Annu Rev Neurosci. 21:47–74.

Callaway EM. 1998b. Prenatal development of layer-specific local circuits in primary visual cortex of the macaque monkey. J Neurosci. 18:1505–1527.

Caporale N, Dan Y. 2008. Spike timing–dependent plasticity: A hebbian learning rule. Annu Rev Neurosci. 31:25–46.

Castellani V, Bolz J. 1997. Membrane-associated molecules regulate the formation of layer-specific cortical circuits. Proc Natl Acad Sci. 94:7030–7035.

Constantinople CM, Bruno RM. 2013. Deep cortical layers are activated directly by thalamus. Science. 340:1591–1594.

Debanne D, Gähwiler BH, Thompson SM. 1998. Long-term synaptic plasticity between pairs of individual ca3 pyramidal cells in rat hippocampal slice cultures. J Physiol. 507:237–247.

Diehl, P. U., Cook, M. (2015) Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9, 99.

Douglas RJ, Martin KA. 2004. Neuronal circuits of the neocortex. Annu Rev Neurosci. 27:419–451.

Egger V, Feldmeyer D, Sakmann B. 1999. Coincidence detection and changes of synaptic efficacy in spiny stellate neurons in rat barrel cortex. Nat Neurosci. 2:1098–1105.

Espinosa JS, Stryker MP. 2012. Development and plasticity of the primary visual cortex. Neuron.

75:230–249.

Feldman DE. 2000. Timing-based ltp and ltd at vertical inputs to layer ii/iii pyramidal cells in rat barrel cortex. Neuron. 27:45–56.

Feldman DE, Brecht M. 2005. Map plasticity in somatosensory cortex. Science. 310:810–815.

Felleman DJ, Van Essen DC. 1991. Distributed hierarchical processing in the primate cerebral cortex. Cereb Cortex. 1:1–47.

Fox K, Wong RO. 2005. A comparison of experience-dependent plasticity in the visual and somatosensory systems. Neuron. 48:465–477.

French, R. M. (1991) Using Semi-Distributed Representations to Overcome Catastrophic Forgetting in Connectioniost Networks. In *Proceedings of the 13th Annual Cognitive Science Society Conference* (pp. 173-178) New Jersey: Lawrence Erlbaum.

French, R. M. (1997) Pseudo-recurrent connectionist networks: an approach to the 'sensitivity-stability' dilemma. *Connection Science*, **9** (4), 353–379.

French, R. M. (1999) Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4), 128–135.

Froemke RC, Letzkus JJ, Kampa BM, Hang GB, Stuart GJ. 2010. Dendritic synapse location and neocortical spike-timing-dependent plasticity. Front Synaptic Neurosci. 2.

Froemke RC, Poo M, Dan Y. 2005. Spike-timing-dependent synaptic plasticity depends on dendritic location. Nature. 434:221–225.

Gilson M, Fukai T. 2011. Stability versus neuronal specialization for STDP: long-tail weight distributions solve the dilemma. PloS One. 6:e25339.

Gutstein, Stump (2015) Reduction Of Catastrophic Forgetting With Transfer Learning And Ternary Output Codes. In *Proceedings 2015 International Joint Conference on Neural Nets* (pp 1-8)

Gütig R. Aharonov R, Rotter S, Sompolinsky H. 2003. Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. The Journal of Neuroscience. 23(9), 3697-3714.

Hebb, D. O. (1949) The organization of behavior: A neurophysiological approach.

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., Masquelier, T. (2017) STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*.

Karmarkar UR, Dan Y. 2006. Experience-dependent plasticity in adult visual cortex. Neuron. 52:577–585.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... Hassabis, D. (2017) Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, **114** (13), 3521-3526.

Kortge, C. A. (1990) Episodic memory in connectionist networks. In *The Twelfth Annual Conference of the Cognitive Science Society*, (pp. 764-771). Hillsdale, NJ: Lawrence Erlbaum.

Kozloski J, Cecchi GA. 2010. A theory of loop formation and elimination by spike timing-dependent plasticity. Front Neural Circuits. 4.

Larsen DD, Callaway EM. 2006. Development of layer-specific axonal arborizations in mouse primary somatosensory cortex. J Comp Neurol. 494:398–414.

Letzkus JJ, Kampa BM, Stuart GJ. 2006. Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. J Neurosci. 26:10420–10429.

Li H, Fertuzinhos S, Mohns E, Hnasko, TS, Verhage M, Edwards R, Sestan N, Crair MC. 2013. Laminar and columnar development of barrel cortex relies on thalamocortical neurotransmission. Neuron. 79:970–986.

Lim S, McKee JL, Woloszyn L, Amit Y, Freedman DJ, Sheinberg DL, Brunel N. 2015. Inferring learning rules from distributions of firing rates in cortical neurons. Nat Neurosci. 18:1804–1810.

Lowel, S., Singer, W. (1992) Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, **255** (5041), 209-212.

Lui JH, Hansen DV, Kriegstein AR. 2011. Development and evolution of the human neocortex. Cell. 146:18–36.

Lund R, Mustari M. 1977. Development of the geniculocortical pathway in rat. J Comp Neurol. 173:289–305.

Markram H, Lübke J, Frotscher M, Sakmann B. 1997. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. Science. 275:213–215.

Masquelier, T., Thorpe, S. J. (2007) Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS computational biology*, **3** (2), e31.

McCloskey, M., Cohen, N. J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* Vol. 24, pp. 109-165. Academic Press.

McRae, K., Hetherington, P. (1993) Catastrophic Interference is Eliminated in Pre-Trained Networks. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society* (pp. 723-728). Hillsdale, NJ: Lawrence Erlbaum

Miller KD. 2003. Understanding layer 4 of the cortical circuit: A model based on cat v1. Cereb Cortex. 13:73–82.

Miyashita-Lin EM, Hevner R, Wassarman KM, Martinez S, Rubenstein JL. 1999. Early neocortical regionalization in the absence of thalamic innervation. Science. 285:906–909.

Musca, S. C., Rousset, S., Ans, B. (2009) Artificial neural network whispering to the brain: Nonlinear system attractors induce familiarity with never seen items. *Connection Science*, **21** (4), 359-377.

Rakic P. 1977. Prenatal development of the visual system in rhesus monkey. Philos Trans R Soc Lond *B* Biol Sci. 278:245–260.

Rakic P. 2009. Evolution of the neocortex: A perspective from developmental biology. Nat Rev Neurosci. 10:724–735.

Ratcliff, R. (1990) Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, **97** (2), 285.

Robins, A. (1995) Catastrophic Forgetting, rehearsal and pseudorehearal. *Connection Science*, 7, 123-146.

Rubin J, Lee DD, Sompolinsky H. 2001. Equilibrium properties of temporally asymmetric hebbian plasticity. Phys Rev Lett. 86:364.

Sengupta, A., Ye, Y., Wang, R., Liu, C., Roy, K. (2018) Going Deeper in Spiking Neural Networks: VGG and Residual Architectures. *arXiv preprint arXiv*: 1802.02627.

Shatz CJ, Luskin MB. 1986. The relationship between the geniculocortical afferents and their cortical target cells during development of the cat's primary visual cortex. J Neurosci. 6:3655–3668.

Silbereis JC, Pochareddy S, Zhu Y, Li M, Sestan N. 2016. The cellular and molecular landscapes of the developing human central nervous system. Neuron. 89:248–268.

Sjöström PJ, Häusser M. 2006. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. Neuron. 51:227–238.

Sjöström PJ, Turrigiano GG, Nelson SB. 2001. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. Neuron. 32:1149–1164.

Song S, Miller KD, Abbott LF. 2000. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nat Neurosci. 3:919–926.

Wess JM, Isaiah A, Watkins PV, Kanold PO. 2017. Subplate neurons are the first cortical neurons to respond to sensory stimuli. Proc Natl Acad Sci. 114:12602–12607.

Zhang LI, Tao HW, Holt CE, Harris WA, Poo MM. 1998. A critical window for cooperation and competition among developing retinotectal synapses. Nature. 395:37.