Learning efficient signal representation in sparse spike-coding networks

Mirjana Maraš

▶ **To cite this version:**

Mirjana Maraš. Learning efficient signal representation in sparse spike-coding networks. Neurons and Cognition [q-bio.NC]. Université Paris sciences et lettres, 2019. English. ⟨NNT : 2019PSLEE023⟩. ⟨tel-02613525⟩

HAL Id: tel-02613525

https://theses.hal.science/tel-02613525v1

Submitted on 20 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Learning Efficient Signal Representation in Sparse Spike-Coding Networks

Soutenue par

Mirjana MARAŠ

Le 5 décembre 2019

École doctorale n°158

Cerveau, Cognition, Comportement

Spécialité

Neurosciences

Composition du jury :

Dr. Jonathan PILLOW Princeton University	*Président, Rapporteur*
Dr. Cristina SAVIN New York University	*Rapporteur*
Dr. Gianluigi MONGILLO Sorbonne Université	*Examinateur*
Dr. Alessandro TORCINI Université de Cergy-Pontoise	*Examinateur*
Dr. Sophie DENÈVE École Normale Supérieure	*Directrice de thèse*
Dr. Christian MACHENS Champalimaud Centre for the Unknown	*Co-directeur de thèse*

ENS | PSL

Abstract

The complexity of sensory input is paralleled by the complexity of its representation in the neural activity of biological systems. To make the relationship between the input and the neural activity more tractable, the activity of sensory neurons is often described in terms of fixed functions of input stimuli. This results in the characterization of the neuron's tuning function and preferred stimulus, which are largely independent of the activity of other neurons. In contrast, principles of efficient coding and control would require that neural populations represent sensory variables cooperatively. Efficient representation would be accomplished with a minimal number of population spikes for a certain level of precision, and the representation would be robust to perturbations. If the brain is efficient and robust, then the activity of a single neuron would depend on the activity of other neurons, as much as it depends on its own preferred stimulus.

Starting from the hypothesis that biological networks are tuned to achieve maximal efficiency and robustness, we investigate how efficient representation can be accomplished in networks with experimentally observed local connection probabilities and synaptic dynamics. We develop a Lasso regularized local synaptic rule, which optimizes the number and efficacy of recurrent connections. The connections that impact the efficiency the least are pruned, and the strength of the remaining ones is optimized for efficient signal representation. Our theory predicts that the local connection probability determines the trade-off between the number of population spikes and the number of recurrent synapses, which are developed and maintained in the network. The more sparsely connected networks represent signals with higher firing rates than those with denser connectivity. The variability of observed connection probabilities in biological networks could then be seen as a consequence of this trade-off, and related to different operating conditions of the circuits.

The learned recurrent connections are structured, with most connections being reciprocal. The dimensionality of the recurrent weights can be inferred from the network's connection probability and the dimensionality of the feedforward input. In a fully connected network, the recurrent synaptic connectivity has the dimensionality of the input space, which is much smaller than the dimensionality of the neural space. As the connections become more rare, the dimensionality of the recurrent matrix increases, until it reaches a maximum in very sparsely connected networks.

In an idealized network with instantaneous synapses, the low-dimensional structure of a fully connected network is a signature of optimality. In contrast, the optimal connectivity of a network with synaptic delays is somewhere at an intermediate level, neither too sparse nor too dense. An intermediate recurrent connection probability acts like a source of noise, which desynchronizes the network and optimizes its efficiency in the presence of synaptic delays. Furthermore, when we add another biological constraint, adaptive regulation of firing rates, our learning rule leads to an experimentally observed scaling of the recurrent weights. When the adaptation of firing thresholds keeps the firing rates approximately constant, then the learned efficacy of recurrent weights scales inversely with one over the square root of the number of recurrent inputs.

Our work supports the notion that biological micro-circuits are highly organized and principled. A detailed examination of the local circuit organization can help us uncover the finer aspects of the principles which govern sensory representation.

Résumé

La complexité de l'entrée sensorielle est parallèle à la complexité de sa représentation dans l'activité neurale des systèmes biologiques. Pour clarifier la relation entre le stimulus et l'activité neurale, l'activité des neurones sensoriels est souvent décrite en termes de fonctions stationnaires du stimulus. Cela se traduit par la caractérisation de la sélectivité neuronale et du stimulus préféré, qui sont largement indépendants de l'activité des autres neurones. En revanche, les principes de codage et de contrôle efficace nécessiteraient que les populations de neurones représentent les variables sensorielles de manière coopérative. Une représentation efficace peut être réalisée avec un nombre minimal de potentiels d'action dans la population pour un certain niveau de précision, et la représentation serait robuste aux perturbations. Si le cerveau est efficace et robuste, l'activité d'un neurone dépend de celle d'autres neurones, autant que de son propre stimulus.

Partant de l'hypothèse que les réseaux biologiques sont réglés pour atteindre une efficacité et une robustesse maximales, nous étudions comment une représentation efficace peut être réalisée dans des réseaux avec des probabilités de connexion locale et une dynamique synaptique observée de manière expérimentale. Nous développons une règle synaptique locale régularisée de type Lasso, qui optimise le nombre et l'efficacité des connexions récurrentes. Les connexions qui affectent le moins le rendement sont élaguées, et la force des connexions restantes est optimisée pour une meilleure représentation du signal. Notre théorie prédit que la probabilité de connexion locale détermine le compromis entre le nombre de potentiels d'action de la population et le nombre de connexions synaptiques qui sont développées et maintenues dans le réseau. Les réseaux plus faiblement connectés représentent des signaux avec des fréquences de déclenchement plus élevées que ceux avec une connectivité plus dense. La variabilité des probabilités de connexion observées dans les réseaux biologiques pourrait alors être considérée comme une conséquence de ce compromis et serait liée à différentes conditions de fonctionnement des circuits.

Les connexions récurrentes sont structurées et la plupart des connexions sont réciproques. La dimensionnalité des poids synaptiques récurrents peut être déduite de la probabilité de connexion du réseau et de la dimensionnalité du stimulus. Dans un réseau entièrement connecté, la connectivité synaptique récurrente a la dimensionnalité de l'espace d'entrée, qui est beaucoup plus petite que la dimensionnalité de l'espace neural. La dimensionnalité de la matrice récurrente augmente dans les réseaux plus faiblement connectés jusqu'à atteindre la dimensionnalité de l'espace neural.

Le réseau optimal sans délais synaptiques est entièrement connecté. En revanche, la connectivité optimale d'un réseau avec des délais synaptiques se situe quelque part à un niveau intermédiaire, ni trop faible ni trop dense. Une probabilité de connexion récurrente intermédiaire agit comme une source de bruit, ce qui désynchronise le réseau et optimise son efficacité en présence d'un délai synaptique. De plus, lorsque nous ajoutons une autre contrainte biologique comme la régulation des taux de décharge par adaptation, notre règle d'apprentissage conduit à une mise à l'échelle observée de manière expérimentale des poids synaptiques.

Nos travaux soutiennent l'idée que les micro-circuits biologiques sont hautement organisés et qu'une étude détaillée de leur organisation nous aidera à découvrir les principes de la représentation sensorielle.

Acknowledgements

I would like to express my deep gratitude to my advisor, Dr. Sophie Denève, for entrusting me with an interesting and challenging research project, and for helping me refine ideas at the right moments. Her creativity and passion for science were inspiring. I am deeply grateful to my co-advisor, Dr. Christian Machens, for his sense of direction and clarity of ideas. It made a big difference at crucial moments during my PhD journey.

During my degree, I was fortunate to have my progress reviewed by Dr. Romain Brette and Dr. Gianluigi Mongillo, during the yearly thesis committee meetings. I am grateful for their pertinent feedback, which helped me focus my research efforts. I would also like to thank all of the members of the thesis defence jury for generously accepting to be on my jury and for reviewing my PhD thesis.

I would like to thank my dear colleagues at the Group for Neural Theory, with whom I shared insightful scientific conversations, but also enjoyed all the lunchtime laughs and summer picnics. The sense of community that we share is so special and I hope that we will keep it in the years to come. I would also like to thank my dear colleagues at the Champalimaud Institute, in Lisbon, who welcomed me with open arms and made me feel so included.

My thanks goes to Héloïse Héro, Vasilisa Skvortsova, and Ivan Gordely, for our machine learning book club. Our weekly discussions and problem solving sessions provided a great format to learn important techniques. I would like to thank Lyudmila Kushnir for her feedback on an early thesis draft and her feedback on the introductory chapter of this thesis. More importantly, I would like to thank her for the many insightful discussions we have have had throughout the years. This includes our weekly discussions of the "Theoretical Neuroscience" book, which made its reading more achievable and rewarding. My thanks goes to Grégory Dumont for reading an early draft of this thesis and providing useful feedback.

I would like to thank Dr. JF Williams, Dr. Rustum Choksi, and Dr. David Muraki, for supporting me wholeheartedely during the application process to this PhD program. They have taught me so much during my days as an Applied Mathematics student at Simon Fraser University, and I will always be grateful to them.

I would like to thank my dear friends and family for always being by my side. Your love and support encouraged me along the way. I thank my parents, Aleksandar and Vojislavka, and my sister Ivana, for their unconditional love. I thank Mehdi for believing in me and cheering me up every day. And I thank my best friends, Asli, Eveline, Gloria, and Trimita, who remind me from far away of how lucky I am to be surrounded by so much love and support.

To my sister Ivana,
mother Vojislavka, and father Aleksandar.

Contents

List of Figures ix

1 Introduction 1

1.1 Experimental observations of synaptic organization in cortical networks . . 3

1.1.1 Connection probability and efficacy of local synapses 3

1.1.2 Balance between excitatory and inhibitory inputs 4

1.2 Observed plasticity of network synaptic connectivity 5

1.2.1 Synaptic restructuring during transient E/I imbalance 5

1.2.2 Developmental synaptic restructuring 7

1.2.3 Emerging functional cell assemblies 8

1.3 Theoretical models of the dynamics of local circuits 9

1.3.1 Balanced networks with sparse connectivity 9

1.3.2 Balanced network with dense connectivity 10

1.3.3 Spike-coding network with full connectivity 10

1.4 Self-organized learning in neural networks 11

1.4.1 From Hebbian to STDP learning 11

1.4.2 Learning in balanced networks 14

1.4.3 Learning in spike-coding networks 15

1.5 Thesis organization 17

2 Review of Coordinated Spike-Coding 19

2.1 Motivation 19

2.2 Efficient representation with coordinated spike-coding 21

2.3 Details of the CSC model: Implementation in a recurrent network 22

2.3.1 Voltage dynamics 23

2.3.2 Interpretation of the network's function 24

2.4 Deriving the network structure from an optimality principle . . . 26
2.4.1 Loss function for the derivation of an efficient network . . . 26
2.4.2 Optimal linear decoder . . . 27
2.4.3 Optimal recurrent connections . . . 27
2.5 Learning the optimal recurrent connections . . . 30
2.5.1 Loss function for the learning of E/I balance . . . 30
2.5.2 Synaptic learning rule . . . 31
2.5.3 Linear decoder during network learning . . . 32
2.6 Conclusion . . . 33

3 Spike Coding Networks with Sparse Connections 35
3.1 Defining the network model . . . 36
3.2 Learning to optimize efficient representation and connection probability in spike-coding networks . . . 39
3.2.1 Loss function for the E/I balance and sparse connectivity . . . 39
3.2.2 Minimization of the loss function . . . 41
3.2.3 Lasso regularized synaptic learning rule . . . 42
3.2.4 Learning E/I balance . . . 42
3.2.5 Learning to distribute spiking . . . 43
3.2.6 Learning to grow sparse recurrent connections . . . 44
3.3 The fixed point of the learning rule . . . 44
3.4 Linking input statistics with connectivity . . . 45
3.5 Convergence to optimal network structure for small inputs . . . 48
3.5.1 Derivation of the fixed point and proof of convergence . . . 49
3.6 The fixed point for large feedforward inputs . . . 54
3.7 Conclusion . . . 57

4 Training Spiking Networks with Sparse Connections 59
4.1 Network model . . . 60
4.2 Training sparse efficient networks . . . 61
4.2.1 Simulation details . . . 61
4.2.2 Evolution of synaptic weights during learning . . . 61
4.2.3 The scaling of synaptic strength with postsynaptic voltage . . . 63
4.2.4 The scaling of synaptic weights with the scalar product of linear encoders . . . 65

4.3 Imbalance between the learned excitatory and inhibitory synaptic strengths 66
4.3.1 Fixing the imbalance . . . 68
4.3.2 Training sparser networks requires higher quadratic firing costs . . . 69
4.4 Simulations with balanced learning of positive and negative connections . . 70
4.5 Structure of recurrent weights in sparse networks . . . 72
4.6 Optimal decoder for sparse networks . . . 80
4.7 The trade-off between firing and wiring . . . 81
4.8 The balance between excitation and inhibition . . . 83
4.9 Sensitivity to network parameters . . . 86
4.10 Alternative sparse spike-coding networks . . . 88
4.10.1 Pruning the weakest and fixing the strongest connections . . . 88
4.10.2 Pruning the weakest and learning the strongest connections . . . 90
4.11 Conclusion . . . 91

5 Learning Spiking Networks with Synaptic Dynamics 93
5.1 Defining the network model . . . 94
5.2 Learning sparse CSC networks with synaptic delays and dynamics . . . 98
5.2.1 The loss function . . . 98
5.2.2 The Lasso regularized synaptic rule with synaptic delays and dynamics 99
5.3 Numerical simulations of the learning rule . . . 100
5.4 Properties of trained sparse networks with synaptic dynamics . . . 102
5.5 Network performance and optimal number of connections . . . 106
5.6 Relation to previous work on coordinated spike coding with synaptic delays 108
5.7 Conclusion . . . 111

6 Performance Trade-off in High Dimensional Sparse Networks 113
6.1 Training high-dimensional sparse networks . . . 113
6.2 Performance of high-dimensional networks . . . 115
6.3 Orthogonality of high-dimensional spaces . . . 118
6.4 Network size and the performance trade-off . . . 122
6.4.1 Varying network size in optimal networks . . . 123
6.4.2 Varying network size in independent networks . . . 126
6.4.3 Performance trade-off between independent and optimal networks as a function of network size . . . 128

6.5 Input scale and the performance trade-off . . . 130
6.6 Relating the performance trade-off to the preferred stimulus . . . 132
6.6.1 Relationship between the preferred angle and network performance . 136
6.7 Conclusion . . . 138

7 Conclusion and Perspectives **141**
7.1 Learning efficient representation in networks with limited connectivity . . . 141
7.1.1 The structure and activity of networks with limited connectivity . . 142
7.1.2 Coding strategies and the trade-off between firing and wiring 143
7.1.3 Efficient representation of high-dimensional inputs . . . 143
7.2 Learning efficient signal representation with synaptic delays . . . 144
7.2.1 Optimal connection probability with synaptic dynamics . . . 145
7.2.2 Robustness of the trained networks . . . 145
7.3 The scaling and distribution of synaptic efficacies . . . 146
7.3.1 The scaling with the number of lateral connections . . . 146
7.3.2 The distribution as a function of input dimensionality . . . 147
7.4 Extension to E/I networks . . . 149
7.5 Future outlook . . . 151

A The Fixed Point of the Learning Rule, when β=1 **153**

Bibliography **155**

List of Figures

1.1 Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex, reproduced from (Levy and Reyes, 2012) 4

1.2 Synaptic modifications induced by nucleus basalis pairing, reproduced from (Froemke et al., 2007) 6

1.3 The different types of synaptic modification via STDP, adapted from (Abbott and Nelson, 2000) and (Caporale and Dan, 2008) 13

2.1 The conundrum of Poisson rate code and E/I balance, reproduced from (Deneve and Machens, 2016) 20

2.2 Input reconstruction by the coordinated spike-coding model 25

2.3 Algorithmic description of coordinated spike-coding 26

2.4 Schematic: Optimal linear encoders and decoders for white inputs 29

3.1 Schematic: Spike-coding network model with sparse recurrent connections . 38

3.2 Schematic: Lasso regression, adapted from (Tibshirani, 1996) 40

3.3 Schematic of the relationship between the signal and encoding vectors of the neural population 46

3.4 Schematic of the relationship between input gain and network activity . . . 47

3.5 Input statistics: signal gain and radial speed 55

4.1 Learning synaptic connections with and without a sparseness constraint . . 62

4.2 Convergence of mean spike triggered voltage and recurrent weights with and without the sparseness constraint 64

4.3 Structure and distribution of recurrent weights in trained networks with varying connection probabilities 66

4.4 Learning sparse networks without a lower voltage bound decreases performance 67

4.5 Learning sparse networks with balanced positive and negative connections . 70

4.6 Training sparse networks on inputs with constant gain, higher quadratic costs, and a fixed voltage lower bound . . . 71
4.7 Structure of recurrent weights in sparse networks . . . 73
4.8 Structure of the leading eigenvectors of sparse recurrent connectivity . . . 75
4.9 Low rank representations of the sparse recurrent matrices . . . 77
4.10 Decomposition of sparse recurrent connectivity as the sum of the error-correcting and L_1 penalization terms . . . 79
4.11 Optimal decoder for sparse networks . . . 81
4.12 Trade-off between the firing rate and the number of synaptic connections . . . 82
4.13 Balance of excitation and inhibition in sparse networks . . . 84
4.14 Membrane potential distribution in *in vivo* intracellular recordings in various areas, reproduced from (Rossant et al., 2011) . . . 85
4.15 Influence of parameter values on connection probability and performance . . . 87
4.16 Characteristics and performance of alternative sparse spike-coding networks 89
4.17 Instability of constructed sparse networks with small linear costs . . . 90

5.1 The effect of a presynaptic spike on the postsynaptic neuron. . . . 96
5.2 Input reconstruction and network activity of a trained sparse network with 1 ms delays and AMPA synaptic dynamics . . . 103
5.3 Structure and performance of trained sparse networks with synaptic delays and dynamics . . . 104
5.4 Recurrent weights are smaller for longer synaptic transmission delays . . . 105
5.5 Performance comparison of sparse networks with and without synaptic delays and dynamics . . . 107
5.6 Impact of synaptic delays on the performance of the optimal network, adapted from (Chalk et al., 2016) . . . 109

6.1 Defining multi-dimensional inputs with constant gain and radial speed . . . 114
6.2 Training higher-dimensional sparse networks of different network sizes . . . 116
6.3 Examples of the dependence of final recurrent connectivity on the network size . . . 117
6.4 Performance with increasing input dimensionality for fixed size networks . . . 119
6.5 The relationship between the encoding vectors and input dimensionality . . . 120
6.6 Performance of the optimal network as a function of its size . . . 124
6.7 Performance of the independent network as a function of its size . . . 127
6.8 Performance trade-off between fully connected and non-connected networks, as a function of network size . . . 129

6.9 Performance trade-off between fully connected and non-connected networks, as a function of input gain . . . 131
6.10 Preferred stimulus of neurons in optimal and independent networks, as a function of network size and input dimensionality . . . 132
6.11 Schematic of the influence of input speed on which neuron spikes next in an optimal network . . . 135

7.1 The scaling of trained synaptic weights with the number of recurrent inputs 146
7.2 The distribution of trained sparse synaptic weights, as a function of input dimensionality . . . 148
7.3 Learning sparse networks of separate excitatory and inhibitory populations with synaptic delays . . . 150

1 | Introduction

Understanding how sensory inputs are represented in the neural activity of biological systems is at the foundation of elucidating the 'neural code'. The complexity of sensory inputs and of the corresponding neural representations makes this a difficult task. To make this task more tractable, the activity of sensory neurons tends to be described in terms of fixed functions of input stimuli. The resulting neuron's tuning function is then primarily determined by the feedforward sensory input, and the neuron's preferred stimulus can be assumed to be independent of the activity of other neurons.

Experimental observations and higher level organizing principles point to more nuanced relationships. For example, the principles of efficient population coding are based on a coordinated representation of sensory variables by a neural population. The organized local circuit distributes the representation across the activity of many neurons, making the representation more robust to perturbations. The representation is considered efficient, in our use of this word, when the local circuit fires a minimal number of population spikes for a certain level of signal precision. If the brain is efficient and robust, then the activity of a single neuron is influenced both by its feedforward sensory inputs and by the activity of other neurons in the local circuit.

In this thesis, we start from the hypothesis that biological networks are tuned to achieve maximal efficiency and robustness, and then ask how this could be accomplished with increasing biological constraints. Specifically, we investigate how efficient representation can be learned in networks with experimentally observed local connection probabilities and synaptic dynamics.

We will show that efficient representation can be learned with local synaptic rules, which progressively shape the efficacy and existence of recurrent synapses. Following these synaptic rules, the recurrent inputs can learn to cancel the network's shared feedforward inputs. The network then represents the feedforward input with minimal network activity. We will show that the network can optimize the number of local synaptic connections, by pruning those connections that contribute the least to the cancelation of the feedforward input.

We will implement efficient representation in coordinated spike-coding networks, in which the precise timing of a spike carries precise information about the stimulus. When the recurrent connections are structured in a specific way, the population spikes are coordinated so that the shared feedforward input is represented with a minimal number of spikes. This particular optimal recurrent structure is related to the statistics of the feedforward

input. As we will show, there is a set of optimal recurrent connections for networks of any recurrent connection probability, and they can be obtained following the synaptic rule that we propose. In the resulting self-organized networks, the activity of other neurons can have a big impact on the activity of any single neuron. Individual neurons share information about their shared stimuli via the available local connections, which are optimized for efficient signal representation at the network level.

This introductory chapter starts with a review of the experimentally observed connection probabilities and efficacies of local synapses. As we will see, these vary based on neuron type, cortical region, and animal species. The local connectivity is also plastic. We will consider examples, which illustrate the dynamics of local connectivity during an organism's development, in response to salient experiences in adult life, and during experimental manipulations.

Since we are interested in relating the activity of a local network to its function, we will then review some of the common theoretical models of the local network activity. First, we will consider the analysis of the local dynamics, given fixed recurrent connectivity profiles. Then, we will turn to the self-organized learning in neural networks, focusing primarily on spike-based learning rules.

In this thesis, we are going to investigate the self-organized learning of efficient representations, in the context of coordinated spike-coding (CSC) networks. The idealized CSC network, with full recurrent connectivity and instantaneous synapses, was introduced in (Boerlin et al., 2013). The local synaptic rule, which can be used to train the idealized CSC network, was introduced in (Brendel et al., 2017). The CSC framework, although highly idealized, provides interesting insights and a functional interpretation of neural activity, which opposes the classical theoretical approaches. Since this framework is important for the understanding of this thesis, and since its details are somewhat less known by the wider community, we will review it in more detail in Chapter 2. The review will particularly focus on the two above-mentioned papers.

After the review, we will progressively present more complete answers to the question at the heart of this thesis. We will show how more biologically realistic networks can learn to represent inputs efficiently. We will show this first in networks with variable connection probabilities, then in networks with realistic synaptic transmission delays and dynamics, and finally in networks with higher-dimensional feedforward inputs. We will also consider the extension to networks with separate excitatory and inhibitory populations, at the very end of this thesis.

Our work brings the coordinated spike-coding model closer to biology, but also examines the more general relationship between the function and connectivity of neural networks. The results from this thesis support the notion that the biological micro-circuits are highly organized, and that this high level of organization is a signature of their operating principles. Given this point of view, much can be gained from a detailed examination of the fine scale organization of the local neural circuits. We hope that this thesis helps refine our understanding of the principles, which govern sensory representation.

1.1 | Experimental observations of synaptic organization in cortical networks

1.1.1 | Connection probability and efficacy of local synapses

The connectivity in local cortical circuits is dominated by recurrent connections with nearby neurons within the same cortical layer. This is especially true in early sensory cortices, which receive input from the sensory organs via thalamic afferents (LeVay, 1986; Peters and Payne, 1993; Peters et al., 1994). The precise architecture of the local connectivity can have a big impact on the tuning properties of individual neurons. It can shape the preferred stimulus orientation in the primary visual cortex, or preferred stimulus frequency in the primary auditory cortex (Douglas et al., 1995; Sompolinsky and Shapley, 1997; Shapley et al., 2007).

Both the synaptic connection probability and the efficacy of individual synapses depend on the cell type, layer, and animal model (Hofer et al., 2011; Bock et al., 2011; Levy and Reyes, 2012; Ko et al., 2013; Jiang et al., 2015; Barral and Reyes, 2016). There is evidence for recurrent connectivity being structured, so that reciprocal synaptic connections occur more often than in a randomly connected circuit (Song et al., 2005), with emerging strongly connected cell assemblies (Yoshimura et al., 2005; Perin et al., 2011; Gal et al., 2017). For example, similarly tuned neurons were recently found to suppress each other's activity more than dissimilarly tuned ones in layer 2/3 of the mouse primary visual cortex (Chettih and Harvey, 2019). This competition could be mediated via more strongly connected subnetworks of similarly tuned excitatory and inhibitory cells.

The recurrent connection profiles and the local distribution of feedforward inputs shape the coding properties of the local circuit (Harris and Mrsic-Flogel, 2013). A general principle in retinotopic and tonotopic systems is that the likelihood of two neighbouring neurons being connected falls off with their distance. Thus, the closer that two neurons are to each other, the more likely it is that they share information with each other.

For example, the connection probability between excitatory pyramidal cells and inhibitory interneurons, in the input layer of the mouse primary auditory cortex, shows a strong dependence on distance (Levy and Reyes, 2012). The connection probability between the pyramidal cells is at most 0.15, for nearby cells (Fig. 1.1a). The connection from pyramidal to PV interneurons can be as high as 0.54, with reverse connection probability being even higher, 0.66 (Fig. 1.1b,c). The connection probability between pyramidal and somatostatin-expressing inhibitory cells is at most 0.5 for outgoing connections, and 0.42 for incoming connections (Fig. 1.1c,e).

Furthermore, the synaptic efficacies depend on neuron type. Single postsynaptic potentials (PSP), elicited or received by inhibitory neurons, are usually larger than excitatory PSPs received by excitatory cells. In the same experiment, (Levy and Reyes, 2012) measured the mean amplitude of the first PSP elicited in the postsynaptic cell by the presynaptic spike. During the regular firing of the presynaptic cell, the first spike elicited the largest effect on the postsynaptic neuron, with short-term depression ensuing thereafter. For synapses from excitatory to local excitatory neurons, the mean amplitude of the first PSP was 0.45 mV. On the other hand, the first excitatory PSPs elicited in PV cells had an

average response of 1.29 mV, and the first inhibitory PSPs elicited in excitatory cells had a mean amplitude of 0.64 mV. There is also evidence that PV cells strongly inhibit each other (Pfeffer et al., 2013).

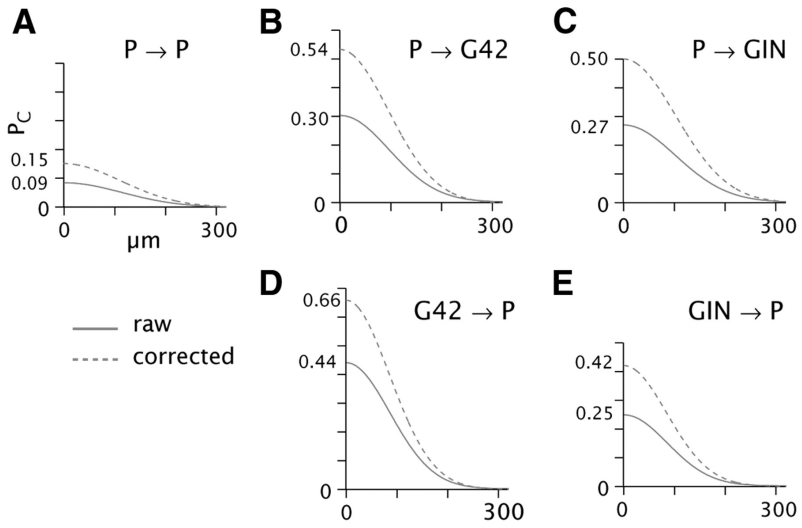


Figure (1.1) Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex, reproduced with the exact caption from (Levy and Reyes, 2012). A-E, Estimation of PC profiles after correction for slicing. Solid lines are Gaussian fits to the physiology data for PC versus distance. Dashed lines are the same curves multiplied by a scale factor representing the ratio of axonal/dendritic overlap for the corrected versus raw process length density maps. Peak values are indicated on left axis. Note that the acronym PC used in this caption from (Levy and Reyes, 2012) stands for the probability of connection.

To make the picture more complex, the postsynaptic response is stochastic, with some presynaptic spikes not eliciting any postsynaptic response. Furthermore, the synaptic connections can be volatile due to spines spontaneously appearing and disappearing. For example, the efficacy of excitatory synapses is highly volatile in the absence of learning, so much so, that the turn-over of excitatory spines during learning is approximately the same as during spontaneous activity (Mongillo et al., 2018).

With this variability in the synaptic efficacies and connection probabilities in local circuits, it seems like the computational capabilities of the circuit would be highly variable and unreliable. However, there are indications that local networks continuously reorganize themselves to produce stable population dynamics.

1.1.2 | Balance between excitatory and inhibitory inputs

There is experimental support for a balance of average excitatory and inhibitory (E/I) currents received by a neuron (Shu et al., 2003; Wehr and Zador, 2003; Poo and Isaacson, 2009; Isaacson and Scanziani, 2011). This indicates that the activity of individual neurons is driven by the fluctuating difference between the strong excitatory and inhibitory inputs,

which cancel each other out on average.

There is also experimental support for a more temporally precise balance in different brain regions and animals, including the rat and mouse primary auditory cortices (Wehr and Zador, 2003; Graupner and Reyes, 2013), rat somatosensory cortex (Okun and Lampl, 2008), rat visual cortex (Tan et al., 2013), primate and mouse retinas (Cafaro and Rieke, 2010), mouse entorhinal cortical slices (Salkoff et al., 2015), ferret prefrontal cortex (Haider et al., 2006), and the zebrafish homologue of the olfactory cortex (Rupprecht and Friedrich, 2018). Furthermore, there is support for excitatory and inhibitory cotuning for stimulus features (Froemke et al., 2007; Rocha et al., 2008), which points to a precise E/I balance.

The mechanism of the precise E/I balance is an area of active research. While as pyramidal cells receive excitatory and inhibitory PSPs, with highly heterogeneous amplitudes, there is evidence that those cells that receive larger excitatory PSPs tend to receive larger inhibitory PSPs. This was observed in simultaneously recorded nearby pyramidal cells in layer 2/3 of the mouse primary visual cortex (Xue et al., 2014). In this experiment, the ratio of the cell's excitatory and inhibitory inputs was close to unity over time and over different pyramidal cells. The PV cells mediated this precise balance. Each PV cell made contacts with many different pyramidal cells, but the elicited inhibitory PSPs were correlated with the temporal excitation of the excitatory targets. The activity of PV cells was such that the inhibitory inputs into pyramidal cells closely matched the pyramidal time varying excitatory inputs, thus resulting in a precise balance.

1.2 | Observed plasticity of network synaptic connectivity

The balancing of excitatory and inhibitory inputs into a neuron could be seen as a signature of a network, which has learned to represent its inputs efficiently, and will spike only during transient imbalances. Larger imbalancing of inputs could be associated with experience-dependent learning, which is mediated by the restructuring of the local cortical networks.

The synaptic restructuring in response to changing excitatory-inhibitory balance is reviewed in (Froemke, 2015), and experience-dependent synaptic plasticity is reviewed in (Feldman, 2009; Cooke and Bear, 2014). Here we describe a few experiments, which illustrate the variety of the contexts in which local networks change their synaptic connectivity in response to experience-dependent or experimentally induced E/I imbalance.

1.2.1 | Synaptic restructuring during transient E/I imbalance

The importance of precise balance, and what happens when it's broken was investigated in the rat primary auditory cortex, with in vivo whole cell recordings (Froemke et al., 2007). This experiment showed that neuromodulation shapes synaptic receptive fields by breaking the synaptic E/I balance in response to the presented stimulus. The transient E/I imbalance then allows a rapid remapping of the original receptive field to one that favours the presented stimulus.

In the experiment, the electrical stimulation of the nucleus basalis, which releases acetylcholine in the primary auditory cortex, was paired with the presentation of a stimulus tone

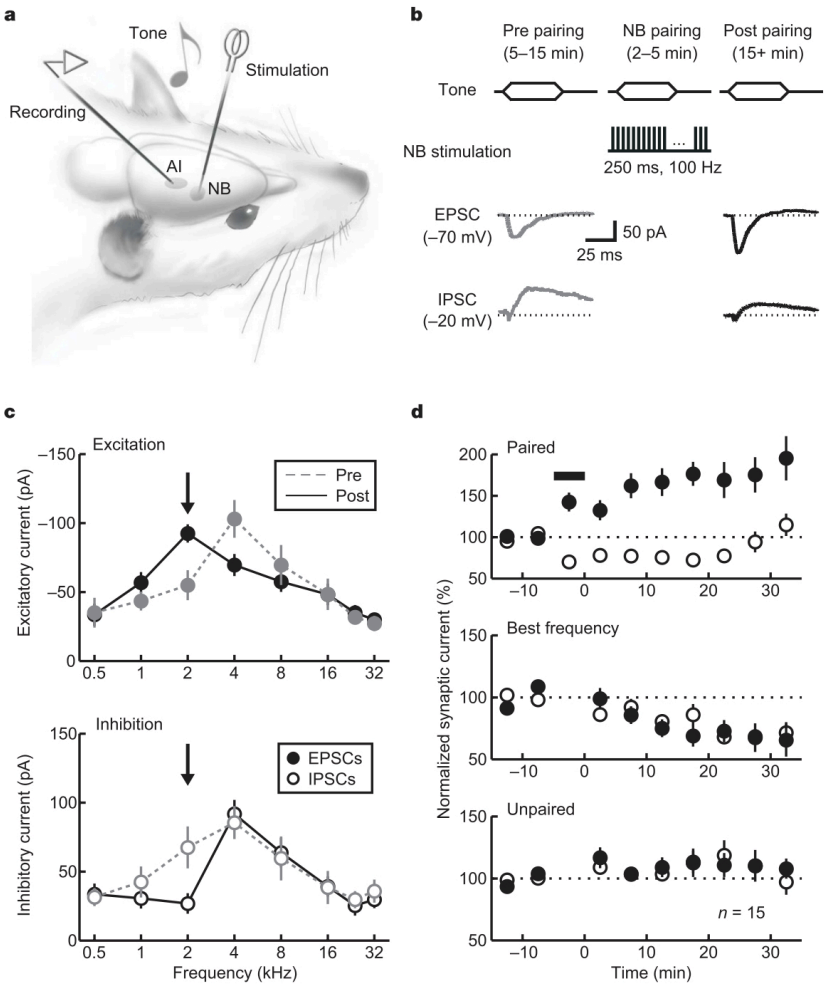


Figure (1.2) Synaptic modifications induced by nucleus basalis pairing, reproduced with the exact caption from (Froemke et al., 2007). **a**, Experimental configuration. **b**, Experimental procedure. **c**, Example of pairing-induced modification of synaptic tuning curves. Upper panel, excitatory tuning. EPSCs at the paired frequency (2 kHz) increased from -54.9±10.9 to −92.4±6.6 pA (68.4%, P<0.006, t-test; filled symbols). Lines, tuning before (dashed grey) and 10 min after (solid black) pairing. Arrow, paired tone. Lower panel, inhibitory tuning. IPSCs at the paired frequency decreased from 67.6 ± 15.2 to 27.0 ± 7.4 pA (-60.1%, P<0.03; open symbols). **d**, Time course. Uppermost panel, paired frequency (excitation: $68.0 \pm 13.9\%$, $n = 15$, P<0.0007; inhibition: $-24.8 \pm 6.0\%$, P<0.0002). Horizontal bar, nucleus basalis pairing. Middle panel, original best frequency. Lowest panel, other unpaired tones. Error bars, s.e.m. NB, nucleus basalis; AI, primary auditory cortex.

of a particular frequency. The paired cholinergic neuromodulation lasted for a few minutes, and was then followed just by the presentation of the stimulus tone. Within twenty seconds of the onset of the paring, there was a decrease in the amplitude of the tone-evoked IPSCs at the paired frequency, and within a minute there was also an increase in the tone-evoked EPSCs at the paired frequency. Here, IPSC and EPSC denote the inhibitory and excitatory postsynaptic currents. The effect of the paired neuromodulation was to break the E/I balance at the stimulus frequency (Fig. 1.2).

Before the pairing, the paired stimulus frequency evoked primarily subthreshold EPSCs, and rarely caused spikes at recording sites. After the pairing, there was a seven-fold increase in the spiking activity of cortical neurons, owing to the imbalance between the excitatory and inhibitory synaptic currents. This caused a remapping of the synaptic receptive fields over the hours following the transient nucleus basalis pairing. The excitatory potentiation increased and then persisted, and the inhibitory response eventually started to increase, until it matched the excitatory response at the paired frequency. The excitatory and inhibitory synaptic currents were balanced at the end of learning.

The depression of the inhibitory response and subsequent potentiation of the excitatory response at the paired frequency, during the transient pairing, allowed the rapid learning of a new stimulus, which could be behaviourally relevant. This experiment indicates that a mechanism for learning salient stimuli could be the transient breaking of the E/I balance.

In another experiment, oxytocin was paired with pup distress calls, to produce a maternal state in virgin female mice (Marlin et al., 2015). After learning, the virgin mice changed their behaviour, and started to retrieve pups who emitted distress calls. The neural representation of the relevant local circuit in the primary auditory cortex also became highly selective for pup calls. The neuromodulatory effect of oxytocin was to introduce a transient E/I imbalance in the relevant circuit. The circuit subsequently reorganized its synaptic connectivity, to re-balance excitation and inhibition, and increase the precision with which it represents the behaviourally salient stimulus. The oxytocin-pup call pairing was brief, and the subsequent network restructuring was completed within minutes to hours. The remapped circuit connectivity was found to be long-lasting.

1.2.2 | Developmental synaptic restructuring

Lack of E/I balance is a known characteristic of the critical developmental period. The developmental increase of GABA-ergic inhibition is correlated with the decrease in experience-dependent plasticity (Sale et al., 2010). So for example, infant mammals which had the vision of one eye experimentally impaired by monocular deprivation during the critical developmental period, are not going to develop normal vision in the deprived eye (Hubel and Wiesel, 1970; Fagiolini et al., 1994). However, pharmacological decrease of GABA-ergic inhibition in the adult brain re-activates the ocular dominance plasticity in adult rats (Harauzov et al., 2010).

Another example of local synaptic restructuring during developmental learning is in whisker-deprived juvenile rats (Shao et al., 2013). Whisker deprivation results in a decreased recurrent excitation of L2/3 circuits, which recruits a decrease in recurrent inhibition. This could be explained by the decreased connection probability between excitatory neurons, which then leads to a structural re-organization of the entire E/I local circuit.

There is also evidence for inhibition being as strong as excitation at the beginning of development, with the E/I imbalance instead being caused by the mismatched selectivities of connected excitatory and inhibitory neurons. Whole-cell recordings in the rat primary auditory cortex show that early on, excitatory and inhibitory neurons have mismatched tuning curves, with inhibitory neurons being less selective to input (Dorrn et al., 2010). During the critical developmental period, the external sensory stimulation shapes the synaptic structure of local circuits, resulting in balanced and co-tuned response properties of the excitatory and inhibitory neurons.

1.2.3 | Emerging functional cell assemblies

Learning influences the tuning properties of different types of cells, and this is in part due to the changing local connectivity structure. For example, functional subnetworks of pyramidal-PV cells can emerge during the learning of visual stimuli (Khan et al., 2018). In this experiment, head-fixed mice were trained to discriminate vertical gratings from angled ones, in a go-no go discrimination task. They were rewarded for licking a spout in response to the vertical gratings, which were displayed on the wall of a virtual approach corridor. There was no punishment for a response to the angled pattern. Two-photon calcium imaging was used to record the activity of L2/3 cells in the primary visual cortex before and after learning. Stained brain sections were later used to identify the simultaneously recorded cells as PV-positive, SOM-positive, VIP-positive, and putative PYR cells.

The response profiles were variable across cell groups and between cell classes, but PV and PYR cells had the most similar average stimulus-triggered response profiles before learning. After learning, both PYR and PV cells significantly increased their selectivity for the rewarded grating pattern. Furthermore, the strength of the effective local synapses from pyramidal neurons onto PV cells was significantly higher when both cells preferred the same grating stimulus, as opposed to cells with different grating preferences. This functional reorganization was also found in PV-PV and PV-PYR effective local connections. The experience-dependent increased selectivity of PV cells for the task relevant stimulus could be predicted by local reorganization of the local connections.

Before learning, the PV cells showed a broad tuning for the task-relevant stimulus. This agrees with the observation that PV cells broadly integrate the heterogeneous inputs from neighbouring excitatory cells in the mouse primary visual cortex (Kerlin et al., 2010; Bock et al., 2011; Hofer et al., 2011; Scholl et al., 2015). After learning, the PV cells became selective for the task-relevant stimulus, and their tuning properties were less related to the average tuning of the surrounding excitatory cells.

This experiment finds that the effective synaptic connections change over learning so that a local network of similarly tuned PYR and PV cells is formed. The dedicated subnetwork is characterized by stronger effective connections, and an increased selectivity of both the excitatory and PV cells for the relevant stimulus.

1.3 | Theoretical models of the dynamics of local circuits

In spite of the experience-dependent restructuring of local networks, the dynamics of cortical networks has certain stable properties. The observed dynamics of local circuits is asynchronous and irregular, with the precise timing and number of spikes *in vivo* being highly variable on different presentations of the same stimulus (Tolhurst et al., 1982; Shadlen and Newsome, 1998). Traditionally, the time-binned spike counts are averaged to obtain average firing rates, which are then seen as reliably encoding the signal.

We will now introduce and provide a brief overview of balanced random networks, which are often used for theoretical analysis of network dynamics. This provides a wider context for the coordinated spike-coding networks, which we use to model local dynamics throughout this thesis.

1.3.1 | Balanced networks with sparse connectivity

The classical approach to theoretical analysis of population dynamics is to assume that neurons are rate units, which have the same firing characteristics and essentially receive the same feedforward input. The recurrent connectivity is sparse and random, which is necessary to ensure self-consistency with irregular firing and decorrelated activity across the population (van Vreeswijk and Sompolinsky, 1996).

Each neuron receives K inhibitory and K excitatory inputs from the local population, and external inputs from K excitatory cells. The number of inputs, K, is much smaller than the network size. The synaptic weights scale with $1/\sqrt{K}$, so that $\sqrt{K}$ concurrent excitatory spikes would drive a postsynaptic neuron to fire a spike. The number of recurrent inputs shared by two neurons in the network is small due to their sparse random connectivity, and their activities are only weakly correlated. Since each cell receives weakly correlated input, the input from each subpopulation (excitatory, inhibitory, or external) can be modeled as a Gaussian distribution with the mean K and the standard deviation $\sqrt{K}$.

Since $\sqrt{K}$ excitatory spikes are needed to cause a postsynaptic neuron to spike, the postsynaptic cell is generally either highly depolarized or hyperpolarized. However, if the total excitatory and inhibitory inputs into a cell cancel on average, then the firing is driven by the input fluctuations rather than the mean input (Shadlen and Newsome, 1994; van Vreeswijk and Sompolinsky, 1996).

In this fluctuation driven regime, the network activity is chaotic. The firing of each cell is irregular and highly nonlinear, but the average population rate is almost a linear function of the external input. In this regime, the average population firing rate responds to changes in external input faster than the individual rate units. Fluctuations in spike rate have been observed to correlate with the decision that macaque monkeys make in a motion discrimination task (Celebrini and Newsome, 1994; Newsome et al., 1995).

The exact conditions under which a random network is in the balanced state were derived from theoretical arguments and verified numerically in networks of leaky integrate-and fire neurons with synaptic transmission delays (Brunel, 2000). A detailed phase diagram was derived, showing predominantly four stable regimes of dynamics. The phase transitions depend on the ratio of the strength of the inhibitory and excitatory synapses,

the external drive, and the distribution of delays. The asynchronous irregular regime was shown to be stable in the inhibition dominated networks, and the region of stability increased when uniform synaptic delays were used. The phase diagram was also characterized for networks in which inhibitory and excitatory neurons are not identical, but have biologically observed time constants and delays. In this case, the phase diagram is more complicated, but the conditions for the stability of the asynchronous irregular state can be characterized.

The asynchronous balanced state supports fast information transmission, but is highly redundant. All neurons have the same constant firing rates and the entire population can be seen as encoding a single variable. However, if the coupling between neurons is made stronger, the network enters a heterogeneous chaotic state, which can support complex computations and learning (Ostojic, 2014; Mastrogiuseppe and Ostojic, 2018).

1.3.2 | Balanced network with dense connectivity

Balanced networks require sparse connectivity in order to reproduce stable activity and irregular firing. Above a certain finite number of connections, these networks start to exhibit collective dynamics (Luccioli et al., 2012). However, as discussed at the beginning of this chapter, biological neural networks are characterized by finite connection probabilities.

If the weak unstructured connections are made stronger, random networks with finite connection probabilities can produce stable asynchronous activity (Renart et al., 2010). The connectivity in this model is dense, meaning that the connection probability is fixed with increasing network size, N. This is in contrast to sparse connectivity, which goes to zero with increased network size.

In the (Renart et al., 2010) model, the number of internal connections scales with N. Furthermore, the recurrent weights are strong and they scale with $1/\sqrt{N}$. The neurons receive shared inputs, but their activity is only weakly correlated. This is a consequence of the fact that the inhibitory currents track excitatory ones more precisely than in classical balanced networks. The resulting activity is thus asynchronous.

1.3.3 | Spike-coding network with full connectivity

Asynchronous activity of cortical networks can also be modeled with networks in which the spike times are coordinated to represent inputs more efficiently than in balanced networks (Boerlin et al., 2013; Deneve and Machens, 2016).

In contrast to the balanced networks, the connections in the coordinated spike-coding model (Boerlin et al., 2013) are structured, and the connectivity is all-to-all. The coupling is very strong, with the synaptic weights being of the same order as the firing threshold. The attained E/I balance is temporally precise, and this corresponds to a highly efficient representation of inputs.

The coordinated spike-coding network is derived from the minimization of a loss function, which measures the representation error and the level of network activity. One of the first times that neural networks were derived from a quadratic loss function was in the context of memory storage (Hopfield, 1982, 1984). In the case of (Boerlin et al., 2013), the

minimization results in a network of leaky integrate-and-fire neurons. Different forms of the loss function could lead to nonlinear neural networks (Olshausen and Field, 1996).

The coordinated spike-coding framework can provide a substrate for linear and nonlinear computations (Boerlin et al., 2013; Alemi et al., 2017), distributed spatio-temporal representation of uncertainty (Savin and Deneve, 2014), and working memory (Boerlin and Deneve, 2011). The framework is reviewed in detail in Chapter 2.

Alternatively, precise spike information could also be used to fit data-driven phenomenological models. Optimal, model based decoding can recover more information about the stimulus, when the correlated activity is taken into account (Pillow et al., 2008). This is in contrast to a modeling approach in which neurons are assumed to act like independent Poisson units, and where any observed correlations in recorded activities are ignored.

1.4 | Self-organized learning in neural networks

The focus of this thesis is learning of local connectivity in networks, which objective is to represent input signals efficiently under the constraints of experimentally observed connection probability and synaptic transmission delays. We now review some of the commonly used learning rules, with a focus on spike-timing dependent learning. This provides context for the synaptic rule that we will introduce in Chapter 3.

1.4.1 | From Hebbian to STDP learning

Synaptic modification during the approximately simultaneous firing of two connected neurons has been postulated as a potential neurobiological basis of learning since the late nineteenth century (James, 1890). This was formulated more precisely by Donald Hebb who postulated that the synapse between two nearby neurons will grow in strength when the activity of the presynaptic neuron repeatedly contributes to the firing of the postsynaptic neuron (Hebb, 1949). The proposed long-term potentiation (LTP) of the synapse was later observed along with long-term depression (LTD) in the hippocampus (Bliss and Lomo, 1973; Bliss and Gardner-Medwin, 1973), and in many other brain regions, including the neocortex (Artola and Singer, 1987; Iriki et al., 1989; Hirsch and Crepel, 1992). The LTP/LTD of synapses was induced in response to a high/low frequency of presynaptic firing and a correlated high/weaker postsynaptic depolarization. It was also observed that the precise order of presynaptic and postsynaptic activity matters, so that the synapse is potentiated when the postsynaptic neuron fires after the presynaptic spike, and depressed when this order is reversed (Levy and Steward, 1983; Debanne et al., 1994, 1997).

Precise spike-timing-dependent plasticity (STDP), where the order and the time difference between spikes matter, was predicted as a model of the emergence of selectivity to sound location in barn owls (Gerstner et al., 1996). Dual whole-cell recordings were then used to control the precise timing of pre- and postsynaptic spikes, and confirmed that the sign and magnitude of long-term synaptic plasticity depend on the temporal difference between the presynaptic and postsynaptic spikes (Markram et al., 1997). Potentiation was induced when the time difference was positive, that is when the postsynaptic spikes

followed the presynaptic ones, and depression was induced when the time difference was negative. This form of STDP, which was originally discovered, was thus a Hebbian form of plasticity. STDP was then reproduced in subsequent experiments (Zhang et al., 1998; Bi and Poo, 1998; Debanne et al., 1998; Sjostrom et al., 2001) and has since been extensively observed in animal models, ranging from insects to mammals, and amply used in computational models of learning in neural networks. A few selected reviews are given in (Dan and Poo, 2004; Caporale and Dan, 2008; Feldman, 2012).

STDP provided a solution to the instability of Hebbian plasticity, which because of its positive feedback leads to divergent synaptic weights. Before STDP was discovered, theoretical models of Hebbian plasticity were supplemented by other forms of plasticity, which rendered it stable and also accounted for competition between synapses. Examples are the classical Oja rule, which multiplicatively scales down all weights proportionally to the square of the postsynaptic activity (Oja, 1982), and the BCM rule, which implements competition via a sliding threshold between LTP and LTD (Bienenstock et al., 1982). Both of these classical rate-based learning rules are expressed as a sum of a Hebbian learning term, which is synapse specific, and a non-Hebbian learning term, which acts across multiple synapses. The Hebbian modification of the synapse is proportional to the product of the presynaptic and postsynaptic firing rates, and the non-Hebbian term stabilizes the network activity by adaptively controlling the output firing rate (Abbott and Nelson, 2000; Zenke et al., 2017; Benucci et al., 2013).

With the discovery of Hebbian STDP, illustrated in (Fig. 1.3a,b), Hebbian learning could naturally be stabilized and induce competition. In Hebbian STDP, LTP is induced when presynaptic spikes lead postsynaptic spikes within a 20 ms time window, with LTD being induced when the order is reversed. This naturally brings about competition between synapses and results in input selectivity. Other forms of STDP were later observed, such as anti-Hebbian STDP (Fig. 1.3c), where LTP is induced by postsynaptic spikes leading presynaptic spikes (Fino et al., 2005, 2008; Roberts et al., 2010). Furthermore, there are other forms of non-Hebbian STDP (Fig. 1.3d,e), which testify to the variability of the actual relationship between the spike order and the sign of the resulting synaptic modification (Han et al., 2000; Lu et al., 2007; Zhao and Tzounopoulos, 2011; Requarth and Sawtell, 2011).

Importantly, the sign and the scale of the observed synaptic modifications depends on the type of the presynaptic and postsynaptic neuron (Fig. 1.3f-h), reviewed in (Caporale and Dan, 2008). Most observed plasticity of excitatory and inhibitory synapses onto pyramidal cells is approximately Hebbian, while the excitatory synapses onto inhibitory interneurons are characterized by Hebbian or some form of non-Hebbian plasticity, depending on the interneuron type and brain region (Feldman, 2012).

Although spike-timing-dependent plasticity has received a lot of experimental and theoretical support, it has also been debated whether spike times are crucial for learning (Lisman and Spruston, 2005, 2010). There are some observations of plasticity being induced by subthreshold postsynaptic voltage depolarizations, and by other mechanisms in the absence of a postsynaptic spike. An example of a voltage based plasticity model is given in (Clopath et al., 2010).

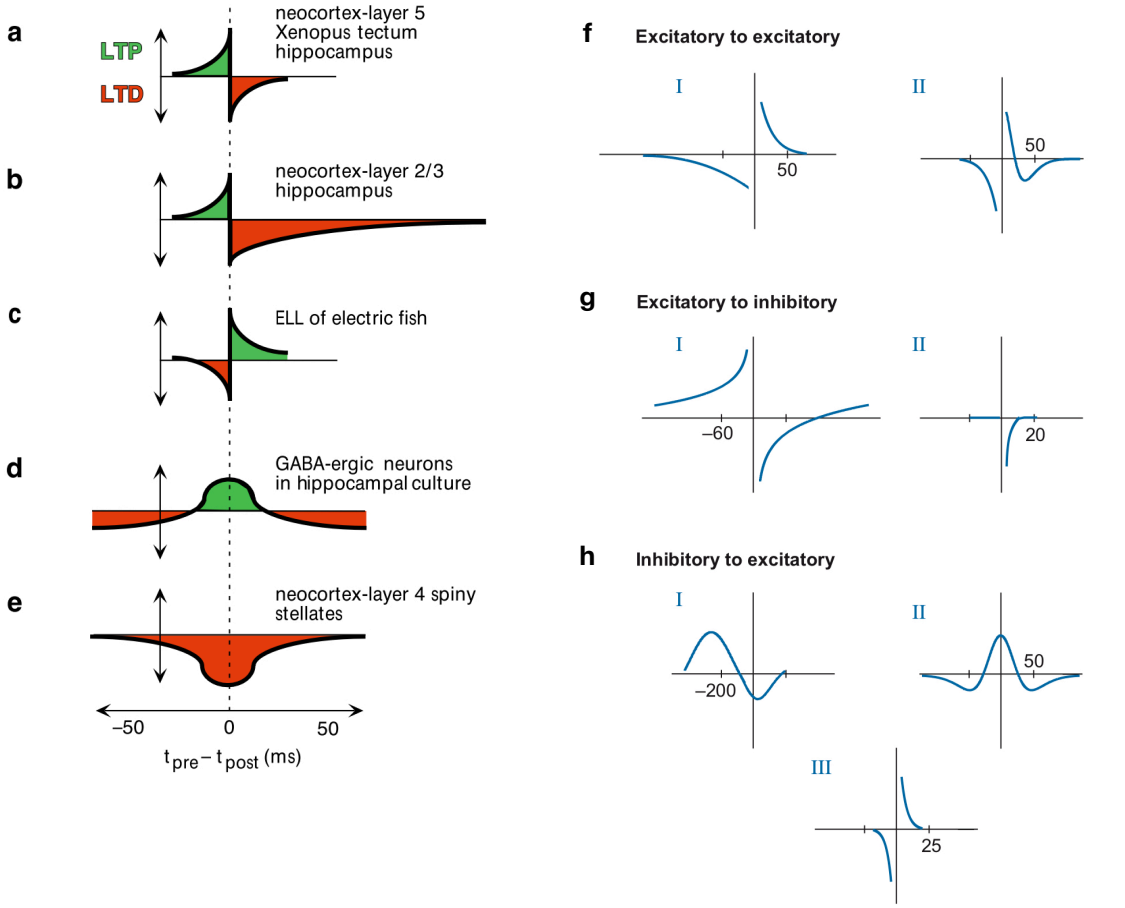


Figure (1.3) The different types of synaptic modification via STDP. **(a-e)** The amount and type of synaptic modification (STDP) evoked by repeated pairing of pre- and postsynaptic action potentials in different preparations, reproduced with the exact caption from (Abbott and Nelson, 2000). The horizontal axis is the difference $t_{pre} - t_{post}$ between the times of these spikes. The numerical labels on this axis are approximate and are only intended to give an idea of the general scale. Results are shown for slice recordings of neocortex layer 5 and layer 2/3 pyramidal neurons (Markram et al., 1997; Feldman, 2000) and layer 4 spiny stellate cells (Egger et al., 1999), in vivo recordings of retino-tectal synapses in Xenopus tadpoles (Zhang et al., 1998), in vitro recordings of excitatory and inhibitory synapses from hippocampal neurons (Levy and Steward, 1983; Gustafsson et al., 1987; Debanne et al., 1994; Magee and Johnston, 1997; Debanne et al., 1998; Bi and Poo, 1998) and recordings from the electrosensory lobe (ELL), a cerebellum-like structure in mormyrid electric fish (Bell et al., 1997). **(f-h)** Diversity of temporal windows for STDP induction, reproduced with the exact caption from (Caporale and Dan, 2008). (f) Windows for excitatory to excitatory connections. (g) Windows for excitatory to inhibitory connections. (h) Windows for inhibitory to excitatory connections. Temporal axis is in milliseconds. Note that positive time intervals in (f-h) correspond to presynaptic spikes occurring before postsynaptic spikes (reverse of graphs in a-e).

1.4.2 | Learning in balanced networks

The connectivity in balanced networks can be learned with Hebbian spike-timing dependent synaptic plasticity rules (Song et al., 2000; Morrison et al., 2007). Different models of learning lead to different final synaptic connectivities, with some developing strongly connected sub-networks (Izhikevich et al., 2004), or synaptic pruning (Iglesias et al., 2005). Most of the studies of balanced networks focus on their dynamics, and assume fixed recurrent connections.

The references cited in the previous paragraph demonstrate the learning of an average balance of excitatory and inhibitory inputs, which is not temporally precise. However, there is one proposed learning rule, which can learn a more temporally precise E/I balance in random networks. We review it in the following subsection, because we are interested in networks which can learn to balance their inputs in a temporally precise manner. A precise E/I balance can be seen as a signature of a network, which represents its inputs with high efficiency.

Learning a more temporally precise balance in random networks

The temporally precise balance between excitatory and inhibitory currents can be maintained with Hebbian learning at inhibitory synapses (Vogels et al., 2011). Following the proposed learning rule, the inhibitory synapses onto an excitatory readout neuron get strengthened when the pre- and post-synaptic spikes occur in close temporal proximity. Also, independently of postsynaptic activity, every presynaptic spike leads to a small weakening of the synapse. The two effects can be summarized with the learning rule:

$$\triangle w = \eta(\text{pre} \times \text{post} - \rho_0 \times \text{pre}), \tag{1.1}$$

where the change in the inhibitory synaptic efficacy, $\triangle w$, is effected at every spike time with a learning rate η. The first term potentiates the synaptic efficacy in proportion to the pre- and post-synaptic activities, when both neurons fire spikes within a short coincidence time window. The second term implements homeostasis, and brings the postsynaptic rate to a target firing rate, ρ_0. This plasticity rule has the general shape of bidirectional Hebbian learning (Fig. 1.3h, panel II).

In a simulation described in this paper, a single integrate-and-fire readout neuron receives 1,000 synapses, which transmit information about eight different signal channels. All synapses are conductance based with transmission delays, with one-fifth of the synapses being inhibitory and plastic. Over the course of learning, the proposed spike-timing-dependent plasticity rule reorganizes an unbalanced network into one which is characterized by a global ('loose') balance state. With the balance being maintained on average, only unbalanced excitatory inputs lead to coincident inhibitory and excitatory spikes, and this sharpens the tuning curves of inhibitory synapses. Eventually, the balancing is precise for all eight input signal channels, and the inhibitory synapses have the same tuning properties as the excitatory ones.

The firing of the readout neuron was sparse after the precise balance was attained, in agreement with experimental observations (Hromadka et al., 2008; Deweese et al., 2003; Yao et al., 2007; Crochet et al., 2011). The spikes were caused by transient changes in the

feedforward input, when the faster dynamics of excitatory synapses lead to a temporary imbalance.

The balance can be maintained even when excitatory synapses are plastic, as in the case of the experimental change of preferred stimulus frequency during a stimulus pairing with acetylcholine release (Froemke et al., 2007). In (Vogels et al., 2011), this is modeled by the excitatory synapses changing their preferred tuning to the paired stimulus, thus causing an E/I imbalance in the output neuron. This then recruits the Hebbian STDP inhibitory plasticity, and the inhibitory synapses learn to balance the new excitatory tuning curve. At the end of learning, the excitatory and inhibitory input currents are co-tuned at the new frequency, with the output neuron having a new preferred stimulus frequency. For this kind of self-organized balancing to work in the presence of excitatory plasticity, the inhibitory plasticity needs to occur on a faster time scale than the excitatory one. Alternatively, the excitatory plasticity events should not be too frequent.

In the absence of synaptic delays, the rate based version of the learning rule (1.1) performs a gradient descent on an energy function. The energy function measures the mean square difference between excitation and inhibition, and the difference of the postsynaptic firing rate from the target rate, ρ_0. However, if the initial state of the network is such that inhibitory neurons do not receive the same information as the one available in excitatory synapses, then the learning rule might only converge to a global balance. If inhibitory neurons have wider tuning properties, or are not tuned at all to the input signals (Hofer et al., 2011), then the inhibitory plasticity leads only to a global balance. This could be an explanation for the variability of observed co-tuning properties of excitation and inhibition in sensory cortices (Wu et al., 2008).

Owing to this learning rule, the network adapts its inhibitory response, so that the network responds to a feedforward input transiently (Murphy and Miller, 2009) during the time lag before inhibitory inputs cancel the excitatory ones entirely. The resulting asynchronous network activity provides a mechanism for reliable signal transmission across multiple layers, in the presence of noise (Vogels and Abbott, 2005; Kumar et al., 2010). The network can also respond persistently when the balance is disrupted in a targeted way (Shu et al., 2003; Vogels and Abbott, 2009). This inhibitory learning rule was later observed in the mouse auditory cortex (Damour and Froemke, 2015).

For a review of other theoretical models of inhibitory STDP and the related experimental observations, see (Hennequin et al., 2017).

1.4.3 | Learning in spike-coding networks

A temporally precise E/I balance can also be learned in coordinated spike-coding networks (Brendel et al., 2017). This is a voltage based rule, which is derived as a minimization of an energy function. The energy function measures the deviation of the difference between excitatory and inhibitory inputs from zero, similar to the energy function, which was used to derive the inhibitory plasticity rule of (Vogels et al., 2011). The learning rule converges to the structured connections from the coordinated spike-coding network (Boerlin et al., 2013), and describes both excitatory and inhibitory plasticity.

Other learning rules, which rely on precise spike coding, have been developed to perform

more complex computations, such as to implement nonlinear dynamics (Alemi et al., 2018; Gilra and Gerstner, 2017; Thalmeier et al., 2016). However, since the topic of this thesis is learning efficient input representation, we are particularly interested in the plasticity rule proposed in (Brendel et al., 2017). This balance restoring plasticity rule will be reviewed in detail in Chapter 2.

1.5 | Thesis organization

Representation of sensory variables in biological spiking networks is limited by the cost of firing action potentials, synaptic dynamics, and the fixed volume that can be filled by synaptic connections. Under the constraints of biologically observed connection probabilities and synaptic dynamics, we ask whether a spiking neural network can learn to represent sensory variables accurately while firing a minimal number of spikes.

In contrast to the idealized coordinated spike-coding framework (Boerlin et al., 2013), in which the optimal signal representation is achieved by a network with symmetric all-to-all recurrent connections, we address optimality in the context of sparser connections. We use the word 'sparse' somewhat casually, to represent connection probabilities which are less than unity. We investigate the properties of spike-coding networks, across the range of connectivity levels: from a purely feedforward network, all the way to a network with full recurrent connectivity. We also consider the impact of delays in synaptic transmission, as well as increased dimensionality of feedforward inputs.

The coordinated spike-coding framework is reviewed in Chapter 2, along with the synaptic rule which converges to the structured all-to-all recurrent connections.

In Chapter 3, we introduce a synaptic rule which optimizes two goals. It aims to balance the excitatory and inhibitory inputs while concurrently pruning those recurrent connections, which affect the accurate signal representation the least. We find an explicit expression for the steady state of this learning rule for particular inputs, and analyze the implicit steady state for more general inputs.

In Chapter 4, we simulate the synaptic rule in recurrent non-Dale networks of leaky integrate-and-fire neurons with instantaneous synapses. We analyze the convergence of the learning rule, and the activity of the trained networks. We also briefly compare the efficiency of signal representation in trained sparse networks, to two alternative sparse networks.

In Chapter 5, we adapt the synaptic rule to networks which are characterized by synaptic transmission delays and dynamics. We simulate the learning rule in non-Dale networks of leaky integrate-and-fire neurons with synaptic delays and dynamics, and analyze the structure of the final recurrent weights. We compare these results to those from Chapter 4.

In Chapter 6, we consider the impact of higher dimensional inputs on the efficiency of representation in sparse networks. We investigate the impact of network size and input gain on the efficiency of representation.

We conclude with Chapter 7, where we discuss the thesis results in the wider context, provide a few additional biologically relevant results, and give a future outlook.

2 | Review of Coordinated Spike-Coding

2.1 | Motivation

Although the action potential, or spike, is considered to be the basic mechanism of information sharing in cortical neural networks, the classical approach to modeling population activity is based on average neural firing rates, and not on precisely timed spikes. The precise timing and number of spikes, recorded *in vivo*, are highly variable on different presentations of the same stimulus, and are thus not reproducible across trials (Tolhurst et al., 1982; Shadlen and Newsome, 1998). Thus, the timing of a spike is generally not seen as a deterministic function of the stimulus, but rather as a realization of a stochastic process (Perkel et al., 1967). In the rate based modeling approach, one generally averages the time-binned spike counts, to obtain the time-varying average firing rates. Spikes are then seen as Poisson point processes, with the rate of the underlying Poisson distribution given by the average firing rate. The average rate itself can then be interpreted as a deterministic function of the stimulus, a reproducible neural correlate of sensory information.

The idea that the firing rate is a good way to summarize the activity of a neuron has been mainstream since the early electrophysiological experiments, where the strength of the external stimulus was found to modulate the number of spikes produced by the recorded neurons (Adrian and Zotterman, 1926). However, the implementation of rate based models in recurrent networks was not straightforward. The implementation relies critically on the balance of inhibitory and excitatory inputs into each neuron, with the proposed models referred to as balanced (rate) networks (Shadlen and Newsome, 1994; van Vreeswijk and Sompolinsky, 1996; Amit and Brunel, 1997; Shadlen and Newsome, 1998; Brunel, 2000). Without the constraint of balance, rate models would not be able to reproduce the irregular spiking of neurons, as integrating random spike trains received by a postsynaptic neuron, usually leads to it spiking at regular time intervals (Softky and Koch, 1993). Motivated by the theoretical constraint of balanced networks, experiments were conducted to test whether it were true, and found that neurons do indeed receive balanced excitatory and inhibitory inputs (Shu et al., 2003; Wehr and Zador, 2003; Isaacson and Scanziani, 2011).

However, it is peculiar that a biological neural network, which represents information in its spike counts, would do so by firing spikes at random times (Deneve and Machens, 2016). This incurs a large metabolic cost (Attwell and Laughlin, 2001), as the precision

of the encoded signal does not scale up well with the number of spikes, M (Fig. 2.1a,b). If a continuous variable is represented in the discrete spike counts, as in analog to digital conversion, then the error of the representation is a consequence of the necessary discretization, and should scale down with $1/M$. However, due to the unreliability of spike counts, the coding error of the Poisson rate code scales down only with $1/\sqrt{M}$. The rate code thus has to fire many more spikes than necessary, for a given level of precision (Fig. 2.1c). Solutions to this problem have been proposed by using the relative timing of spikes and interpreting a neuron as a coincidence detector (Abeles, 1982; Hopfield, 1995). However, indication of an implementation in recurrent networks was usually missing.

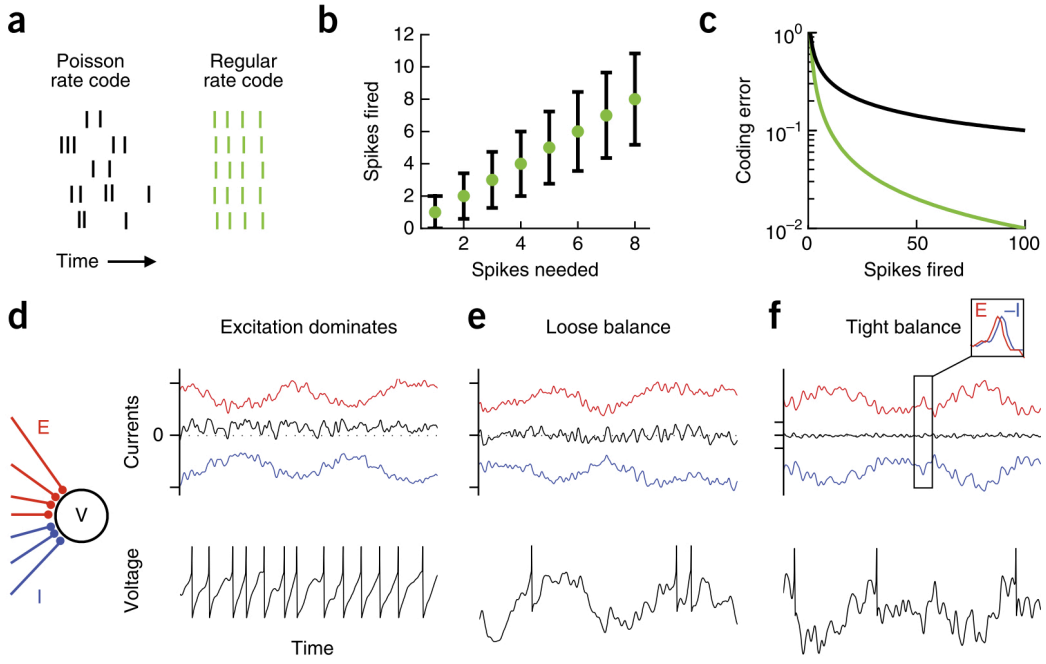


Figure (2.1) The conundrum of Poisson rate codes and E/I balance, reproduced with the exact caption from (Deneve and Machens, 2016). **(a)** When seeking to produce an average rate of 4 spikes per second, the number of spikes in each trial will fluctuate for a Poisson rate code (left), yet remains constant for a perfect, regular rate code (right). **(b)** As the average rate or number of spikes per second needed grows, the variability of the Poisson rate code (s.d.; black bars) grows as well. **(c)** When encoding an analog number, any rate code will cause coding errors due to the unavoidable discretization of the spike count. While these coding errors drop as $1/M$ for M spikes (green), the Poisson rate coding errors drop only as $1/\sqrt{M}$ owing to the additional problems caused by unreliability (black). **(d)** In a neuron receiving both excitatory and inhibitory currents, even a small imbalance toward excitation will cause a spike train output that is far more regular than a Poisson spike train. **(e)** If excitatory and inhibitory currents are balanced on a slow time scale yet exhibit fast and uncorrelated (loose) fluctuations, then the net input current will depend on these faster fluctuations only, causing a random walk to threshold in the membrane potentials. **(f)** If the faster fluctuations are strongly correlated but shifted in time-for example, by inhibition trailing excitation (see inset)-then the system moves into a regime of tight E/I balance.

The type of balance needed to produce irregular spiking by balanced networks is often referred to as 'loose', as inhibition and excitation need only balance on larger time scales (Fig. 2.1e vs. d). There is also experimental evidence for a temporally precise balance, in which inhibitory inputs balance excitatory ones with a millisecond resolution delay (Okun and Lampl, 2008; Graupner and Reyes, 2013; Tan et al., 2013; Salkoff et al., 2015), as illustrated in (Fig. 2.1f). Balanced networks can incorporate this 'tight' balance (Renart et al., 2010; Vogels et al., 2011). But the question still remains why information would be represented in noisy spike trains, and not in the precise timing of the spikes which, as we will see in the rest of this Chapter, can lead to more efficient signal representation. For details on the history of the rate versus spike coding debate, see (Brette, 2015), review; for details on efficient codes and balanced networks, see (Deneve and Machens, 2016), review.

2.2 | Efficient representation with coordinated spike-coding

A recurrent network, which uses the precise timing of its spikes to communicate information, can represent a signal more efficiently than a Poisson rate code (Boerlin et al., 2013). This was shown to be true for networks which recurrent connections are not random like in balanced networks, but instead have a specific structure, derived from an optimality principle. This kind of a spike coding recurrent network can represent a signal precisely with a minimal number of spikes.

This model was originally introduced under the name of predictive coding, because the recurrent network can learn the structure of the input signal and subtract it via negative feedback (Rao and Ballard, 1999). It can also be related to sparse coding with an overcomplete basis set (Olshausen and Field, 1997).

However, the essence of this model is the coordination of spikes for efficient signal representation, and we therefore refer to it as coordinated spike-coding (CSC). The optimal recurrent weights were derived from a top-down normative approach, in which the goal of a neural network is to represent a signal precisely with a minimal number of spikes (Boerlin et al., 2013).

The activity of the network was proposed to minimize a loss function, which measures the error and the cost of spiking. The abstract energy minimization process was then related to the membrane potential, as a measure of the coding error, and to spikes, as the error-correcting mechanism. A spike indicates that the error has exceeded a threshold. When the recurrent weights are set to their optimal values, the effect of the fired spike on the postsynaptic neurons is inhibitory for similarly tuned neurons, and excitatory for dissimilarly tuned neurons. This prevents similarly tuned neurons from firing their own spikes to correct the error, and ensures that the voltage of dissimilarly tuned neurons is at the resting state. The rest state here is equated with a zero coding error.

The recurrent synaptic currents are modeled as impulse functions, and so the balance between excitation and inhibition, is instantaneous. Although, synaptic delays and dynamics were added to the model, and still produced more efficient signal representation than a Poisson rate code (Schwemmer et al., 2015; Chalk et al., 2016). With the efficient signal representation being achieved with fast synapses, computations in this network are accomplished with a set of synapses with finite temporal dynamics. They can be used to

implement linear dynamical systems, such as a harmonic oscillator and a 2D arm controller (Boerlin et al., 2013), as well as nonlinear dynamics (Alemi et al., 2018). Both linear and nonlinear computations can be learned via local supervised learning rules (Bourdoukan and Deneve, 2015; Alemi et al., 2018). Furthermore, this network can also perform optimal sensory cue combination and implement persistent working memory on a time scale much longer than the time constant of a neuron (Boerlin and Deneve, 2011).

The coordinated spike-coding network produces biologically observed asynchronous and irregular spike trains, even though the spiking is coordinated. Furthermore, the subthreshold membrane potentials of similarly tuned neurons are correlated, as experimentally observed (Yu and Ferster, 2010; Gentet et al., 2010).

This model proposes an alternative view of network firing variability to that of the classical rate based approaches, and it posits "the brain as an efficient and robust adaptive learner" (Alemi et al., 2017). In this framework, a disease that kills neurons, depending on the number of neurons that were affected, would not prevent a neural circuit from performing its function (Barrett et al., 2016). The remaining neurons would compensate for the neuron loss, and increase their firing rates to keep the signal representation precise. The essence of its optimality relies on the optimal recurrent connections, which can be learned with a local unsupervised plasticity rule, using rates (Vertechi et al., 2014) and spikes (Brendel et al., 2017). We will discuss the details of this learning rule in this chapter, but first we will introduce the details of the model itself. We will derive the CSC model for a network, which function is efficient signal representation. For details of computations with CSC networks, see the relevant articles cited above.

2.3 | Details of the CSC model: Implementation in a recurrent network

The coordinated spike-coding model was originally derived from an optimality principle (Boerlin et al., 2013). The goal of a neural network is postulated to be the minimization of a loss function, which measures the coding error and the cost of firing. The network is derived from this computational goal, first as an algorithmic solution based on an autoencoder with an error feedback, and then the error feedback is implemented by recurrent connections of the network. The model thus provides descriptions at all three levels of Marr's analysis (Marr, 1982), and is derived in a top-down fashion. Here, we present the CSC model in the reverse order, from the implementational level, and then show how it can accomplish the computational goal of efficient signal representation.

We thus begin with a network of recurrently connected neurons, which membrane potential dynamics is described by the leaky integrate-and-fire (LIF) model. The LIF model captures the subthreshold voltage dynamics well without the added complexity of modeling the stereotypical action potentials, which occur on a much faster time scale (Lapicque, 1907; Hill, 1936), see (Burkitt, 2006) for a review of different variants of the LIF model.

2.3.1 | Voltage dynamics

The temporal dynamics of the subthreshold membrane voltage, $V_i(t)$, of neuron i, is given by:

$$\dot{V}_i(t) = -\lambda V_i(t) + \sum_{k=1}^{K} F_{ik} c_k(t) + \sum_{j=1}^{N} \Omega_{ij} o_j(t), \tag{2.1}$$

for $t > 0$ and $i = \{1, 2, ..., N\}$. Here, λ is the inverse of the membrane time constant, F_{ik} is the linear encoder weight, $c_k(t)$ the feedforward input current, Ω_{ij} the strength of the recurrent current received from neuron j, and $o_j(t)$ is the spike train, which equals the Dirac delta function if neuron j spiked at time t, and is zero otherwise. A spike is modeled as an instantaneous event, which occurs when $V_i(t)$ reaches the firing threshold, T. At that point, the spike train $o_i(t)$ is set to equal the Dirac delta function:

$$o_i(t) = \begin{cases} 0 & \text{if } V_i(t) < T, \\ \delta(t) & \text{if } V_i(t) = T, \end{cases} \tag{2.2}$$

and the subsequent resetting of the voltage is modeled by the negative self-connection, Ω_{ii}, in (2.1). We also note that the linear encoding weights, F_{ik}, are picked randomly, and then the length of the encoding vector, $\mathbf{F}_i = (F_{i1}, F_{i2}, ..., F_{iK})$, is normalized to equal one. The input currents, $c_k(t)$, could be smoothly varying or constant.

Integrating the voltage dynamics shows that the neural voltage equals the sum of the filtered feedforward and recurrent currents, at any point in time:

$$V_i(t) \;=\; \sum_{k=1}^{K} F_{ik} \int_0^t c_k(t') e^{-\lambda(t-t')} dt' + \sum_{j=1}^{N} \sum_{n=1}^{M_j} \Omega_{ij} e^{-\lambda(t-t_j^n)},$$

where t_j^n are the spike times of neuron j, with $t_j^n < t$. Since spikes are modeled as delta functions, each new spike contributes to the postsynaptic voltage a synaptic kernel, or filtered spike, $e^{-\lambda(t-t_j^n)}$, which is weighted by the recurrent strength, Ω_{ij}.

We will now introduce two new variables, and rewrite the neural voltage as:

$$V_i(t) = \sum_{k=1}^{K} F_{ik} x_k(t) + \sum_{j=1}^{N} \Omega_{ij} r_j(t). \tag{2.3}$$

Here, $x_k(t)$ is the leaky integral of the feedforward input current, $c_k(t)$:

$$x_k(t) = \int_0^t c_k(t') e^{-\lambda(t-t')} dt', \tag{2.4}$$

and $r_j(t)$ is the instantaneous firing rate, of filtered spike train, of neuron j:

$$r_j(t) = \sum_{n=1}^{M_j} e^{-\lambda(t-t_j^n)}. \tag{2.5}$$

If the recurrent weights have a very specific low dimensional structure, and can be represented as:

$$\Omega_{ij} = -\sum_{k=1}^{K} F_{ik} D_{kj}, \tag{2.6}$$

where the weights, D_{kj}, will be defined shortly, then (2.3) can be rewritten as:

$$\begin{aligned} V_i(t) &= \sum_{k=1}^{K} F_{ik} x_k(t) - \sum_{j=1}^{N} \sum_{k=1}^{K} F_{ik} D_{kj} r_j(t), \\ &= \sum_{k=1}^{K} F_{ik} \Big(x_k(t) - \sum_{j=1}^{N} D_{kj} r_j(t) \Big), \\ &= \sum_{k=1}^{K} F_{ik} \big(x_k(t) - \hat{x}_k(t) \big). \end{aligned} \tag{2.7}$$

Here, we have introduced a new variable, $\hat{x}_k(t) = \sum_{j=1}^{N} D_{kj} r_j(t)$.

2.3.2 | Interpretation of the network's function

The coordinated spike-coding framework makes the following interpretation of the above variables, and the overall function of the neural network.

The goal of the network is to represent a dynamic sensory variable $x_k(t)$, which is a leaky integral of the feedforward current, $c_k(t)$, and has the following dynamics:

$$\dot{x}_k(t) = -\lambda x_k(t) + c_k(t). \tag{2.8}$$

The network should represent the variable in its spike trains, with the reconstructed variable, $\hat{x}_k(t)$, being decoded from the filtered spike trains:

$$\hat{x}_k(t) = \sum_{j=1}^{N} D_{kj} r_j(t). \tag{2.9}$$

Here, D_{kj}, are the decoding weights used by a readout neuron. Differentiating the above expression, gives the dynamics of the network reconstruction of the sensory variable:

$$\dot{\hat{x}}_k(t) = -\lambda \hat{x}_k(t) + \sum_{j=1}^{N} D_{kj} o_j(t). \tag{2.10}$$

To summarize, the goal of the network is to represent a continuous variable, $x_k(t)$, in its discrete population spikes. The reconstructed variable, $\hat{x}_k(t)$, can be decoded linearly from the population spike train. An example of input reconstruction with the subtheshold voltage profiles is shown in (Fig. 2.2).

With this interpretation of variables, $x_k(t)$ and $\hat{x}_k(t)$, we now see that (2.7) represents the voltage of each neuron as the projection of the reconstruction error, $x_k(t) - \hat{x}_k(t)$, onto

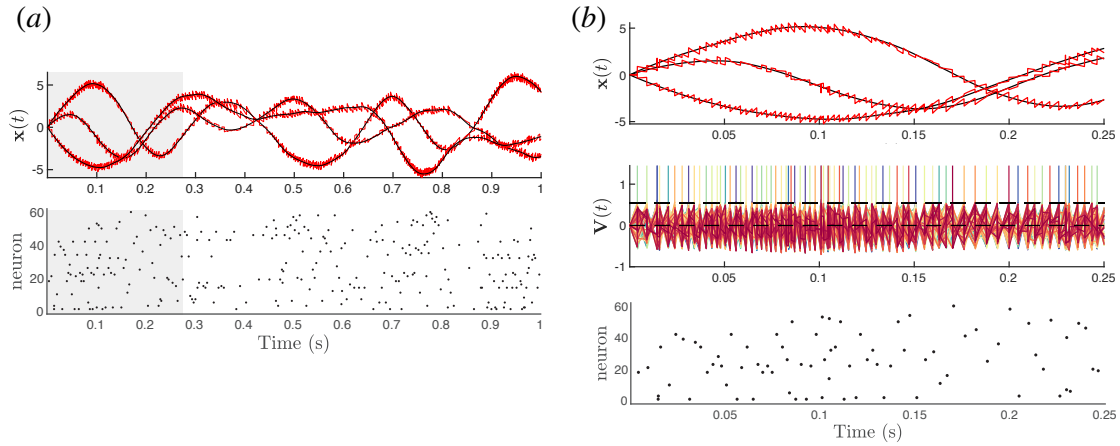


Figure (2.2) Input reconstruction by the Coordinated Spike-Coding model. A three-dimensional variable is represented by a network of 60 neurons in a CSC network with optimal recurrent connections. **(a)** Top: Input variable $x_k(t)$ (black), network reconstruction $\hat{x}_k(t)$ (red). Bottom: The raster plot showing the spike trains of the 60 neurons. The grey shaded area corresponds to the first 250 ms and is shown in a close-up in (b). **(b)** A close-up for the first 250 ms of input reconstruction, shown in (a). The middle panel is added to show the population subthreshold voltages.

its encoding weight. The neuron fires a spike when the projection of the error onto its encoding weight exceeds the threshold (Fig. 2.3, purple loop). The spike acts to correct the reconstruction, and the voltage of other neurons in the population is immediately updated to take this into account. At an algorithmic level, one can replace the recurrent connections with feedback connections as follows. One can imagine a network with no recurrent connections, which receives a feedforward input, given by the analog/continuous variable, $x_k(t)$, that it needs to represent in its spiking activity. The network also receives a negative feedback input, the reconstructed digital/discrete variable, $\hat{x}_k(t)$. The overall input into the network is the reconstruction error, $x_k(t) - \hat{x}_k(t)$.

Algorithmically, coordinated spike-coding is set up like an autoencoder (Hinton and Zemel, 1994), and its activity is driven by the error, which is proportional to the signal, but much smaller than it. This is key to the efficiency of the CSC model. The fact that its activity is driven by the error, rather than by the signal, leads to a minimal number of fired spikes. This is in contrast to a feedforward LIF network, without any negative feedback, which activity is driven by the full signal (Fig. 2.3, orange loop). This network will fire many more spikes, as the global information about the error is not shared with individual neurons. In the independent LIF model, each neuron will fire error-correcting spikes as if it were acting alone, and not as a part of a population with the same overarching goal.

At an implementational level, the feedback loop is implemented by the optimal recurrent connections, which we will discuss along with the optimal decoder in the next section. The effect of the optimal recurrent connections on the population activity is the same as the effect of the feedback loop. Both mechanisms update the local error seen by each neuron in the CSC network, where the local error is measured by $V_i(t)$.

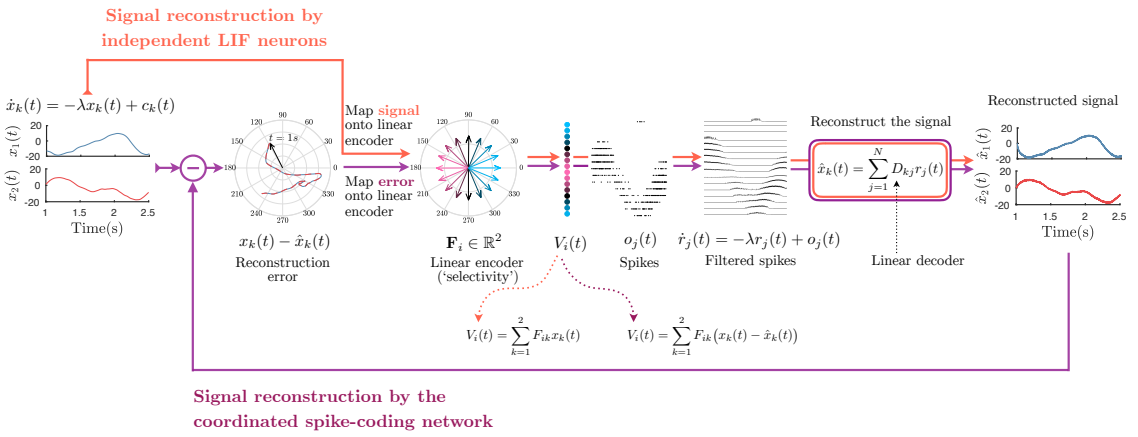


Figure (2.3) Algorithmic description of coordinated spike-coding (purple loop). Description of the signal reconstruction by independent LIF neurons is shown for contrast (orange loop). Signal representation in coordinated spike-coding is implemented via a recurrent network of LIF neurons. To understand how this network reconstructs a signal with a minimal number of spikes, we analyze its behaviour at the algorithmic level. Here, the sensory variable of interest is a two dimensional signal $x_k(t)$, $k = \{1, 2\}$, which is a leaky integral of the input current $c_k(t)$, with the time constant given by $\frac{1}{\lambda}$. The network's estimate of this signal is given by $\hat{x}_k(t)$, which is a leaky integral of the linearly decoded population spike train. The activity of neuron i is driven by the reconstruction error $x_k(t) - \hat{x}_k(t)$, which is mapped onto the linear encoder $\mathbf{F}_i \in \mathbb{R}^2$. The subthreshold voltage is interpreted as a measure of the global reconstruction error, and a spike is fired when the error exceeds a threshold. $\hat{x}_k(t)$ can then be linearly read out from the filtered spike trains.

2.4 | Deriving the network structure from an optimality principle

2.4.1 | Loss function for the derivation of an efficient network

The optimal decoder and optimal recurrent connections for the CSC model, are derived from an optimality principle. The network is assumed to minimize a loss function, which measures the coding error, regularized by the cost of firing:

$$L = \left\langle \sum_{k=1}^{K} \left(x_k(t) - \hat{x}_k(t)\right)^2 + \mu \sum_{i=1}^{N} r_i^2(t) + \nu \sum_{i=1}^{N} r_i(t) \right\rangle_t . \tag{2.11}$$

Here, the first term measures the reconstruction error, the second term measures the quadratic cost of firing, weighted by a parameter μ, and the third term measures the linear cost of firing, weighted by a parameter ν. The linear firing cost term measures the total population firing rate, and the larger the ν, the fewer population spikes will be fired. The

quadratic firing cost regularizer is added because the optimal network activity should be distributed among all neurons over time.

2.4.2 | Optimal linear decoder

The function of the CSC network is to represent the time varying input signal as accurately as possible, while minimizing the number of population spikes. Under the assumption that the input variable is reconstructed linearly from the filtered population spike trains, the optimal linear decoder can be found by minimizing the loss function, (2.15), with respect to the decoder:

$$\begin{aligned}\frac{\partial L}{\partial D_{kj}} &= \frac{\partial}{\partial D_{kj}}\Big\langle \sum_{k=1}^{K}\Big(x_k(t)-\sum_{j=1}^{N}D_{kj}r_j(t)\Big)^2\Big\rangle_t,\\ &= -2\Big\langle \sum_{k=1}^{K}\Big(x_k(t)-\sum_{i=1}^{N}D_{ki}r_i(t)\Big)r_j(t)\Big\rangle_t. \end{aligned} \tag{2.12}$$

Setting the gradient to zero,

$$0=\Big\langle \sum_{k=1}^{K}x_k(t)r_j(t)\Big\rangle_t-\Big\langle \sum_{i=1}^{N}D_{ki}r_i(t)r_j(t)\Big\rangle_t,$$

we see that the optimal linear decoder has to satisfy:

$$\Big\langle \sum_{i=1}^{N}D_{ki}r_i(t)r_j(t)\Big\rangle_t=\Big\langle \sum_{k=1}^{K}x_k(t)r_j(t)\Big\rangle_t. \tag{2.13}$$

This is precisely the linear regression solution, which can be written in matrix form as follows:

$$\mathbf{D}=\Big\langle \mathbf{x}(t)\mathbf{r}^T(t)\Big\rangle_t\Big\langle \mathbf{r}(t)\mathbf{r}^T(t)\Big\rangle_t^{-1}. \tag{2.14}$$

Here, the decoder matrix, $\mathbf{D}$, is a $K\times N$ matrix, with its elements given by the decoding weights, D_{kj}. Given that the optimal linear decoder is the linear regression solution, it can be fitted given an input signal, $\mathbf{x}(t)$, and some network activity, $\mathbf{r}(t)$. The optimal decoder can also be used to evaluate the performance of networks with different activities, $\mathbf{r}(t)$.

2.4.3 | Optimal recurrent connections

Having found the optimal decoder, we can now find the optimal recurrent weights, by minimizing the loss function (2.15) with respect to recurrent weights, Ω_{ij}. Since the recurrent weights shape the temporal dynamics of the network, we will find their optimal values by minimizing the time dependent loss function:

$$l(t)=\sum_{k=1}^{K}\big(x_k(t)-\hat{x}_k(t)\big)^2+\mu\sum_{i=1}^{N}r_i^2(t)+\nu\sum_{i=1}^{N}r_i(t), \tag{2.15}$$

where the average loss function, (2.15), is related to $l(t)$, via $L = \langle l(t) \rangle_t$.

The optimal recurrent weights are determined by minimizing this loss function with respect to the weights, in a greedy sense, at the time of each spike.

We suppose that neuron j fires a spike at some time, t. The effect of its spike on the time-dependent loss function, is to add a decoding weight, D_{kj}, to the reconstructed signal, and a unit mass, δ_{ij}, to the filtered population spike train:

$$l(t|\text{neuron } j \text{ spikes}) = \sum_{k=1}^{K} \big(x_k(t) - \hat{x}_k(t) - D_{kj}\big)^2 + \mu \sum_{i=1}^{N} \big(r_i(t) + \delta_{ij}\big)^2 + \nu \sum_{i=1}^{N} \big(r_i(t) + \delta_{ij}\big).$$

Here, δ_{ij} is the Kroenecker delta which equals 1 when $i = j$, and is zero otherwise.

Since the loss function is minimized at the time of each spike, a neuron j should fire a spike if $l(t|\text{neuron } j \text{ spikes}) < l(t|\text{neuron } j \text{ does not spike})$.

This then leads to the spiking condition:

$$-2\sum_{k=1}^{K} \big(x_k(t) - \hat{x}_k(t)\big) D_{kj} + \sum_{k=1}^{K} D_{kj}^2 + 2\mu r_j(t) + \mu + \nu < 0,$$

which is equivalent to:

$$\sum_{k=1}^{K} D_{kj}\big(x_k(t) - \hat{x}_k(t)\big) - \mu r_j(t) > \frac{1}{2}\Big(\sum_{k=1}^{K} D_{kj}^2 + \mu + \nu\Big). \tag{2.16}$$

In the coordinated spike-coding framework, the LHS is defined as the time-varying membrane potential of neuron j, and the RHS as the firing threshold of that neuron. That is,

$$V_j(t) = \sum_{k=1}^{K} D_{kj}\big(x_k(t) - \hat{x}_k(t)\big) - \mu r_j(t), \tag{2.17}$$

$$T_j = \frac{1}{2}\Big(\sum_{k=1}^{K} D_{kj}^2 + \mu + \nu\Big). \tag{2.18}$$

The voltage of each neuron is the global reconstruction error projected onto encoding weights, and there is an additional firing cost term, which we ignored in section 2.3 for simplicity.

Taking the time derivative of voltage, we recover the LIF dynamics (2.1), but this time with the optimal recurrent connections and optimal encoding weights:

$$\begin{aligned}
\dot{V}_j(t) &= \sum_{k=1}^{K} D_{kj}\big(\dot{x}_k(t) - \sum_{i=1}^{N} D_{ki}\dot{r}_i(t)\big) - \mu\dot{r}_j(t), \\
&= \sum_{k=1}^{K} D_{kj}\big(-\lambda x_k(t) + c_k(t)\big) - \sum_{k=1}^{K}\sum_{i=1}^{N} D_{kj}D_{ki}\big(-\lambda r_i(t) + o_i(t)\big) \\
&- \mu(-\lambda r_j(t) + o_j(t)), \\
&= -\lambda V_j(t) + \sum_{k=1}^{K} D_{kj}c_k(t) - \sum_{k=1}^{K}\sum_{i=1}^{N} \big(D_{kj}D_{ki} + \mu\delta_{ij}\big) o_i(t).
\end{aligned} \tag{2.19}$$

Therefore, the optimal recurrent weights and optimal encoding weights are given by:

$$\Omega_{ij} = -\sum_{k=1}^{K} D_{ki} D_{kj} - \mu\delta_{ij}, \tag{2.20}$$

$$F_{ik} = D_{ki}, \tag{2.21}$$

for $i, j = \{1, 2, ..., N\}$ and $k = \{1, 2, ..., K\}$, with $K \ll N$. The optimal encoding weights for white inputs equal the decoding weights, see (Brendel et al., 2017) for optimal encoding weights for non-white inputs. The optimal recurrent weights equal the negative of the product of the encoding and decoding vectors. The self connections have an extra μ term added, to ensure that firing is distributed amongst a population which has many neurons with the same encoding vectors.

The decoding vectors, $\mathbf{D}_i = (D_{1i}, D_{2i}, ..., D_{Ki})$, are the rows of the decoder matrix, and the encoding vectors, $\mathbf{F}_i = (F_{1i}, F_{2i}, ..., F_{Ki})$, are the columns of the encoder matrix (Fig. 2.4). For white inputs, which have zero pair-wise correlations for all $k \in \{1, ..., K\}$, the optimal decoding vectors, $\mathbf{D}_i$, cover the K-sphere, as equally as possible, given the total number of neurons N. This also holds true for the optimal encoding vectors.

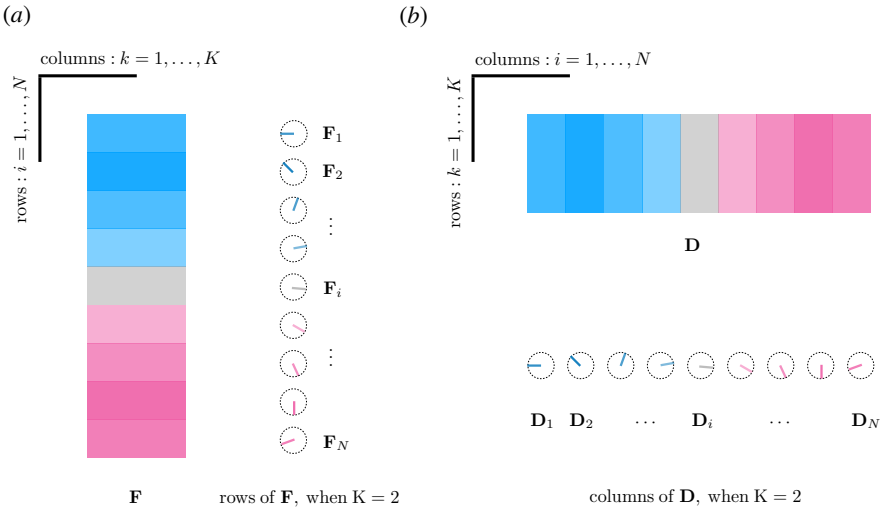


Figure (2.4) Schematic: Optimal linear encoders and decoders for white inputs. The illustration is done for $K = 2$ inputs, which are represented by a network of N neurons, with $N \gg K$. Each neuron has K encoding and decoding weights, which can be organized into vectors, $\mathbf{F}_i$ and $\mathbf{D}_i$. For white inputs, these vectors are optimal when they cover the 2D input space equally. When these vectors are random, the network represents the signal efficiently. The vectors can then be summarized by the encoder and decoder matrices, $\mathbf{F}$ and $\mathbf{D}$. **(a)** The optimal encoder matrix, $\mathbf{F}$, is a $N \times K$ matrix (a tall matrix). Its rows are the linear encoding vectors, $\mathbf{F}_i$. **(b)** The optimal decoder matrix, $\mathbf{D}$, is a $K \times N$ matrix (a flat matrix). Its columns are the linear decoding vectors, $\mathbf{D}_i$. For white inputs, $\mathbf{F} = \mathbf{D}^T$ and the recurrent weights are $\mathbf{\Omega} = -\mathbf{F}\mathbf{D} - \mu\mathbf{I}$, where $\mathbf{I}$ is a diagonal matrix with ones on the diagonal.

We also define a matrix of recurrent weights, $\mathbf{\Omega}$, with elements Ω_{ij}. Each column j of the recurrent matrix contains the synaptic weights sent from neuron j to neuron i.

Since the optimal recurrent weights can be expressed as a scalar product of the encoding and decoding vectors, the N recurrent weight vectors, $\mathbf{\Omega}_j \in \mathbb{R}^N$, span a K dimensional space. Equivalently, the matrix of recurrent weights has rank $K < N$. The signature of optimality is precisely the low rank K of the recurrent matrix, which implements the negative feedback depicted in the algorithmic description of the framework (Fig. 2.3).

For more details of the derivation of the coordinated spike-coding network, as well as to see how it can optimally implement arbitrary linear dynamical systems, see (Boerlin et al., 2013).

2.5 | Learning the optimal recurrent connections

The optimal recurrent connections can be learned with a local synaptic rule (Brendel et al., 2017). Starting from suboptimal recurrent connections, this synaptic rule updates the weights during spikes, until they converge to the optimal weights (2.20). The learning rule is motivated by the minimization of the loss function, (2.15), which was used to derive the network structure from an optimality principle of efficient coding. However, the loss function used to derive the synaptic learning rule is based on biophysical quantities, the neural membrane potentials, and its minimization is equivalent to enforcing excitatory-inhibitory (E/I) balance, as we will now see.

2.5.1 | Loss function for the learning of E/I balance

Following the supplementary information in (Brendel et al., 2017), the synaptic rule for optimal recurrent weights in the absence of quadratic costs is derived by minimizing the following loss function:

$$L = \Big\langle \sum_{i=1}^{N} V_i^2(t) \Big\rangle_t . \tag{2.22}$$

This loss functions penalizes the divergence of voltage from the resting state, and its minimization thus restores the E/I balance. The authors assume that neurons can minimize their voltage fluctuations precisely at the time of a presynaptic spike. This is motivated by the desire to derive a biologically feasible synaptic rule, which relies only on local information.

The loss function can then be approximated as an average over spike times:

$$\begin{aligned} L &\approx \Big\langle \sum_{i=1}^{N} \Big(\frac{1}{2}\big(V_i(t^-) + V_i(t^+)\big)\Big)^2 \Big\rangle_{\text{spikes}} \\ &= \Big\langle \sum_{i=1}^{N} \Big(V_i(t^-) + \frac{1}{2}\sum_{j=1}^{N} \Omega_{ij}(t^-)\tilde{o}_j(t)\Big)^2 \Big\rangle_{\text{spikes}} \end{aligned} \tag{2.23}$$

Here, t^- and t^+ denote the time right before and right after the spike time t of a presynaptic neuron j; and $\tilde{o}_j(t)$ equals one if neuron j spiked at time, t, and is zero

otherwise. In going from the second to the third line, the following identity was used:

$$V_i(t^+) = V_i(t^-) + \sum_{j=1}^{N} \Omega_{ij}(t^-)\tilde{o}_j(t), \tag{2.24}$$

which holds true when the quadratic cost parameter $\mu = 0$.

The gradient of the approximated loss function, (2.23), with respect to a recurrent weight, $\Omega_{ij}(t)$, equals:

$$\frac{\partial L}{\partial \Omega_{ij}(t)} = \Big\langle 2V_i(t^-) + \Omega_{ij}(t)\tilde{o}_j(t) \Big\rangle_{\text{spikes } j}, \tag{2.25}$$

where we have approximated $\Omega_{ij}(t^-)$ with $\Omega_{ij}(t)$, as the synaptic weights change on a much slower time scale than the membrane potential.

The temporal dynamics of recurrent synaptic weights is then defined to be proportional to the negative gradient of the approximated loss function:

$$\big\langle \dot{\Omega}_{ij}(t) \big\rangle_{\text{spikes } j} = -\varepsilon \frac{\partial L}{\partial \Omega_{ij}(t)}, \tag{2.26}$$

where, ε represents the learning rate.

2.5.2 | Synaptic learning rule

The synaptic learning rule for optimal recurrent weights in the absence of quadratic firing costs is given by:

$$\dot{\Omega}_{ij}(t) = -\varepsilon \Big\{ 2V_i(t^-) + \Omega_{ij}(t)\tilde{o}_j(t) \Big\}, \text{ when } \tilde{o}_j(t) > 0, \tag{2.27}$$

which is combined with the following initial conditions: $\Omega_{ij}(0) = 0$, or $\Omega_{ij}(0) = \mathcal{N}(0, \sigma)$. Here, σ is ideally small compared to the firing threshold. If the initial recurrent weights are on the order of the threshold, the learning rule could potentially lead to divergent synaptic weights.

Setting the dynamics (2.27) to zero, shows that the steady state solution for the recurrent weights is:

$$\big\langle \Omega_{ij} \big\rangle_{\text{spikes } j} = -2\big\langle V_i(t^-) \big\rangle_{\text{spikes } j}. \tag{2.28}$$

Following the proposed synaptic rule, the recurrent weights converge to values which act to suppress the activity of neurons that are similarly tuned to neuron j, and to depolarize those that are dissimilarly tuned. Thus the effect of the learned recurrent activity is to bring the voltage of the entire population close to zero, during each spike. This is equivalent to having E/I balance.

Furthermore, substituting the expression for $V_i(t)$, given by (2.3), into (2.28), and rearranging the terms, gives:

$$\sum_{m=1}^{N} \Omega_{im} \big\langle 2r_m(t^-) + \delta_{jm} \big\rangle_{\text{spikes } j} = -2 \sum_{k=1}^{K} F_{ik} \big\langle x_k(t^-) \big\rangle_{\text{spikes } j}. \tag{2.29}$$

Writing this in matrix form and solving for $\mathbf{\Omega}$ shows that the final recurrent weights have the optimal low rank structure:

$$\mathbf{\Omega} = -\mathbf{F}\mathbf{D}, \tag{2.30}$$

where $\mathbf{D}$ is given by:

$$\mathbf{D} = 2\big\langle \mathbf{x}(t^-)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{spikes}} \big\langle \big(2\mathbf{r}(t^-) + \tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t)\big\rangle^{-1}_{\text{spikes}}. \tag{2.31}$$

Here, $\mathbf{r}(t^-)$ is a $N \times 1$ vector of population filtered spikes at time t^-, just before a population spike, and $\tilde{\mathbf{o}}(t)$ is a $N \times 1$ vector, with a one in position j, if neuron j fires a spike at time t, and zero otherwise. As we will show in the following section, this expression is equivalent to the length constrained optimal decoder, (2.35), up to a scaling factor.

The steady state, (2.30), of the synaptic learning rule, (2.27), is given by the optimal recurrent connections, (2.20), up to the quadratic cost term, which was ignored in the learning rule.

To include the quadratic cost term, the authors then modify the synaptic learning rule as follows:

$$\dot{\Omega}_{ij}(t) = -\varepsilon \Big\{ 2\big(V_i(t^-) + \mu r_i(t^-)\big) - \big(\Omega_{ij}(t) + \mu\delta_{ij}\big)\tilde{o}_j(t) \Big\}, \text{ when } \tilde{o}_j(t) > 0. \tag{2.32}$$

The steady state of the full learning rule is precisely the optimal recurrent connectivity, (2.20). With this learning rule a suboptimal network can learn to represent signals efficiently, using only local information.

2.5.3 | Linear decoder during network learning

We now briefly address the expression for the linear decoder, (2.31), which was used to define the steady state, (2.30), of the learning rule.

Substituting $\hat{\mathbf{x}}(t) = \mathbf{D}\mathbf{r}(t)$ into (2.31), and rearranging the terms, shows that an element of the decoder matrix, $\mathbf{D}$, can be expressed as follows:

$$D_{kj} = 2\big\langle x_k(t) - \hat{x}_k(t)\big\rangle_{\text{spikes } j}. \tag{2.33}$$

The linear decoder is proportional to the average reconstruction error during spike events, and as we will now see, this expression is related to the linear regression solution, (2.14), with an additional length constraint.

The expression for firing thresholds, (2.18), imposes a constraint on the length of the optimal decoding vector for each neuron. If the thresholds are fixed throughout learning, then the optimal decoder for neuron j has to satisfy:

$$\sum_{k=1}^{K} D_{kj}^2 = 2T_j - \mu - \nu. \tag{2.34}$$

The optimal decoder, under this length constraint, is obtained by minimizing the original loss function used to derive the efficient network, (2.15), with an added length constraint. The extra term is: $\sum_{j=1}^{N} \lambda_j \big(\sum_{k=1}^{K} D_{kj}^2 - 2T_j + \mu + \nu \big)$. Setting the gradient of

the modified loss function to zero, gives the implicit solution for the length-constrained optimal decoder:

$$D_{kj} = \frac{1}{\lambda_j}\big\langle \big(x_k(t) - \hat{x}_k(t)\big) r_j(t) \big\rangle_t. \tag{2.35}$$

This expression is related to (2.33), with the difference of the multiplier, and the averaging over spikes vs. filtered spikes.

For completeness, we also show the explicit solution:

$$\mathbf{D} = \Big\langle \mathbf{x}(t)\mathbf{r}^T(t) \Big\rangle_t \Big\langle \mathbf{r}(t)\mathbf{r}^T(t) + \mathbf{\Lambda} \Big\rangle_t^{-1}, \tag{2.36}$$

where the matrix, $\mathbf{\Lambda}$, is a diagonal matrix with Lagrange multipliers, λ_j, on the diagonal. The length constraint adds a regularizer to the original (unconstrained) least squares solution, (2.14).

For more details about the properties of the learning rule, and how it can be extended to non-white inputs, see (Brendel et al., 2017).

We also note that the greedy minimization of the biophysically motivated loss function (2.22) can be turned into a continuous gradient descent problem (Huh and Sejnowski, 2018). They accomplish this by minimizing the original network loss function (2.15) via supervised error-backpropagation in time. The minimization of the error between the input signal and the reconstructed signal is made continuous by substituting the spiking delta function with a differentiable gate function with narrow finite support, motivated by (Lajoie et al., 2013). This approach is interesting from the analysis point of view, while as the greedy minimization of voltage deviations (Brendel et al., 2017) is more biologically motivated.

2.6 | Conclusion

The coordinated spike-coding model represents input signals optimally, with a minimal number of spikes for a given precision level. This idealized model has instantaneous synapses and all-to-all recurrent connectivity.

The model is derived from an efficient representation optimality principle, which allows a very specific functional meaning to be given to the network variables. The subthreshold membrane potential of each neuron measures the signal error, and the function of each spike is to correct this error.

The recurrent connections are structured, and the effect of the recurrent drive is to keep the network in a state of balanced inhibitory and excitatory inputs. In the coordinated spike-coding framework, this is equivalent to minimizing the number of spikes fired to represent the signal. The optimal structured connections for the idealized network can be learned with a local synaptic rule.

3 | Spike Coding Networks with Sparse Connections

Neural populations, which share information about their common feedforward inputs via spikes, can achieve efficient signal representation. This was shown for the coordinated spike-coding network in Chapter 2 (Boerlin et al., 2013). However, this efficient representation relies on each neuron sharing bidirectional recurrent connections with all other neurons in the network. This might be a strong assumption, as local cortical networks are known to have intermediate connection probabilities. The connection probability of local circuits in the early sensory systems, such as for example, in the primary visual and auditory cortices, depends on the cell type, layer, and animal model (Hofer et al., 2011; Bock et al., 2011; Levy and Reyes, 2012; Barral and Reyes, 2016). A general principle in retinotopic and tonotopic systems is that the likelihood of two neighbouring neurons being connected falls off with their distance. Thus, the closer that two neurons are to each other, the more likely it is that they share information. Overall, the reported connection probabilities tend to be in the 0.2-0.6 range.

Given the variability of local connection probabilities, we ask how is the network function affected by its synaptic connectivity profile. We are interested in particular in spike-coding recurrent networks, which function is to represent signals in their population spike trains. A read-out neuron which receives inputs from this kind of a network can linearly decode the population spike train, and recover the original feedforward input.

Specifically, we ask whether a spike-coding network with an intermediate level of connection probability can learn to represent a signal efficiently? We know that fully connected spike-coding networks can learn the optimal recurrent connections, which are required for efficient signal representation (Brendel et al., 2017). The optimal connections in this case are structured, and mediate competition between similarly tuned neurons.

In contrast to the optimal coordinated spike-coding network, we address optimality in networks, which respect the observed local connection probabilities. We propose a local synaptic rule which optimizes the efficient signal representation and the developmental cost of growing synapses. We derive this synaptic rule from an optimality principle, which minimizes the coding error and the number of recurrent connections. The synaptic rule has a fixed point, which is stable for certain kinds of feedforward inputs. The steady state connectivity, proposed by this synaptic rule, is structured. Depending on the coding properties of individual neurons, some connections are grown, while as others are pruned. The similarly tuned neurons learn to send inhibitory recurrent connections to each other,

while as the orthogonally tuned neurons prune their connections over time. In this way, a spike-coding network with intermediate connectivity, learns to balance the efficiency of its signal representation with the cost of growing recurrent connections. Because of the efficiency principle, the network will grow those connections that contribute the most to an optimal signal representation.

When we refer to sparse connections in the context of coordinated spike-coding, we mean connections which are not all-to-all, like in the idealized network. Throughout this thesis, sparsely connected CSC networks refer to networks with a recurrent connection probability less than one. This should be differentiated from the use of the word 'sparse' in balanced networks. There, sparse connectivity means that the recurrent connection probability decreases with increasing networks size, and is essentially very close to zero.

3.1 | Defining the network model

Network dynamics

We study a network of leaky integrate-and-fire neurons, which collectively represent a multi-dimensional variable in their spike trains. The subthreshold potential, $V_i(t)$ of neuron i has the following dynamics:

$$\dot{V}_i(t) = -\lambda V_i(t) + \sum_{k=1}^{K} F_{ik} c_k(t) + \sum_{j=1}^{N} \Omega_{ij} o_j(t), \tag{3.1}$$

for $t > 0$ and $i = \{1, 2, ..., N\}$. Here, λ is the inverse of the membrane time constant, F_{ik} is the linear encoder weight, $c_k(t)$ the feedforward input current, Ω_{ij} the strength of the recurrent current received from neuron j, and $o_j(t)$ is the spike train, which equals the Dirac delta function if neuron j spiked at time t, and is zero otherwise. A spike is modeled as an instantaneous event, which occurs when $V_i(t)$ reaches the firing threshold, T. At that point, the spike train $o_i(t)$ is set to equal the Dirac delta function, and the subsequent resetting of the voltage is modeled by the negative self-connection, Ω_{ii}, in (2.1). The linear encoding weights, F_{ik}, are picked randomly, and then the length of the encoding vector, $\mathbf{F}_i = (F_{i1}, F_{i2}, ..., F_{iK})$, is normalized to equal one. The input currents, $c_k(t)$, could be smoothly varying or constant.

Integrating the voltage dynamics, we get:

$$V_i(t) = \sum_{k=1}^{K} F_{ik} x_k(t) + \sum_{j=1}^{N} \Omega_{ij} r_j(t). \tag{3.2}$$

Here, $x_k(t)$ is the leaky integral of the feedforward input current, $c_k(t)$:

$$x_k(t) = \int_0^t c_k(t') e^{-\lambda(t-t')} dt', \tag{3.3}$$

and $r_j(t)$ is the instantaneous firing rate, or filtered spike train, of neuron j:

$$r_j(t) = \sum_{n=1}^{M_j} e^{-\lambda(t-t_j^n)}, \tag{3.4}$$

where t_j^n are the spike times of neuron j, with $t_j^n < t$.

Decoding the network activity

The goal of the network is to represent a dynamic sensory variable $x_k(t)$, which is a leaky integral of the feedforward current, $c_k(t)$, and has the following dynamics:

$$\dot{x}_k(t) = -\lambda x_k(t) + c_k(t). \tag{3.5}$$

The network should represent the variable in its spike trains, with the reconstructed variable, $\hat{x}_k(t)$, being decoded from the filtered spike trains:

$$\hat{x}_k(t) = \sum_{j=1}^{N} D_{kj} r_j(t). \tag{3.6}$$

Here, D_{kj}, are the decoding weights used by a readout neuron, and are set to equal the optimal decoder, (2.14). This is the least-squares linear regression solution, given a population filtered spike vector, $\mathbf{r}(t)$, and the vector of sensory variables, $\mathbf{x}(t)$:

$$\mathbf{D} = \Big\langle \mathbf{x}(t)\mathbf{r}^T(t) \Big\rangle_t \Big\langle \mathbf{r}(t)\mathbf{r}^T(t) \Big\rangle_t^{-1}. \tag{3.7}$$

Linear decoding is commonly used in studies, owing to its simplicity and biological plausibility (Bialek et al., 1991; Seung and Sompolinsky, 1993; Warland et al., 1997; Rieke et al., 1997; Dayan and Abbott, 2001).

In summary, the goal of the network, just like in Chapter 2, is to represent a continuous variable, $x_k(t)$, in its discrete population spikes. The reconstructed variable, $\hat{x}_k(t)$, is decoded linearly from the filtered spike trains of all neurons in the network (Fig. 3.1a). We consider white inputs, so the optimal encoding and decoding vectors should cover the input space evenly (Fig. 3.1b,c).

Network structure

The recurrent weights, Ω_{ij}, in (3.1) have not yet been specified. In this chapter we are going to consider both building and learning recurrent connections of various connection probabilities, $p \in [0, 1]$. This is in contrast to the fully-connected coordinated spike-coding network, which we analyzed in Chapter 2.

We recall that the optimal weights for efficient signal representation in a fully connected network are structured (Boerlin et al., 2013):

$$\Omega_{ij} = -\sum_{k=1}^{K} F_{ik} D_{kj} - \mu\delta_{ij}. \tag{3.8}$$

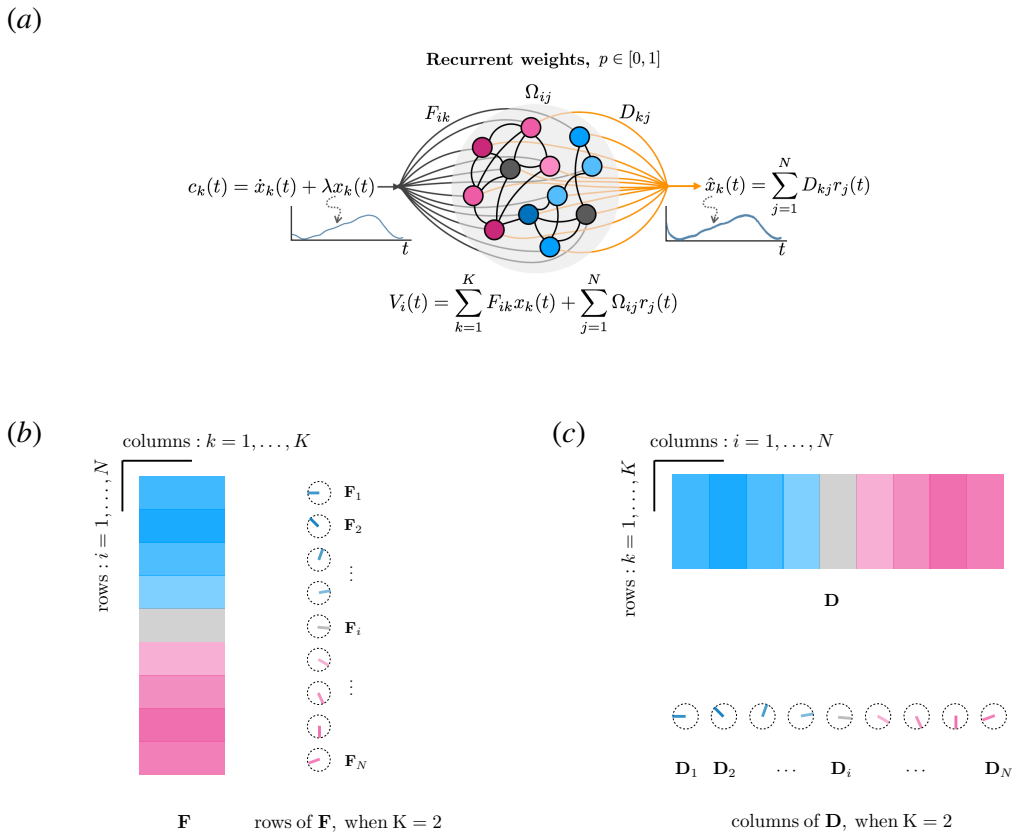


Figure (3.1) Schematic: Spike-coding network model with sparse recurrent connections, with the optimal linear encoders and decoders for white inputs. **(a)** A schematic of the spike-coding network model with recurrent connection probability, p, ranging from 0 to 1. Details of the network set-up are presented in the main text. (b-c) Optimal linear encoders and decoders for white inputs (originally presented in Chapter 2). The illustration is done for $K = 2$ inputs, which are represented by a network of N neurons, with $N \gg K$. Each neuron has K encoding and decoding weights, which can be organized into column vectors, $\mathbf{F}_i$ and $\mathbf{D}_i$. For white inputs, these vectors are optimal when they cover the 2D input space equally. When these vectors are random, the network represents the signal efficiently. The vectors can then be summarized by the encoder and decoder matrices, $\mathbf{F}$ and $\mathbf{D}$. **(b)** The optimal encoder matrix, $\mathbf{F}$, is a $N \times K$ matrix (a tall matrix). Its rows are the linear encoding vectors, $\mathbf{F}_i$. **(c)** The optimal decoder matrix, $\mathbf{D}$, is a $K \times N$ matrix (a flat matrix). Its columns are the linear decoding vectors, $\mathbf{D}_i$.

The optimal recurrent connection, Ω_{ij}, from neuron j to neuron i equals the negative scalar product of neuron i's linear encoding vector and neuron j's linear decoding vector. The self-connection, Ω_{ii}, has an added quadratic firing cost term, which depends on the parameter, μ. Furthermore, for white inputs, $F_{ik} = D_{ki}$.

We also recall that when the connection probability is equal to one, and the recurrent weights are the optimal ones, then the voltage of each neuron measures the projection of the global error onto its encoding weights, up to the quadratic firing cost regularization:

$$V_i(t) = \sum_{k=1}^{K} F_{ik}\big(x_k(t) - \hat{x}_k(t)\big) - \mu r_i(t). \tag{3.9}$$

On the other end of the spectrum there is the independent network of leaky integrate-and-fire neurons, which is a purely feedforward network with no recurrent connections. When $p = 0$, the voltage of each neuron is given by:

$$V_i(t) = \sum_{k=1}^{K} F_{ik} x_k(t), \tag{3.10}$$

as all of the recurrent weights, Ω_{ij}, in (3.2) have been set to zero.

3.2 | Learning to optimize efficient representation and connection probability in spike-coding networks

We introduce a principled way of learning synaptic connections in spike-coding recurrent networks with connection probabilities ranging anywhere from zero to one. The network's function is efficient signal representation, and so the learning should guide the growth and optimization of synaptic connections with this goal in mind. The network structure should be optimized both for efficient signal representation and for efficient network development. The recurrent connections should have the stabilizing *negative feedback* influence, which is necessary for efficient signal representation, but at the same time only a subset of all potential synapses should be grown in the network, thus minimizing its developmental costs.

The learning rule that we now introduce is based on the minimization of a loss function, which measures the two goals that the network has: efficient representation and efficient development.

3.2.1 | Loss function for the E/I balance and sparse connectivity

We define a loss function, which measures the divergence of regularized voltage from zero and the strength of recurrent connections:

$$L = \sum_{i=1}^{N} \Big\langle \big(V_i(t) + \mu r_i(t)\big)^2 + \kappa \sum_{j=1}^{N} |\Omega_{ij}(t)|(1 - \delta_{ij}) \Big\rangle_t. \tag{3.11}$$

The first term is similar to the loss function, (2.22), which penalizes the divergence of voltage from its resting state, thus restoring E/I balance (Brendel et al., 2017). The difference is in the added quadratic firing cost term, weighted by the parameter μ.

The second term in the loss function adds an L_1 penalty constraint on the synaptic strengths. It implements the lasso regularization (least absolute shrinkage and selection operator), which leads to sparse solutions while keeping the optimization problem convex (Tibshirani, 1996). This approach is also related to optimal basis pursuit (Chen et al., 1998).

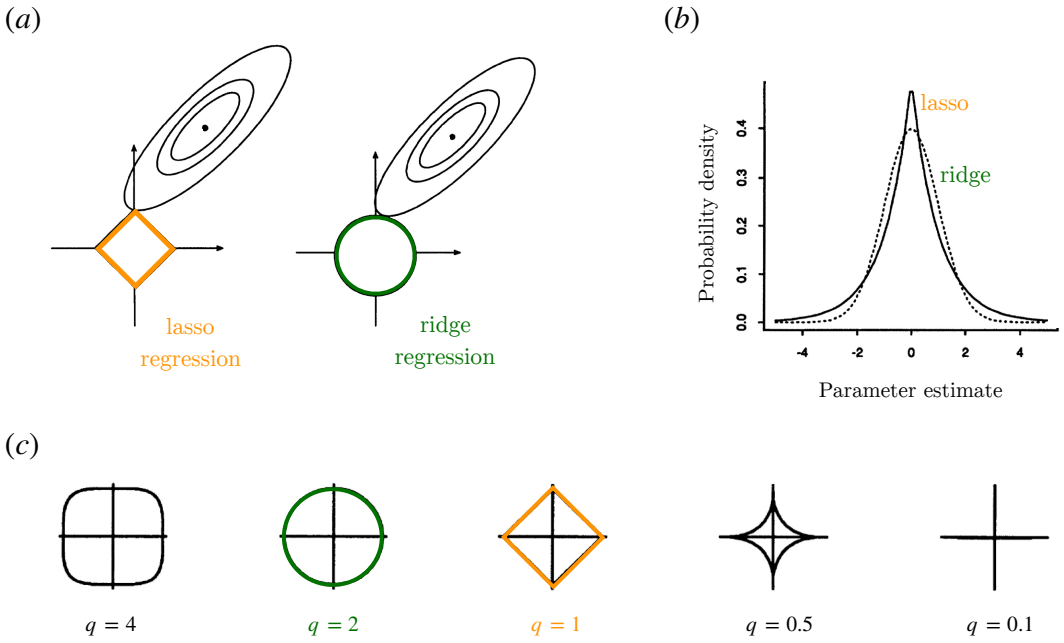


Figure (3.2) Schematic: Lasso regression, adapted from (Tibshirani, 1996). **(a)** Contours of constant value for lasso (square) and ridge (circle) parameter regularizations of ordinary least squares (ellipses). **(b)** The Bayesian view of lasso and ridge regression. The lasso regression is equivalent to a Laplace prior distribution (full line), which is more peaked around zero and has heavier tails than the Gaussian prior (dashed line). **(c)** Contours of constant value of L_q norms for given values of q. As q approaches zero, the shrinking of parameter values goes to zero, and this is equivalent to best subset selection. However, the optimization problem is no longer convex. The L_1 constraint is the convex relaxation of the $\lim_{q\to 0} L_q$ problem.

The lasso regularization constrains the magnitude of the recurrent weights, with the strength of the constraint being measured by the sparseness parameter, κ. This constraint penalizes the magnitude of all recurrent connections sent out by the spiking neuron, but it is not active for self-resets. This is indicated by the $(1 - \delta_{ij})$ multiplier, where the Kronecker delta, δ_{ij}, equals one when $i = j$, and is zero otherwise.

The derivative of the L_1 norm is given by the sign function, and so any quantity that is minimized, will be penalized equally by the regularizing parameter, κ. This is in contrast to the L_2 regularization which would penalize larger quantities more, and smaller

quantities less (Bishop, 2006). Since the L_1 cost penalizes all quantities equally, the small ones converge to zero.

This effect can also be explained by expressing the minimization of the loss function as a constrained quadratic programming problem, with a lasso constraint:

$$\text{argmin}\Big\{\sum_{i=1}^{N}\Big\langle\Big(\mathbf{F}_i^T\mathbf{x}(t)+\mathbf{\Omega}_i(t)\mathbf{r}(t)+\mu r_i(t)\Big)^2\Big\rangle_t\Big\}\ \text{sub. to}\ \sum_{i=1}^{N}\big\langle|\mathbf{\Omega}_i(t)|\big\rangle_t\leq\alpha.$$

Here, we substituted the integral of the voltage dynamics (3.2) into (3.11). The optimal recurrent weights are a solution of the ordinary least squares problem, subject to the weights being smaller than some value α, which could be related to the parameter κ.

Furthermore, the L_1 regularization is equivalent to having a Laplace prior distribution on the recurrent weights:

$$f(\Omega_{ij})=\frac{\kappa}{2}e^{-\kappa|\Omega_{ij}|}, \tag{3.12}$$

with $|\Omega_{ij}|$ being proportional to the negative log density of the double-exponential distribution (3.12) (Murray et al., 1981). Since the Laplace prior puts more weight around zero and has heavier tails, it leads to sparse solutions (Fig. 3.2).

3.2.2 | Minimization of the loss function

Motivated by biological plausibility, we assume that the network cannot minimize the loss function continuously, but only during spike times. This assumption is also made to increase the effectiveness of the optimization, as minimizing during the post-synaptic potentials was found to be sometimes unstable. So we will approximate the loss function by taking the average over spikes in (3.11).

Also, we will evaluate the voltage and filtered spike train at spike times, by averaging their quantities just before and just after a spike. Here, t^- and t^+ denote the time right before and right after the spike of a presynaptic neuron j, which spikes at some time t.

$$\begin{aligned}\big(V_i(t)+\mu r_i(t)\big)^2 &= \Big(\frac{1}{2}\big(V_i(t^-)+V_i(t^+)+\mu r_i(t^-)+\mu r_i(t^+)\big)\Big)^2\\ &= \Big(V_i(t^-)+\mu r_i(t^-)+\frac{1}{2}\Omega_{ij}(t)+\frac{1}{2}\mu\delta_{ij}\Big)^2,\end{aligned} \tag{3.13}$$

where, following (3.1), the effect of a spike of neuron j on the postsynaptic neuron i, is to instantaneously add $\Omega_{ij}(t)$ to its voltage. We have also used the fact that the filtered spike of neuron j will get a unity jump, δ_{ij}. Substituting this into (3.11), we get an approximation of the loss function:

$$L\approx\sum_{i=1}^{N}\Big\langle\Big(V_i(t^-)+\mu r_i(t^-)+\frac{1}{2}\Omega_{ij}(t)+\frac{1}{2}\mu\delta_{ij}\Big)^2+\kappa|\Omega_{ij}(t)|(1-\delta_{ij})\Big\rangle_j, \tag{3.14}$$

where the averaging is done over spike times of neuron j, for $j=\{1,2,...,N\}$. We assume that no two spikes occur precisely at the same time, and so have dropped the summation over j in front of the L_1 cost term.

The derivative of the loss function with respect to the recurrent weight, Ω_{ij}, is given by:

$$\frac{\partial L}{\partial \Omega_{ij}(t)} = \Big\langle 2V_i(t^-) + 2\mu r_i(t^-) + \Omega_{ij}(t) + \mu\delta_{ij} + \kappa\,\mathrm{sgn}\big(\Omega_{ij}(t)\big)(1-\delta_{ij})\Big\rangle_j. \tag{3.15}$$

3.2.3 | Lasso regularized synaptic learning rule

The synaptic weights are learned with a small learning rate, ε, with the following rule:

$$\dot{\Omega}_{ij}(t) = -\varepsilon\Big\{2\big(V_i(t^-) + \mu r_i(t^-)\big) + \Omega_{ij}(t) + \kappa\,\mathrm{sgn}\big(\Omega_{ij}(t)\big)\Big\},\ \text{for } i \neq j, \tag{3.16}$$

$$\dot{\Omega}_{jj}(t) = -\varepsilon\Big\{2\big(V_j(t^-) + \mu r_j(t^-)\big) + \Omega_{jj}(t) + \mu\Big\},\ \text{for autapses.} \tag{3.17}$$

The weights are updated at time instants, t, when neuron j fires a spike, and the updates are weighted by ε, which is small. The initial weights should be set to zero or $\mathcal{N}(0,\sigma)$, where σ is small compared to the firing threshold. This is needed to ensure the stability of recurrent weights over the course of learning.

The learning rule was derived by setting the average synaptic dynamics to be proportional to the negative gradient of the loss function. A network which learns its connections following the synaptic rule, (3.16-3.17), is thus also minimizing the loss function, (3.11).

Setting the average synaptic dynamics, (3.16), to zero, gives an implicit steady state of the recurrent weights:

$$\big\langle \Omega_{ij}(t)\big\rangle_j = -\big\langle 2\big(V_i(t^-) + \mu r_i(t^-)\big) + \kappa\,\mathrm{sgn}\big(\Omega_{ij}(t)\big)\big\rangle_j. \tag{3.18}$$

On average, the effect of a spike of neuron j is to reflect the voltage of neuron i around zero, from the state it had just before the spike, $V_i(t^-)$, to a new state, $-bV_i(t^-)$. The parameter, b, measures the amount by which voltage is reflected around zero. In the absence of the firing rate and synaptic cost regularizations, $b = 1$, and the synaptic learning rule is enforcing E/I balance and efficient coding. With the regularizations, $b < 1$, as the network is also optimizing distributing its spiking activity across the entire population, and growing fewer synaptic connections.

3.2.4 | Learning E/I balance

Let's first consider the special case in which $\mu = 0$ and $\kappa = 0$. We then consider the voltage, $V_i(t^-)$, of neurons which encoding vectors are almost co-directed with that of neuron j. At the time, t^-, just before neuron j fires a spike, their voltages will also be close to the firing threshold. In the absence of a negative recurrent input, neurons i would be in a depolarized state, ready to spike. The recurrent input implements a negative feedback, which hyperpolarizes similarly tuned neurons. When neuron j fires an error-correcting spike, it also optimally changes the postsynaptic voltage from $+V_i(t^-)$ to $-V_i(t^-)$, preventing these neurons from spiking unnecessarily.

Similarly, the effect of an error-correcting spike is to also reflect the voltages of dissimilarly tunes neurons around zero. In the absence of a positive recurrent input, these

neurons would be in a hyperpolarized state, as the projection of the input onto their encoding weights is negative. The recurrent input then changes their negative voltage state, $V_i(t^-)$, to the depolarized value, $-V_i(t^-)$.

The input currents are assumed to vary slowly compared to the membrane time constant, and so after this recurrent input the voltage of all neurons will start to move closer to zero again. The negative stabilizing feedback implemented by the recurrent weights, ensures that the voltages of all neurons spend more time close to the resting state, which is also equated with a minimal coding error.

In the light of ensuring E/I balance, one could argue that the recurrent input should reset the voltages of the population to zero, rather than reflecting them around zero. However, when input currents vary slowly in time, reflecting the voltages around zero cuts the number of needed population spikes in half. This also recovers the optimal CSC network, with no firing costs.

To allow the possibility of recurrent weights resetting the voltage of the population to zero, or other states, the multiplication by two in (3.16), should be changed to a multiplication by a parameter β. $\beta = 2$ prescribes that the postsynaptic voltage should be reflected around zero, and $\beta = 1$ that it should be reset to zero.

In fact, β can be related (Brendel et al., 2017) to the firing threshold, T, and the linear cost parameter, ν, which was introduced in the context of the optimal CSC model in Chapter 2:

$$\beta = \frac{2T - \mu - \nu}{T + \mu\langle r_i \rangle_i}. \tag{3.19}$$

$\beta = 2$ corresponds to zero quadratic and linear costs, which recovers the optimal CSC network. To include small firing cost terms, one can use $\beta = 1.8$ (Brendel et al., 2017).

3.2.5 | Learning to distribute spiking

Let's now consider the quadratic cost term, $\mu r_i(t^-)$, in (3.18). This term has the biggest impact on neurons, which encoding vectors are approximately co-directed with that of the presynaptic neuron, j. If the filtered spike, $r_i(t^-)$, is non-zero and perhaps even close to one, that means that neuron i fired recently. The effect of the synaptic rule is then to increase the strength of the negative feedback that neuron j will send to neuron i. The stronger recurrent input decreases the likelihood of neuron i firing, and makes it more likely for another neuron, which has not fired recently, to fire instead.

The firing cost term, $\mu r_i(t^-)$, will not affect the steady-state recurrent weights, which are sent to dissimilarly tuned neurons. When inputs change slowly in time, there will always be about half of the neurons in the population which are hyperpolarized by the feedforward input. The neuron j which spikes, is in the other half of the population, where the neurons are depolarized by the feedforward input. What group a neuron belongs to, at any time, depends on the product of its encoding vector with the feedforward input. If this product is positive, then the neuron will be depolarized by the feedforward input, and if it is negative, the neuron will be hyperpolarized instead. Since the filtered spikes of neurons in the hyperpolarized subgroup are going to be zero, the $\mu r_i(t)$ regularization does not affect the recurrent weights that they receive from the other half of the population.

3.2.6 | Learning to grow sparse recurrent connections

Lastly, we consider the sparseness regularization, $\kappa\,\mathrm{sgn}\big(\Omega_{ij}(t)\big)$, in the steady state (3.18). The approximate effect of this term is to decrease the magnitude of stronger synaptic connections by κ, and to remove those weights, which magnitude is smaller than κ. We will investigate this in more detail in the next section, where we prove the convergence of the synaptic learning rule, under certain conditions.

3.3 | The fixed point of the learning rule

We now re-examine the fixed point, (3.18), and express it in terms of the linear encoders and linear decoders. This is important, as it allows us to address the dimensionality of the steady state recurrent connections.

Following the fixed point analysis (Brendel et al., 2017), reviewed in Chapter 2, we substitute the expression for $V_i(t)$, given by (3.2), into (3.18). Just like before, we ignore the quadratic cost term, for simplicity. Then, the fixed point equation, (3.18), can be re-expressed as follows:

$$\Big\langle 2\sum_{k=1}^{K} F_{ik}x_k(t) + \sum_{m=1}^{N}\Omega_{im}\big(2r_m(t)+\delta_{jm}\big) + \kappa\mathrm{sgn}\big(\Omega_{ij}\big)\Big\rangle_j = 0. \tag{3.20}$$

Here, we use the notation t instead of t^-, for simplicity.

We now introduce $\mathbf{r}(t)$ as an $N\times 1$ vector of population filtered spike trains, $r_j(t)$, and $\tilde{\mathbf{o}}(t)$ as an $N\times 1$ vector, which j^{th} element equals one if neuron j spiked at time t, and is zero otherwise. This allows us to rewrite the fixed point condition in matrix form:

$$\mathbf{\Omega}\big\langle\big(2\mathbf{r}(t)+\tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}} = -2\mathbf{F}\big\langle\mathbf{x}(t)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}} - \kappa\big\langle\mathrm{sgn}(\mathbf{\Omega})\tilde{\mathbf{o}}(t)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}.$$

Under the condition that the sum on the LHS has full rank, we can express the implicit steady state solution as:

$$\mathbf{\Omega} = -\mathbf{FD} - \kappa\big\langle\mathrm{sgn}(\mathbf{\Omega})\tilde{\mathbf{o}}(t)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}\big\langle\big(2\mathbf{r}(t)+\tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}^{-1}, \tag{3.21}$$

where the linear decoder, $\mathbf{D}$, which is implicitly implemented by the network, is defined as:

$$\mathbf{D} = 2\big\langle\mathbf{x}(t)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}\big\langle\big(2\mathbf{r}(t)+\tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}^{-1}. \tag{3.22}$$

This expression for the implicit linear decoder is related to the optimal decoder, (3.7), up to a length constraint. This relationship was reviewed in Chapter 2, where we also derived (2.36) as the least squares linear regression solution under a length constraint (Brendel et al., 2017). In fact, if the E/I balancing part of the synaptic rule was defined to bring all voltages to zero, rather than to reflect them around zero, the implicit decoder would be exactly equal to the optimal decoder (Appendix A).

As reviewed in Chapter 2, we recall that an element of the decoder matrix, $\mathbf{D}$, given by (3.22), satisfies the following relationship with the reconstruction error:

$$D_{kj} = 2\big\langle x_k(t) - \hat{x}_k(t)\big\rangle_j. \tag{3.23}$$

The linear decoder is proportional to the average reconstruction error during spike events. It is important to make a distinction however between the specific decoding weights that will be optimal in more sparsely connected networks, to those that are optimal in the fully-connected CSC network. As we will see in Chapter 4, networks with different connection probabilities lead to different network activities, and different optimal decoding weights.

The first term on the RHS of (3.21) is a low rank matrix, its rank being given by the dimensionality of the input space. The second term, on the other hand, is a full rank matrix. The greater the sparseness parameter, κ, the further away are the recurrent weights from the low-dimensional structure, which is a signature of optimality in CSC networks.

We will return to this implicit expression at the end of this chapter, after we have analyzed how different input statistics influence the network activity, and proven the convergence of the learning rule to an explicit fixed point for particular input statistics.

3.4 | Linking input statistics with connectivity

Input statistics influences the recurrent connectivity that the network will learn by following the learning rule, (3.16-3.17). To illustrate how, we consider a feedforward network which consists of N neurons with no recurrent connections. Each neuron i receives the same two-dimensional input current, $\mathbf{c}(t)$, projected onto its two-dimensional encoding vector $\mathbf{F}_i$. The input is filtered by the neuron's membrane with the leak constant, λ, and the contribution of the input to the neural voltage is then given by $\mathbf{F}_i^T\mathbf{x}(t)$. Here, as before, $\mathbf{x}(t)$ is the leaky integral of the input current. Neurons fire a spike when their voltage crosses the firing threshold, and we assume that all neurons have the same threshold, T. The reset value for the voltage after the spike is $-T$. The firing threshold and reset values could be mapped onto biologically meaningful values (Boerlin et al., 2013), but we keep the rescaled values. They allow us to express the threshold and reset values in relation to the reference rest state, which is set at zero.

In summary, the voltage of each neuron evolves in time following the leaky integrate-and-fire ODE:

$$\dot{V}_i(t) = -\lambda V_i(t) + \mathbf{F}_i^T\mathbf{c}(t) - 2To_i(t), \tag{3.24}$$

where the spike train, $o_i(t) = \delta(t)$, if neuron i fired a spike at time t, and is zero otherwise. Integrating (3.24), we see that the voltage of each neuron is given by:

$$V_i(t) = \mathbf{F}_i^T\mathbf{x}(t) - 2Tr_i(t), \tag{3.25}$$

where $r_i(t)$ is the leaky integral of the spike train $o_i(t)$.

If the network starts receiving the input current at time $t = 0$, then the unfolding of the network's spiking activity will depend on the strength and variability of the input. Given that there was no recent input and thus, that there was no recent spiking activity, which neuron fires the first spike will depend only on $\mathbf{F}_i^T\mathbf{x}(t)$:

$$V_i(t) = \mathbf{F}_i^T\mathbf{x}(t), \quad \text{if } r_i(t)=0. \tag{3.26}$$

This will also hold true when the influence of previous spikes, $r_i(t)$, has decayed to zero.

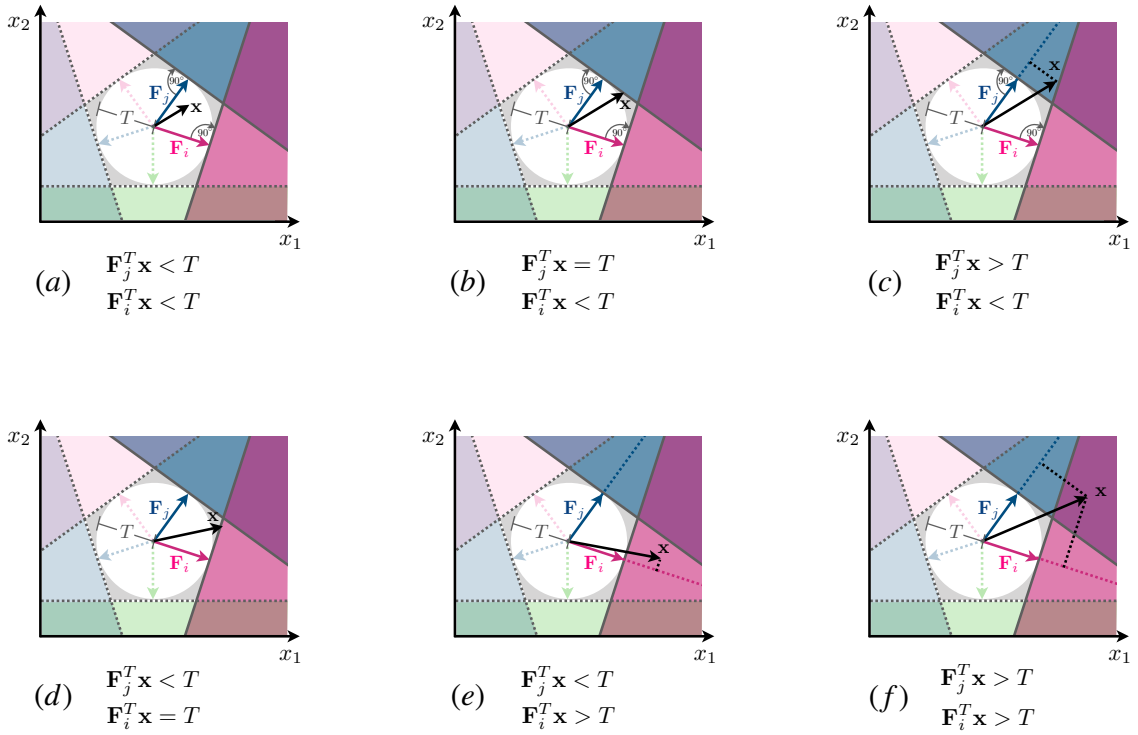
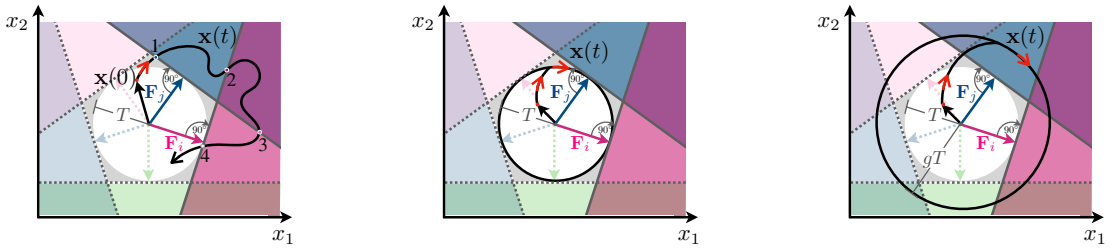


Figure (3.3) Schematic of the relationship between the signal and encoding vectors of the neural population.

The magnitude and angle of the signal $\mathbf{x}(t)$ will then determine which neurons fire the first spikes. This is illustrated in Figure (3.3), in the example network consisting of five neurons. The five encoding vectors span the input space evenly, and the thresholds for each neuron are shown perpendicular to the corresponding encoding vectors. Following the geometric interpretation of CSC models in (Calaim et al., 2019), we observe that neighbouring threshold vectors form closed regions in the input space. We focus on neurons i and j specifically, and consider what happens when the input is in the part of the space that is spanned by their encoding vectors.

If the signal $\mathbf{x}(t)$ is a linear combination of the encoding vectors $\mathbf{F}_j$ and $\mathbf{F}_i$, but the signal magnitude is smaller than the firing threshold T, then neither neuron j nor i will fire a spike (Fig. 3.3a). However, the subthreshold membrane potentials of all neurons will reflect the presence of the input, with the exception of those whose encoding vectors are perpendicular to the signal. The white circular region, with the center $(x_1, x_2) = (0, 0)$ and radius T, defines the part of the input space that does not lead to any spiking activity.

If the signal $\mathbf{x}(t)$ now grows in strength and its scalar product with the encoding vector of neuron j reaches the threshold T, then the neuron j will fire a spike, (Fig. 3.3b). In fact, the neuron j will fire a spike as long as the signal is in the part of the input space bounded by three lines: the threshold vector of neuron j, and the threshold vectors of the two closest neurons, blue section in (Fig. 3.3c). This is where the scalar product, $\mathbf{F}_j^T\mathbf{x}(t)$, is always greater than the threshold, and the scalar product of the signal with other encoding vectors is smaller than T. Thus, the blue region defines the part of the



(*a*) variable gain (*b*) constant small gain (*c*) constant large gain

Figure (3.4) Schematic of the relationship between input gain and network activity.

input space, that can only lead to neuron j firing spikes. As long as the signal is in the blue region, neuron j will fire a new spike as soon as the hyperpolarizing influence of its previous spike has decayed enough. From (3.25), we get that this will be true when:

$$r_j(t) \leq 2\left(\frac{\mathbf{F}_j^T\mathbf{x}(t)}{T} - 1\right), \tag{3.27}$$

and so the next spike of neuron j will be emitted when the time since the last spike exceeds the following limit:

$$\triangle t \geq \frac{1}{\lambda}\log\left(\frac{2T}{\mathbf{F}_j^T\mathbf{x}(t) - T}\right). \tag{3.28}$$

Neuron i was silent so far, but will fire a spike when the signal hits its threshold vector (Fig. 3.3d). This neuron will fire spikes as long as the signal is in the pink domain which is outlined by the threshold vectors of neuron i, and its two neighbouring neurons, and as long as the influence of its last spike has decayed enough (Fig. 3.3e). Lastly, a signal which is a linear combination of $\mathbf{F}_i$ and $\mathbf{F}_j$ could be such that its projections on both encoding vectors are larger than T, purple region in (Fig. 3.3f). This would lead to both neurons i and j firing spikes.

Having discussed what kinds of signals lead to the firing activity of leaky integrate-and-fire neurons in a feedforward network, we now return to the original question of how the signal dynamics interacts with the learning of the recurrent connections. We use the same network of initially independent neurons, and ask what kinds of recurrent connections the repeated application of the sparseness learning rule leads to depending on the signal characteristics in time.

We always assume that the signal is smooth, but there are two characteristics that strongly influence the learned connections. They are namely, the magnitude of the signal $\mathbf{x}(t)$ in relation to the firing threshold, and the speed with which it changes its angle in time. We call the ratio between the strength of the signal and the threshold, the input gain:

$$g(t) = \frac{\left\|\mathbf{x}(t)\right\|_2}{T}. \tag{3.29}$$

The least restricting assumption on the signal dynamics in two-dimensions is to allow its magnitude and angle to vary freely in time (Fig. 3.4a). Here, the signal starts its

temporal evolution inside the white region, thus leading to no spiking activity at $t = 0$. After some time it leaves the white region (point of interest 1), and meanders through the blue region with variable gain. During this time, neuron j will be the only one firing spikes. It then, enters the purple region (point of interest 2), and sends enough excitation to both neurons i and j, and so it will be both of them firing spikes. Once the signal enters the pink region (point of interest 3), it sends enough excitation only to neuron i, and then returns to the white region (point of interest 4), thus bringing the network back to a state of no firing activity.

The variable gain as well as the speed with which it moves through these regions (which is not indicated in the diagram) have a big impact on the firing rate of neurons. If the gain $g(t)$ is very high, this could lead to unrealistically high firing rates of neurons. When the gain varies a lot, this could lead to too large of a range of firing rates that a neuron can realistically handle. More importantly, it is hard to address the exact influence of so many variables on the recurrent learning rule, so we isolate them as follows. We remove the influence of gain variability by considering signals with constant gain. We illustrate an example of a signal with a gain of unity in (Fig. 3.4b), and a gain which is larger than unity in (Fig. 3.4c).

3.5 | Convergence to optimal network structure for small inputs

In this section we provide the explicit fixed point and prove convergence for particular input statistics. We modify the original synaptic learning rule, (3.16), to a more general one:

$$\dot{\Omega}_{ij}(t) = -\varepsilon \left\{ \beta\big(V_i(t^-) + \mu r_i(t^-)\big) + \Omega_{ij}(t) + \kappa \operatorname{sgn}\big(\Omega_{ij}(t)\big) \right\}, \text{ for } i \neq j, \tag{3.30}$$

where we have replaced the scaling by two with the scaling by the parameter β, given by (3.19). This scaling parameter is used in the E/I balance restoring rule in (Brendel et al., 2017).

In what follows, we prove by induction that the synaptic rule (3.30) converges to the fixed point:

$$\Omega_{ij} = \begin{cases} -\beta T \mathbf{F}_i^T \mathbf{F}_j - \kappa, & \text{for } \mathbf{F}_i^T \mathbf{F}_j < -\frac{\kappa}{\beta T}, \\ -\beta T \mathbf{F}_i^T \mathbf{F}_j + \kappa, & \text{for } \mathbf{F}_i^T \mathbf{F}_j > \frac{\kappa}{\beta T}, \\ 0, & \text{for } |\mathbf{F}_i^T \mathbf{F}_j| < \frac{\kappa}{\beta T}, \end{cases} \tag{3.31}$$

when the inputs are slow and have unity gain.

Furthermore, we show that the steady state connection probability of a network, as a function of the sparseness parameter, κ, is given by:

$$p(\Omega_{ij}) = 1 - \frac{\kappa}{\beta T}. \tag{3.32}$$

As a special case, when $\kappa = 0$, this also proves the convergence to the optimal CSC structure with connection probability, $p = 1$.

3.5.1 | Derivation of the fixed point and proof of convergence

We consider the synaptic dynamics, (3.30), and examine whether an initially unconnected network will converge to a fixed point, given a slowly varying input signal with a constant gain of unity.

In this special case, the feedforward network which initially has no connections, will never have two co-occurring spikes. With a gain of unity, at any point in time there will be at most one neuron which will receive just enough excitation to fire a spike. This will happen when the signal alines exactly with the encoding vector of that neuron. If we further assume that the input moves slowly around the unit circle, so that the influence of any population spike will have decayed by the time of the next population spike, then the voltage of neuron i at the time of the spike of neuron j will always be equal to:

$$V_i(t|j \text{ spikes}) = \mathbf{F}_i^T \mathbf{F}_j T, \tag{3.33}$$

since at the time of the spike of neuron j:

$$\mathbf{x}(t) = T\mathbf{F}_j. \tag{3.34}$$

Thus, since the learning rule (3.30) is applied at the time of a spike of the presynaptic neuron j, for slow unity gain inputs, it becomes equivalent to:

$$\dot{\Omega}_{ij}(t) = -\varepsilon \left\{ \beta T \mathbf{F}_i^T \mathbf{F}_j + \Omega_{ij}(t) + \kappa \operatorname{sgn}\big(\Omega_{ij}(t)\big) \right\}, \text{ for } i \neq j. \tag{3.35}$$

The learning rule is actually discrete, as it is applied at the times of the spikes of neuron j. So, we introduce the notation, Ω_{ij}^m, to indicate the value of the recurrent connection after the update following the m^{th} spike of neuron j. Then,

$$\begin{aligned} \Omega_{ij}^{m+1} &= \Omega_{ij}^m + \dot{\Omega}_{ij}^m, \\ &= \Omega_{ij}^m - \varepsilon \left\{ \beta T \mathbf{F}_i^T \mathbf{F}_j + \Omega_{ij}^m + \kappa \operatorname{sgn}\big(\Omega_{ij}^m\big) \right\}, \\ &= (1-\varepsilon)\Omega_{ij}^m - \varepsilon\beta T \mathbf{F}_i^T \mathbf{F}_j - \varepsilon\kappa \operatorname{sgn}\big(\Omega_{ij}^m\big). \end{aligned}$$

We now use this iterative expression to evaluate the recurrent connection, Ω_{ij}, for the first three spikes of neuron j. We will analyze these first three iterations of the learning rule, and then generalize to an M^{th} spike. We start from an unconnected network, with zero initial conditions, although we could have also started from small initial weights. The recurrent weights get updated as follows:

$$\begin{aligned} \Omega_{ij}^0 &= 0 \\ \Omega_{ij}^1 &= -\varepsilon\beta T \mathbf{F}_i^T \mathbf{F}_j \\ \Omega_{ij}^2 &= -\varepsilon(2-\varepsilon)\beta T \mathbf{F}_i^T \mathbf{F}_j - \varepsilon\kappa \operatorname{sgn}\big(\Omega_{ij}^1\big) \\ \Omega_{ij}^3 &= -\varepsilon(3-3\varepsilon+\varepsilon^2)\beta T \mathbf{F}_i^T \mathbf{F}_j - \varepsilon(1-\varepsilon)\kappa \operatorname{sgn}\big(\Omega_{ij}^1\big) - \varepsilon\kappa \operatorname{sgn}\big(\Omega_{ij}^2\big) \end{aligned}$$

At the beginning of learning, the synapse from neuron j to a neuron i changes from its initial zero value to a small value (birth of a synapse). This new value is negative if $\mathbf{F}_i^T \mathbf{F}_j > 0$, and positive if $\mathbf{F}_i^T \mathbf{F}_j < 0$:

$$\operatorname{sgn}\big(\Omega_{ij}^1\big) = \begin{cases} 1, & \text{for } \mathbf{F}_i^T\mathbf{F}_j < 0, \\ -1, & \text{for } \mathbf{F}_i^T\mathbf{F}_j > 0, \\ 0, & \text{for } \mathbf{F}_i^T\mathbf{F}_j = 0. \end{cases} \tag{3.36}$$

Let's now see what happens to the sign of the synapse during the second spike of neuron j. The condition for the synapse, Ω_{ij}^2, being positive is:

$$\Omega_{ij}^2 > 0 \text{ when } -\varepsilon(2-\varepsilon)\beta T\mathbf{F}_i^T\mathbf{F}_j - \varepsilon\kappa\operatorname{sgn}\big(\Omega_{ij}^1\big) > 0.$$

That is,

$$\Omega_{ij}^2 > 0 \text{ when } \mathbf{F}_i^T\mathbf{F}_j < -\frac{\kappa\operatorname{sgn}\big(\Omega_{ij}^1\big)}{(2-\varepsilon)\beta T}.$$

Similarly, reversing the inequalities provides the condition for the synapse attaining a negative value. Combining this with (3.36), we can then define the sign of the synapse after the second spike of neuron j, as follows:

$$\operatorname{sgn}\big(\Omega_{ij}^2\big) = \begin{cases} 1, & \text{for } \mathbf{F}_i^T\mathbf{F}_j < -\frac{\kappa}{(2-\varepsilon)\beta T} < 0 \text{ or } 0 < \mathbf{F}_i^T\mathbf{F}_j < \frac{\kappa}{(2-\varepsilon)\beta T}, \\ -1, & \text{for } \mathbf{F}_i^T\mathbf{F}_j > \frac{\kappa}{(2-\varepsilon)\beta T} > 0 \text{ or } -\frac{\kappa}{(2-\varepsilon)\beta T} < \mathbf{F}_i^T\mathbf{F}_j < 0, \\ 0, & \text{for } \mathbf{F}_i^T\mathbf{F}_j = 0. \end{cases} \tag{3.37}$$

Comparing (3.36) and (3.37), we identify two specific scenarios: 1. the synapse does not change its sign from the first to the second iteration, 2. the synapse flips its sign after the second spike:

$$\begin{aligned} &1. \quad \text{Sign stays constant: } |\mathbf{F}_i^T\mathbf{F}_j| > \frac{\kappa}{(2-\varepsilon)\beta T} \\ &2. \quad \text{Sign flips: } |\mathbf{F}_i^T\mathbf{F}_j| < \frac{\kappa}{(2-\varepsilon)\beta T} \end{aligned} \tag{3.38}$$

Let's now see what happens to the sign of the synapse during the third spike of neuron j. The condition for the synapse, Ω_{ij}^3, being positive is:

$$\Omega_{ij}^3 > 0 \text{ when } -\varepsilon(3-3\varepsilon+\varepsilon^2)\beta T\mathbf{F}_i^T\mathbf{F}_j - \varepsilon(1-\varepsilon)\kappa\operatorname{sgn}\big(\Omega_{ij}^1\big) - \varepsilon\kappa\operatorname{sgn}\big(\Omega_{ij}^2\big) > 0.$$

That is,

$$\Omega_{ij}^3 > 0 \text{ when } \mathbf{F}_i^T\mathbf{F}_j < -\frac{(1-\varepsilon)\kappa\operatorname{sgn}\big(\Omega_{ij}^1\big) + \kappa\operatorname{sgn}\big(\Omega_{ij}^2\big)}{(3-3\varepsilon+\varepsilon^2)\beta T}.$$

Since there are four possible combinations of the signs of the synapse during the first and second spike, we consider all four of them:

$$\Omega_{ij}^3 > 0 \text{ when } \mathbf{F}_i^T\mathbf{F}_j < \begin{cases} -\frac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}, & \text{for } \operatorname{sgn}\big(\Omega_{ij}^1\big) = \operatorname{sgn}\big(\Omega_{ij}^2\big) = 1, \\ \frac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}, & \text{for } \operatorname{sgn}\big(\Omega_{ij}^1\big) = \operatorname{sgn}\big(\Omega_{ij}^2\big) = -1, \\ -\frac{\varepsilon\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}, & \text{for } \operatorname{sgn}\big(\Omega_{ij}^1\big) = -1,\ \operatorname{sgn}\big(\Omega_{ij}^2\big) = 1, \\ \frac{\varepsilon\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}, & \text{for } \operatorname{sgn}\big(\Omega_{ij}^1\big) = 1,\ \operatorname{sgn}\big(\Omega_{ij}^2\big) = -1. \end{cases}$$

Combining this with (3.36) and (3.37), we get the conditions under which the synapse attains a positive value.

$$\Omega_{ij}^3 > 0 \text{ when } \begin{cases} \mathbf{F}_i^T\mathbf{F}_j < -\frac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T} \text{ and } \mathbf{F}_i^T\mathbf{F}_j < -\frac{\kappa}{(2-\varepsilon)\beta T}, \\ \mathbf{F}_i^T\mathbf{F}_j < \frac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T} \text{ and } \mathbf{F}_i^T\mathbf{F}_j > \frac{\kappa}{(2-\varepsilon)\beta T}, \\ \mathbf{F}_i^T\mathbf{F}_j < -\frac{\varepsilon\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T} \text{ and } 0 < \mathbf{F}_i^T\mathbf{F}_j < \frac{\kappa}{(2-\varepsilon)\beta T}, \\ \mathbf{F}_i^T\mathbf{F}_j < \frac{\varepsilon\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T} \text{ and } -\frac{\kappa}{(2-\varepsilon)\beta T} < \mathbf{F}_i^T\mathbf{F}_j < 0. \end{cases}$$

Note that the third condition is never satisfied. This condition corresponds to the case when the sign of the connection was negative in the first iteration, then flipped to a positive value in the second iteration, and now would have needed to stay positive in the next iteration. Thus, once a sign has started to flip from one iteration to the next, it will continue to do so at all successive iterations. We now rewrite the remaining three conditions, which fully define when the synapse is going to be positive,

$$\Omega_{ij}^3 > 0 \text{ when } \begin{cases} \mathbf{F}_i^T\mathbf{F}_j < -\frac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}, \\ \frac{\kappa}{(2-\varepsilon)\beta T} < \mathbf{F}_i^T\mathbf{F}_j < \frac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}, \\ -\frac{\kappa}{(2-\varepsilon)\beta T} < \mathbf{F}_i^T\mathbf{F}_j < 0. \end{cases}$$

The conditions for the negative values of the synapse after the third spike can be obtained following the same process, and we now summarize our observations. We identify three specific scenarios: 1. the sign of the synapse stays the same as in all previous iterations, 2. the sign of the synapse flips once again, 3. the sign of the synapse flips for the first time:

1. $|\mathbf{F}_i^T\mathbf{F}_j| > \dfrac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}$
2. $|\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\kappa}{(2-\varepsilon)\beta T}$
3. $\dfrac{\kappa}{(2-\varepsilon)\beta T} < |\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{(2-\varepsilon)\kappa}{(3-3\varepsilon+\varepsilon^2)\beta T}$

A pattern is starting to emerge, and we will generalize it now by considering what the synaptic connection look like after an M^{th} spike of neuron j. For ease of writing out the long sums, we temporarily adopt the following notation: $a = \beta T\mathbf{F}_i^T\mathbf{F}_j$, and $s^m = \operatorname{sgn}\big(\Omega_{ij}^m\big)$. Then,

$$
\begin{aligned}
\Omega_{ij}^{1} &= -(1-\varepsilon)^{0}\varepsilon a \\
\Omega_{ij}^{2} &= -(1-\varepsilon)^{1}\varepsilon a - \varepsilon a - \varepsilon\kappa s^{1} \\
\Omega_{ij}^{3} &= -(1-\varepsilon)^{2}\varepsilon a - (1-\varepsilon)^{1}\varepsilon a - \varepsilon a - (1-\varepsilon)^{1}\varepsilon\kappa s^{1} - \varepsilon\kappa s^{2} \\
\Omega_{ij}^{4} &= -\big\{(1-\varepsilon)^{3} + (1-\varepsilon)^{2} + (1-\varepsilon) + 1\big\}\varepsilon a \\
&- (1-\varepsilon)^{2}\varepsilon\kappa s^{1} - (1-\varepsilon)\varepsilon\kappa s^{2} - \varepsilon\kappa s^{3} \\
&\vdots \\
\Omega_{ij}^{M} &= -\varepsilon a \sum_{m=0}^{M-1}(1-\varepsilon)^{m} - \varepsilon\kappa \sum_{m=1}^{M-1}(1-\varepsilon)^{M-m-1}s^{m}
\end{aligned}
\tag{3.39}
$$

The scenarios in which the sign stays the same or flips take the form:

1. Sign stays constant: $|\mathbf{F}_i^T\mathbf{F}_j| > \dfrac{\kappa\sum_{m=0}^{M-2}(1-\varepsilon)^m}{\beta T\sum_{m=0}^{M-1}(1-\varepsilon)^m}$
2. Sign always flipping: $|\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\kappa}{(2-\varepsilon)\beta T}$

2.* Sign flips again: $\dfrac{\kappa}{(2-\varepsilon)\beta T} < |\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\kappa\sum_{m=0}^{M-3}(1-\varepsilon)^m}{\beta T\sum_{m=0}^{M-2}(1-\varepsilon)^m}$

3. Sign 1st flip: $\dfrac{\kappa\sum_{m=0}^{M-3}(1-\varepsilon)^m}{\beta T\sum_{m=0}^{M-2}(1-\varepsilon)^m} < |\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\kappa\sum_{m=0}^{M-2}(1-\varepsilon)^m}{\beta T\sum_{m=0}^{M-1}(1-\varepsilon)^m}$

We have the new condition 2.*, which corresponds to the cases where the sign is flipping now, but it wasn't at the beginning of learning. This is in juxtaposition to the second condition which corresponds to the cases where the sign has been flipping since the second spike, and to the third condition which specifies which synapses will be flipping their sign for the first time during the M$^{\text{th}}$ spike. This is because every new spike leads to another increasingly smaller subset of weights flipping their signs for the first time (where up to that point, their signs had been constant).

We now notice that many of the terms depend on the geometric series, which sum can be expressed as follows:

$$
\sum_{m=0}^{M}(1-\varepsilon)^m = \frac{1-(1-\varepsilon)^{M+1}}{1-(1-\varepsilon)} = \frac{1-(1-\varepsilon)^{M+1}}{\varepsilon}
\tag{3.40}
$$

We rewrite the four conditions, in closed form:

1. Sign stays constant: $|\mathbf{F}_i^T\mathbf{F}_j| > \dfrac{\{1-(1-\varepsilon)^{M-1}\}\kappa}{\{1-(1-\varepsilon)^{M}\}\beta T}$
2. Sign always flipping: $|\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\kappa}{(2-\varepsilon)\beta T}$

2.* Sign flips again: $\dfrac{\kappa}{(2-\varepsilon)\beta T} < |\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\{1-(1-\varepsilon)^{M-2}\}\kappa}{\{1-(1-\varepsilon)^{M-1}\}\beta T}$

3. Sign 1st flip: $\dfrac{\{1-(1-\varepsilon)^{M-2}\}\kappa}{\{1-(1-\varepsilon)^{M-1}\}\beta T} < |\mathbf{F}_i^T\mathbf{F}_j| < \dfrac{\{1-(1-\varepsilon)^{M-1}\}\kappa}{\{1-(1-\varepsilon)^{M}\}\beta T}$

As we are interested in the long-term behaviour of the synapses, we can take the limit $M \to \infty$, and reduce the four conditions to two of real interest:

$$\begin{aligned} &1. \quad \text{Sign stays constant: } |\mathbf{F}_i^T \mathbf{F}_j| > \frac{\kappa}{\beta T} \\ &2. \quad \text{Sign flips: } |\mathbf{F}_i^T \mathbf{F}_j| < \frac{\kappa}{\beta T} \end{aligned} \tag{3.41}$$

Here, we have used the fact that $\lim_{M\to\infty}(1-\varepsilon)^M = 0$, when $\varepsilon < 1$. And since $\varepsilon \ll 1$, this limit can be attained quickly in practice.

We have merged the original three conditions for sign flipping into the second condition in (3.41). This condition allows us to summarize that the synapses for which $|\mathbf{F}_i^T \mathbf{F}_j| < \frac{\kappa}{\beta T}$, will progressively keep getting smaller in strength over the course of learning. These weights will keep diminishing until even those close to the upper bound, defined by $\frac{\kappa}{\beta T}$, have become so small that they start to flip their sign. These synapses will thus be of the order $\mathcal{O}(\varepsilon)$.

We now return to the expression (3.39) for the value of the synapse after the M^{th} spike, and evaluate it for the remaining synapses, which never change their sign over the course of learning.

$$\begin{aligned} \Omega_{ij}^M &= -\varepsilon\beta T \mathbf{F}_i^T \mathbf{F}_j \sum_{m=0}^{M-1} (1-\varepsilon)^m - \varepsilon\kappa \sum_{m=1}^{M-1} (1-\varepsilon)^{M-m-1} \operatorname{sgn}\big(\Omega_{ij}^m\big), \\ &= -\varepsilon\beta T \mathbf{F}_i^T \mathbf{F}_j \sum_{m=0}^{M-1} (1-\varepsilon)^m - \varepsilon\kappa \operatorname{sgn}\big(\Omega_{ij}^1\big) \sum_{m=0}^{M-2} (1-\varepsilon)^m, \\ &= -\varepsilon\beta T \mathbf{F}_i^T \mathbf{F}_j \frac{1-(1-\varepsilon)^M}{\varepsilon} - \varepsilon\kappa \operatorname{sgn}\big(\Omega_{ij}^1\big) \frac{1-(1-\varepsilon)^{M-1}}{\varepsilon}, \\ &= -\beta T \mathbf{F}_i^T \mathbf{F}_j \{1-(1-\varepsilon)^M\} + \kappa \operatorname{sgn}\big(\mathbf{F}_i^T \mathbf{F}_j\big)\{1-(1-\varepsilon)^{M-1}\}, \end{aligned} \tag{3.42}$$

where we have used the expression for the sum of the geometric series, (3.40), once again. We also used the fact that $\operatorname{sgn}\big(\Omega_{ij}^m\big) = \operatorname{sgn}\big(\Omega_{ij}^1\big) = -\operatorname{sgn}\big(\mathbf{F}_i^T \mathbf{F}_j\big)$, for all $m > 1$, as we are considering only those synapses for which the sign never changes.

Taking the limit $M \to \infty$:

$$\lim_{M\to\infty} \Omega_{ij}^M = -\beta T \mathbf{F}_i^T \mathbf{F}_j + \kappa \operatorname{sgn}\big(\mathbf{F}_i^T \mathbf{F}_j\big), \tag{3.43}$$

which gives the value that the synapses will converge to over learning, for neurons for which $|\mathbf{F}_i^T \mathbf{F}_j| > \frac{\kappa}{\beta T}$. This is a $\mathcal{O}(1)$ quantity, so we can approximate the remaining synapses, which are of $\mathcal{O}(\varepsilon)$, with zero.

In summary, the fixed point to which the synaptic weight from neuron j to neuron i converges to is:

$$\Omega_{ij} = \begin{cases} -\beta T \mathbf{F}_i^T \mathbf{F}_j - \kappa, & \text{for } \mathbf{F}_i^T \mathbf{F}_j < -\frac{\kappa}{\beta T}, \\ -\beta T \mathbf{F}_i^T \mathbf{F}_j + \kappa, & \text{for } \mathbf{F}_i^T \mathbf{F}_j > \frac{\kappa}{\beta T}, \\ 0, & \text{for } |\mathbf{F}_i^T \mathbf{F}_j| < \frac{\kappa}{\beta T}. \end{cases} \tag{3.44}$$

And the steady state connection probability of a network as a function of the sparseness parameter, κ, is given by:

$$p(\Omega_{ij}) = 1 - \frac{\kappa}{\beta T}. \tag{3.45}$$

This proof applies to signals with unity gain, which vary slowly in time. We recall that the assumption of slow variation of the signal in time was necessary to be able to neglect the contributions of $\Omega_{ij}(t)r_j(t)$ and $r_i(t^-)$ at the time of the spike of neuron i. Thus, in order for this assumption to hold true throughout learning, the signal has to become progressively slower, as the recurrent weights grow in magnitude over learning. Or alternatively, the signal can be assumed to be very slow at the very start of learning, so that even when the recurrent weights have converged to their fixed point, the contribution of previous spikes would have not interfered with the learning process.

It is important to point out that the case for which we have just derived the fixed point and proved convergence does not correspond to the conditions which are associated with efficiency. The signal is so slow that the inhibitory recurrent input never ends up actually preventing similarly tuned neurons from spiking one after another. In fact, similarly tuned neurons fire spikes in succession, albeit with a large inter-spike interval, because the radial angle of the input changes so slowly that all recurrent input has decayed to zero by the time of the next population spike.

3.6 | The fixed point for large feedforward inputs

The more relevant case to consider is a network which receives input much larger than its firing threshold. This allows the network to represent a signal with reasonable precision, as the reconstruction error in the optimal CSC network scales with . It also places the network in a regime where recurrent inhibition of similarly tuned neurons can lead to an efficient signal representation.

The important characteristics of the filtered feedforward input are its gain compared to the firing threshold, and its radial speed, compared to the membrane leak constant. The input can have both variable gain and variable speed, as illustrated for the same 2D input in the phase plot in (Fig. 3.5a), and as a function of time in (Fig. 3.5b). The case for which we have proven convergence is for constant unit gain (Fig. 3.5c), with a slow radial speed (Fig. 3.5d, top panel). Ideally, we would like to derive analytical results for larger input gain (Fig. 3.5e) and see how different input derivatives (Fig. 3.5f) affect the convergence and the explicit fixed point.

However, analytical results are difficult to obtain in the more general case. We are going to see in the following chapter, that numerical simulations of the synaptic rule (3.30) lead to stable networks, which can represent larger inputs accurately.

For now, we write the more general form of the implicit fixed point condition (3.21):

$$\big\langle \mathbf{\Omega} \big\rangle_{\text{sp.}} = -\mathbf{FD} - \kappa \big\langle \text{sgn}(\mathbf{\Omega}) \tilde{\mathbf{o}}(t) \tilde{\mathbf{o}}^T(t) \big\rangle_{\text{sp.}} \big\langle \big(\beta \mathbf{r}(t) + \tilde{\mathbf{o}}(t)\big) \tilde{\mathbf{o}}^T(t) \big\rangle_{\text{sp.}}^{-1}, \tag{3.46}$$

which is the fixed point condition for the general synaptic rule, (3.30). We have replaced the scaling by two, with the more general scaling by the parameter, β, (3.19).

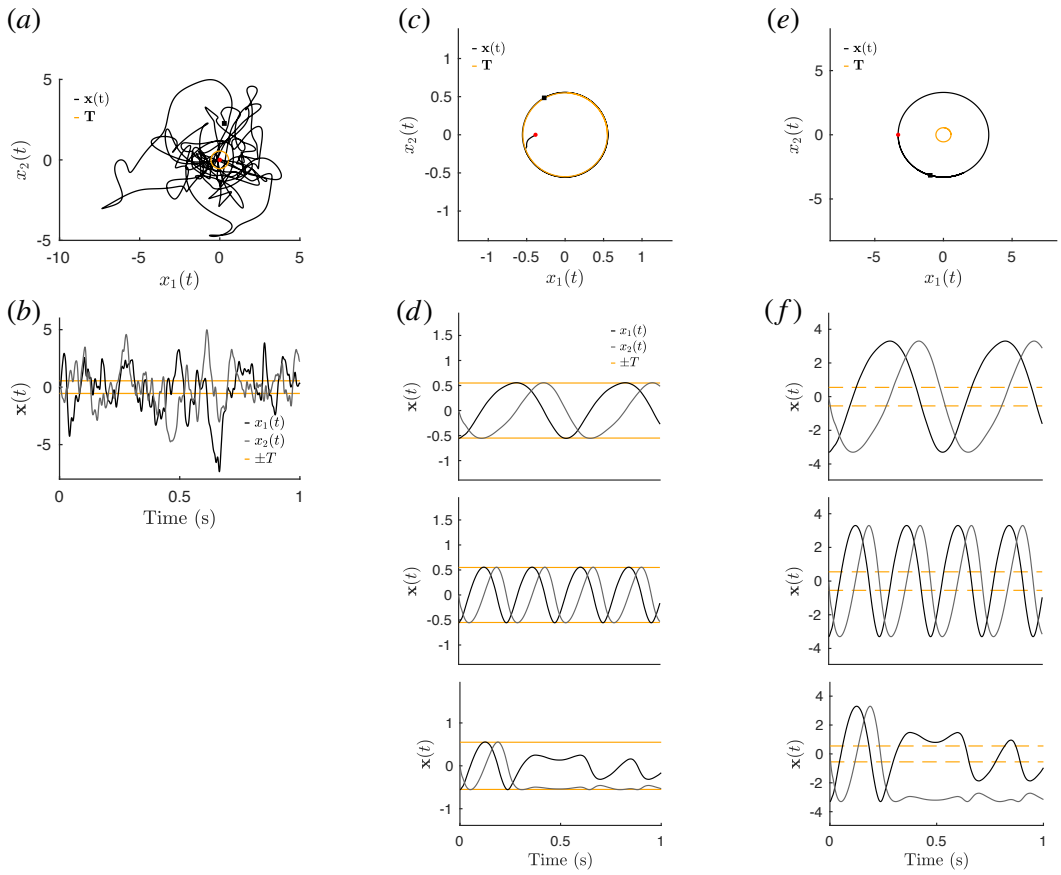


Figure (3.5) Input statistics: signal gain and radial speed. We show 2D inputs with different kinds of gain and radial speed. Phase plots **(a,c,e)** show the input trajectory, with the starting point at $t = 0$ (red dot), and the final point at the end of time (black square). The firing threshold, T, and its negative, are shown for comparison (orange). The input has variable gain (a), unity gain (c), and larger constant gain (e). **(b)** Same signal as in (a), but plotted over time. This particular signal also has variable radial speed. **(d)** Examples of a signal with unity gain, but different choices of radial speed: constant small, high, and variable radial speed. **(f)** Examples of a signal with a larger gain, and different choices of radial speed: constant small, high, and variable radial speed.

The linear decoder, $\mathbf{D}$, which is implicitly implemented by the network, then becomes:

$$\mathbf{D} = \beta \big\langle \mathbf{x}(t) \tilde{\mathbf{o}}^T(t) \big\rangle_{\text{sp.}} \big\langle \big(\beta \mathbf{r}(t) + \tilde{\mathbf{o}}(t)\big) \tilde{\mathbf{o}}^T(t) \big\rangle_{\text{sp.}}^{-1}. \tag{3.47}$$

The linear decoder is related to the optimal decoder, (3.7), up to a length constraint, and when $\beta = 1$, it becomes equivalent to the optimal decoder (Appendix A).

The general fixed point condition, (3.46), has as its special case the explicit fixed point, (3.44), which we derived for unit gain slow inputs. In this special case, the recurrent connectivity does not actually impact the network activity, and so, $\mathbf{r}(t) = 0$, just before the arrival of a new spike. The expression for the general linear decoder, (3.47), then simplifies so that a decoder element is given by:

$$\begin{aligned} D_{kj} &= \beta \big\langle x_k(t) \big\rangle_j, \\ &= \beta T F_{jk}, \end{aligned} \tag{3.48}$$

where we used (3.34) to express the spike-triggered input in terms of the linear encoder.

Substituting (3.48) into (3.46), we get:

$$\big\langle \Omega_{ij} \big\rangle_{\text{sp.}} = -\beta T \sum_{k=1}^{K} F_{ik} F_{jk} - \kappa \big\langle \text{sgn}(\Omega_{ij}) \big\rangle_{\text{sp.}}. \tag{3.49}$$

When the first term on the right hand side is larger in magnitude than κ, and when $\langle \text{sgn}(\Omega_{ij}) \rangle = \pm 1$, then we recover the κ penalized non-zero synaptic connections in (3.44). Otherwise, we consider $|\mathbf{F}_i^T \mathbf{F}_j| < \frac{\kappa}{\beta T}$, and the condition (3.49) without the averaging signs, for a connection, Ω_{ij}, which is positive at some spike time. Then the second term on the RHS of (3.49), will subtract κ, leading to a negative Ω_{ij}, thus flipping its sign. On average, these connections will thus be zero, as derived more formally in (3.44).

The implicit fixed point condition, (3.46), is satisfied by fixed points of synaptic connectivity in different kinds of networks, and is not limited to networks which represent slowly varying inputs of unit gain. To see what these synaptic connections look like, we will turn to numerical simulations in the next chapter.

3.7 | Conclusion

In this chapter, we have introduced a local synaptic rule which allows networks to learn to represent signals efficiently, while also distributing the spiking across the population, and optimizing the number of recurrent synaptic connections. In the absence of the sparseness regularization, our synaptic rule recovers the E/I balance restoring rule (Brendel et al., 2017), which learns the optimal all-to-all connectivity in idealized coordinated spike-coding networks (Boerlin et al., 2013).

With the sparseness constraint, the Lasso regularized synaptic rule balances the efficiency of representation with a smaller recurrent connection probability. The synaptic rule has a fixed point, which we have proven to be stable for small slowly varying feedforward inputs. Numerical simulations are necessary to investigate the fixed points for more general inputs.

4 | Training Spiking Networks with Sparse Connections

Sparsely connected local networks can learn to represent signals efficiently in their population spikes. In the previous chapter, we focused on the derivation of the Lasso regularized synaptic rule, which grows those recurrent connections which are the most important for efficiency, and prunes the rest. The resulting synaptic connections are structured and their specific values depend on the characteristics of the feedforward input.

In this chapter, we investigate the exact structure of the learned synaptic connections, for different kinds of inputs. We also analyze the activity of the resulting networks, and assess how efficiently they represent signals for varying recurrent connection probabilities. We investigate the behaviour of networks all along the connectivity spectrum, from a purely feedforward network to one with full connectivity, given by the idealized coordinated spike-coding network.

Many variables influence the strength and structure of the recurrent weights that get learned. Different input statistics, dimensionality of the input, number of neurons, parameters of the learning rule, all influence what the final recurrent connections look like.

In order to understand how different parameters influence the recurrent connectivity, as well as to analyze the performance of networks with various connection probabilities, we need to simulate them numerically. We will verify whether recurrent weights converge to a fixed point, and analyze how changing input statistics affect the fixed point.

Numerical simulations also allow us to assess how network's connection probability affects the efficiency of signal representation. With a fully connected optimal network on one end of the spectrum, and a network of independent leaky integrate-and-fire neurons on the other end, simulations of the learning rule give us a tool to study intermediate networks.

4.1 | Network model

As in previous chapters, the networks that we consider consist of leaky integrate-and-fire neurons. The subthreshold potential, $V_i(t)$ of neuron i is given by:

$$V_i(t) = \sum_{k=1}^{K} F_{ik} x_k(t) + \sum_{j=1}^{N} \Omega_{ij} r_j(t), \tag{4.1}$$

for $t > 0$ and $i = \{1, 2, ..., N\}$. Here, F_{ik} is the linear encoder weight, $x_k(t)$ is the leaky integral of the feedforward input current, Ω_{ij} the strength of the recurrent current received from neuron j, and $r_j(t)$ is the instantaneous firing rate of neuron j. A spike is fired when $V_i(t)$ reaches the firing threshold, T. At that point, the filtered spike train $r_i(t)$ is increased by one, and the voltage is reset by the negative self-connection, Ω_{ii}, in (4.1). The linear encoding weights, F_{ik}, are picked randomly, and then the length of the encoding vector, $\mathbf{F}_i = (F_{i1}, F_{i2}, ..., F_{iK})$, is normalized to equal one.

As introduced before, $x_k(t)$ is the leaky integral of the feedforward input current, $c_k(t)$:

$$x_k(t) = \int_0^t c_k(t') e^{-\lambda(t-t')} dt', \tag{4.2}$$

where λ is the inverse of the membrane time constant. The input currents, $c_k(t)$, could be smoothly varying or constant.

The instantaneous firing rate, or filtered spike train, of neuron j is given by:

$$r_j(t) = \sum_{n=1}^{M_j} e^{-\lambda(t-t_j^n)}, \tag{4.3}$$

where t_j^n are the spike times of neuron j, with $t_j^n < t$.

Synaptic rule

The recurrent weights, Ω_{ij}, are initially set to zero, and are learned following the synaptic rule, which was derived in Chapter 3:

$$\dot{\Omega}_{ij}(t) = -\varepsilon \Big\{ \beta\big(V_i(t^-) + \mu r_i(t^-)\big) + \Omega_{ij}(t) + \kappa \operatorname{sgn}\big(\Omega_{ij}(t)\big) \Big\}, \text{ for } i \neq j, \tag{4.4}$$

$$\dot{\Omega}_{jj}(t) = -\varepsilon \Big\{ \beta\big(V_j(t^-) + \mu r_j(t^-)\big) + \Omega_{jj}(t) + \mu \Big\}, \text{ for autapses.} \tag{4.5}$$

The weights are updated at time instants, t, when neuron j fires a spike, and the updates are weighted by a small learning rate, ε. The initial weights should be set to zero or $\mathcal{N}(0, \sigma)$, where σ is small compared to the firing threshold. This is needed to ensure that the learning is always stable.

Decoding the network activity

The goal of the network is to represent the dynamic sensory variable, $x_k(t)$, in its spike trains. The reconstructed variable, $\hat{x}_k(t)$, is decoded from the filtered spike trains:

$$\hat{x}_k(t) = \sum_{j=1}^{N} D_{kj} r_j(t). \tag{4.6}$$

Here, D_{kj}, are the decoding weights used by a readout neuron, and are set to equal the optimal decoder, (2.14). This is the least-squares linear regression solution, given a population filtered spike vector, $\mathbf{r}(t)$, and the vector of sensory variables, $\mathbf{x}(t)$:

$$\mathbf{D} = \Big\langle \mathbf{x}(t)\mathbf{r}^T(t) \Big\rangle_t \Big\langle \mathbf{r}(t)\mathbf{r}^T(t) \Big\rangle_t^{-1}. \tag{4.7}$$

4.2 | Training sparse efficient networks

4.2.1 | Simulation details

We simulated the learning rule (4.4-4.5) in networks with various characteristics. The synaptic rule was discretized using the feedforward Euler method, and the synaptic weights were updated at each time step in which there was a new spike. The subthreshold membrane potential (4.1) was evaluated at every time step. At any time step in which $V_i(t) \geq T$, the filtered spike (4.3), was updated by adding one.

The discretization time step was always $dt = 10^{-4}$, although we tried smaller time steps, and this did not change the results. The leak constant was always, $\lambda = 50s^{-1}$, which corresponds to the characteristic time constant of 20ms. Thus 10 time steps correspond to 1ms. The learning rate was $\varepsilon = 10^{-4}$.

The recurrent connections, $\mathbf{\Omega}_{ij}$, are initialized at 0, although we tried random values smaller than one half of the firing threshold, T, and this did not change the final results. The self-resets are initialized with $\Omega_{ii} = -2T$. The scaling parameter in the synaptic rule was set to $\beta = 1.8$. The quadratic parameter was usually $\mu = 0.1$, and the sparseness parameter κ was varied, and it was usually in the range $[0, 1]$.

The learning rule was simulated in networks consisting of $N = 60$ neurons, which receive K feedforward input currents. We considered two cases: $K = 2$ and $K = 3$. The feedforward encoding vectors, $\mathbf{F}_i$, were picked as random vectors from the K-dimensional input space, and normalized to have length one. The firing thresholds of all neurons were set to $T = 0.55$, which is slightly above one half of the encoder length. The input variable, $\mathbf{x}(t)$, has constant magnitude, gT, where we usually picked $g = 5$. The input makes a smooth trajectory on the surface of the K-sphere of radius gT.

4.2.2 | Evolution of synaptic weights during learning

The learned recurrent connections between two neurons are generally inhibitory when the encoding vectors of two neurons have a positive scalar product, and generally excitatory

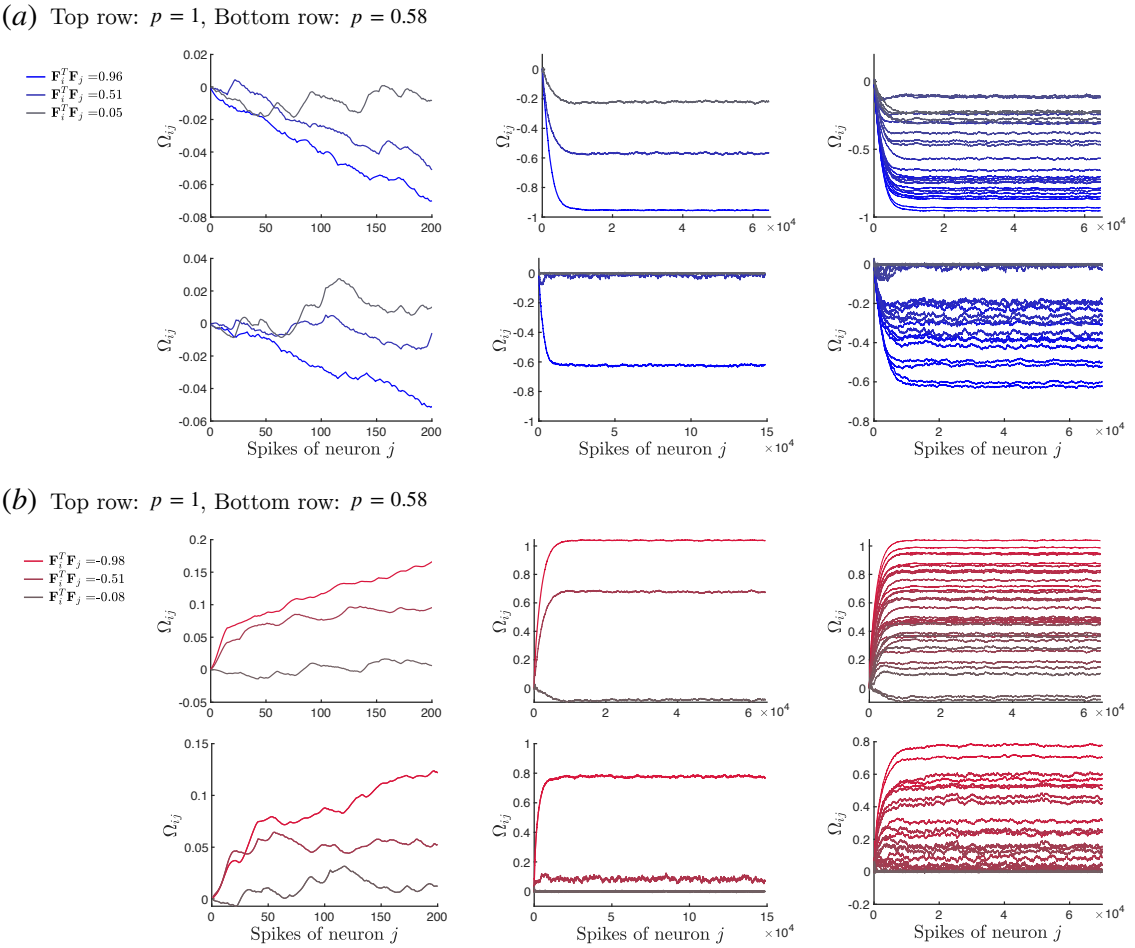


Figure (4.1) Learning synaptic connections with and without a sparseness constraint. The two example networks consist of 60 neurons and receive 3D inputs. One was trained to be fully connected (top row), and the other to have a connection probability, $p = 0.58$ (bottom row). The presynaptic neuron j was picked at random, and the plots look similar for other presynaptic neurons. **(a)** Learning of recurrent connections, which are sent out by the presynaptic neuron j to neurons with which it has *positively correlated* encoding weights. Left: the early evolution of three example weights sent from neuron j to three neurons i, whose encoding vectors are almost parallel, at a 45° angle, and almost orthogonal to that of neuron j. Middle: the same evolution, but over the entirety of training. Right: the evolution of all synaptic weights sent by neuron j to neurons with which it has positively correlated encoding weights (blue to grey = high to low positive correlation). **(b)** Same as (a), but here the neurons i have *negatively correlated* encoding weights with those of neuron j (red to grey = high to low negative correlation).

if this product is negative. This is illustrated in Figure (4.1), which shows the evolution of synaptic connections sent by an example presynaptic neuron to example postsynaptic neurons in a network of sixty neurons, which are trained to represent three-dimensional inputs. We consider two specific networks, one which was trained to have full connectivity, and another which connectivity is trained to be $p = 0.58$. Figure (4.1) (a) shows the evolution of synaptic connections between neurons which encoding vectors have a positive scalar product. In a network which was trained with the sparseness parameter, $\kappa = 0$, the strength of the learned negative connection is proportional to the scalar product of the encoding vectors. It is the greatest for the postsynaptic neuron i which is similarly tuned to neuron j ($\mathbf{F}_i^T\mathbf{F}_j = 0.96$), and almost zero for the postsynaptic neuron i, which is almost orthogonally tuned to neuron j ($\mathbf{F}_i^T\mathbf{F}_j = 0.05$). All other neurons in between develop synaptic connections whose strength falls between these extremes, proportional to the scalar product of encoding vectors.

When we train the network with the sparseness parameter $\kappa = 0.15$, the final connections follow a similar trend, with the more similarly tuned neurons developing stronger inhibitory connections. However, here the final connection weights are smaller for neurons which are similarly tuned, compared to the fully connected network, and many weights converge to zero. This is caused by the non-zero value of κ. However, the exact relationship, derived in the previous chapter, for unit gain slow inputs:

$$\Omega_{ij} = \begin{cases} -\beta T \mathbf{F}_i^T \mathbf{F}_j - \kappa, & \text{for } \mathbf{F}_i^T\mathbf{F}_j < -\frac{\kappa}{\beta T}, \\ -\beta T \mathbf{F}_i^T \mathbf{F}_j + \kappa, & \text{for } \mathbf{F}_i^T\mathbf{F}_j > \frac{\kappa}{\beta T}, \\ 0, & \text{for } |\mathbf{F}_i^T\mathbf{F}_j| < \frac{\kappa}{\beta T}, \end{cases} \tag{4.8}$$

does not hold here. This is because the input used to train the networks had variable gain and radial speed, as in (Brendel et al., 2017).

Figure (4.1) (b) shows similar results for three neurons i, which encoding weights are negatively correlated with that of the presynaptic neuron j. As in (a), the final connection weights are stronger the greater the magnitude of the scalar product of the two encoding vectors. Comparing the final strengths attained in (a) and (b) for the sparsely connected network, we observe that the connection strength is penalized less for dissimilarly tuned neurons than for similarly tuned ones. This is related both to the statistics of the input, and to the fact that the balance between excitatory and inhibitory inputs becomes much less tight in sparsely connected networks. Thus, the voltage can decay much more below the rest state, and the greater the magnitude of the input, the greater this deviation. This will be addressed in more detail later in this chapter.

4.2.3 | The scaling of synaptic strength with postsynaptic voltage

We now turn to the relationship between the steady state solution for the recurrent connectivity and the spike-triggered postsynaptic voltage, which was derived in the previous chapter. We ignore the quadratic cost term for simplicity.

$$\langle \Omega_{ij} \rangle = -\big\langle \beta V_i(t^-) + \kappa \operatorname{sgn}\big(\Omega_{ij}\big) \big\rangle_{\text{spikes } j}, \text{ for } i \neq j. \tag{4.9}$$

This relationship should hold regardless of the variability of input gain or radial speed.

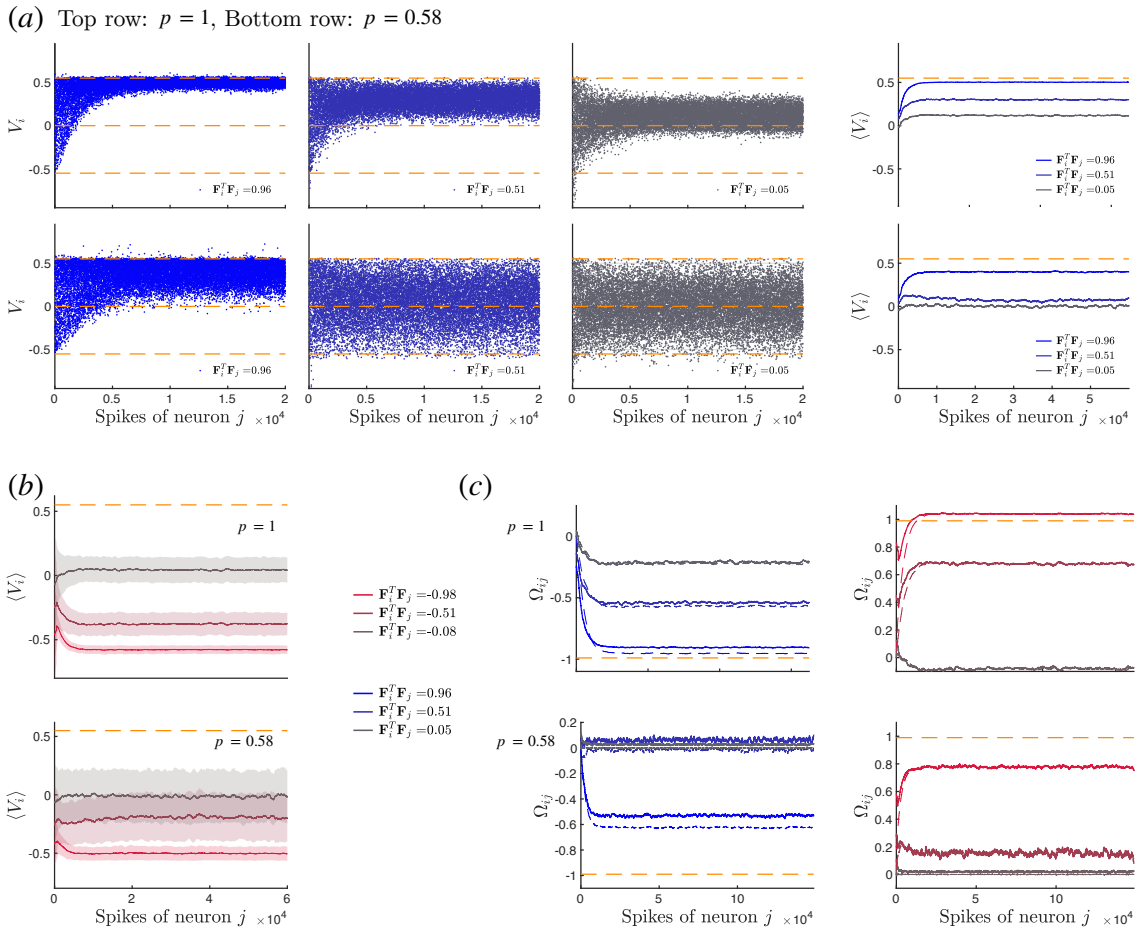


Figure (4.2) Convergence of mean spike triggered voltage and recurrent weights with and without the sparseness constraint. The two example networks consist of 60 neurons and receive 3D inputs. One was trained to be fully connected (top row), and the other to have a connection probability, $p = 0.58$ (bottom row). All shown voltage values are sampled at the time of a spike of neuron $j = 1$, but before the application of the synaptic input from that neuron. The spike triggered voltage values are shown from the onset of training. The neuron $j = 1$ was picked at random, and the plots look similar for other presynaptic neurons. The orange lines in (a) and (b) show the firing threshold T, zero (rest) state, and $-T$. **(a)** Postsynaptic voltage V_i over network training, for three neurons i, which encoding weights are *positively correlated* with that of the presynaptic neuron j. The first neuron i is *similarly* tuned to neuron j, the second one has an encoding vector which is at a 45° angle to that of neuron j, and the third one is almost *orthogonally* tuned to neuron j. We also show the mean voltage for the three example postsynaptic neurons. **(b)** Mean and standard deviation of the postsynaptic voltage V_i, for three neurons i, which encoding weights are *negatively correlated* with that of the presynaptic neuron j. The measures are calculated in moving windows over 1,000 consecutive spikes of neuron j. **(c)** Evolution of trained recurrent weights (dashed line) versus the analytical voltage-dependent average value, $\Omega_{ij} = -\langle \beta V_i \rangle_j$ (full line), derived from the learning rule. The orange line shows $-\beta T$, for $\beta = 1.8$.

We consider whether this relationship holds true in the two networks mentioned above, one which was trained to have full connectivity, and the other which was trained to have intermediate connectivity, $p = 0.58$. Figure (4.2) (a) shows the spike-triggered postsynaptic voltage of example neurons from the two networks. All shown voltage values are sampled at the time of a spike of a presynaptic neuron j, but before the application of the synaptic input from that neuron. The voltage takes on a wide range of values at the beginning of training for a neuron i which is similarly tuned to neuron j ($\mathbf{F}_i^T\mathbf{F}_j = 0.96$), but then starts to converge to a mean value close to the firing threshold. This mean value is smaller in the sparsely connected network, and the variance around the mean is larger. Postsynaptic neurons i with $\mathbf{F}_i^T\mathbf{F}_j$ equal to 0.51 and 0.05, also show a wide range of spike-triggered voltage values at the beginning of training, and then converge to mean values between the rest state and T in the fully connected network. In the sparsely connected network, these voltage values converge to the vicinity of the rest state.

Fig. (4.2) (b) shows the mean and standard deviation of the postsynaptic voltage V_i, for three neurons i, which encoding weights are negatively correlated with that of the presynaptic neuron j. The measures are calculated in moving windows over 1,000 consecutive spikes of neuron j. Again, we observe greater magnitudes of mean voltage values over training in fully connected networks. The smaller variance around the mean spike-triggered voltage value, and its approximate equality to $T\mathbf{F}_i^T\mathbf{F}_j$, are both signatures of optimality in fully connected networks.

Having calculated the mean spike-triggered voltages for the representative neurons in the network, we now compare the evolution of the recurrent weights Ω_{ij} to the voltage-dependent steady state, given by (4.9). As can be seen in Fig. (4.2) (c), the long term numerical value of Ω_{ij} alines fully with that predicted from the mean voltage, $\langle V_i \rangle_j$, for neurons which receive excitatory connections from neuron j. For those which receive inhibitory connections, the numerical value of Ω_{ij} is offset somewhat from the voltage derived predictions. This is caused by the μr_i term, which can be positive for similarly tuned neurons, but was not incorporated in the measures shown in Fig. (4.2).

4.2.4 | The scaling of synaptic weights with the scalar product of linear encoders

A good way to summarize the steady state recurrent connections is as a function of the tuning similarity between two neurons. As expected, in the absence of the sparseness constraint, the recurrent weights have a linear relationship with tuning similarity: $\Omega_{ij} = -\beta T\mathbf{F}_i^T\mathbf{F}_j$, Figure (4.3) (a). Adding the sparseness constraint, $\kappa = 0.15$, leads to the learning of synaptic connections with high tuning similarity and dissimilarity, while as the connections corresponding to neurons with small $|\mathbf{F}_i^T\mathbf{F}|$ get pruned, Fig. (4.3) (b). Lastly, increasing the sparseness constraint further leads to a final network which has no connections, Fig. (4.3) (c).

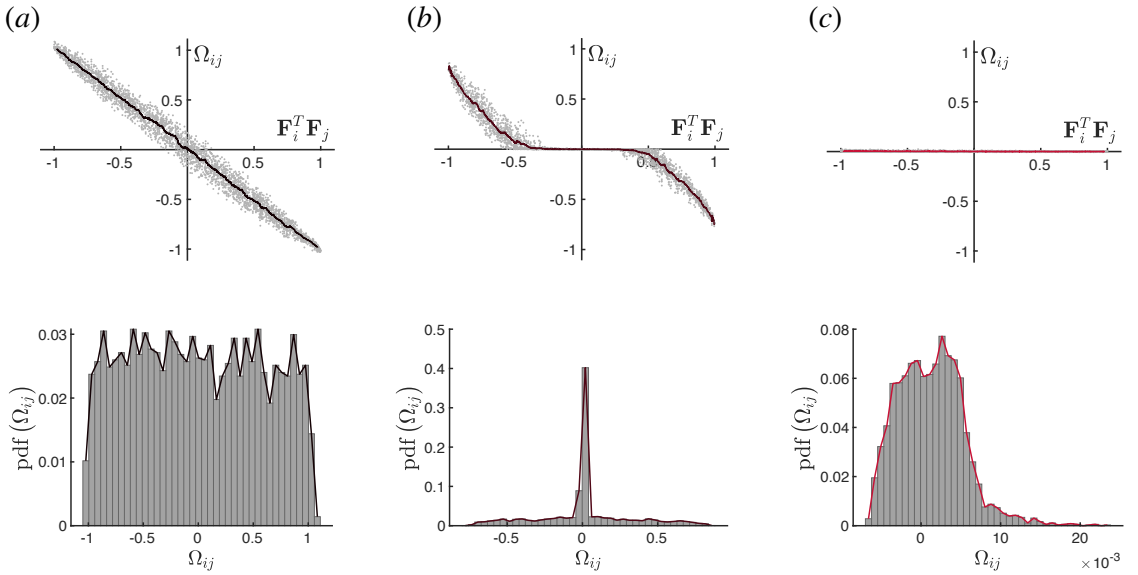


Figure (4.3) Structure and distribution of recurrent weights in trained networks with varying connection probabilities. The networks consist of 60 neurons, and are trained to represent general 3D inputs. The final recurrent connections are shown as a function of neuronal tuning similarity (top graphs), along with the distribution of the final recurrent strengths (bottom graphs), for three different connection probabilities: **(a)** $p = 0.99$, $\kappa = 0$, **(b)** $p = 0.58$, $\kappa = 0.15$, and **(c)** $p = 0.07$, $\kappa = 5.8$.

4.3 | Imbalance between the learned excitatory and inhibitory synaptic strengths

As we train progressively less connected networks, we start to encounter problems caused by an imbalance between the learned positive and negative synaptic strengths. The trained excitatory connections start to outweigh the inhibitory ones, and can lead to spurious spikes, Figure (4.4) (a). The distribution of population inter-spike intervals (ISI) shows a peak at one time step, meaning that some spikes are fired immediately one after another. This peak at one dt becomes higher the less connected the network is.

In order to investigate further, we look at the encoding vectors of the pairs of neurons which fire consecutive spikes. If the relative tuning, $\mathbf{F}_i^T\mathbf{F}_j$, of neuron i which spikes one time step after neuron j is positive and close to one, then the consecutive spike could have been caused by decreased inhibition in sparser networks. Alternatively, it could have been caused by a numerical discretization issue, dt not small enough, as in these simulations only one neuron is allowed to fire a spike per time step.

On the other hand, if $\mathbf{F}_i^T\mathbf{F}_j$ is negative and close to -1, then the consecutive spikes are caused by excessively strong excitation, and decrease the performance of the network. This was referred to as 'ping-pong' in earlier work (Boerlin et al., 2013), where optimal networks could fire spurious spikes when imposed linear costs were too small for the gain of the feedforward input.

In the trained fully connected network, 5% of ISIs equal one times step, and 10% of

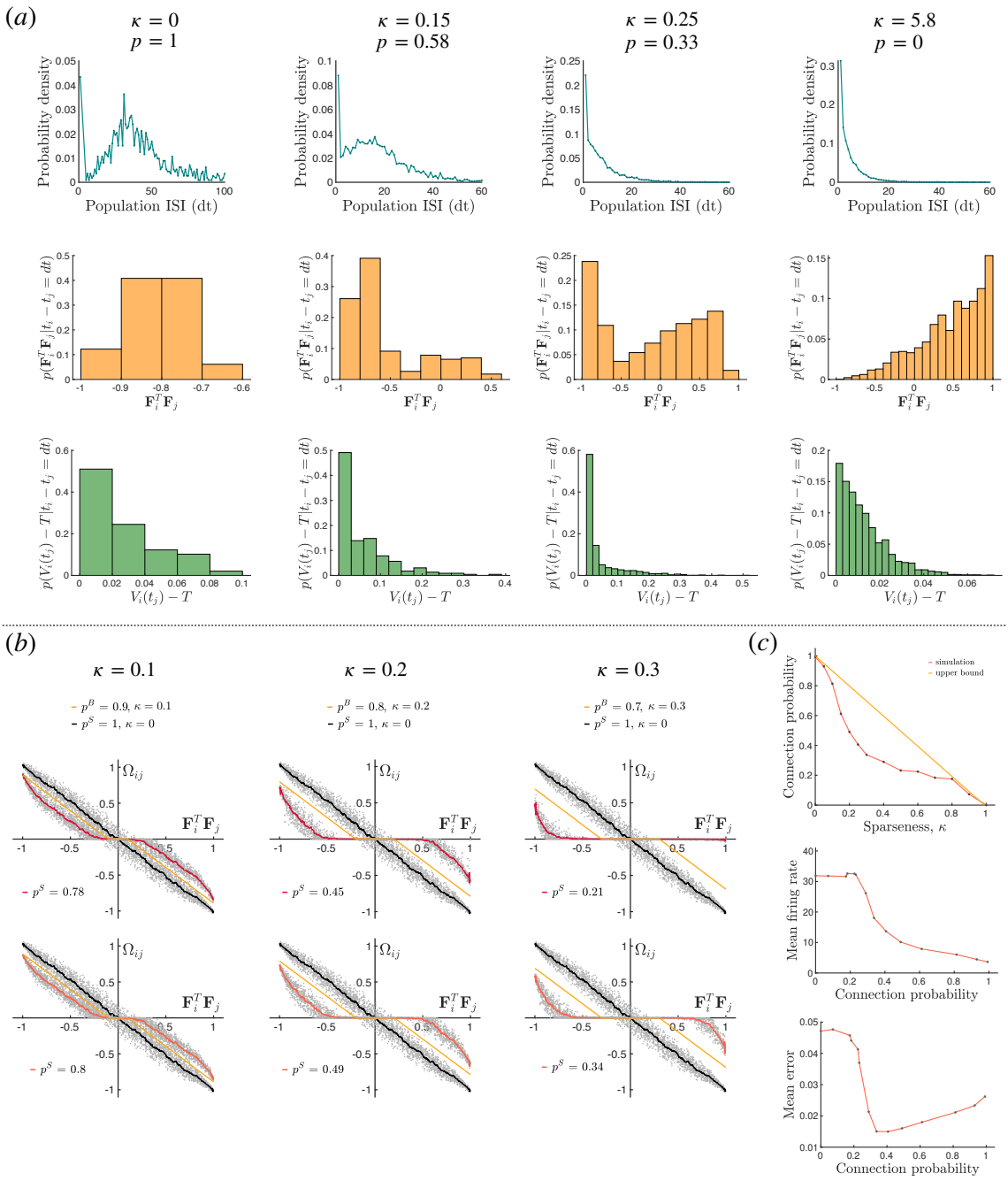


Figure (4.4) Learning sparse networks without a lower voltage bound decreases performance. All networks consist of 60 neurons and were trained to represent 3D variable gain inputs, with different connection probabilities. **(a)** Top: Population inter-spike interval of networks with varying connection probabilities. Middle: Relative tuning of the second neuron which spikes within one time step of a previous population spike. Bottom: $V_i(t_j) - T$ for a neuron i which spikes immediately after neuron j, evaluated at time t_j of the spike of neuron j. **(b)** Top: Learned connections without a lower voltage bound. Bottom: Learned connections with a lower voltage bound, $\min(V) = -T$. Reference data: trained fully connected recurrent weights (black), analytic recurrent weights for unit gain slow inputs (orange): $\Omega^B_{ij} = -\beta T \mathbf{F}_i^T\mathbf{F}_j + \kappa\,\mathrm{sgn}(\mathbf{F}_i^T\mathbf{F}_j)$, with $p^B = 1 - \frac{\kappa}{\beta T}$. **(c)** Summary of results for networks trained with a lower voltage bound.

those are caused by spurious spikes. As the network becomes more sparse, there are more consecutive spikes, and a greater proportion of these is caused by spurious spikes (up to 6% of all spikes are caused by strong excitatory connections in the 33% connected network). In the trained unconnected network, this problem of course no longer exists.

We also checked the voltage values of neurons i when they fire a consecutive population spike, and evaluated it at the time of the previous spike, one time step in the past. All of these values were a bit higher than the threshold, indicating a discretization issue: the time step was at times too large, as when the input gain was larger than usual two neurons could have spiked during one time step. This discretization issue was fixed in later simulations, and does not change the fact that some spikes were fired by neurons untuned to the present input.

4.3.1 | Fixing the imbalance

Adding a lower voltage bound, $\min(V) = -T$, decreases the imbalance between the trained excitatory and inhibitory connections in sparse networks, Figure (4.4) (b). As the value of the sparseness parameter increases, in the absence of a lower voltage bound, we observe all inhibitory connections being pruned, with a significant proportion of the excitatory connections still in the network ($\kappa = 0.3$, top plot). However, when we introduce the lower voltage bound, excitatory and inhibitory connections are pruned more equally ($\kappa = 0.3$, bottom plot).

Furthermore, the connection probability of the trained network increases for a given sparseness constraint, and gets a bit closer to the value predicted for unit gain slow inputs, $p^B = 1 - \frac{\kappa}{\beta T}$. This value can be considered to be an upper bound on the connection probability for a given sparseness constraint value. Checking the connection probability of a network trained with $\kappa = 0.3$, shows in simulations that its connection probability p^S goes up from 0.21 to 0.34 after we add the lower voltage bound. The dependence of the trained recurrent connections on $\mathbf{F}_i^T \mathbf{F}_j$, also gets closer to the analytic prediction for unit gain slow inputs: $\Omega_{ij}^B = -\beta T \mathbf{F}_i^T \mathbf{F}_j + \kappa \operatorname{sgn}(\mathbf{F}_i^T \mathbf{F}_j)$.

The intuition behind the minimum voltage becoming unbounded in sparse networks is the following. The maximum subthreshold voltage always equals the firing threshold, T, save for discretization issues in practical simulations. However, there is no strict lower bound for the minimum subthreshold voltage. In a fully connected network, the recurrent weights are strong and balanced, and so the lower bound ends up being approximately $-T$. However, as we introduce the sparseness parameter, all weights that get learned start to decrease and with this the tight balance gets lost. The lower bound on the subthreshold voltage then starts to be mainly driven by the strength of the signal, and is no longer tightly balanced by the recurrent input. Thus, when the network learns recurrent connections from a neuron j, which codes for the present input, to neurons i, which code for the opposite part of the input space, then the learnt connection can be quite strong and unbounded. The negative valued voltage V_i can be much lower than $-T$, depending on the strength of the feedforward input.

On the other hand, the recurrent connections from neuron j to similarly tuned neurons i are bounded, as V_i cannot be larger than the threshold T. Since the same parameter κ is used for the learning of all connections in the network, the value of inhibitory connec-

tions gets penalized more, and thus they end up being pruned more. This then decreases the performance of the network, because it is the inhibitory connections that reduce the number of spikes that a network fires for the same accuracy of the reconstructed variable, $\hat{\mathbf{x}}(t)$. Furthermore, the imbalance between the strengths of the excitatory an inhibitory connections leads to spurious spikes which do not improve the signal representation.

Adding the lower voltage bound goes in the direction of solving this problem. It also makes sense from a biological point of view, as it allows us to continue to treat the zero voltage state as a reference rest state. Otherwise, allowing the voltage to have arbitrarily large deviations from this reference point, would change the meaning of the reference.

We added the lower voltage bound in simulations, and trained networks with various values of the sparseness constraint parameter, κ. We then computed the connection probability, mean firing rate, and mean reconstruction error for the resulting networks, Figure (4.4) (c). The connection probability is lower than the analytical prediction for unit gain slow inputs. However, we are now able to get more meaningful results, as an unconnected network is obtained with $\kappa = 1$. This is equivalent to $\kappa = \beta T$, as this product is approximately one in our simulations ($\beta = 1.8$, $T = 0.55$). And so we can use the results derived for unit gain slow inputs, to bound the sparseness parameter that we need in order to train a completely unconnected network. With $\kappa = 1$, we obtain $p = 1 - \kappa/\beta T = 0$, as expected. This is in contrast to the results obtained without the lower voltage bound, Fig. (4.4) (a), where an unconnected network was obtained with $\kappa = 5.8$, reflecting the strength of the feedforward input.

4.3.2 | Training sparser networks requires higher quadratic firing costs

We now consider the mean firing rate and reconstruction error, as a function of connection probability, Fig. (4.4) (c). The rate and error were calculated in response to the same input test function for all trained networks. As the connection probability decreases from $p = 1$ to $p = 0.5$, the firing rate slowly increases, and the reconstruction error slowly decreases. The spikes come from neurons which encoding vectors have a high scalar product with the input vector. Thus, an increased number of spikes improves the accuracy of the decoded signal.

However, at $p = 0.3$, the mean firing rate jumps to 32 Hz per neuron and stays there for smaller connection probabilities. The error simultaneously increases to a peak of about 5% of the signal input value on average. This sharp decrease in performance is caused by the pruning of all inhibitory connections, with the excitatory ones being the only connections still present in the network. Adding the lower voltage bound helped alleviate the imbalance between excitatory and inhibitory connections for higher connection probabilities. However, this is not enough to resolve the problem in the more sparsely connected networks.

To see why the imbalance persists in networks with a low connection probability, we have to make a connection between the increased firing rate and how this impacts the learning of the inhibitory connections. Given that the learning rule makes inhibitory connections progressively weaker in more sparsely connected networks, the firing rate of similarly tuned neurons goes up. This means that the repetitive application of the synaptic learning rule, which we recall depends on the postsynaptic voltage, may now find some

neurons in their hyperpolarized state, even though they are similarly tuned and code for the present input. This decreases the inhibitory connections ever more, and has a snowballing effect with ever increasing firing rate.

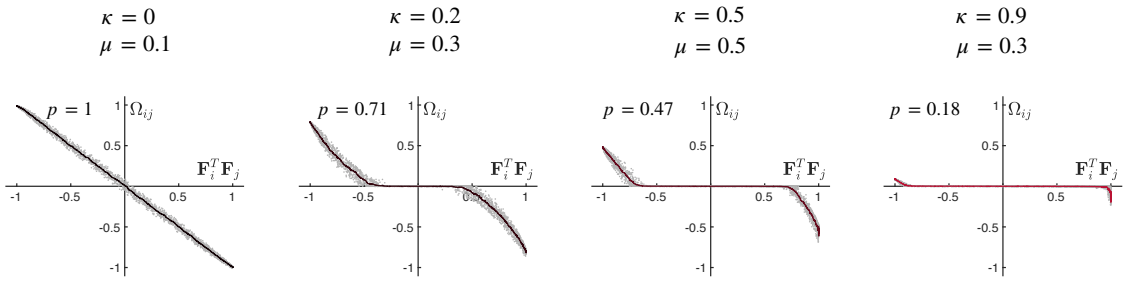


Figure (4.5) Learning sparse networks with balanced positive and negative connections. The quadratic cost term, $\mu r_i(t)$, with higher μ ensures that an equal number of excitatory and inhibitory weights is pruned.

This too can be resolved. The learning rule has a term which depends on the filtered spike train of the postsynaptic neuron, $\mu r_i(t)$, and thus has a memory of the recent spiking of the postsynaptic neuron. Increasing the importance of this term, by increasing μ, as the network gets sparser will lead to stronger inhibitory connections which can balance the excitatory ones, Figure (4.5).

4.4 | Simulations with balanced learning of positive and negative connections

We now simulate learning in networks, while making the following important changes: we fix the voltage lower bound, set higher quadratic costs when learning networks with low connection probabilities, and train networks on feedforward inputs with a fixed gain, Figure (4.6). The connection probability is now very close to $p = 1 - \kappa/\beta T$, see panel (c). The mean reconstruction error does not increase for very sparsely connected networks, see panel (d).

The loss function,

$$l = \left\langle \left\| \mathbf{x}(t) - \hat{\mathbf{x}}(t) \right\|_2^2 + \mu \left\| \mathbf{r}(t) \right\|_2 \right\rangle_{\text{spikes}}, \tag{4.10}$$

shows a general increase with decreased connection probability, as expected.

The mean value of the connections that do not get pruned becomes lower with increased sparseness, as expected. The mean coefficient of variation, CV_2, of neuronal inter-spike intervals, decreases with decreased connection probability. But if the sparser networks receive faster feedforward input, their spiking activity would become more irregular and attain $\text{CV}_2 = 1$. Lastly, for these simulations we pick 2D inputs, as it is convenient for the visualization of the input structure, and the encoding vectors of all neurons, see panels (a-b). We can see the gradual recruitment of neurons, as the input rotates in the input phase space, with input magnitudes five times bigger than the firing threshold.

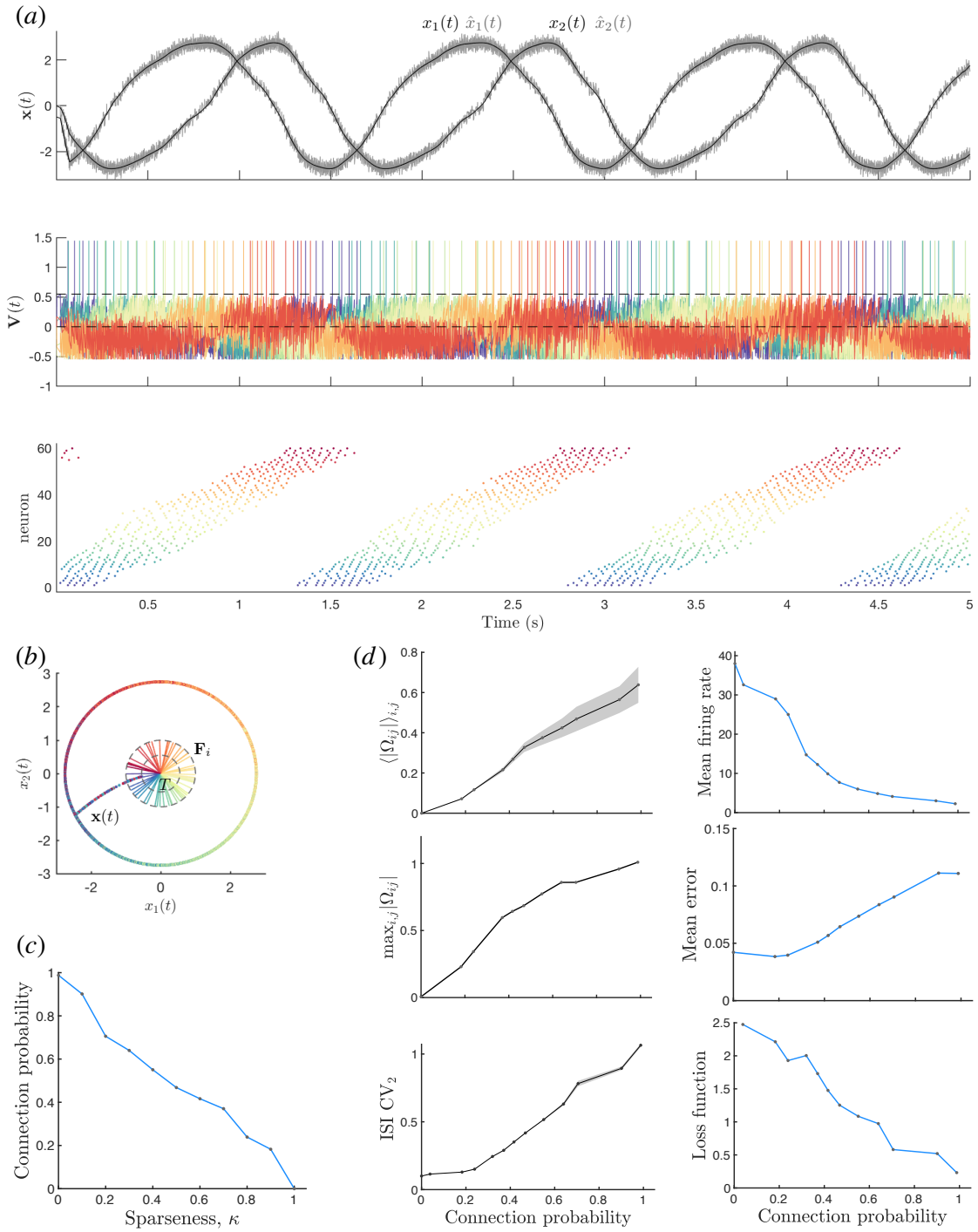


Figure (4.6) Training sparse networks on inputs with constant gain, higher quadratic costs, and a fixed voltage lower bound. **(a)** Reconstruction of a 2D input, $N = 60$, $p = 64\%$. Voltage traces are shown for select ten neurons. The colours represent the angle of each encoding vector, shown in (b). **(b)** Phase plot of the input signal, shown in (a), with the encoding vectors of all neurons. The signal is overlaid with the colour of the neuron which fired a spike. Input gain: $g = 5T$. **(c)** Connection probability of the trained networks as a function of the sparseness parameter, κ. **(d)** Network measures as a function of their connection probability. All measures were calculated for the test input, shown in (a).

4.5 | Structure of recurrent weights in sparse networks

The structure of the learnt recurrent connections in sparsely connected networks resembles the structure of the optimal connections for a fully connected network in the following ways.

First of all, the connections that are not pruned, are reciprocal. If a neuron j sends an inhibitory connection to neuron i, then almost always the neuron i will also be sending an inhibitory connection back to neuron j. Not only is the connection reciprocal, but it also respects the sign most of the time. Furthermore, the strength of the reciprocal connections is approximately the same. More specifically, the strength of the bidirectional connection between neurons i and j, is still proportional to the negative product of their linear encoding vectors, $\mathbf{F}_i^T \mathbf{F}_j$. Thus, the strongest connections in the network will still be found between neurons which have similar tuning, and between neurons which preferred stimulus orientations are 180° apart.

These observations can be summarized by noting that the recurrent connectivity matrix, $\mathbf{\Omega}$, is approximately symmetric, Figure (4.7) (a). The measure of symmetry that we use is given by:

$$\text{sym}(\mathbf{\Omega}) = \frac{\left\| \frac{1}{2} \left(\mathbf{\Omega} - \mathbf{\Omega}^T \right) \right\|}{\left\| \frac{1}{2} \left(\mathbf{\Omega} + \mathbf{\Omega}^T \right) \right\|}. \tag{4.11}$$

Here, the numerator is the norm of the *skew-symmetric* part of the matrix, which is measured as one half of the difference between the matrix and its transpose. The denominator is the norm of the *symmetric* part of the matrix, defined as the sum of the matrix and its transpose, divided by two.

The sum of the anti-symmetric and symmetric parts equals the original matrix. The ratio of their norms then indicates how much of the matrix can be described by the symmetric part, and how much by the skew-symmetric part. The closer the value of the ratio is to zero, the more symmetric the matrix is.

A symmetric matrix would have $\text{sym}(\mathbf{\Omega}) = 0$, which indeed is not so far from what we get for the trained fully connected recurrent matrix, Figure (4.7) (a), for $p = 1$. On the other hand, a skew-symmetric matrix would have $\text{sym}(\mathbf{\Omega}) = 1$. Since the symmetry measure takes on values smaller than 0.1 for positive connection probabilities, $p > 0$, we can then conclude that the trained recurrent matrices are predominantly symmetric. We do not show the symmetry measure for an unconnected network, as this would not be very meaningful.

Lastly, we note that the matrix norm used in the formula, (4.11), is the spectral norm, which gives the maximum singular value of the matrix. Similar results are obtained with other norms.

It is also enough to look at visualizations of the recurrent matrices for various connection probabilities to get a sense of their high level of symmetry. Figure (4.7) (a) shows four

example connectivity matrices for different connection probabilities. The rows and columns of the matrix are arranged so that similarly tuned neurons are next to each other. For networks which represent two-dimensional inputs, this then makes it quite easy to visualize the structure of the trained connections.

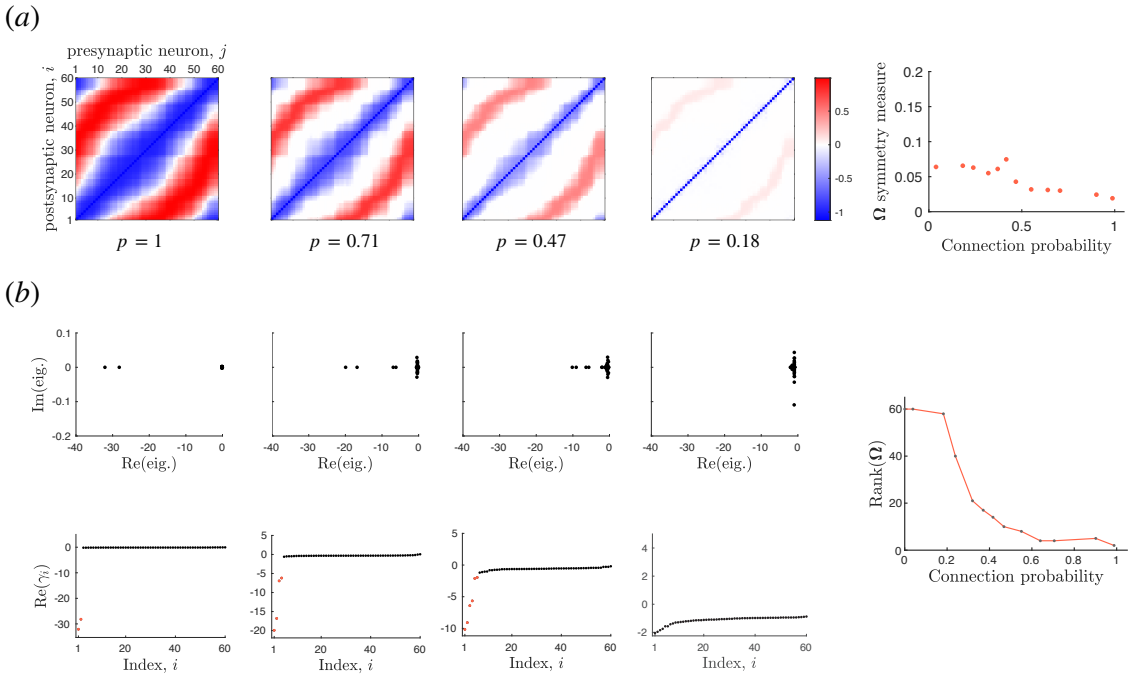


Figure (4.7) Structure of recurrent weights in sparse networks. All results were obtained from networks of 60 neurons, which represent 2D inputs. **(a)** Recurrent weights, Ω_{ij}, shown for all neuron pairs as a matrix $\mathbf{\Omega}$, for four example network connection probabilities. Each matrix row shows the synaptic input from neuron j into neuron i. The excitatory (inhibitory) connections are shown in red (blue), with the intensity of the colour matching the connection strength. Right plot: Measure of symmetry of the trained recurrent connections. **(b)** The eigenvalue spectrum of $\mathbf{\Omega}$ for the example networks in (a), and the rank of $\mathbf{\Omega}$, as a function of connection probability. It was calculated with the Matlab function, rank($\mathbf{\Omega}$,tol), where the number of singular values of $\mathbf{\Omega}$ that are larger than tol was obtained with a tolerance level of 0.9. The real part of the eigenvalues, γ_i, is plotted to show the leading eigenvalues (red).

Having discussed the ways in which the structure of the learnt recurrent connections in sparsely connected networks resembles the structure of the optimal all-to-all network, we now turn to how the two diverge from each other.

In contrast to the optimal connectivity matrix for a fully connected network, the number of non-zero eigenvalues of the sparse recurrent matrix is greater than the dimensionality of the input, Fig. (4.7) (b). The eigenvalue spectrum of the optimal $\mathbf{\Omega}$, with connection probability $p = 1$, shows two non-zero eigenvalues when the network represents two-dimensional inputs. However, as the sparseness constraint progressively changes the recurrent connectivity, more non-zero eigenvalues appear, and simultaneously the strongest eigenvalues become smaller. The number of non-zero eigenvalues goes up to four for $p = 0.71$, and up

to five or six for $p = 0.47$. We will call these eigenvalues the leading eigenvalues of the matrix.

The appearance of new leading eigenvalues is a consequence of the pruning of some connections, while as the decreased magnitude of the greatest eigenvalues is a consequence of learning recurrent connections with the L_1 constraint on their values. When the sparseness constraint is so strong that most recurrent connections have been pruned away, the real part of the entire eigenvalue spectrum gets very close to zero. This can be observed for $p = 0.18$, where there are many non-zero eigenvalues, but they are all very small in magnitude.

We summarize these observations by noting that in contrast to the optimal connectivity matrix for a fully connected network, the sparse recurrent matrix progressively loses the low rank structure with increasing sparseness, Fig. (4.7) (c). While as the number of singular values greater than zero equals exactly the dimensionality of the input for $\mathbf{\Omega}$, when $p = 1$, this number increases to the number of neurons for the unconnected network $\mathbf{\Omega}$, with $p = 0$. Because the rank was calculated numerically, the tolerance level for deciding how many singular values were greater than zero was set at 0.9. It could be argued that a zero tolerance level would have produced a more accurate measure of rank. However, a rank computation with this tolerance level would have picked up on the slight deviations of the smallest singular values away from zero, and the rank would have been equal to N for all connection probabilities. The tolerance level that we picked allows us to ignore the noise of the small singular values around zero. It is a fair way to calculate the rank, as the tolerance level of 0.9 is about thirty times smaller than the largest singular value of the fully connected matrix.

We can get more insight by analyzing the properties of the eigenvectors of the recurrent matrices. We begin by calculating the pairwise scalar products of the N eigenvectors for the four example recurrent matrices, corresponding to $p = 1$, 0.71, 0.41, and 0.18, Figure (4.8) (a). We sort the eigenvectors in a descending order, with respect to the absolute value of the eigenvalue to which they correspond, and can then read off the pairwise products of the i^{th} and j^{th} eigenvector, as the (i, j) element of the product matrix (top row, a). We recall that there were two, four, and six leading eigenvalues for the first three example recurrent matrices, and we can see that the inner products of these vectors were approximately zero. In fact, these scalar products equaled at most about 0.005, 0.03, and 0.02 in the three cases. While as the inner product is increasing a bit with decreasing p, we can still consider it to be approximately zero. So we can say that the leading eigenvectors are approximately linearly independent. The non-leading eigenvectors have non-zero scalar products. However, these eigenvectors correspond to almost vanishing eigenvalues, so we can ignore them. We will discuss the case corresponding to $p = 0.18$ later.

We also note that there are overall more linearly independent eigenvectors as the connection probability decreases (bottom row, a). The probability density of the scalar products of eigenvectors $\mathbf{e}_i$ and $\mathbf{e}_j$ always has a peak around zero, but the number of eigenvectors which are linearly independent is higher the smaller the connection probability. For perfectly symmetric matrices, all eigenvectors should be orthogonal, but here they are not (in part due to numerics).

In order to see what the leading eigenvectors look like, we plot their N coefficients,

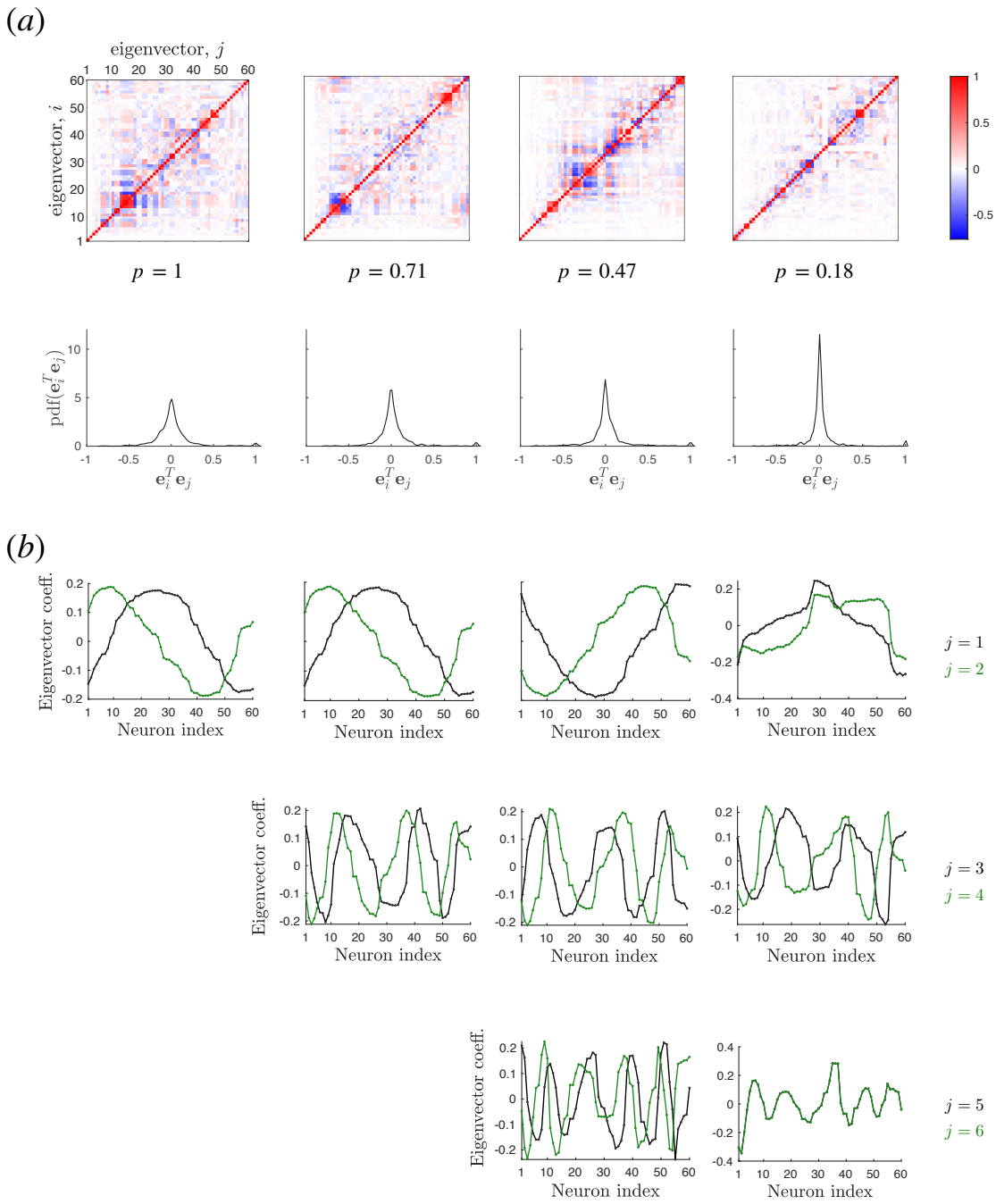


Figure (4.8) Structure of the leading eigenvectors of sparse recurrent connectivity. All results were obtained from networks of 60 neurons, which represent 2D inputs. **(a)** The pairwise product of eigenvectors of the recurrent matrix $\mathbf{\Omega}$, for four example network connection probabilities. Top: Each matrix row shows the real part of the scalar product of the i^{th} and j^{th} eigenvectors. The eigenvectors are sorted in a descending order, with respect to the absolute value of the real part of the eigenvalue to which they correspond. Bottom: The probability density function of the product of the i^{th} and j^{th} eigenvectors, $\mathbf{e}_i$ and $\mathbf{e}_j$. **(b)** The leading eigenvectors of $\mathbf{\Omega}$ for four example connection probabilities in (a). Top: The coefficients of the two leading eigenvectors, $j = 1, 2$. Middle: Same for $j = 3, 4$. Bottom: Same for $j = 5, 6$.

Figure (4.8) (b). We see that the two leading eigenvectors for the first three example connection probabilities are the same, with the third set having its sign reversed (top row, b). The next two leading eigenvectors for $p = 0.47$ and 0.71, also look the same if we ignore the sign flip for $j = 3$ (middle row, b). They look like a higher order harmonic of the first two leading eigenvectors. Lastly, the fifth and the sixth leading eigenvectors for $p = 0.71$ are a higher order harmonic of the third and the fourth eigenvectors (bottom row, b). In fact, the pair of leading eigenvectors, corresponding to $j = \{1, 2\}$, look like the first Fourier mode, $j = \{3, 4\}$ the third Fourier mode, and $j = \{5, 6\}$ the seventh Fourier mode. Each pair has a different phase shift applied to it.

We now return to the consideration of the eigenvectors of the recurrent matrix with $p = 0.18$. The eigenvectors corresponding to $j = \{3, 4\}$ resemble the same indexed eigenvectors for the higher connection probabilities (middle row, b). Otherwise, there is not much structure in its first eigenvectors. We recall that this matrix has all of its eigenvalues very close to zero, so there are no leading eigenvectors. We show it here to contrast the lack of structure in the very sparsely connected networks, with the presence of structure in the more highly connected networks.

As already discussed, the recurrent matrices are approximately symmetric. This means that they can approximately be represented in terms of their eigenvalues and eigenvectors:

$$\begin{aligned} \mathbf{\Omega} &= \mathbf{E}\mathbf{\Gamma}\mathbf{E}^T, \\ &= \sum_{i=1}^{N} \gamma_i \mathbf{e}_i \mathbf{e}_i^T, \text{ for } i = \{1, ..., N\}, \end{aligned} \tag{4.12}$$

where $\mathbf{E}$ is a matrix whose columns are the eigenvectors, $\mathbf{e}_i$, of $\mathbf{\Omega}$, and $\mathbf{\Gamma}$ is a diagonal matrix with the associated eigenvalues, γ_i, along its diagonal. As shown on the second line, this matrix decomposition can be written as a sum of the outer products of the eigenvectors, weighted by the eigenvalues. That is, the matrix $\mathbf{\Omega}$ can be written as a sum of N rank one matrices, (4.12).

Furthermore, if some of the eigenvalues are zero, then this sum will be taken just over the number of eigenvalues that are non-zero, approximated by the rank of the matrix. So, in the case of the fully connected matrix, which rank equals the dimensionality of the input, K, we need only consider the first K eigenvectors. In the case of $K = 2$, we can represent the matrix as a sum of two rank one matrices, as illustrated in Fig. (4.9) (a).

For the case of $\mathbf{\Omega}$ for $p = 0.71$, we can write it as a sum of four matrices of rank one, as this matrix has four non-zero eigenvalues. Considering that its first two eigenvectors are the same as for the fully connected matrix, we can substitute the sum of the first two matrices of rank one with the rank two fully connected matrix, which has been scaled down appropriately. The rank two matrix has to be scaled down to account for the fact that the two leading eigenvalues for $p = 0.71$ are smaller than for the fully connected matrix. Another way of saying this is that the sum of the first two matrices of rank one equals a scaled down version of the fully-connected matrix. Fig. (4.9) (b) illustrates that this rank four approximation recovers the original $\mathbf{\Omega}$ for $p = 0.71$.

Lastly, we follow the same logic for the case of $\mathbf{\Omega}$ for $p = 0.47$, which has six leading eigenvalues. Its low rank approximation is a sum of a scaled down rank two fully connected

matrix, and four matrices of rank one, Fig. (4.9) (c). The outer products over the third and fourth eigenvectors are shown as a single matrix, to illustrate the higher order patterning that they add on top of the underlying fully connected matrix. They also equal the scaled down version of the rank two sum of the third and fourth eigenvectors in (b). Again, this low rank approximation recovers the original matrix.

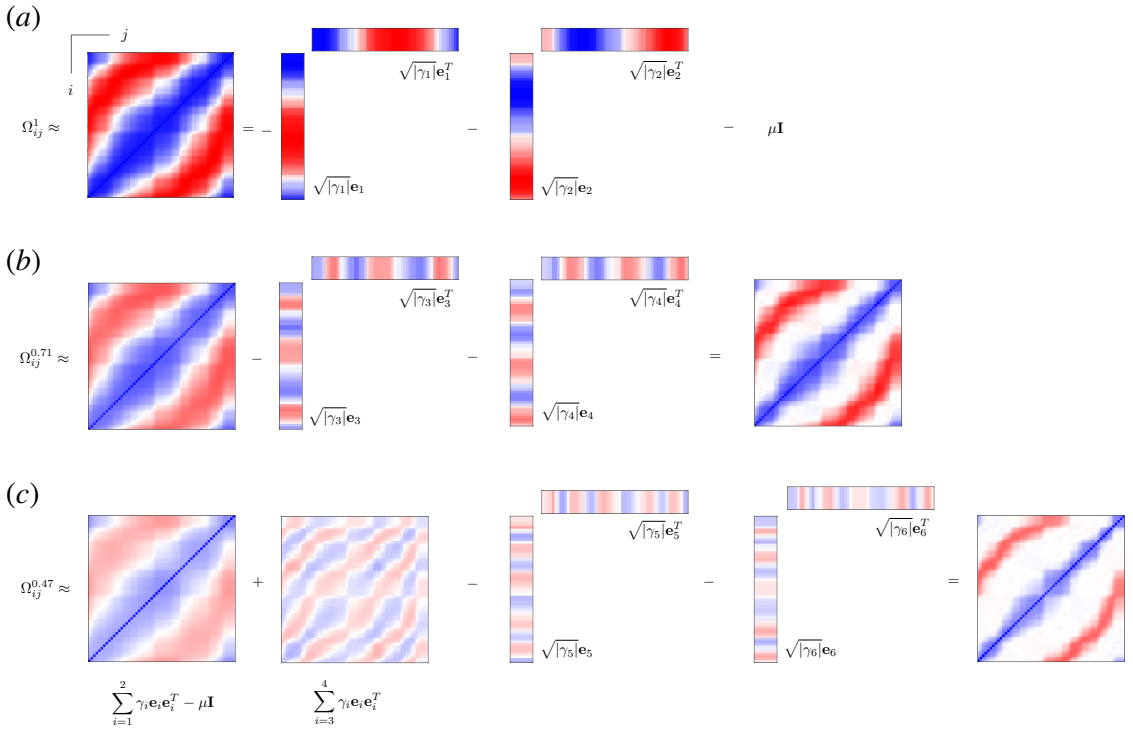


Figure (4.9) Low rank representations of the sparse recurrent matrices. All results were obtained from networks of 60 neurons, which represent 2D inputs. Three representative connection matrices, $\mathbf{\Omega}$, are considered for $p = 1$ (a), $p = 0.71$ (b), and $p = 0.47$ (c). The eigenvectors of each matrix in turn are denoted by $\mathbf{e}_i$, and the associated eigenvalues by γ_i. **(a)** The trained fully connected matrix can be approximated with a rank two matrix, which is a sum of the outer products of its leading eigenvectors, scaled by the two nonzero eigenvalues. **(b)** Same for $\mathbf{\Omega}$ with $p = 0.71$. We denote this matrix with $\mathbf{\Omega}^{0.71}$, and illustrate that its low rank approximation equals the sum of a scaled down version of the fully connected $\mathbf{\Omega}^1$, and weighted outer products over the third and fourth eigenvectors. **(c)** Same for $\mathbf{\Omega}$ with $p = 0.47$. We denote this matrix with $\mathbf{\Omega}^{0.47}$, and illustrate its low rank approximation as the sum of a scaled down version of the fully connected $\mathbf{\Omega}^1$, weighted outer products over the third and fourth eigenvectors (shown as a rank two matrix), and weighted outer products over its fifth and sixth eigenvectors.

One of the conclusions from the above discussion is that the low rank approximations of recurrent matrices, $\mathbf{\Omega}$, with intermediate to high connectivity, have a leading contribution equal to the scaled down version of the full matrix. This was a consequence of the fact that the leading eigenvectors of these recurrent matrices equal those of the full matrix.

We would like to formalize this relationship, so we recall an expression for the steady

state solution, (3.46), which we derived in the previous chapter. We rewrite it here for convenience, and include the full expression for diagonal self-resets too:

$$\begin{aligned}\boldsymbol{\Omega} + \mu\mathbf{I} = \quad &- \quad \beta\mathbf{F}\big\langle \mathbf{x}(t)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}\big\langle \big(\beta\mathbf{r}(t) + \tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}^{-1} \\ &- \quad \kappa(1-\delta_{ij})\big\langle \text{sgn}(\boldsymbol{\Omega})\tilde{\mathbf{o}}(t)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}\big\langle \big(\beta\mathbf{r}(t) + \tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t)\big\rangle_{\text{sp.}}^{-1}.\end{aligned} \tag{4.13}$$

Here, $\mathbf{r}(t)$ is a $N \times 1$ vector of population filtered spikes at time t, and $\tilde{\mathbf{o}}(t)$ is similar to the population spike train at time t, but it has the value one instead of the Dirac delta function in the position of the index of the neuron that fired a spike.

We can rewrite the above steady state solution more succinctly, as follows:

$$\boldsymbol{\Omega} + \mu\mathbf{I} = -\mathbf{F}\mathbf{X}\mathbf{R}^{-1} - \mathbf{S}\mathbf{R}^{-1}, \tag{4.14}$$

where $\mathbf{R}^{-1} \in \mathbb{R}^{N\times N}$ is the inverse of the β weighted firing rate matrix that appears in both terms on the right hand side of (4.13), $\mathbf{X} \in \mathbb{R}^{K\times N}$ is the β weighted average feedforward input at the time of population spikes, and $\mathbf{S} \in \mathbb{R}^{N\times N}$ is the κ weighted average sign of the outgoing recurrent connections at the time of population spikes. More specifically, the j^{th} column of the $K \times N$ matrix, $\mathbf{X}$, contains the feedforward input at the time of the spike of neuron j, averaged over all spikes of neuron j. The j^{th} column of the $N \times N$ matrix, $\mathbf{R}$, contains the average filtered spikes of all neurons, just before the spike of neuron j, weighted by β, and with a one added to the (j, j) element, to account for the sudden jump in the filtered spike of neuron j. The j^{th} column of the $N \times N$ matrix, $\mathbf{S}$, is the average sign of the connections sent out from neuron j to all other neurons, weighted by the parameter κ. The average is taken over all spikes of neuron j.

The first term on the right hand side of (4.14), is the scaled down all-to-all recurrent matrix, of rank K. We call this term the error-correcting term. The second term is the L_1 penalization matrix, which removes some of the connections, and increases the rank of the final recurrent matrix for lower connection probabilities. This is illustrated in Figure (4.10) (a), for $p = 0.71$.

Let's focus first on the structure and scale of the first term, $-\mathbf{F}\mathbf{X}\mathbf{R}^{-1}$. The structure of this matrix is identical to that of the matrix, $-\mathbf{F}\mathbf{X}$, and is scaled down by $\mathbf{R}^{-1}$ (top row, a). On the other hand, the second term, $\mathbf{S}\mathbf{R}^{-1}$, is influenced equally by the structure of the sign matrix, $\mathbf{S}$, and the inverse of the population firing rate, $\mathbf{R}^{-1}$. This is true for this connection probability, but the relative strength of the terms will change with increased sparseness.

As the connection probability decreases, the scale of the firing rate matrix $\mathbf{R}$ changes. Structurally, it always looks like a band matrix, but with decreasing p, the elements around the diagonal increase, and the non-zero band becomes wider. This is a consequence of neurons firing more spikes, and thus similarly tuned neurons (around the diagonal) will have larger non-zero filtered spikes, with increasing sparseness. This in turn influences the scale and structure of the $\mathbf{R}^{-1}$ matrix (bottom row, b).

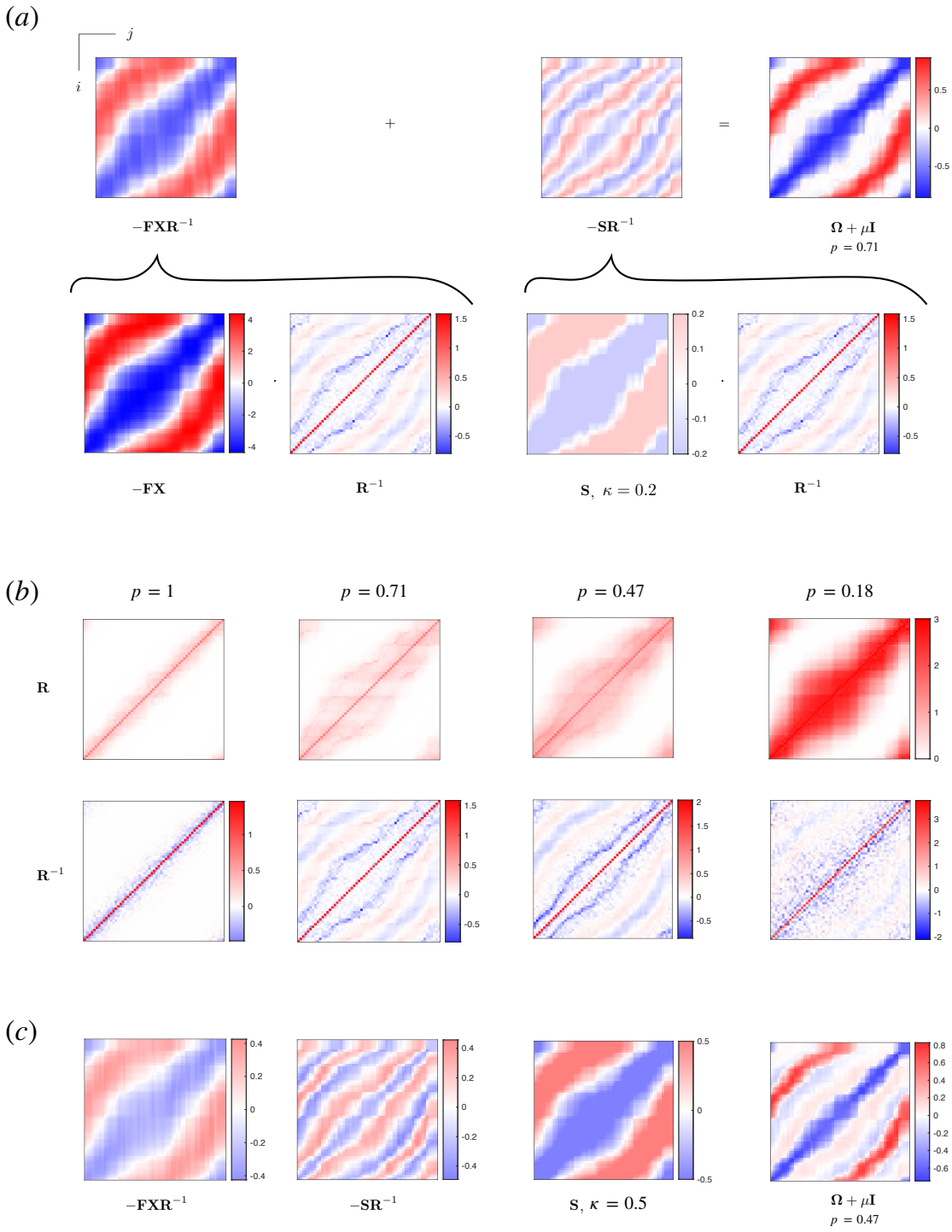


Figure (4.10) Decomposition of sparse recurrent connectivity as the sum of the error-correcting and L_1 penalization terms. Illustration of the solution, (4.14). Simulations were performed with trained networks of 60 neurons, which represent 2D inputs. **(a)** Top row: The recurrent matrix for $p = 0.71$, $\kappa = 0.2$, shown as a sum of two matrices: the error-correcting term (scaled-down full connectivity) and the L_1 penalization. Bottom row: The negative of the average feedforward input (scaled by β); the inverse of the average firing rate matrix (scaled by β); the κ weighted average sign of recurrent connections, and $\mathbf{R}^{-1}$ again, as it scales both terms in the sum shown in the top row. The averages in column j are taken over the spike times of the neuron j. **(b)** Matrices $\mathbf{R}$ and $\mathbf{R}^{-1}$, for four different connection probabilities, p. **(c)** Matrix decomposition for $p = 0.47$, $\kappa = 0.5$.

The changing scale and structure of the $\mathbf{R}^{-1}$ matrix, along with the increase in the strength of the sign matrix with κ, eventually lead to a change in the relative strength of the feedforward and L_1 penalization terms, Figure (4.10) (c). For $p = 0.47$ and $\kappa = 0.5$, the L_1 term is of the same order as the feedforward term. As κ gets closer to unity, both terms start to be dominated by the $\mathbf{R}^{-1}$, and decrease together toward zero.

4.6 | Optimal decoder for sparse networks

The decomposition (4.14) of the sparse recurrent matrix can also be related to its optimal decoder. We begin by recalling that we have already defined the optimal decoder for sparse connection matrices in the previous chapter,

$$\begin{aligned} \mathbf{D} &= \beta \big\langle \mathbf{x}(t)\tilde{\mathbf{o}}^T(t) \big\rangle_{\text{sp.}} \big\langle \big(\beta \mathbf{r}(t) + \tilde{\mathbf{o}}(t)\big)\tilde{\mathbf{o}}^T(t) \big\rangle_{\text{sp.}}^{-1}, \\ &= \mathbf{X}\mathbf{R}^{-1}, \end{aligned} \tag{4.15}$$

where we use the matrix notation from the previous section in the second line.

For $\beta = 2$, this expression is the same as for the optimal fully connected network. However, the element-wise values of the optimal linear decoder in a sparse network will be different from that for the all-to-all network, because of the different firing activity characteristics, embodied by the matrix $\mathbf{R}$.

We have confirmed numerically that the expression (4.15) leads to the same result as the linear least squares solution:

$$\mathbf{D} = \big\langle \mathbf{x}(t)\mathbf{r}^T(t) \big\rangle_{\text{sp.}} \big\langle \mathbf{r}(t)\mathbf{r}^T(t) \big\rangle_{\text{sp.}}^{-1}, \tag{4.16}$$

which defines the optimal linear decoder for a given input stimulus and the observed population firing rate.

In order to see how the optimal linear decoder changes with decreased connection probability, we consider what the product of the linear encoder and decoder looks like as a function of tuning similarity. Figure (4.11) (a) shows the negative product of the linear encoder and optimal linear decoder (green), as well as the learned connection weights (black). The plots show that the product $\mathbf{FD}$ is always a linear function of $\mathbf{F}^T\mathbf{F}$ (x-axis), and so the optimal linear decoder of each neuron is parallel to its linear encoder. For $p = 1$, $\mathbf{D} = \mathbf{F}^T$, and the negative of their product defines the recurrent connections, as expected. For $p < 1$, the linear decoder is a fraction of the linear encoder, $\mathbf{D} = a\mathbf{F}^T$, for some $0 < a < 1$. The factor a decreases with decreasing connection probability.

If we consider the columns, $\mathbf{D}_i$, of the decoder matrix, then $\|\mathbf{D}_i\|_2$ increases monotonically with connection probability, Figure (4.11) (b). This is equivalent to saying that it increases with the average recurrent strength in the network and with the average total recurrent input into a neuron. This makes sense, as the two latter variables increase monotonically with p. The decoder length decreases with the increase in the average firing rate, which is inversely proportional to p. In fact, the decoder length is inversely proportional to the firing rate, Figure (4.11) (c).

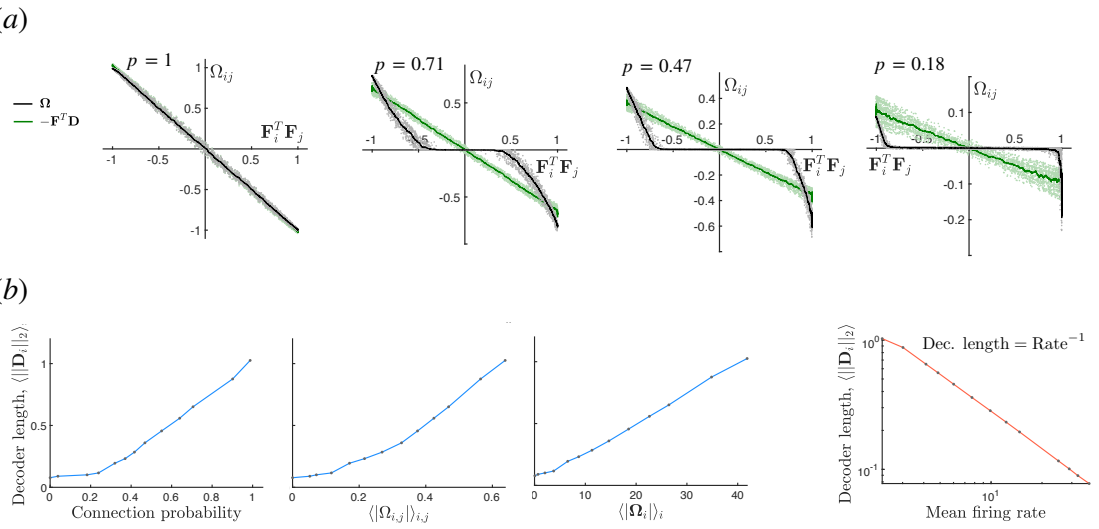


Figure (4.11) Optimal decoder for sparse networks. All results were obtained from networks of 60 neurons, which represent 2D inputs. **(a)** Comparing the trained recurrent weights to the negative product of the linear encoder and decoder, for increasingly more sparsely connected networks. The data is shown as a function of the tuning similarity of pairs of neurons. **(b)** Mean length of the linear decoder, as a function of network's connection probability, mean recurrent weight, and mean total recurrent input into a cell. **(c)** Mean length of the linear decoder vs. mean neuronal firing rate (log scale).

4.7 | The trade-off between firing and wiring

All networks which recurrent connections were trained with the sparseness learning rule, have shown an inverse relationship between the resulting connection probability and the network's firing rate. In order to summarize these results, we consider three example networks: an independent leaky-integrate-and fire network on one end of the spectrum, a recurrent network with an intermediate connection probability, and a recurrent network with all-to-all connectivity, on the other end of the spectrum, Figure (4.12) (a). We see that the networks that have fewer recurrent connections, fire more spikes, see panel (b).

In order to compare their performance systematically, we need to take care of the fact that the optimal linear reconstruction of the signal requires a decoder length which is inversely proportional to the firing rate. This means that as sparsely connected networks fire more population spikes, each spike contributes a smaller correction vector to the reconstructed signal. Consequently, the reconstruction error becomes somewhat smaller for more sparsely connected networks, but at the expense of a higher firing cost.

Given this inverse relationship between the error and rate in sparsely connected networks, one way to systematically assess their performance is to modulate the input signal strength so that the reconstruction error is fixed. We can then compare the performance of networks with different connection probabilities by comparing their average firing rates.

In the simulations presented in Figure (4.12) (b), we modulate the signal strength, so that it increases with decreasing connection probability. This ensured that the mean

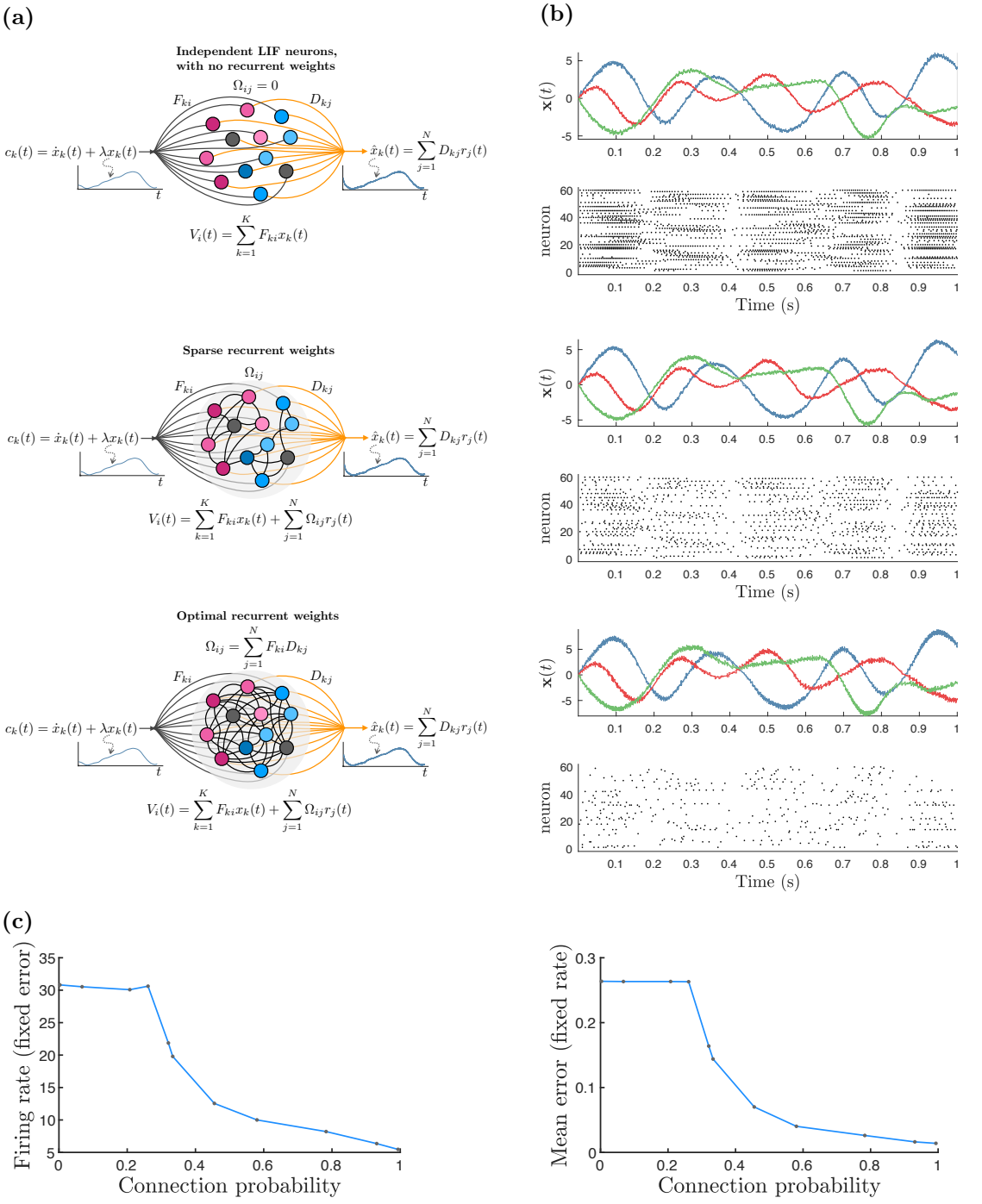


Figure (4.12) Trade-off between the firing rate and the number of synaptic connections. Simulations were performed with trained networks of 60 neurons, which represent 3D inputs. **(a)** Schematics of networks with no recurrent connections, intermediate connection probability, and all connections, respectively. **(b)** Trained recurrent networks with connection probabilities: $p = 0$, $p = 0.4$, and $p = 1$, have a firing rate of 31 Hz, 15.7 Hz, and 5.5 Hz, per neuron. The input strength is modulated so that the mean error is fixed at 0.014 of the signal strength. **(c)** Left: Firing rate as a function of connection probability (error fixed at 0.014). Right: Mean error as a function of connection probability (rate fixed at 5.5 Hz).

reconstruction error in all three simulations was indeed the same. The observed increase in the firing rate with increased sparseness, then means that there is a trade-off between the number of connections that a network has, and its performance, Figure (4.12) (c).

The decrease in the performance can be related to the rank of the recurrent connections. Given (4.15), we can rewrite the steady state $\mathbf{\Omega}$ as:

$$\mathbf{\Omega} = -\mathbf{FD} - \mathbf{SR}^{-1}, \tag{4.17}$$

where the first term on the RHS is a low rank matrix, its rank being given by the dimensionality of the input space. The second term, on the other hand, is a full rank matrix. The rank increase curve in Fig. (4.7) (b) follows closely the increase in the average firing rate with increased sparseness, Fig. (4.6) (d). In fact, the two can be seen as signatures of the decreased performance in comparison to the optimal all-to-all spike-coding framework.

4.8 | The balance between excitation and inhibition

Experiments in the early sensory systems have shown that there is a balance between excitatory and inhibitory inputs that a single neuron receives (Shu et al., 2003; Wehr and Zador, 2003; Isaacson and Scanziani, 2011). There is also experimental evidence for a temporally precise balance, in which inhibitory inputs balance excitatory ones with a millisecond resolution delay (Okun and Lampl, 2008; Graupner and Reyes, 2013; Tan et al., 2013; Salkoff et al., 2015). The optimal spike-coding model (Boerlin et al., 2013) is characterized by this kind of 'tight' balance, which can also be achieved in balanced networks (Renart et al., 2010; Vogels et al., 2011), with both dense and sparse connections. The classical balanced networks, with sparse connectivity, are characterized by 'loose' balance on larger time scales.

We will show the extension to Dalian networks, with true excitatory and inhibitory connections, at the end of this thesis, in Chapter 7. The non-Dale networks give a good idea about the level of E/I balance. We consider positive inputs into a cell to be excitatory, and negative inputs to be inhibitory, even though the synapses do not respect Dale's law.

The excitatory contribution to neural voltage can be written as,

$$V_i^E(t) = \max\left(\mathbf{Fx}(t), 0\right) + \max\left(\mathbf{\Omega r}(t), 0\right), \tag{4.18}$$

and the inhibitory contribution as,

$$V_i^I(t) = \min\left(\mathbf{Fx}(t), 0\right) + \min\left(\mathbf{\Omega r}(t), 0\right). \tag{4.19}$$

We study networks of 60 neurons which receive 2D inputs, and pick trained example networks with $p = 0.71$ and $p = 0.47$ connection probability. The excitatory input, $V_i^E(t)$, is negatively correlated with the inhibitory input, $V_i^I(t)$, in both of these sparsely connected spike coding networks. This is shown for an example neuron in each network, with the excitatory contribution in red and inhibitory contribution in blue (Fig. 4.13a,b). Summing these two contributions, recovers the voltage traces of these neurons over time, up to the imposed lower limit, $V_i \geq -T$. The more time that a neuron's voltage spends close to zero, the more detailed is the E/I balance. We recall that $V_i = 0$ corresponds to the resting

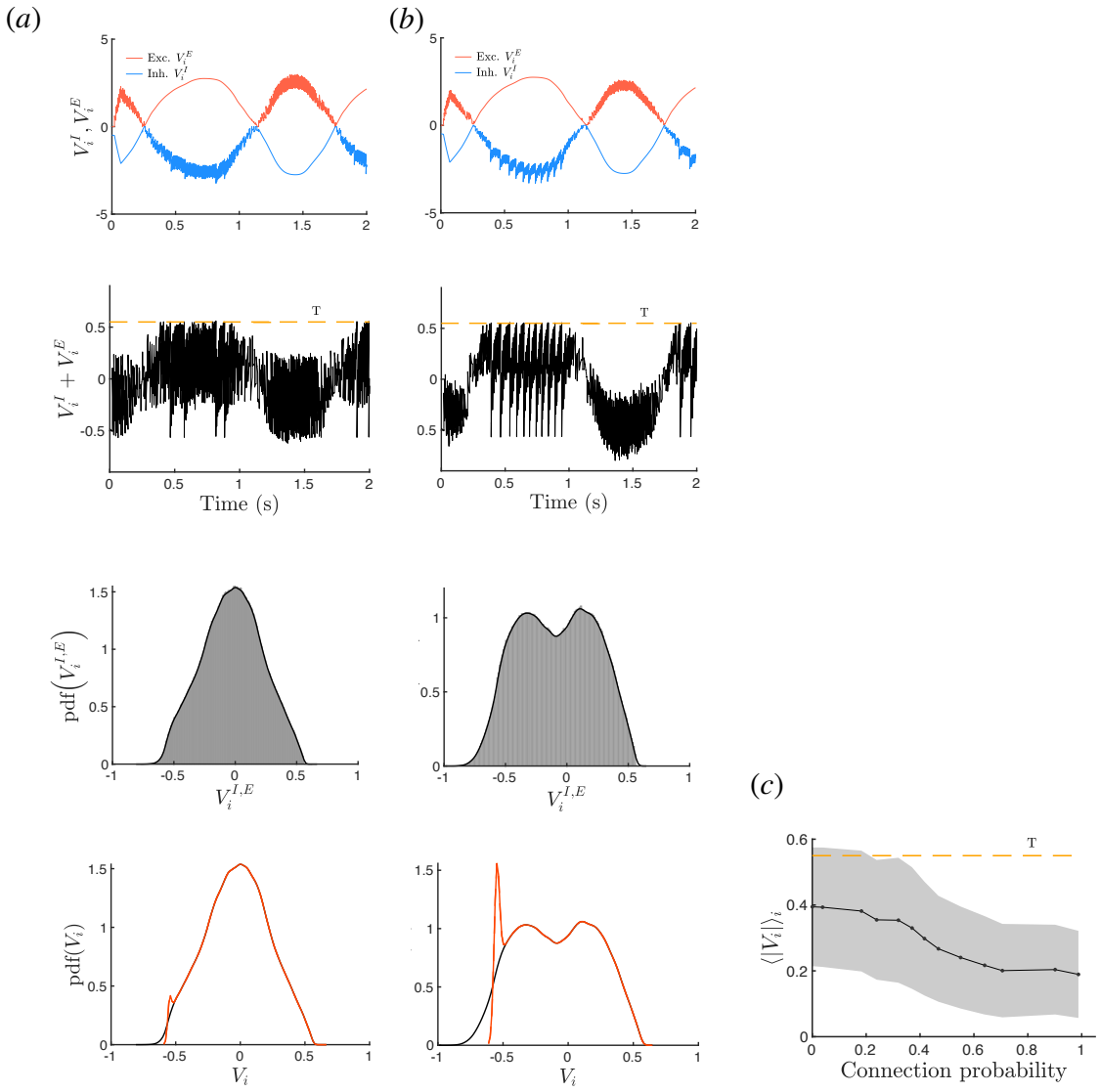


Figure (4.13) Balance of excitation and inhibition in sparse networks. All results were obtained with networks of $N = 60$ neurons, which receive two-dimensional inputs. Training was done with: $\beta = 1.8$, $g = 5$. The neural voltage, V_i, can be written as a sum of the contributions from excitatory and inhibitory currents (if we ignore the imposed lower limit, $V_i \geq -T$, for all i). Then, the excitatory contribution is, $V_i^E(t) = \max\left(\mathbf{Fx}(t), 0\right) + \max\left(\mathbf{\Omega r}(t), 0\right)$, and the inhibitory contribution is, $V_i^I(t) = \min\left(\mathbf{Fx}(t), 0\right) + \min\left(\mathbf{\Omega r}(t), 0\right)$. **(a)** Excitatory and inhibitory contributions to V_i in a network with $p = 0.71$. Top: $V_i^I(t)$ and $V_i^E(t)$ shown for an example neuron. Second row: The sum of the excitatory and inhibitory contributions, for the same example neuron. Third row: The distribution of the values of V_i^I and V_i^E for all neurons in the network. Bottom: The distribution of the values of V_i (red), with the distribution of V_i^I and V_i^E (black), for all neurons in the network. The distributions are the same, with the exception of V_i having a lower bound, $V_i \geq -T$, for all i. **(b)** Same as (a), for a network with $p = 0.47$. **(c)** Measure of E/I balance, as a function of connection probability. The measure is the mean and standard deviation of absolute neural voltages, taken over all neurons in a network.

potential, -65 mV, with $V_i = -T$ corresponding to the reset potential of about -70mV, and T denoting the spiking threshold at about -45mV.

The distribution of the sum of excitatory and inhibitory contributions, $V_i^I(t) + V_i^E(t)$, for all neurons i over time, measures the E/I balance in the whole network. This distribution has a peak at zero in the $p = 0.71$ network, with most of its mass in the vicinity of the zero point. On the other hand, the distribution has a different shape in the $p = 0.47$ network. Here, there are two peaks close to $\pm T/2$, and we expect that the peaks would move closer to $\pm T$ in very sparsely connected networks.

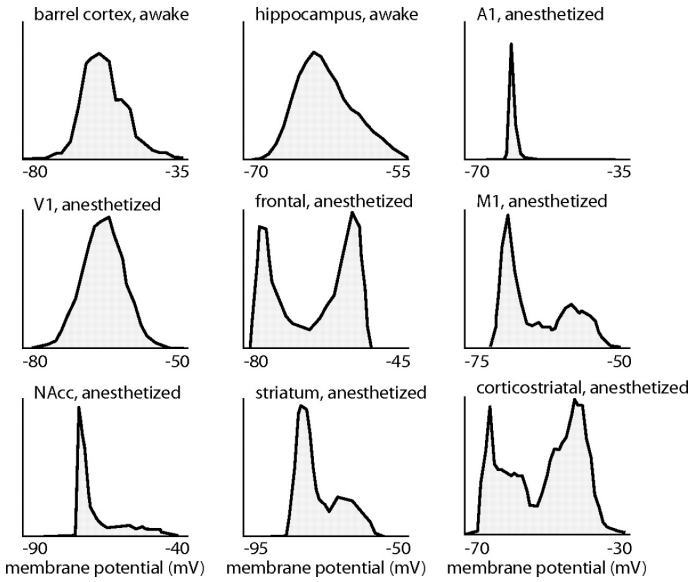


Figure (4.14) Membrane potential distribution in *in vivo* intracellular recordings in various areas, reproduced with the exact caption from (Rossant et al., 2011). From left to right and top to bottom: barrel cortex (Crochet and Petersen, 2006), hippocampus (Harvey et al., 2009), primary auditory cortex (DeWeese and Zador, 2006), primary visual cortex (Azouz and Gray, 1999), frontal cortex (Leger et al., 2005), primary motor cortex (Brecht et al., 2004), nucleus accumbens (ventral striatum) (Goto and O'Donnell, 2001), neostriatum (Wilson and Kawaguchi, 1996), corticostriatal neurons (Stern et al., 1997).

In fact, as the network becomes less connected and less coordinated, two subpopulations of neurons start to emerge. One group is depolarized by the present input, and the other group is hyperpolarized by it. The recurrent connectivity is not strong enough to counteract this and bring all voltages toward zero. The two peaks thus correspond to the two subpopulations. This corresponds to some experimental recordings, see (Fig. 4.14) from (Rossant et al., 2011).

The absolute mean voltage over time and over the whole population is a measure of the average level of E/I balance. As the connection probability decreases, the balance becomes more poor (Fig. 4.13c). This might indicate that networks with different connection probabilities are characterized by different levels of E/I balance. We will investigate this as a potential way to explain the variability in experimental observations.

The loss of tight balance with sparser connectivity will be impacted by input statistics, and it would be interesting to study this impact. An appropriate way of assessing the level of balance in that case would be with the mean sum of excitatory and inhibitory inputs, normalized by their difference. The former is given by the mean subthreshold voltage, and the latter by the scales of the absolute feedforward and absolute recurrent inputs. We will investigate this as part of future work.

4.9 | Sensitivity to network parameters

The final connection probability and performance of trained networks depend on various parameters. Figure (4.15) (a-d) shows the dependence on the number of neurons in the network, N, the synaptic scaling parameter, β, the input gain, g, and the quadratic cost, μ.

Increasing the network size N leads to a decreasing final connection probability, given a fixed sparseness parameter $\kappa = 0.2$ (Fig. 4.15a). Increasing the parameter β, when κ is fixed, leads to more connected networks (Fig. 4.15b). This is because the influence of κ is scaled by $1/\beta$. Increasing input gain, first leads to larger connection probabilities, when the gain is closer to unity, but then after $g = 5$ it starts to level off at a smaller connection probability (Fig. 4.15c). Increasing the quadratic cost parameter, μ, increases the final connection probability (Fig. 4.15d). We also observed that a smaller quadratic cost is needed for distributed spiking in networks with more neurons.

Lastly, we checked what happens if we suddenly increase input gain in a network which was trained with $g = 5$ (Fig. 4.15f). This shows whether a sparse network trained on smaller gain inputs remains stable when we suddenly increase the gain. The network functions well, and as expected increases its firing rate in response to larger inputs.

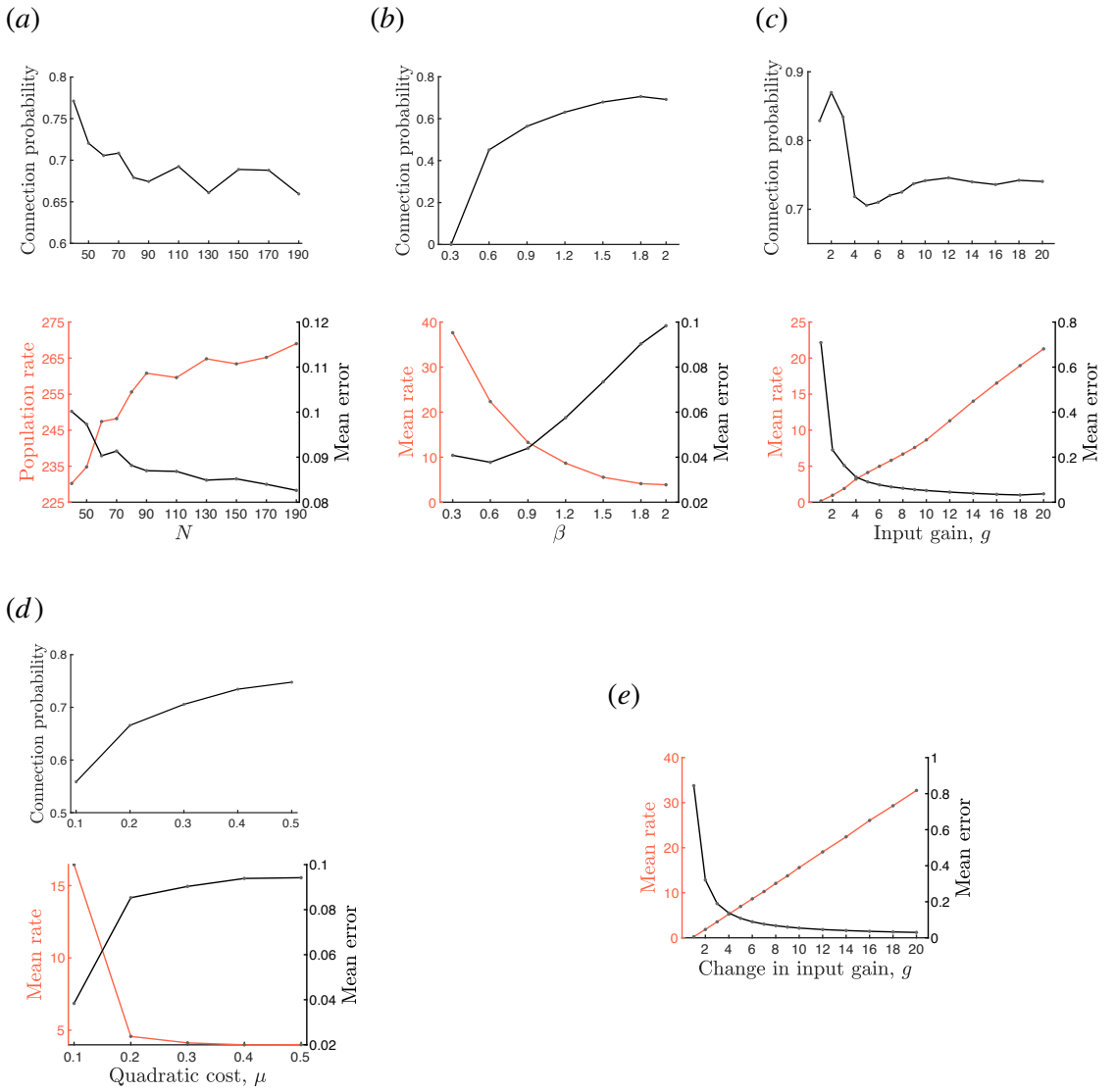


Figure (4.15) Influence of parameter values on connection probability and performance. All results were obtained with networks of neurons which receive a two-dimensional input, with the sparseness constraint fixed at $\kappa = 0.2$. (a-d) show results from networks which were trained with different values of the parameter, shown on the x-axis. The default parameter values were: $N = 60$, $\beta = 1.8$, $\mu = 0.3$, and $g = 5$. **(a)** Influence of network size, N. **(b)** Influence of the synaptic scaling parameter, β. **(c)** Influence of input gain, g. **(d)** Influence of quadratic cost, μ. **(e)** Influence of a sudden change of input gain, g, in a network which was trained with $g = 5$. There is no learning after the input suddenly changes its gain. If the new gain is constant for some time, then the learning in the network at the new gain level is described by the results shown in (c).

4.10 | Alternative sparse spike-coding networks

4.10.1 | Pruning the weakest and fixing the strongest connections

We showed in this chapter that it is possible to learn sparse spike-coding networks. However, what would have happened if we had tried to construct networks with this goal in mind?

We consider the optimal coordinated spike-coding model (Chapter 2), which prescribes specific recurrent connections, (3.8), as optimal ones for signal representation. We can then build networks with different connection probabilities by pruning the optimal connections, Ω_{ij}, which are the smallest in magnitude. This is motivated by the idea that the strongest connections are the most important for efficient representation.

An optimal connection in a fully connected network equals the negative scalar product of two random vectors of length one, namely, the linear encoder of the postsynaptic neuron and the linear decoder of the presynaptic neuron. So, the absolute value of the recurrent weights is in the range from zero to one, and the magnitude of self-connections will be given by, $|\Omega_{ii}| = 1 + \mu$. A network which receives a two-dimensional feedforward input has a uniform distribution of recurrent weights when $p = 1$ (Fig. 4.16a, purple, left panel). We build networks which have $N = 60$ neurons, and prune the smallest connections to build progressively sparser networks (Fig. 4.16a, purple).

The linear cost, and thus the threshold, have a big impact on the performance of sparse networks built in this way (Fig. 4.16b). If we pick a small linear cost that leads to the smallest firing rate in fully connected networks, and use that same linear cost to define recurrent connections of increasingly less connected networks, the tracking of the input signal becomes very poor, orange line in (Fig. 4.16b). This significant degradation in performance is caused by the many additional spikes that are fired by all neurons in the network.

These additional spikes result from strong excitatory connections between dissimilarly tuned neurons, which cause a spike of a neuron that does not correct the error immediately after a spike that does decrease the error, red spikes in (Fig. 4.17). If we instead keep increasing the linear cost as we build more sparsely connected networks, then the resulting networks do keep the error in check. The larger linear cost would have made the fully connected network less efficient, but it helps keep the error and firing rate bounded as we build sparsely connected networks. There is a gradual increase in the firing rate with increased sparseness, but we confirmed that these spikes were not a consequence of the instability caused by strong excitatory connections.

We also found the best linear costs for each connection probability, as the smallest linear costs which did not lead to instabilities, and the resulting firing rate and reconstruction error are shown in red in (Fig. 4.16b). These networks have the best performance among the networks with strong structured connections, where the sparseness was achieved by the pruning of the smallest weights. The histograms of weights correspond to these best linear costs (Fig. 4.16a, purple).

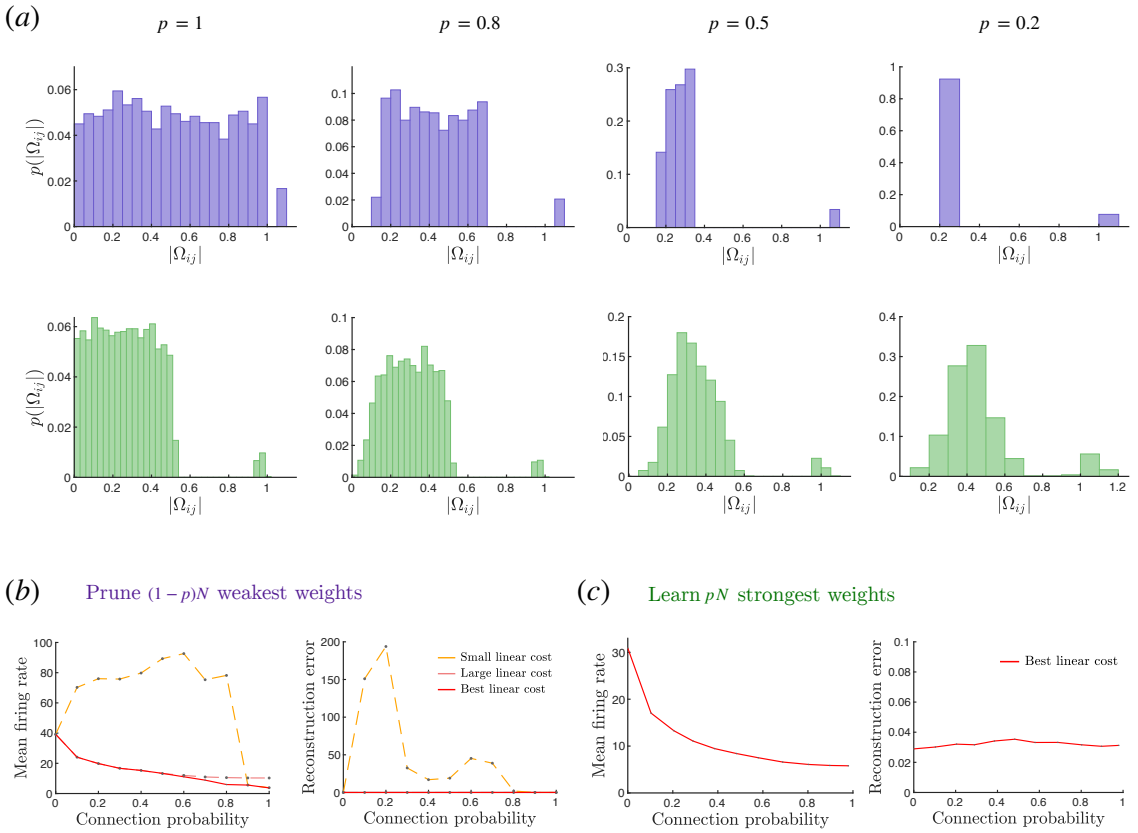


Figure (4.16) Characteristics and performance of alternative sparse spike-coding networks. **(a)** Comparison of the recurrent strengths in constructed and learned networks of different connections probabilities. The example networks consist of 60 neurons and their function is to represent 3D inputs. The top row (purple) shows the recurrent strengths in constructed networks, where a percentage of connections was pruned. The bottom row (green) shows the strengths in networks which were learned using the local learning rule for select connections. Here, depending on the desired final connection probability, a subset of recurrent connections were trained using the local learning rule, which in a fully-connected network would converge to optimal connections. The remaining (small) connections were set to zero and were never learned. **(b)** The performance of the constructed sparse networks (top row in a), with three different linear costs. **(c)** The performance of the learned sparse networks (bottom row in a), with the best linear cost.

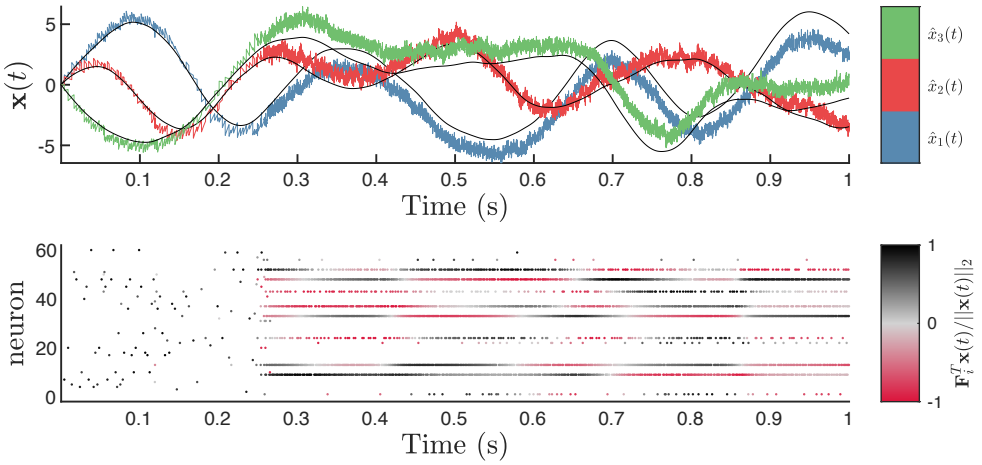


Figure (4.17) Instability of constructed sparse networks with small linear costs (desired low error tolerance). The example network consists of 60 neurons which collectively represent 3D inputs in their spiking activity. The recurrent connections were set to the optimal values (with small linear costs), and a network of 88% connection probability was built by pruning 12% of the smallest weights in the network.

4.10.2 | Pruning the weakest and learning the strongest connections

Alternatively, we could have applied the local balance-restoring learning rule, which in a fully connected network converges to the optimal weights (Brendel et al., 2017).

Analogous to the pruning of the smallest connections, here we do not allow the smallest connections to ever be learned. The optimal connectivity matrix already gives us the information about which pairs of neurons share the smallest weights (orthogonally tuned ones), and so we can fix those connections to always have a weight of zero. The remaining connections are learned using the local balance restoring rule. In order to learn stable networks, which do not have divergent firing rates, we have to use progressively smaller values of β as we decrease the connection probability. (Fig. 4.16c) shows the resulting neuronal firing rates and reconstruction errors, as a function of connection probability, for these best linear costs.

Figure (4.16a) compares the recurrent strengths in constructed and learned networks, where a certain percentage of connections were either set to zero, or never allowed to be learned, respectively. The distributions of the constructed recurrent weights are a lot more uniform, at least in the low-dimensional input coding scenarios. In both cases, the connections that were not present in the network were the ones that would have been the smallest in magnitude in the fully connected optimal network. If we had done the reverse for example and built sparsely connected networks by not allowing the strongest connections to exist in the network, then the performance would have been much worse. For example, a fifty percent connected network consisting of sixty neurons which collectively represent three-dimensional inputs, would have had a reconstruction error which is almost twenty times higher if the connections that are removed are the strongest optimal ones, as opposed to the smallest ones. Thus, which particular connections do not exist in a network has a significant influence on the performance of the network.

In summary, we see that we can construct sparse spike-coding networks, as well as learn them using the balance restoring rule. However, this requires a lot of fine tuning, as the networks are not derived in a principled way. Both of these two methods approximate the L_0 constrained sparsification of network connections. They are very prone to instabilities, and require a lot of fine tuning to ensure stability. These two methods are not a good alternative to learning optimal connectivity with our Lasso-regularized synaptic rule, which optimizes both efficient representation and recurrent connection probability.

4.11 | Conclusion

We have shown that the synaptic rule, which optimizes efficient representation with the number of recurrent connections, leads to stable networks of any connection probability. The learning converges to a steady state, in which the connections that impact the performance the least are pruned, and the remaining ones optimized for efficient signal representation. The learned synaptic connections are structured and share many common features with the optimal weights in the fully connected networks. They can be decomposed into a sum of a low rank matrix, signature of optimality, and additional rank one matrices, which become more dominant, as the connectivity becomes more sparse.

The networks fire more spikes when they have fewer recurrent connections, and their efficiency decays. However, they still perform better than a network of independent leaky integrate-and-fire neurons. There is a trade-off between the operational cost, associated with the firing activity of the network, and the developmental cost, associated with the growth of synapses. The synaptic rule provides a way to minimize the firing cost in a network that can only grow a certain number of connections. Similarly, it provides a way to minimize the number of synapses that are developed in a network that can only afford a certain level of firing activity.

5 | Learning Spiking Networks with Synaptic Dynamics

All models are necessarily *idealizations* of the complex phenomena that they describe. Knowing which variables are essential, and which ones can be ignored, is crucial in the search for simple underlying principles, which govern complex systems. In our work so far, we have been neglecting synaptic delays and dynamics, which characterize synaptic communication in biological neural networks. The classical spike-coding framework makes the simplifying assumption of instantaneous synapses, in part because it makes possible the derivation of an optimal network. The optimal network has the greatest coding accuracy for a fixed population firing rate (Chapter 2), and an optimal network in the context of realistic synaptic dynamics has not been proposed. What has been studied, however, is the impact of synaptic delays and finite time synaptic dynamics on the performance of the network, which is optimal when the synaptic interactions are instantaneous. The result was spiking synchronization, which degrades the network performance, but can be improved by adding an optimal level of synaptic noise (Chalk et al., 2016; Schwemmer et al., 2015). The biological detail, in this case, makes an important impact on the performance of the idealized model.

In this chapter, we are going to add the biological detail of realistic synapses to spike coding networks with sparse recurrent connectivity. We are going to learn these connections based on a learning rule, which we adapt from the learning rule proposed in the context of instantaneous interactions (Chapter 3). We will show that the trained sparse networks represent multi-dimensional variables more accurately than an independent network. Furthermore, we will see that the application of the learning rule to the edge case of full recurrent connectivity leads to synaptic weights which are smaller than those proposed by the idealized optimal network with instantaneous interactions. Lastly, we will see that sparse connectivity can act to desynchronize population activity. Certain levels of sparseness might thus act as a source of noise, effectively improving the performance of a spiking network in the presence of realistic synaptic delays.

5.1 | Defining the network model

Network dynamics

We study a network of leaky integrate-and-fire neurons, which collectively represent a multi-dimensional variable in their spike trains. The subthreshold potential of each neuron has the following dynamics:

$$\dot{V}_i(t) = -\lambda V_i(t) + \sum_{k=1}^{K} F_{ik} c_k(t) + \sum_{n=1}^{T_j} \sum_{\substack{j=1 \\ j \neq i}}^{N} \Omega_{ij} h(t - t_j^n) + \Omega_{ii} o_i(t). \tag{5.1}$$

Here, a neuron $i \in \{1, 2, ..., N\}$, receives K feedforward input currents, $c_k(t)$, projected onto its encoding weights, F_{ik}, and at most $N - 1$ synaptic currents, $h(t - t_j^n)$, weighted by the recurrent weights, Ω_{ij}. The linear encoding weights, F_{ik}, are picked randomly, and then the length of the encoding vector, $\mathbf{F}_i = (F_{i1}, F_{i2}, ..., F_{iK})$, is normalized to equal one. The input currents, $c_k(t)$, could be smoothly varying or constant.

The synaptic kernel, $h(t - t_j^n)$, is a function of the time elapsed since the n^{th} spike of neuron j, given by t_j^n, and we define it as the following double-exponential function:

$$h(t) = \begin{cases} \frac{1}{\tau_d - \tau_r} \left(e^{-(t-\tau_{tr})/\tau_d} - e^{-(t-\tau_{tr})/\tau_r} \right), & \text{if } t > \tau_{tr}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Here, τ_r and τ_d are the finite rise and decay times of the synaptic kernel, and τ_{tr} is the spike transmission delay. The synaptic kernel, $h(t)$, models the unweighted postsynaptic current from neuron j to neuron i, and its integral equals one. Weighting by the synaptic weight, Ω_{ij}, then gives the synaptic current received by neuron i from neuron j, over the course of duration of $h(t)$. The particular choice of the synaptic kernel is motivated by the fact that, in spite of its simplicity, it provides a good description of the time course of many synaptic conductances (Roth and van Rossum, 2009). An example of the synaptic kernel, $h(t)$, with a 1 ms transmission delay, is shown in Figure 5.1 (right column, middle panel).

The voltage dynamics, (5.1), also includes a membrane potential leak term, with the time scale determined by the time constant, λ. Neurons spike when their voltage reaches a fixed firing threshold, T, which is the same for all neurons. Their voltage is then instantaneously reset to a hyperpolarized level by an autaptic current, $\Omega_{ii} o_i(t)$. Here, $o_i(t) = \sum_n \delta(t - t_i^n)$ is the spike train of neuron i, and δ is the Dirac delta function.

Membrane potential

The exact value of the membrane potential, $V_i(t)$, at any time, $t > 0$, can be derived by integrating the network dynamics, (5.1):

$$V_i(t) = \sum_{k=1}^{K} F_{ik} \int_0^t c_k(t') e^{-\lambda(t-t')} dt' + \sum_{n=1}^{T_j} \sum_{\substack{j=1 \\ j \neq i}}^{N} \Omega_{ij} \int_0^t h(t' - t_j^n) e^{-\lambda(t-t')} dt'$$
$$+ \Omega_{ii} \sum_{n=1}^{T_i} e^{-\lambda(t-t_i^n)}.$$

We are going to express this solution in terms of the signal, $x_k(t)$, the filtered synaptic kernel, $\tilde{r}_j(t)$, and the filtered self-reset current, $r_i(t)$, which we describe here.

As in previous chapters, the variable that is represented by the network is $x_k(t)$, which is a leaky integral of the input current, $c_k(t)$:

$$x_k(t) = \int_0^t c_k(t') e^{-\lambda(t-t')} dt'. \tag{5.3}$$

The double-exponential synaptic kernel, $h(t)$, is filtered by the membrane of the postsynaptic neuron to obtain the unweighted postsynaptic potential (PSP), or filtered synaptic kernel, $\tilde{r}_j(t)$. The filtered synaptic kernel has the following dynamics:

$$\dot{\tilde{r}}_j(t) = -\lambda \tilde{r}_j(t) + \sum_{n=1}^{T_j} h(t - t_j^n). \tag{5.4}$$

Integrating its temporal dynamics, we obtain the unweighted PSP contribution to $V_i(t)$, caused by the spikes of neuron j:

$$\tilde{r}_j(t) = \sum_{n=1}^{T_j} \int_0^t h(t' - t_j^n) e^{-\lambda(t-t')} dt'. \tag{5.5}$$

Lastly, the filtered self-reset current, $r_i(t)$, is the unweighted hyperpolarizing PSP contribution to $V_i(t)$, from the neuron i itself:

$$r_i(t) = \sum_{n=1}^{T_i} e^{-\lambda(t-t_i^n)}, \tag{5.6}$$

where t_i^n are the spike times of neuron i. In the previous chapters, all filtered currents were described by this expression, but now with recurrent synaptic dynamics, the only instantaneous currents are the intrinsic self-resetting ones. Given that filtering by the membrane occurs on a time scale that is much longer than that of synaptic current dynamics, $r_i(t)$ and $\tilde{r}_j(t)$ are quite similar (Fig. 5.1, bottom row, plotted for a single presynaptic spike). The most important difference between them is in fact the synaptic transmission delay.

Putting this together, we now express the subthreshold membrane potential as follows:

$$V_i(t) = \sum_{k=1}^{K} F_{ik} x_k(t) + \sum_{\substack{j=1 \\ j \neq i}}^{N} \Omega_{ij} \tilde{r}_j(t) + \Omega_{ii} r_i(t). \tag{5.7}$$

The neural voltage is given by the sum of the filtered feedforward input, weighted by the neuron's encoding weights, the filtered double-exponential synaptic input from the neural population, weighted by the recurrent weights, and the filtered intrinsic input, which resets the voltage after neuron's own spikes.

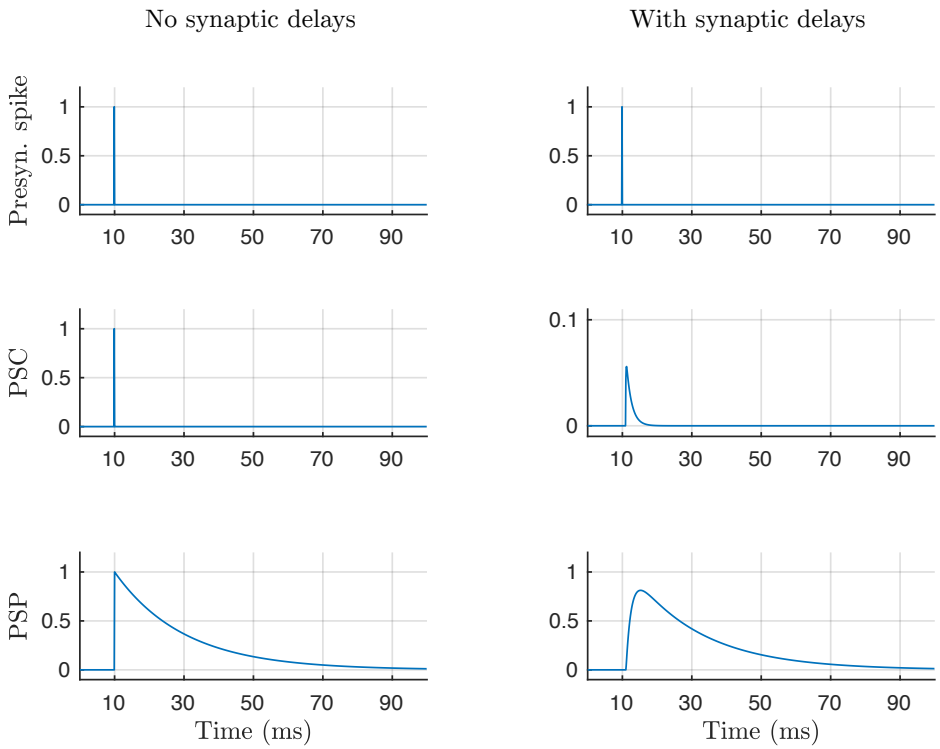


Figure (5.1) The effect of a presynaptic spike on the postsynaptic neuron. The plots on the left show the effect in the model which assumes instantaneous interactions, while as the ones on the right show the effect in the model which incorporates 1 ms delays with AMPA synaptic dynamics. Top: The example presynaptic spike occurs at $t = 10$ ms, and is modeled as an instantaneous event. Middle: The postsynaptic current in the model with instantaneous interactions is modeled like an instantaneous event. The postsynaptic current in the model with synaptic dynamics has a delay of 1 ms, and AMPA dynamics. Bottom: Postsynaptic potential in the model with instantaneous interactions, and the model with synaptic dynamics, respectively. The PSPs in both models are the corresponding PSCs, filtered by the membrane with the time constant, $1/\lambda = 20$ ms.

Decoding the network activity

Depending on the specific recurrent connectivity, the spiking activity of the network can encode an input variable, $x_k(t)$, which is a leaky integral of the feedforward input current, $c_k(t)$.

The ability of the network to perform this function is measured by the difference between the true value of the variable, $x_k(t)$, and its estimated value, $\hat{x}_k(t)$, derived from the network's spiking activity. As described in Chapters 2 and 3, the reconstructed variable is decoded linearly from the filtered spikes of the network. Given that we introduced realistic recurrent synaptic dynamics, we could have defined the reconstructed variable based on the filtered synaptic kernel, $\tilde{r}_j(t)$. This would have given us a reconstructed variable which tracks the true variable with a small lag, equal to the synaptic transmission delay.

However, we keep the old definition:

$$\hat{x}_k(t) = \sum_{j=1}^{N} D_{kj} r_j(t), \tag{5.8}$$

where $r_j(t)$ is the filtered spike. The particular choice of the spike filtering kernel is not essential to the assessment of the network's decoding accuracy. We suspect that the more realistic choice of the kernel would have increased the decoding error slightly. The effect should be equivalent to that of delaying an error correcting spike by 1ms, which should be negligible. However, it would be good to revisit this assumption, as an extension of this work.

The linear decoder is set to equal the optimal decoder, (2.14). This is the least-squares linear regression solution, given a population filtered spike vector, $\mathbf{r}(t)$, and the input vector $\mathbf{x}(t)$:

$$\mathbf{D} = \left\langle \mathbf{x}(t)\mathbf{r}^T(t) \right\rangle_t \left\langle \mathbf{r}(t)\mathbf{r}^T(t) \right\rangle_t^{-1}. \tag{5.9}$$

Stable coordination of network activity

A network with synaptic delays can suffer from unstable activity, which reflects negatively on its ability to represent a signal. Here, we briefly discuss what it means for a network to be stable or unstable, in the context of coordinated spike coding.

Given the random distribution of encoding vectors, a spiking population will always be split into two groups: about one half of the neurons are activated by the present input, and the other half is hyperpolarized by it. If the spikes are coming from the subset of neurons which are activated by the input, then these are generally error-correcting spikes. On the other hand, if they come from the set of neurons which are depressed by the input, then these spikes increase the error. In the latter case, the neurons are not spiking in response to the feedforward input. They emit spikes because of the destabilizing recurrent inputs from neurons in the active half of the population. We characterize networks which have this behaviour as unstable. The error-increasing spikes can further cause neurons from the active subpopulation to fire more spikes, thus leading to greater instability.

In the context of coordinated spike coding, we define a *stable* network to be one in which the spiking activity is generated in response to excitatory *feedforward* inputs. In contrast, an *unstable* network is one in which the spiking activity is generated in response to excitatory *recurrent* inputs.

Network structure affects decoding accuracy

The structure of the recurrent connections has a significant impact on the performance of the network in the presence of synaptic delays. We will discuss this in more detail at the end of this chapter, where we consider previous work which investigated this impact. For now, we recall that the optimal recurrent connections, Ω_{ij}, in the absence of delays, have a low rank structure, with the rank being given by the dimensionality of the input.

The optimal recurrent weight between two neurons can be written as the negative product of the linear encoder of the postsynaptic neuron and the linear decoder of the presynaptic neuron. Ignoring quadratic costs, this allows us to write the voltage of each neuron as the projection of the network's reconstruction error onto the neuron's encoding weights:

$$V_i(t) = \sum_{k=1}^{K} F_{ik}\Big(x_k(t) - \sum_{j=1}^{N} D_{kj} r_j(t)\Big). \tag{5.10}$$

This is key to the precise and efficient representation of the input by the optimal network, which fires a minimal number of spikes. The optimal recurrent weights are not known for the network which has synaptic delays, but we expect that they should still keep the symmetry and low rank properties that are key in the idealized network. Recurrent weights that lead to well behaving networks with synaptic delays can in fact be learnt, and we develop this approach in the following section.

5.2 | Learning sparse CSC networks with synaptic delays and dynamics

Having succeeded at learning sparsely connected coordinated spike coding networks in Chapters 3 and 4, we now adapt the learning rule to the case of synaptic delays and dynamics. As before, we will define a loss function, which motivates the derivation of the synaptic learning rule.

5.2.1 | The loss function

We define the loss function in the context of synaptic delays and dynamics, as follows:

$$L = \Big\langle \sum_{i=1}^{N} \big(V_i(t) + \mu r_i(t)\big)^2 + \kappa \sum_{i,j=1}^{N} |\Omega_{ij}(t)| h_j(t) \Big\rangle_t, \tag{5.11}$$

where we introduce the notation, $h_j(t) = \sum_n h(t - t_j^n)$, to denote the sum of all unweighted synaptic currents induced by neuron j. As in the case of the loss function for networks with instantaneous connections, the loss function here measures the average deviations from zero of neural voltages penalized by firing rate quadratic costs, and the average magnitude of all recurrent synaptic weights. The sparseness parameter, κ, controls how much weight is given to either term.

We make a modeling choice of scaling the synaptic weight term by the synaptic kernel, which means that the minimization of weights will be done over the duration of the postsynaptic current. Alternatively, we could have scaled it by the filtered synaptic kernel, but we will see shortly that this would have not been a good choice. As defined, the loss function relies only on locally available information.

Taking the derivative with respect to $\Omega_{ij}(t)$, gives us the gradient of the loss function:

$$\frac{\partial L}{\partial \Omega_{ij}(t)} \approx \Big\langle 2\big(V_i(t) + \mu r_i(t)\big) \frac{\partial V_i(t)}{\partial \Omega_{ij}(t)} + \kappa \operatorname{sgn}(\Omega_{ij}(t)) h_j(t) \Big\rangle_t + \text{non-local terms.}$$

In the above expression, we have made the assumption that we can neglect the partial derivative of $r_i(t)$ with respect to Ω_{ij}. If we also ignore the non-local terms, as biologically plausible learning should be based on local computations, then the approximation of the gradient of the loss function becomes:

$$\frac{\partial L}{\partial \Omega_{ij}(t)} \approx \left\langle 2\big(V_i(t) + \mu r_i(t)\big)\tilde{r}_j(t) + \kappa \operatorname{sgn}(\Omega_{ij}(t)) h_j(t) \right\rangle_t, \tag{5.12}$$

where we have substituted in the expression for the partial derivative of $V_i(t)$ by differentiating (5.7). In the above approximation, $\tilde{r}_j$ is replaced by $r_j(t)$, for autapses, when $i = j$.

We could now define the temporal evolution of synaptic weights to be proportional to the negative of the sum, which is inside the averaging brackets on the right-hand side. If we were to define the synaptic weight change to be driven by the negative of the approximate gradient, we would find that this learning rule actually does not work well in simulated neural networks. The reason behind this is that the learning would be occurring during the entire length of the postsynaptic potential, $\tilde{r}_j(t)$, on the order of tens of milliseconds. This is a long enough time that other neurons will be firing spikes and sending synaptic currents to a postsynaptic neuron i. The neuron i would then be concurrently updating multiple connections, Ω_{ij}, in relation to its membrane voltage, V_i, which includes the sum of multiple filtered synaptic currents from various neurons j. This leads to unstable learning, but it might work in a detailed model of a neuron, which includes dendrites. Then, the learning would occur locally at the synapse itself, with a lessened influence of other synaptic currents. We go around this problem by requiring the learning to occur over the duration of the postsynaptic current, and not the postsynaptic potential. This shortens the time interval over which a recurrent weight is being updated and allows a neuron to update its incoming weights in a more stable manner.

5.2.2 | The Lasso regularized synaptic rule with synaptic delays and dynamics

We assume that neurons can minimize their voltage fluctuations and gradually remove connections only during the postsynaptic current, which is induced by the arrival of a presynaptic spike after some transmission delay. The synaptic learning rule for recurrent connections then becomes:

$$\dot{\Omega}_{ij}(t) = -\varepsilon h_j(t) \left\{ \beta\big(V_i(t) + \mu r_i(t)\big) + \kappa \operatorname{sgn}(\Omega_{ij}(t)) \right\}, \text{ for } j \neq i, \tag{5.13}$$

where ε is a small learning rate. In contrast to learning with idealized Dirac delta synapses in Chapters 3 and 4, the learning with more realistic synapses occurs over the course of the PSC kernel, $h_j(t)$.

The learning rule for autapses remains the same as the one that was derived in Chapter 3:

$$\dot{\Omega}_{ii}(t) = -\varepsilon o_i(t) \left\{ \beta(V_i(t) + \mu r_i(t)) + \Omega_{ii}(t) + \mu \right\}, \text{ for autapses.} \tag{5.14}$$

An autapse is updated at the time of the neuron i's spike, when $o_i(t)$ is given by the Dirac delta function.

The steady-state of the synaptic learning rule

The implicit steady state of the recurrent weights is given by:

$$\Omega_{ij}(t) = -\big\langle h_j(t)\big\{\beta\big(V_i(t) + \mu r_i(t)\big) + \kappa \operatorname{sgn}(\Omega_{ij}(t))\big\}\big\rangle_t. \tag{5.15}$$

The recurrent weight sent by a presynaptic neuron j is the negative of the sum of two terms: the average state of the postsynaptic neuron i (measured by its voltage and how recently it spiked), and the average sign of the recurrent weight. The averages are weighted by the PSC kernel, $h_j(t)$.

5.3 | Numerical simulations of the learning rule

Algorithm

We evaluated the analytic solution for the membrane potential, (5.7), at consecutive time steps. The magnitude of the input variable, $x_k(t)$, for $k = \{1, ..., K\}$, is generally kept fixed at gT, for some gain $g > 1$ and threshold T. The only exception is at the beginning of all simulations, where its magnitude increases gradually from a sub-threshold value to an above-threshold value. When the neural voltage(s) reach the firing threshold, they fire a spike with a firing probability, which depends on the connection probability level. The firing probability is measured per discretization time step (1 ms). So, a firing probability of 0.5, means that there is a 50% chance that a neuron will fire a spike in this time step. This probability is the same in the following time steps, as long as the voltage is still above the threshold. For neurons that spike, the dynamics of $\tilde{r}_j(t)$, given by (5.4), is updated by adding the synaptic kernel, $h(t)$. Similarly, the dynamics of the filtered spike, $r_j(t)$, which is needed for the self-reset, is updated by adding one. The neuron enters a short refractory period after its spike. The autapse is updated at the time of the spike, as prescribed by the self-reset dynamics, (5.14). The recurrent weights are updated at every time step for which the evoked postsynaptic current is non-zero, as prescribed by the learning rule, (5.13).

The dynamical equations are solved with the forward Euler discretization scheme, in Matlab. The program is compiled into C, for faster run time. The learning algorithm is run until the synaptic weights converge to a final state. The performance of the trained network is assessed by running the network with a new input, recording the instantaneous filtered spikes, and using the least squares regression to fit a linear decoder. The network is then run with a test input, and the network reconstruction, $\hat{x}_k(t)$, is linearly decoded from filtered spikes, as given by (5.8).

Simulation details

The learning rule was simulated in networks consisting of $N = 60$ neurons, which receive $K = 2$ feedforward input currents. The feedforward encoding vectors, $\mathbf{F}_i$, were picked as random vectors from the K-dimensional input space, and normalized to have length one. The firing thresholds of all neurons were set to $T = 0.55$, which is slightly above one half of the encoder length. The constant magnitude of the input variable, $\mathbf{x}(t)$,

was gT, with $g = 5$. The initial values of the recurrent weights were set to equal zero. We also tried small initial values (small compared to the threshold), and this did not change the results. The self-connections were initialized as $\Omega_{ii} = -2T$. The refractory period used was 1 ms, and the firing probability was in the range $[0.2, 1]$. The membrane potential was not allowed to go below a minimum value, given by $-T$.

The discretization time step was always $dt = 10^{-4}$, although we tried smaller time steps, and this did not change the results. The leak constant was always, $\lambda = 50s^{-1}$, which corresponds to the characteristic time constant of 20ms. Thus 10 time steps correspond to 1ms. The synaptic learning rate was set at $\varepsilon = 10^{-4}$. For the learning rule, we used the quadratic cost, $\mu = 0.3$, and the scaling $\beta = 1.8$. We could have removed this scaling, but that would have changed the scaling of the sparseness parameter κ. Thus, we kept it for ease of comparison with the results from Chapter 4. The sparseness parameter κ took on values in the range $[0, \beta T] = [0, 1]$.

To define the double-exponential synaptic kernel, $h(t)$, given by (5.2), we used: the transmission delay, $\tau_{tr} = 1$ ms, the characteristic rise time, $\tau_r = 0.09$ ms, and the characteristic decay time, $\tau_d = 1.5$ ms. These parameter values agree with the general characteristics of AMPA synapses (Gerstner and Kistler, 2002). The synaptic kernel was normalized, so that the discrete sum of its values at consecutive time steps equals one. Multiplying it by the recurrent weight then gives the total recurrent input current.

Stability of learning

Simulating learning in networks with synaptic delays and dynamics is more prone to instabilities. The network can become unstable, with high firing rates, and the recurrent weights can diverge to infinity. Two things that were critical to ensuring stable learning were having a refractory period and not allowing the voltage to be hyperpolarized below $-T$.

The instability is caused by too much recurrent excitation being sent to the subpopulation which is hyperpolarized by the feedforward input. This is not a problem at the beginning of learning, when the recurrent weights are still small. However, over the course of learning, the magnitude of a recurrent weight starts to scale with the magnitude of the postsynaptic voltage. Hyperpolarized neurons will have large negative voltages, bounded from below by -gT. The depolarized neurons will have much smaller voltages, bounded by T. Thus, the learned positive connections will be much larger than the negative ones, and will destabilize the network.

A necessary ingredient for resolving this is to require the lower and upper voltage bounds to be equal. This requirement also ensures that the rest state continues to be meaningfully linked to biology. We recall our convention by which the voltage is measured in reference to the membrane rest state (rescaled to equal zero). The rest state should be approximately midway between the firing threshold and the hyperpolarized level, as is the case in real neurons. This would not be the case if the neural voltage was allowed to attain values as low as $-gT$, for any $g > 1$.

Furthermore, because of the nature of the learning rule, the final connection weights can still lead to an unstable network activity. This is caused by the variable population

spike count during the synaptic delay, which can in turn be linked to input characteristics (scale and derivative) and to the network input space coverage. The variable number of spikes during the synaptic delay means that, when this number is higher than the average, too many synaptic currents can be sent to the hyperpolarized neurons. The refractory period limits the duration of this instability, if it occurs. We also introduce probabilistic firing, which decreases the number of spikes that occur during the synaptic delay. This acts as a source of noise.

Finally, to limit potential instabilities, we used slow inputs and the same input scale for the learning of the recurrent connections and for the testing of network performance. Unlike the network with instantaneous connections, the network with synaptic delays works best for input scales for which it was trained. One way to ensure stability for a range of input scales is to increase the firing threshold, or to set it adaptively. The details of the influence of different input scales should be investigated in the future.

We also note that instabilities cannot be removed by not learning the positive weights. Our simulations indicate that the learning rule works only when both positive and negative weights are learned concurrently.

5.4 | Properties of trained sparse networks with synaptic dynamics

The learning rule can train stable networks, which represent input variables reliably. An example of the network reconstruction and its activity is given in Figure (5.2a), for a trained network with intermediate connection probability, 1 ms synaptic delays, and AMPA synaptic dynamics. The reconstruction error is small and the raster plot shows only a few error-increasing spikes (red). Voltage traces of a few example neurons show that the voltage is approximately in the $[-T, T]$ range, with bursts of spikes during synaptic delays (Fig. 5.2b). The few times when the neural voltage crosses the firing threshold but the neuron doesn't spike are a result of probabilistic firing.

The trained recurrent weights have the same qualitative structure as those that were trained in networks without synaptic delays, in Chapter 4. The symmetry of connections is preserved (Fig. 5.3a, top row), but the synaptic strengths are smaller than they are in networks without delays, given the same connection probability. This is a direct consequence of delays.

Synaptic strength scales inversely with the synaptic delay

Two similarly tuned neurons will sometimes fire spikes during the delay, thus adding a small positive update to their synaptic weight; and sometimes they won't, thus adding a negative update to their synaptic strength. Overall, the learned connection will be negative, as required for efficient computing.

Similarly, the weights are also smaller between neurons, which encoding vectors point in opposite directions. A neuron, which is not activated by the feedforward input, receives multiple spikes from the activated population during their synaptic delay. Thus, the outgoing weights from these neurons will be scaled down by the average number of spikes that

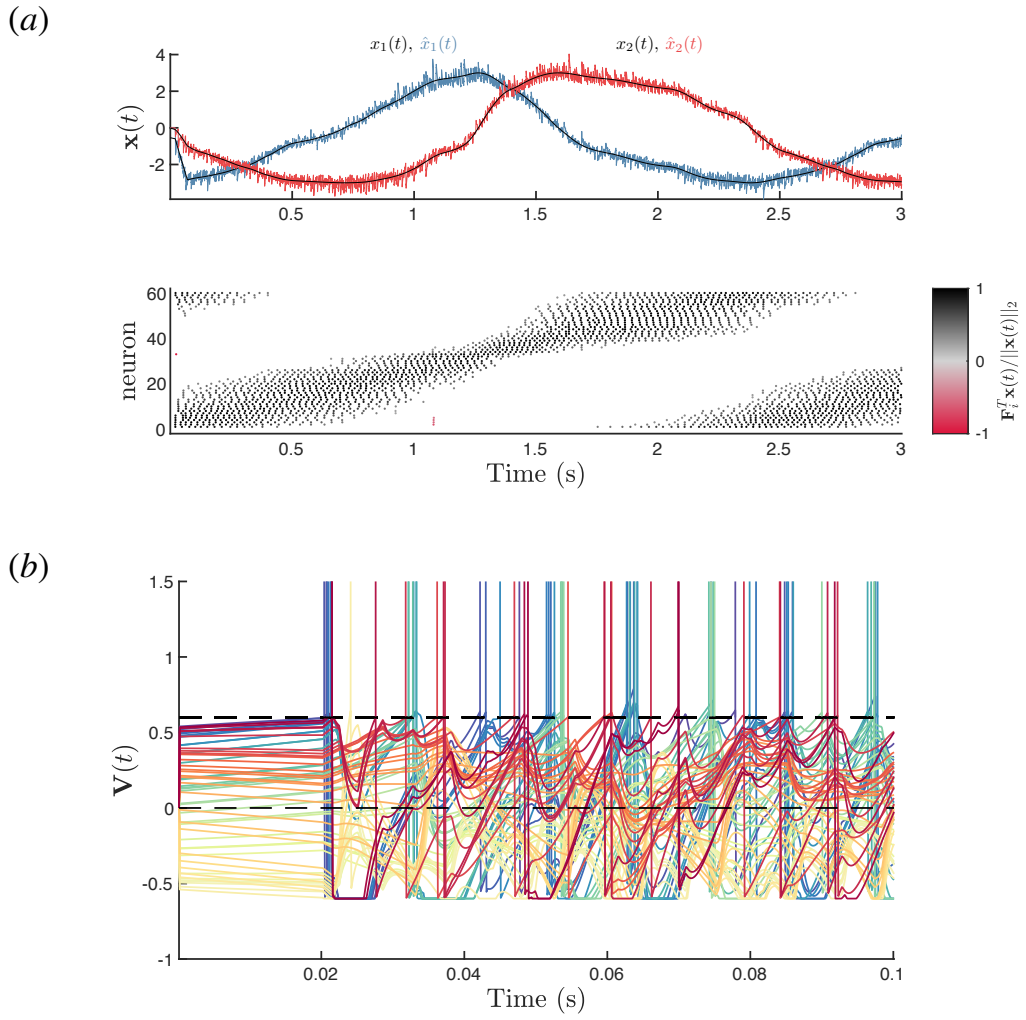


Figure (5.2) Input reconstruction and network activity of a trained sparse network with 1 ms delays and AMPA synaptic dynamics. The network has 60 neurons and was trained to represent 2D inputs. Its connection probability is, $p = 0.42$. **(a)** Top: Test input (black) and network reconstruction (blue, red). Bottom: Raster plot of the spiking activity. Only a few neurons fire spikes which increase the reconstruction error (red). **(b)** Voltage traces for a few example neurons during the first 0.1s of the test input. The voltage changes gradually from subthreshold values, and first spikes are fired around $t = 0.02s$, at which point a few similarly tuned neurons spike. There is a spurious (yellow) voltage discharge soon after, caused by too much recurrent excitation. Thereafter, error-decreasing spikes occur in bursts, because of synaptic delays. The voltage exceeds the threshold sometimes because the firing probability is less than one.

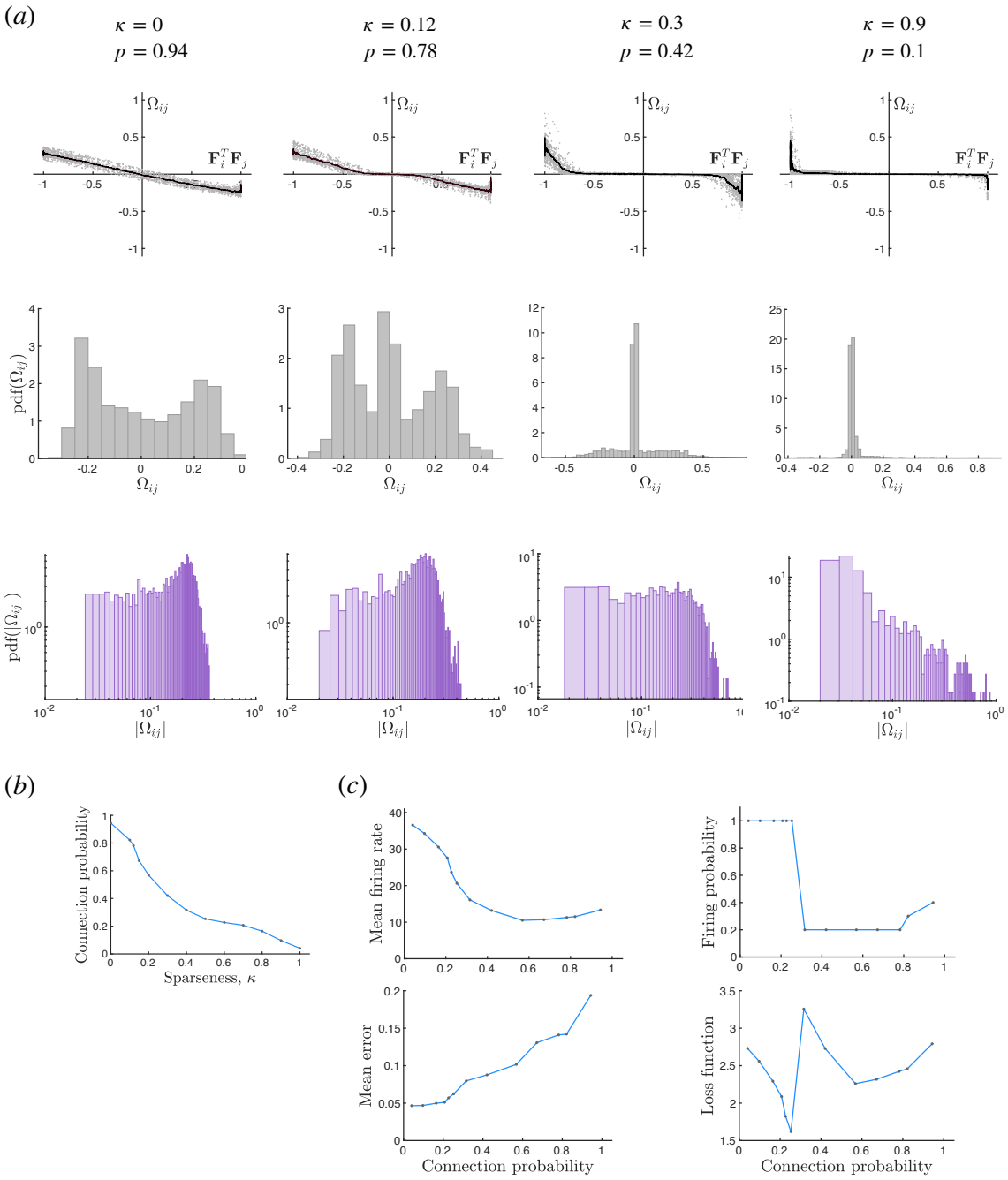


Figure (5.3) Structure and performance of trained sparse networks with synaptic delays and dynamics. The networks consist of 60 neurons, and are trained to represent inputs with 1 ms synaptic delays and AMPA synaptic dynamics. The inputs in (a) are 3D, and in (b-c) 2D. **(a)** Top: examples of synaptic connections, Ω_{ij}, as a function of the scalar product of neural encoding vectors, $\mathbf{F}_i^T \mathbf{F}_j$. As the sparseness parameter, κ, is increased, the final connection probability, p, goes toward zero. Middle: the distribution of synaptic weights for the four examples above. Bottom: The distributions of absolute weights on a log scale. **(b)** Final connection probability as a function of the sparseness parameter κ. **(c)** The performance of trained networks, as a function of connection probability: mean firing rate, mean relative error, firing probability, and loss function. The firing probability is picked as the largest probability that allowed a stable learning of the networks. It could have been higher if we had used a higher threshold. The loss function is calculated, as explained in Chapter 2.

they fire during the synaptic delay.

The longer the delay, the more spikes are fired by the population, and the smaller the learned synaptic weights (Fig. 5.4).

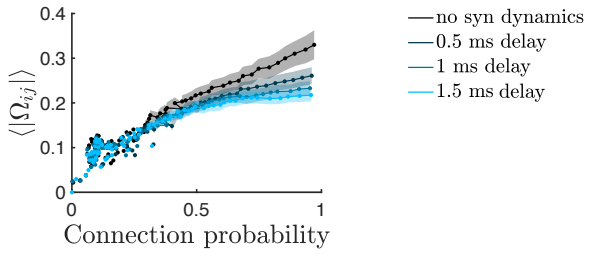


Figure (5.4) Recurrent weights are smaller for longer synaptic transmission delays. As the synaptic delay increases, the likelihood of a similarly tuned neuron firing a spike during the delay increases. This has the effect of decreasing the optimal strength of the recurrent connections.

Quadratic costs are important for stable connections

Two similarly tuned neurons should always learn negative reciprocal connections. However, if their encoding vectors are almost parallel, then they will almost always fire during the delay. After the delay, both of their voltages will be hyperpolarized, and thus they will learn a destabilizing positive connection. The quadratic cost term, $\mu r_i(t)$, in the learning rule ensures that the firing rate is taken into account, and that a stabilizing negative connection gets learned here. A higher quadratic cost was used to ensure this ($\mu = 0.3 \approx T/1.8$).

It is possible that this will not be so important in networks which receive more input variables. As we will see in Chapter 6, the scalar product of random unit vectors has a peak at the extremes (-1 and 1) in 2D, which corresponds to more strong recurrent connections. This peak moves from the extremes toward zero, as the dimensionality of the input grows.

Distribution of synaptic weights

The distribution of trained synaptic weights depends in general on the dimensionality of the input, as will be discussed in Chapter 6. However, we note here that the distribution is similar to that observed for networks without delays. One exception is the presence of two peaks for larger positive and larger negative weights (Fig. 5.3a, middle row). As mentioned in the paragraph above, this might be a consequence of how scalar products of random unit vectors are distributed in 2D. We also note that the synaptic weights do not follow the lognormal distribution, which is what is observed in cortical networks (Fig. 5.3a, bottom row, log scale).

5.5 | Network performance and optimal number of connections

Network performance

As in the case of networks with instantaneous connections, when the sparseness parameter, κ, takes on values in the range $[0, \beta T]$, the trained networks attain connection probabilities, p, in the range $[1, 0]$ (Fig. 5.3b).

The mean firing rate generally decreases with greater sparseness, while as the mean relative error increases (Fig. 5.3c). The fact that the mean error decreases is a signature of the stability of the network for all connection probabilities. If the increased firing rate measures an increased number of error-correcting spikes, as is the case here, then the optimal decoder has a smaller length, and the error decreases.

We also notice a small dip in the firing rate for intermediate connectivity levels. This is mediated by the firing rate probability, which we set a priori. We picked the largest firing probability which lead to stable learning. We could have trained networks with higher firing probabilities than the ones shown here, if we had set a larger firing threshold. The threshold that we picked was only slightly bigger than one half of the encoder length ($T = 0.55$), which meant that the precision of the network was high, but also that the network would be more prone to instabilities.

Optimal recurrent connectivity level

The dip in the firing rate for intermediate connection probabilities might mean that this level of connectivity is optimal in the presence of synaptic delays. Removing connections from a network can act to desynchronize the population spiking activity. This can improve the performance and was observed in earlier work with idealized optimal networks, discussed at the end of this chapter. However, we would need to differentiate between the influence of the variable firing probability and that of the changing connection probability.

The loss function indicates that the decreased firing rate is related to the connection probability. While as the firing probability is fixed at 0.2 for connection probability, p, between 0.3 and 0.8, the loss function looks like a u-shaped curved, with a minimum between $p = 0.5$ and $p = 0.6$. The exact optimal connection probability will depend on various network characteristics, and could be explored.

We also note that the loss function, for p smaller than 0.3, first jumps to its lowest value and then starts increasing. This is because it is influenced both by the regularity of population spiking (quadratic cost), the total number of spikes (linear cost), and the absolute error. In fact, if we control for the fact that the firing probability is one now, this looks like a continuation of an increasing u-shaped curve.

Synaptic delays do not matter for small connection probabilities

We can now compare the performance of trained networks with synaptic delays and dynamics, to those with instantaneous interactions, that we computed in the previous

chapter. The firing rate and error is generally better in networks with instantaneous interactions, but is about the same for connection probabilities smaller than 0.3 (Fig. 5.5). This means that for $p < 0.3$, trained networks with synaptic delays do not perform worse than those with instantaneous interactions. In fact, in this connectivity regime, the coding is driven more by the feedforward excitation, than by the recurrent inhibition. So, synaptic delays do not matter here.

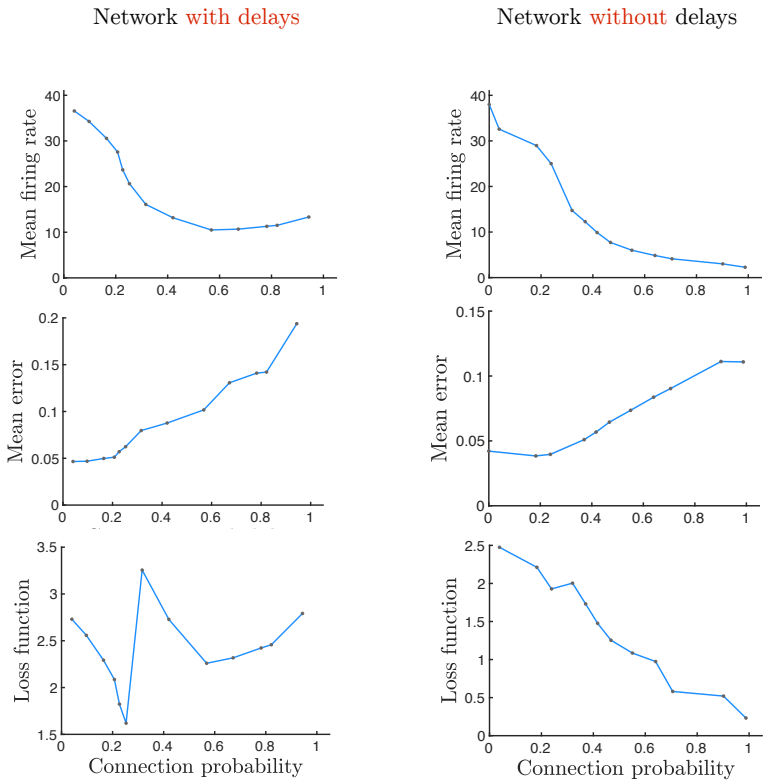


Figure (5.5) Performance comparison of sparse networks with and without synaptic delays and dynamics. Left: results from trained networks with synaptic delays and dynamics, from this chapter. Right: results from trained networks with instantaneous synaptic connections, from Chapter 4. The networks had 60 neurons in both cases and receive the same type of 2-dimensional inputs.

As mentioned already, we also see that the more realistic networks, which incorporate delays, show an intermediate optimal level of connectivity. In contrast, the optimal level of connectivity in networks with instantaneous connections is $p = 1$, and the performance gradually decreases as p goes to zero.

Might it be the case that sparse connections are not just a biological constraint, but a way that local circuits optimize signal representation in the presence of synaptic delays?

5.6 | Relation to previous work on coordinated spike coding with synaptic delays

Previous work addressed the question of performance in networks with fixed optimal recurrent connectivity and synaptic delays (Chalk et al., 2016). They considered a one-dimensional network, with recurrent connections set to equal the optimal connections for a network with instantaneous interactions. Adding synaptic delays caused a synchronization of the network, where similarly tuned neurons fired in unison in response to the feedforward input. In contrast to the optimal network, where instantaneous inhibition prevents all but one neuron from firing, the network with delayed inhibition cannot hyperpolarize similarly tuned neurons before they have fired their own spikes. For networks of fixed population firing rate, this synchronization leads to a worsening of the network's coding error. The authors proposed an optimal level of noise (e.g. membrane potential white noise), which desynchronizes the individual spike trains just enough to decrease the coding error to a minimum, given the presence of synaptic delays. Increasing the noise beyond this level desynchronizes the spiking activity too much, leading to an increase of the coding error. They find that there is an optimal level of spiking coordination, which allows efficient and precise signal representation in the presence of synaptic delays. This optimal level of spiking coordination is attained by modulating the level of noise.

We single out a few ideas from their work, which are of interest to our analysis of sparse networks with synaptic delays. The authors compare the coding error of an optimal network with synaptic delays to that of an independent network in which neurons fire with Poisson variability. The underlying firing rate of the independent (Poisson) network is determined by the feedforward input. The input is constant and one-dimensional in majority of the work, but time varying inputs are considered in a few instances. Since the systematic analysis and results are based on network simulations with constant inputs, we will focus on those.

When the input is constant, the optimal network produces regular population spikes, at an interval which depends on the membrane leak constant and the input gain. Individual neurons, on the other hand, fire irregularly, thanks to the instantaneous inhibition (Fig. 5.6a, left panel). With the introduction of synaptic delays, the irregularity of individual spike trains starts to break down and neurons start to synchronize their spike times. For a fixed population firing rate (which is obtained by varying the firing threshold), increased synchronization leads to an increasing network coding error (Fig. 5.6a, middle and right panels). In fact, there can be so much synchronization that the coding error exceeds that of an independent network, which fires asynchronously (comparison with the Poison network in Fig. 5.6a).

A proposed solution for the increasing coding error is adding an optimal level of noise, which desynchronizes neural membrane voltages just enough so that spike times are more asynchronous (Fig. 5.6b). Increasing the noise level above the optimal level would desynchronize the activity too much and increase the coding error. This is because the population spiking is not coordinated anymore, as the membrane voltages no longer encode the global coding error in the presence of the high level of noise.

This gives an intuition behind what happens in an optimal network when synaptic

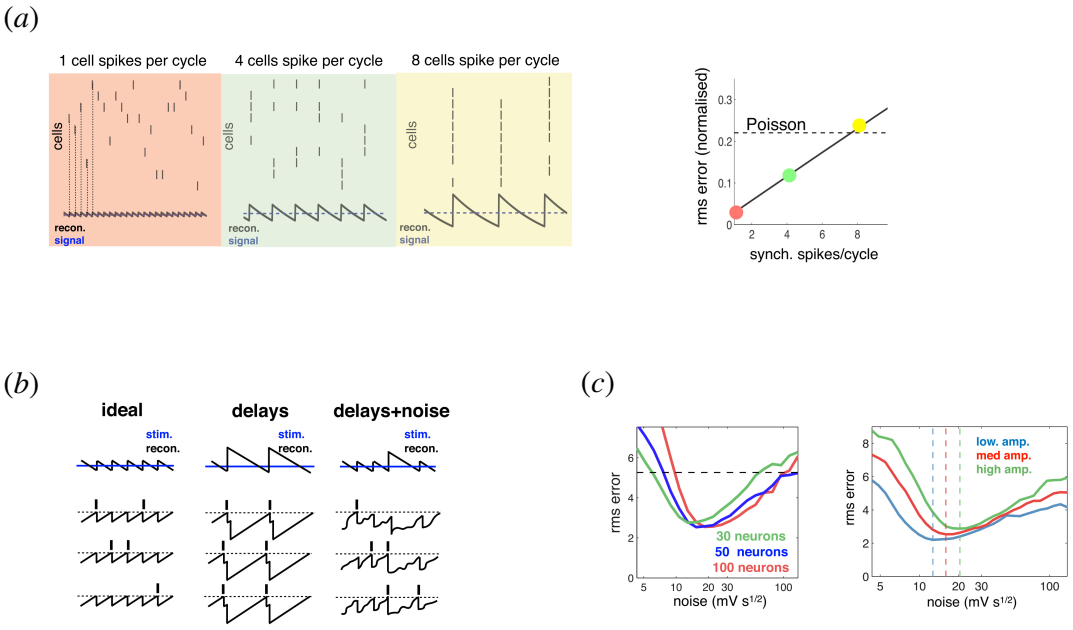


Figure (5.6) Impact of synaptic delays on the performance of the optimal network. Reproduced along with captions from Figures 1, 3, 9 and 10 of (Chalk et al., 2016). **(a)** Relationship between synchrony and coding accuracy. Left: Each panel illustrates the response of 10 neurons. An encoded sensory variable is denoted by a horizontal blue line. Each spike fired by the network increases the sensory reconstruction by a fixed amount, before it decays. Greatest accuracy is achieved when the population fires at regular intervals, but no two neurons fire together (left). Coding accuracy is reduced when multiple neurons fire together (middle and right panels). Right: Reconstruction error (root-mean squared error divided by the mean) for a regular spiking network shown in panel a, versus the number of synchronous spikes on each cycle. A horizontal line denotes the performance when neurons fire with independent Poisson variability. **(b)** Schematic, illustrating how delays affect the decoding performance of the network. **(c)** Left: Effect of varying population size. Root-mean-squared (rms) reconstruction error versus added membrane noise, for three population sizes. A horizontal line denotes the performance of a Poisson spiking network with matching population firing rate. Right: Effect of varying the input amplitude. Root-mean-square (rms) reconstruction error versus injected noise amplitude with three different input amplitudes. The optimal noise level in each condition is denoted by vertical dashed lines.

delays are added. However, this intuition relies on very specific assumptions, which are different from the ones we make in our work. First, the authors consider a network in which all neurons have the same encoding vectors, or more precisely the same encoding weights, as the network receives a single input variable. The decoding weights are set to equal the encoding weights. In our work, the encoding vectors are random and the decoding vectors are computed using linear least squares regression, so they are optimal. In the discussed work, all neurons receive exactly the same input, given the same encoding weights. Consequently, the number of concurrent spikes is determined by the strength of the input, in the absence of noise, and these spikes occur precisely at the same time. On the other hand, in our work, the number of concurrent spikes is close to zero, as there are almost never two neurons which reach the threshold at precisely the same time. However, there is a variable number of spikes which are fired during the synaptic transmission delay. The number is variable because the coverage of the input space is not precisely symmetric (although it is random). Thus, depending on the trajectory of the input, a varying number of neurons can reach the threshold before the inhibitory recurrent input is communicated. We compute an optimal decoder, and its length is roughly inversely proportional to the mean number of spikes which occur during the synaptic transmission delay. We suspect that an optimal decoder would have led to a better coding error in the discussed work too.

Secondly, the results from the cited work are based on networks with no opposing negative weights. Since the input is constant and positive, the necessary encoding weights cover only one half of the input space (the positive real line). We suspect that this makes the results better, as it is the opposite weights that can introduce significant coding inaccuracies. As already discussed, if too many spikes fire during a synaptic delay, the oppositely tuned neurons in a fine tuned optimal network (small firing costs) could receive enough recurrent excitation to fire their own spikes. This could be resolved with a high linear cost, which would increase the firing threshold. A better solution might be to have an adaptive threshold which is increased during this spurious spiking, and then adaptively reduced again when the input signal strength decreases. This might lead to a smaller coding error over all input strengths.

This study also found that greater network size and higher input amplitude require a higher level of noise for an optimal coding error (Fig. 5.6c). This should hold true in our network too, and could be investigated as part of future work. In our network, the higher the coverage of the input space, the higher the number of spikes fired during the synaptic delay. Given the not exactly symmetric covering of the input space by the encoding vectors, the variability around this mean number might increase with N. The reasoning is similar for high input gain. In our approach, noise is provided both by probabilistic firing, and by an intermediate connection probability, which desynchronizes the network activity.

Another work (Schwemmer et al., 2015) addresses realistic synaptic dynamics in the spike-coding framework. They start from the classical framework (Boerlin et al., 2013), and replace the instantaneous synapses with exponential "fast" synapses, which have finite temporal dynamics. Furthermore, they incorporate intrinsic biophysical currents, which are modeled as spike-generating Hodgkin-Huxley-type ionic currents. The spike generating currents add biological realism, but they also break the relationship between the subthreshold voltage and the coding error, on the time scale of a single spike. The authors remedy this problem by adding compensating synapses to the voltage and decoder dynam-

ics, which are fitted so that they restore the relationship between the neural voltage and the projected coding error. Although the biophysical model is derived for a network which can perform arbitrary linear computations, the main results in this paper are obtained based on a network of 400 neurons, which performs perfect integration (without any leak) of its one-dimensional input. The network is homogeneous, meaning that the magnitude of the encoding weights is the same for all neurons, with an equal number of positive and negative weights in the population.

In this work, the authors add white noise current to the voltage equations, to remedy synchronization due to synaptic delays. They find that there is an optimal level of noise for a given synaptic input scale, for which the coding error and population synchrony are the smallest. They also consider the influence of network size, while ensuring a fixed total synaptic input (synaptic connections scale with $1/N$). They find that the coding error, firing rate, and level of synchrony, all decrease with increasing network size. However, it is not clear whether there was an optimal level of noise in the later simulations.

5.7 | Conclusion

Adding synaptic delays to coordinated spike coding networks leads to a synchronization of network activity and poorer network performance. Depending on input statistics and network characteristics, the network activity could become unstable and no longer be able to represent inputs. At best, the network is able to encode an input in its activity, but it does so with much smaller precision than an idealized network without delays. Previous work found that adding an optimal level of synaptic noise desynchronizes the network, and improves its performance. No learning in networks with synaptic delays has previously been done, and no optimal recurrent connectivity has been proposed. Furthermore, no previous work addressed coordinated spike coding networks with both sparse connectivity and synaptic delays.

Here we show that optimal recurrent connectivity in the presence of synaptic delays and dynamics can be learned with a biologically plausible synaptic learning rule. We found the conditions under which both the learning process and the activity of the resulting trained network are stable. Learning the network structure improves the decoding accuracy, and allows the network to adapt itself to the presence of delays and sparser connectivity.

We find that the optimal synaptic strengths scale inversely with the length of the synaptic delay. Since synaptic delays scale with the distance between neurons, the synaptic delay between the most similarly tuned neurons might be smaller, thus increasing the overall performance of the network. There also seems to be an optimal connectivity level, for which the performance of the network is the best. This is contrasted by the idealized network with instantaneous synapses, for which the fully connected network is the most performant. With the added biological constraint of more realistic synapses, the optimal network connectivity is sparse. The smaller recurrent connection probability acts like a source of noise, which desynchronizes the network and optimizes its performance.

6 | Performance Trade-off in High Dimensional Sparse Networks

Biological neural networks represent high-dimensional sensory stimuli in their complex neural firing activities. However, the majority of recorded neural activities from early sensory cortices are available from experiments, which used a low number of stimuli.

The recorded neural responses tend to be low-dimensional (Archer et al., 2014; Chapin and Nicolelis, 1999; Bathellier et al., 2008; Gao and Ganguli, 2015). This could be a consequence of the low complexity of the input stimulus, as the dimensionality of the population response cannot exceed that of the stimulus (Gao et al., 2017). In fact, recordings from a large number of neurons in mouse V1, found the neural code to be high-dimensional in response to high dimensional natural image stimuli (Stringer et al., 2019).

Throughout this thesis we developed synaptic rules, which allow sparse networks to learn efficient representations of low-dimensional signals. We dedicate this chapter to the examination of sparse spike coding networks, which represent high-dimensional inputs. The learning rules extend naturally to this biologically motivated scenario. We will begin this chapter by training sparse networks with increasing numbers of feedforward inputs. As we will see, increasing the input dimensionality decreases the trade-off between firing and wiring, for fixed connection probability and network size.

However, when we allow the network size to change, the trade-off becomes more complex. In order to facilitate a complete understanding of the impact of network size and other network variables, we will take the following approach. Rather than training sparse networks of various connection probabilities, we will compare two edge cases: a purely feedforward network with no connections and a fully connected coordinated spike-coding network. Understanding the impact of network variables on the efficiency of these edge cases, can then be extended to intermediate sparsely connected networks, without the added computational costs incurred by a large number of training simulations.

6.1 | Training high-dimensional sparse networks

We begin this chapter by training spike coding networks to represent high-dimensional inputs, with the constraint of sparse connectivity. We consider networks with instantaneous synapses, with the simulation details described in Chapter 4. In order to be able to compare the reconstruction error and firing rate of networks with varying numbers of

input dimensions, K, we ensure that the input gain is constant for all K.

Defining the high-dimensional input variable

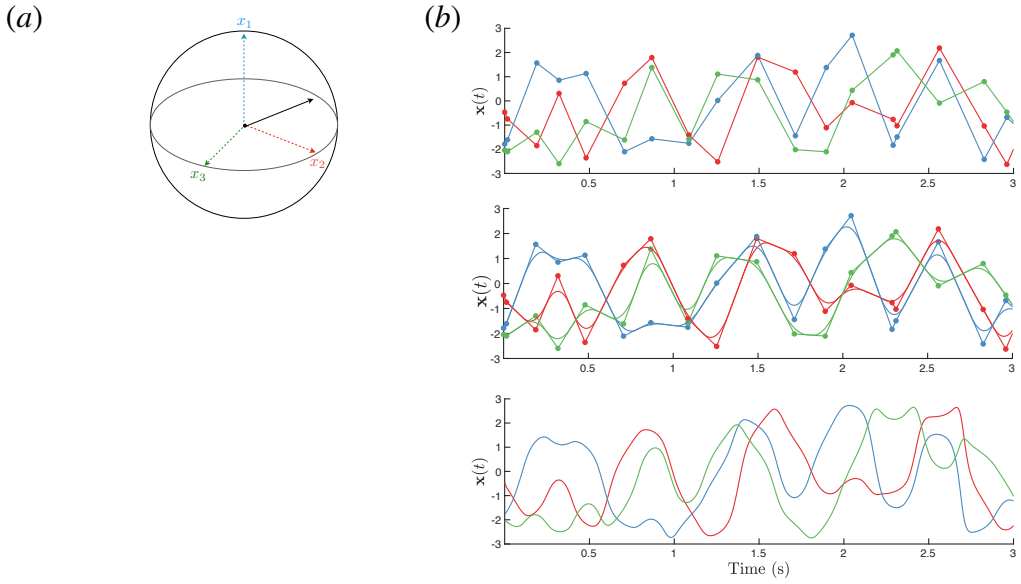


Figure (6.1) Defining multi-dimensional inputs with constant gain and radial speed. Illustration for a $K = 3$ dimensional input. **(a)** The input is created by first picking random K-dimensional vectors, cantered at the origin (single example vector shown). All vectors are rescaled, so that their magnitude equals gT. Here, g is the desired gain of the input, and T, the neural firing threshold. All vectors have their end points on the K-dimensional sphere, of radius gT. **(b)** Three steps of input transformation, until the final version is obtained. Top: Given a sequence of K-dimensional vectors, difference vectors are calculated between consecutive vectors. The input, $\mathbf{x}(t)$, is then defined to have its initial value equal to the first random vector. Its evolution in time is given by the piece-wise linear trace of the consecutive difference vectors. The vectors are followed at a fixed radial speed. The three curves show the three inputs, with the dots showing the coordinates of the random vectors. Middle: The input is then transformed so that the corners are smooth. Bottom: Finally, the magnitude of the smooth input is rescaled to ensure that it equals gT.

The training input is created in four steps. First, we pick random K-dimensional vectors, $\mathbf{F}_i^K$, of length one, centered at the origin. This is the same way that we generate the encoding vectors, $\mathbf{F}_i$, for $i = \{1, 2, ..., N\}$; only the number of random vectors that we use for training is much larger than N. Given the similarity between the random vectors we use to define the training input, and the encoding vectors, we use the superscript K to talk specifically about the former. Figure (6.1a) shows an example of a random vector for $K = 3$. Before using these random vectors to create the training input, we change their magnitude to equal the gain times the threshold, gT. We then define the training input $\mathbf{x}(t)$ to equal $\mathbf{F}_i^K$, for specific time instances, t, indicated with circles in the top panel of Fig. (6.1b). The input is made continuous by drawing straight lines between the consecutive random vectors, that is by following the difference vectors, $\mathbf{F}_{i+1}^K - \mathbf{F}_i^K$.

The speed of moving along these difference vectors is constant, so that two vectors which happen to be further apart will lead to a training input which takes longer to traverse the path between them. We then smooth the training input, and then rescale the input so that its magnitude always equals gT, Fig. (6.1) (b), middle and bottom panels. This gives us the final training input which traverses the K-dimensional sphere of radius gT.

Next, we must consider how many neurons should be in each network as the dimensionality of the input grows. Simulations of networks which receive 2D and 6D inputs show that training networks with a larger number of neurons leads to a smaller final connection probability, Fig. (6.2), panels (b) and (e). This agrees with the parameter study in Chapter 4, where we studied the influence of the network size N on the final connection probability. There we considered 2D inputs only, but the relationship holds in higher dimensions too.

We also note that the population rates in the larger networks grow faster than in the smaller networks, once we enter the sparser regime, Fig. (6.2) (c) and (f). This is not entirely compensated for by the smaller reconstruction error, and consequently, the loss function of larger networks starts to outweigh that of the smaller ones. This happens around $p = 0.3$ in 2D, and $p = 0.1$ in 6D.

The sparseness parameter, κ, used to train the example 6D network was smaller than that used to train the 2D network. Yet, the connection probability of the 6D network is smaller. Looking at the histograms of the products of encoding vectors in 2D and 6D gives a hint as to why we need larger κ in higher dimensions to achieve the same level of connection probability as with lower dimensional inputs. The products of encoding vectors in higher dimensions are more concentrated around zero, and the learning rule will thus prune more of the recurrent connections which are associated with the orthogonal encoding vectors.

The final sparse synaptic connections have the same characteristic dependence on the product of encoding and decoding vectors in 2D and 6D, Figure (6.3).

6.2 | Performance of high-dimensional networks

We would like to systematically compare networks which receive inputs of different dimensionality, K. So a good place to start is to fix the network size, and vary K. We can consider later whether there is an optimal N for each K, and use that network size for the performance comparison.

We fix the network size at $N = 60$ neurons, and vary the number of input dimensions (Fig. 6.4a). We consider inputs of dimensionality, $K = \{2, 3, 6, 9, 12\}$, and use them to train networks of varying connection probabilities (Fig. 6.4b). We will refer to the networks which receive K-dimensional inputs, as K-dimensional networks (not to be confused with the dimensionality of the activity of the network).

We make two observations: for a fixed connection probability, p, higher dimensional networks require smaller values of the sparseness parameter, κ, and the mean value of their recurrent connections is smaller than for lower dimensional networks. We also consider the performance of the networks (Fig. 6.4c), and make following observations. All networks have the same firing rate when p is close to one. As expected, their firing rate

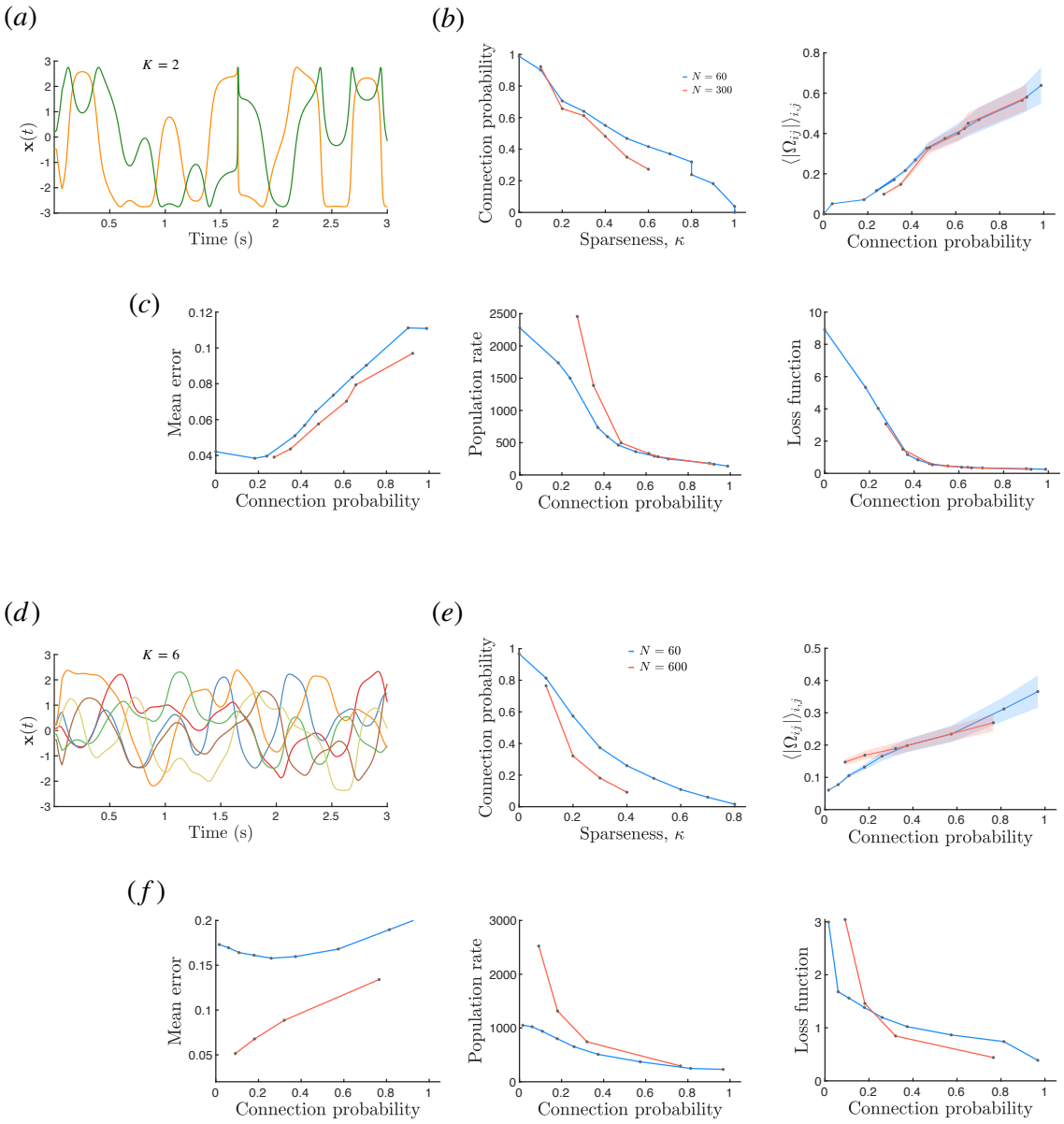


Figure (6.2) Training higher-dimensional sparse networks of different network sizes. (a-c) show the results from 2-D networks of two different network sizes. (d-f) show the results from 6-D networks. **(a)** Example 2-D test input (3 s of total 100 s shown). **(b)** Left: The final connection probability as a function of the sparseness parameter, κ, for networks of different sizes: $N = 60$ neurons (blue) and $N = 300$ neurons (red). Right: Mean and standard deviation of synaptic connections which have not been pruned away, for networks of different connection probabilities. **(c)** Performance of the 2D networks, as a function of connection probability. Left: Mean reconstruction error of the test input given in (a). Middle: Population firing rate. Right: Loss function. **(d-f)** same as (a-c) for 6D networks of two different sizes: $N = 60$ (blue), and $N = 600$ (red).

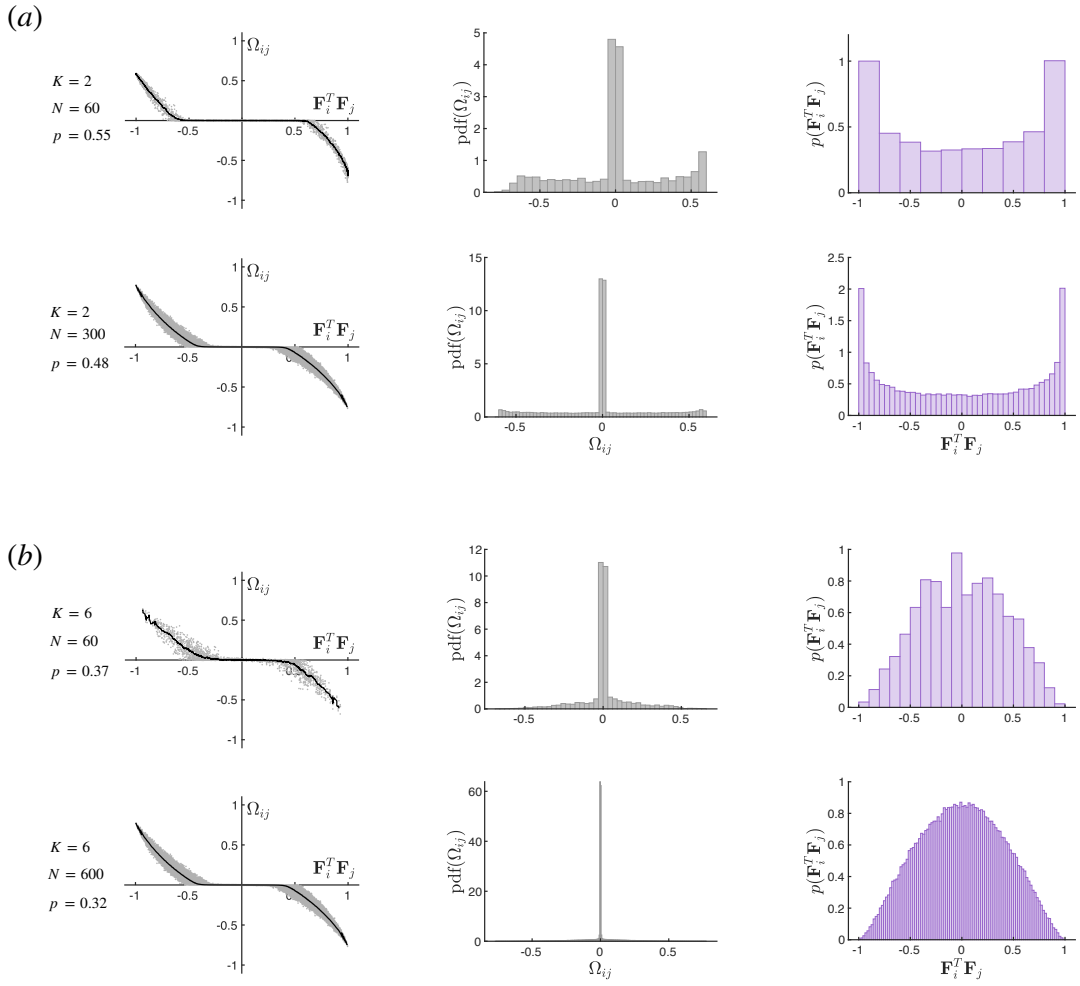


Figure (6.3) Examples of the dependence of final recurrent connectivity on the network size. **(a)** Trained synaptic connections in a network of 60 (top) and 300 (bottom) neurons, with $\kappa = 0.4$. The input is 2D. Left: The trained synaptic values as a function of tuning similarity between two neurons. Middle: The histogram of trained synaptic connections. Right: The histogram of the scalar products of encoding vectors. **(b)** Same as (a) but for a 6D input and a network of 60 (top) and 600 (bottom) neurons, with $\kappa = 0.3$.

grows with decreasing p. However, this growth is slower for higher dimensional networks. The 12-dimensional network fires the least number of spikes for p close to zero. Another performance metric, the reconstruction error, has the opposite behaviour. It is larger for higher dimensional networks.

A true assessment of the performance has to consider both the firing rate and the error. So we calculate the performance as the inverse of the average population firing rate multiplied by the average relative error:

$$\text{Performance} = \frac{1}{\langle \text{Pop. firing rate} \rangle_t \langle \text{Error} \rangle_t}. \tag{6.1}$$

Here, the reconstruction error is measured relative to the signal strength, as is the case throughout this thesis.

For high connection probability, performance of higher dimensional networks is worse than that of lower dimensional ones. This is not surprising given the fixed network size. However, as we approach the non-connected regime, all networks have the same level of performance. The low-dimensional networks have enough neurons to cover the input space well. This means that in the non-connected regime, all neurons fire a lot more spikes and the reconstruction error is small. The high dimensional networks on the other hand do not have enough neurons to cover the much larger input space. In the non-connected regime, they fire fewer spikes than the low-dimensional ones, and the error is much larger. However, over all, the performance is the same.

We also calculate the loss function, as the mean absolute error plus the μ weighted mean L_2 norm of the firing rate:

$$\text{Loss} = \left\langle \left\| \mathbf{x}(t) - \hat{\mathbf{x}}(t) \right\|_2^2 \right\rangle_t + \mu \left\langle \left\| \mathbf{r}(t) \right\|_2^2 \right\rangle_t. \tag{6.2}$$

As expected, the loss function increases with decreasing p for all K, and its growth is higher for low dimensional networks. For high p, the higher dimensional networks have a higher loss function, but its growth with decreasing p is slower than for low-dimensional networks. And so around $p \in (0.2, 0.3)$, the loss of the higher dimensional networks becomes smaller. This is probably a consequence of more regular spiking of the lower dimensional networks in this regime (CV_2 of the population inter-spike intervals in Fig. 6.4c).

6.3 | Orthogonality of high-dimensional spaces

We now return to two of our earlier observations, which contrast higher and lower dimensional networks. Firstly, higher dimensional networks require smaller values of the sparseness parameter, κ, to attain a set connection probability, p. Secondly, the mean recurrent strength for a fixed p is smaller for high-dimensional than for low-dimensional networks. These two properties are related, and both are a consequence of the nature of higher dimensional spaces.

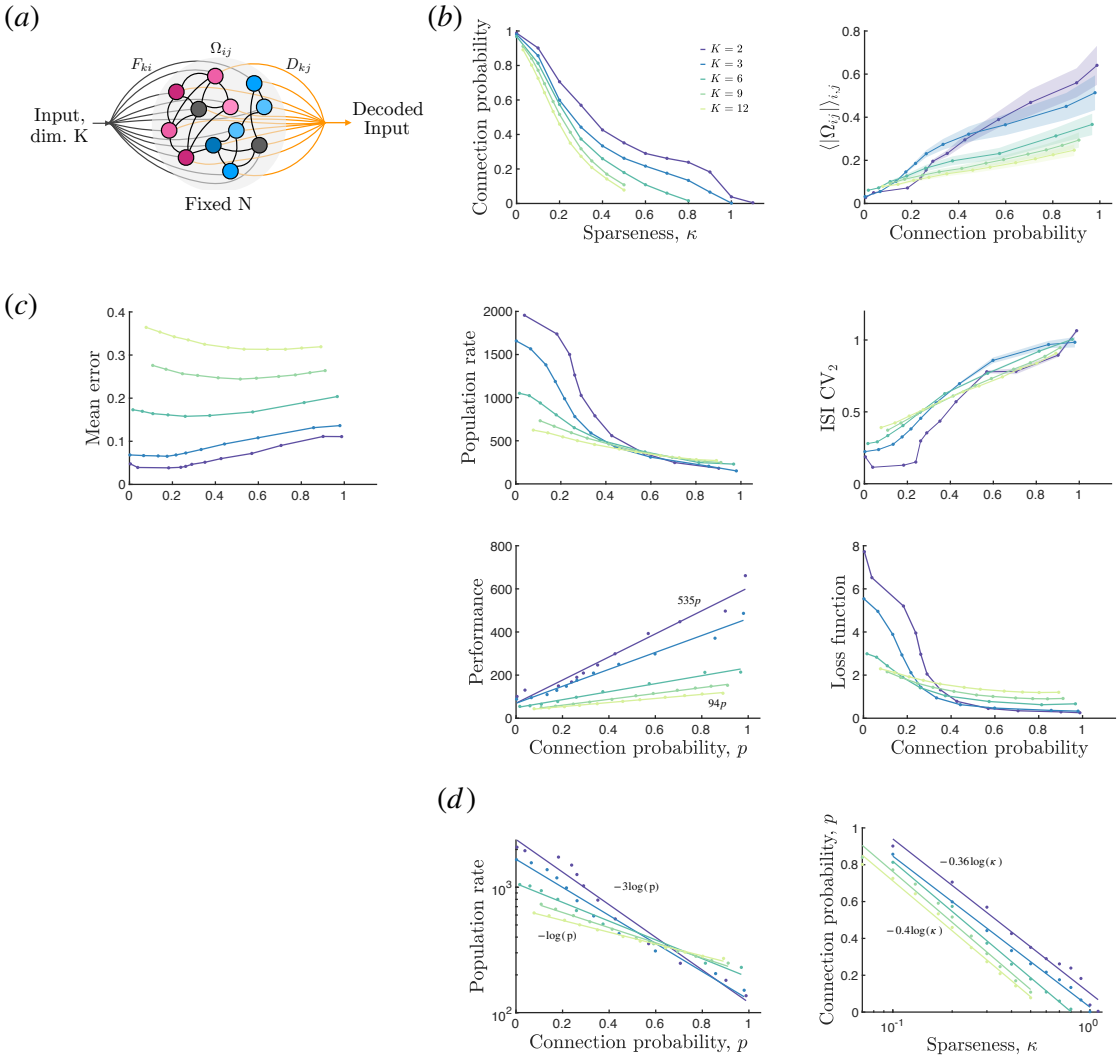


Figure (6.4) Performance with increasing input dimensionality for fixed size networks. **(a)** The size of all networks is fixed to equal 60 neurons. We pick different dimensionalities, K, of the input signal, and train networks of various connection probabilities. The considered inputs have dimensionality: $K = \{2, 3, 6, 9, 12\}$. **(b)** Left: Final connection probability as a function of the sparseness parameter, κ, for K-dimensional inputs. Right: Mean and standard deviation of synaptic connections which have not been pruned away, as a function of connection probability. All networks were trained with $\mu = 0.3$ for all κ values, except for $\kappa = 0.1$, where $\mu = 0.1$. The large quadratic cost in the learning rule leads to a more balanced pruning of excitatory and inhibitory connections (smaller μ would have favoured the pruning of inhibitory over excitatory connections). **(c)** All the performance metrics were calculated based on the activity of a network in response to a 100 s long K-dimensional test input, of fixed gain and radial angle, for all K. The mean error is calculated relative to input strength. The performance metric is calculated as the inverse of the product of the mean error and population firing rate. The loss function is calculated as $\left\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right\|_2^2 + 0.1\left\|\mathbf{r}(t)\right\|_2^2$, as the quadratic cost used to define the threshold equals 0.1. **(d)** Left: The mean firing rate as a function of connection probability (log-log scale). Right: Final connection probability as a function of the sparseness parameter, κ, for K-dimensional inputs (log-log scale).

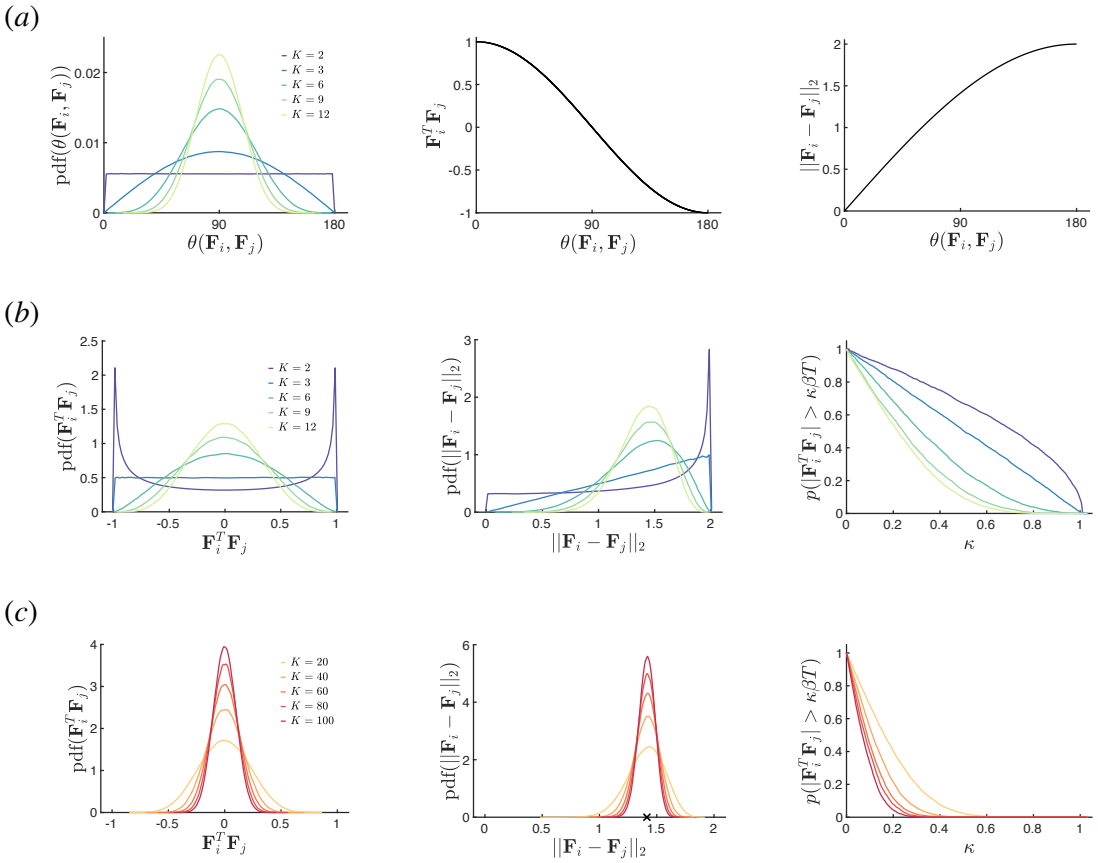


Figure (6.5) The relationship between the encoding vectors and input dimensionality. **(a)** Left: The angle between two random unit-length encoding vectors has different probability densities with increasing input space dimensionality, K. As K increases, the angle starts to follow a Gaussian distribution. Middle: The scalar product of two unit vectors as a function of their angle. Right: The distance between two unit vectors as a function of their angle. **(b)** Left: The scalar product of two random unit-length encoding vectors has different probability densities with increasing input space dimensionality, K. As K increases, the scalar product starts to follow a Gaussian distribution. Middle: The length of the distance between two unit-length encoding vectors is always in the $[0, 2]$ range. But the shape of the distribution changes with increasing K. Right: The probability of the scalar product of encoding vectors being larger than $\kappa\beta T$ for different values of κ. For a fixed value of κ, the higher dimensional encoding vectors are increasingly less likely to have their scalar product be larger than $\kappa\beta T$. **(c)** Same as (b), but for larger values of K.

To see how, we consider K-dimensional random vectors, which emanate from the origin and end on the surface of a K-dimensional sphere. These vectors can be thought of as the encoding vectors, $\mathbf{F}_i$, of unit length. A property of high-dimensional spheres is that two encoding vectors are most likely to be orthogonal to each other (Fig. 6.5a, left panel). This is contrasted by the fact that the angle between two random vectors in 2D is equally likely to take on any possible value.

Consequently, the distribution of scalar products, $\mathbf{F}_i^T\mathbf{F}_j$, is also a function of K (Fig. 6.5b, left panel). In 2D, the distribution has peaks at the extremes, namely at -1 and 1. The peaks at the edges are absent in the distribution of the angles, and show up because the scalar product is proportional to the cosine of the angle between the vectors (Fig. 6.5a, middle panel). For the same reason, the scalar product in 3D is uniformly distributed in the $[-1, 1]$ range, and not bell-shaped like the angles between the vectors. For $K > 3$, the distribution starts to get a bell-curved shape, with a peak at zero.

As K grows, it is ever more likely that two random vectors are going to have a zero scalar product. That also means that the majority of recurrent connections in a fully connected optimal network are going to be close to zero. We recall that the optimal recurrent connections between two neurons are proportional to the negative scalar product of their encoding vectors.

In contrast, when $K = 2$, the distribution of the scalar products (Fig. 6.5b, left panel), and thus of the optimal weights, has peaks at the extremes. There are more strong than weak connections, and it is the strong inhibitory connections that contribute the most to the efficient representation in optimal networks. As K grows, the distribution of optimal weights changes, and the majority of the weights are weak, with the strong connections (± 1) almost being non-existent.

Because of this, we hypothesize that trained sparse networks do not perform much worse than fully connected ones, when K is large. We would expect the performance to decay more slowly with decreasing connection probability, p, because the all-to-all optimal connections are already highly concentrated around zero. This agrees with the simulation results with a fixed number of neurons and varying K (Fig. 6.4c), where the performance does fall off more slowly with decreasing p, as K gets larger. To make a definitive conclusion it is necessary to make sure that the coding regime does not change as we increase K. Fixed network size means that as K gets larger, the number of encoding vectors is too small to cover the space well, and so the error is quite large. We will address the issue of obtaining equal coverage with changing K in the following section, where we study the influence of network size on the performance trade-off.

Just as the distribution of the scalar products of encoding vectors changes with K, so does their pairwise distance (Fig. 6.5b, middle panel). Since we consider unit hyperspheres, the length of the difference vector between any two encoding vectors, $\mathbf{F}_i$ and $\mathbf{F}_j$, has to be contained in the $[0, 2]$ range. Because of the relationship between the distance between two vectors, and their angle (Fig. 6.5a, right panel), there are many oppositely directed vectors (distance 2), in 2D. This holds true in 3D as well, although the peak is smaller. As the input space becomes higher-dimensional, the peak starts to move closer to an intermediate value, close to $\sqrt{2}$. This is equivalent to the fact that the product of random encoding vectors starts to have a peak at zero, as K grows. In other words, more

and more vectors are orthogonal to each other.

Finally, the analysis of encoding vectors in higher dimensions allows us to understand the two observations made at the beginning of this section: higher dimensional networks require smaller values of the sparseness parameter, κ, to attain a set connection probability, p; and, the mean recurrent strength for a fixed p is smaller than in low-dimensional networks.

Regarding the first observation, we recall the importance of the value $\kappa\beta T$ in the training of sparse networks: changing κ from zero to one allowed us to train networks of increasingly sparser connectivity. How high κ needs to be to train a non-connected network depends on the distribution of the scalar product of random encoding vectors. As K grows, the likelihood that the magnitude of this product is greater than $\kappa\beta T$, for fixed κ, becomes smaller (Fig. 6.5b, right panel).

In fact, the probability, $p(|\mathbf{F}_i^T\mathbf{F}_j| > \kappa\beta T)$, is a sort of an upper bound on the final connection probability, p. Comparing this analytic bound on p as a function of κ, with the final connection probabilities of trained sparse networks (Fig. 6.4b), shows a close resemblance.

The second observation is related to the first, and is also a consequence of the changing distribution of $\mathbf{F}_i^T\mathbf{F}_j$. As K grows, the scalar products become smaller and so does the mean of the recurrent strengths for a fixed p.

We also consider much higher dimensional spaces, for which we have not trained sparse networks yet (Fig. 6.5c). As K continues to grow, the product of random encoding vectors has a normal distribution, centered at zero, with a decreasing variance. Similarly, the distance between vectors has a peak at $\sqrt{2}$, corresponding to orthogonal vectors (cross in Fig. 6.5c, middle panel). Lastly, the κ needed to train non-connected networks continues to get smaller.

6.4 | Network size and the performance trade-off

Given that the dimensionality of the input influences the distribution of optimal weights, we would like to know whether it also impacts the trade-off between performance and connection probability. In higher-dimensional networks, optimality does not require strong connections, and so we hypothesize that removing synaptic connections will have a smaller impact on the performance of sparser networks. This is precisely what we observed in numerical simulations of fixed size networks (Fig. 6.4). However, our observations were based on fixed size networks, and we need to consider the influence of the network size, N.

The numerical simulations comparing 2- to 12-dimensional fixed size networks (Fig. 6.4) were obtained with $N = 60$ neurons. This number of neurons is large enough to track a 2D signal with a $0.5 - 1\%$ relative error, but too small to track a 12 dimensional signal accurately (the error was around 0.3-0.4%). This means that the networks might be in different performance regimes: the 2D network is precise and fires more spikes than the 12D one. So, if possible, we should also study the performance of high-dimensional networks in the precise regime.

One way of ensuring precise coding is to increase the network size so that there are

enough neurons to cover the input space well. However, since training high-dimensional networks with many neurons is computationally expensive, we will first consider what happens to the performance of the optimal fully-connected network as we increase the network size. We will then compare it to the performance of the non-connected network. Knowing the performance at the extremes of connection probability will give us the lower and upper bounds on the performance of trained sparse networks.

6.4.1 | Varying network size in optimal networks

We begin by considering the influence of the network size on the performance of optimal fully-connected networks (Fig. 6.6a). We see that for fixed K, adding more neurons to the optimal network leads to a smaller mean error and a smaller population firing rate. The later can be seen in the firing rate panel, where we see that the neuronal firing rate decays faster than $1/N$. Decreasing error and population rate are not surprising, as additional neurons lead to a better coverage of the input space. The global error of the signal representation is smaller with more neurons: there are more encoding vectors and the angle between the error vector and the closest encoding vector gets smaller, as N grows.

We illustrate the influence of coverage in 2D for $N = 60$ and $N = 500$ neurons (Fig. 6.6b). We sorted the neurons in the raster plot by the angle of their encoding vector. It is then easy to see how successive neurons get recruited, as the signal moves clock-wise or anti-clockwise along its circular trajectory. The input gain is fixed at gT, where $g = 5$, and $T = 0.55$ is the firing threshold. Both the smaller and larger networks are large enough to encode the signal very accurately. But the raster plots show that the neurons in the larger network fire fewer spikes, and they do so when their encoding vectors are better alined with the signal.

We also note that there is a network size, which we will denote by N_K, for which the error and population firing rate start to level off. This is somewhere around 300 neurons in 2D, which amounts to a 1.2° angle between neighbouring encoding vectors (Fig. 6.6a, dashed line in bottom panel). For this network size, the error decays as $N^{-0.02}$ and the population firing rate decays as $N^{-0.05}$. Having a network size larger than this value will then incur resource costs: too many neurons are dedicated to the representation of the 2D signal, where they could have been dedicated to another functionality. Thus, there is a maximum number of neurons, which allows accurate, robust, and efficient representation of variables. Extending the network size beyond this number would improve its performance only marginally. This number of neurons is somewhere around 1,000 in 3D, and our simulations do not show what it is for the 6D network.

We can however estimate what the maximum network size, N_K, for a 6D or a general K-dimensional network should be based on the known value in 2D (Fig. 6.6c). What we need is a number of neurons which cover the surface of the K-sphere, just as well as the 300 neurons cover the perimeter of the circle in the 2D case.

To this end, we note that the surface area of a K-dimensional sphere with radius R, is given by the following formula:

$$A_K = \frac{2\pi^{K/2}}{\Gamma(K/2)} R^{K-1}, \tag{6.3}$$

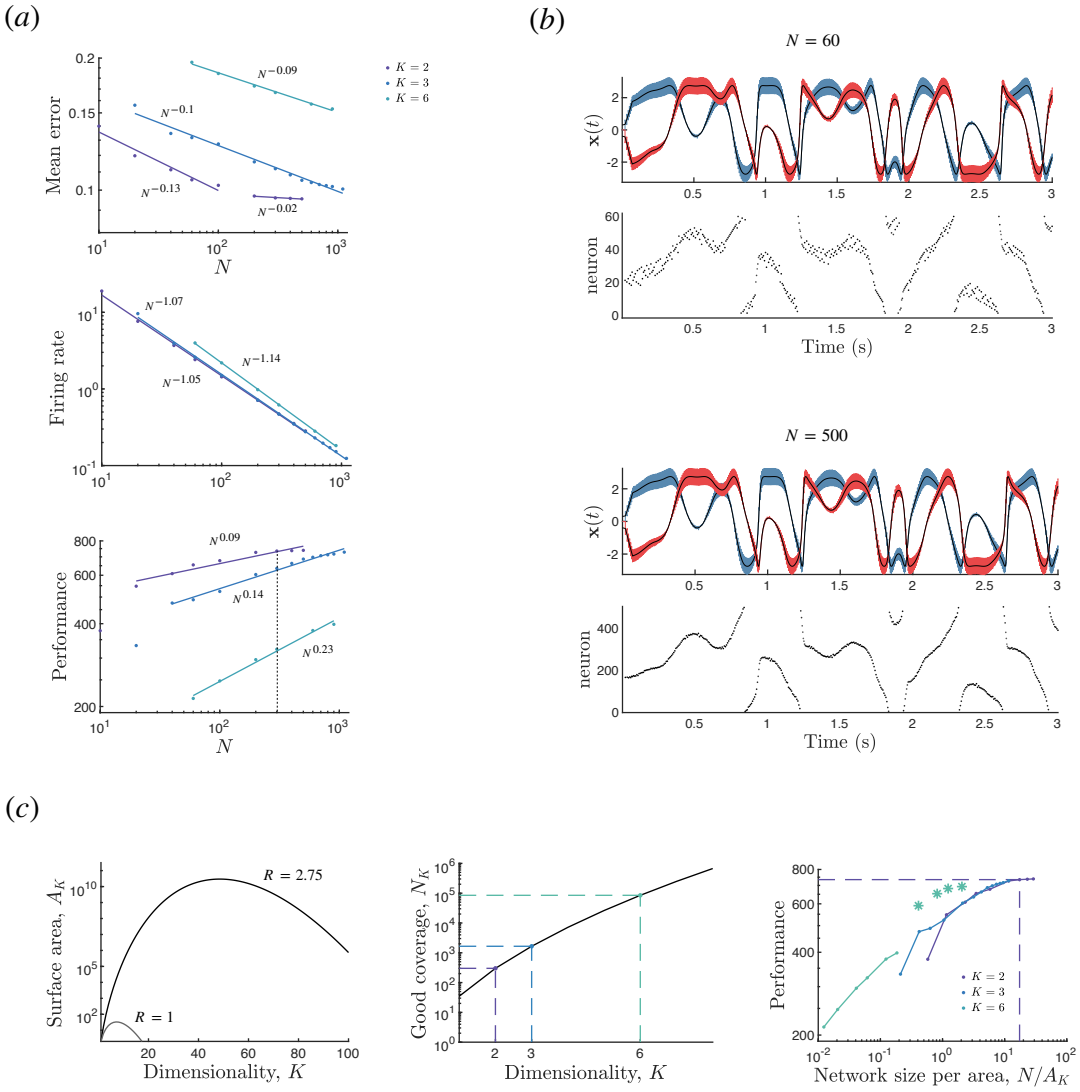


Figure (6.6) Performance of the optimal network as a function of its size. We consider optimal fully connected networks, of increasing size N. Their performance is assessed for $\{2, 3, 6\}$-dimensional inputs of fixed gain, $g = 5$. **(a)** Mean error, firing rate, and performance, as a function of network size, for three different input dimensionalities. The performance is calculated as the inverse of the product of the mean population spike count and the mean relative reconstruction error. The means are taken over all time steps. The 100 s long test function has a fixed gain, $g = 5$, and fixed radial speed, 500 steps per unit length covered, for all K. **(b)** Top: The reconstruction of a 3 second interval of the 2D test input, and the raster plot of the network consisting of $N = 60$ neurons. Bottom: Same, with a network consisting of $N = 500$ neurons. **(c)** Left: The surface area, A_K, of a K-dimensional sphere of radius R, formula (6.3) in the text. Values for two different radii shown: $R = 2.75$, corresponding to the sphere which surface is traversed by our test input, with the scale $R = gT = 5 \times 0.55 = 2.75$; and, $R = 1$, for comparison. Middle: Estimated maximum size of networks, beyond which performance improvement is marginal. The three circles indicate recommendations for $\{2, 3, 6\}$-dimensional networks. Right: Performance values from (a) plotted versus the number of neurons per unit surface area of the K-sphere. The results for $K = 6$ are extended closer to the recommended ratio, $N/A_K = 17.4$ (4 green stars). Details in the text.

where Γ is the gamma function. We plot the surface area as a function of K with the radius, R, given by the scale of the input, $gT = 5 \times 0.55 = 2.75$. (Fig. 6.6c, left panel). The surface area initially grows with R^{K-1}. Consequently, it is orders of magnitude greater than the area of the unit sphere ($R = 1$), which we show for comparison. For a large enough K, the area reaches a peak and then starts to asymptotically converge to zero. This is caused by the growth of the gamma function in the denominator of (6.3). The decay of the surface area as K goes to infinity is a known property of high-dimensional spheres. The value of K for which there is an onset of decay is much larger in the case of the test input, with $g = 5$, than for the unit sphere.

The appropriate coverage of the surface of a K-dimensional sphere can then be derived from the ratio of the recommended number of neurons in 2D, and the surface (or more precisely, the perimeter) in 2D. We observed a levelling off of performance around the network size, $N = 300$, which is equivalent to 17.4 neurons per unit perimeter in 2D. We estimate the maximum number of neurons for a good coverage in higher-dimensional networks as:

$$N_K = 17.4 A_K. \tag{6.4}$$

Using this formula, we estimate N_K to equal 1,100 neurons in 3D (Fig. 6.6c, middle panel). This value is in agreement with our numerical observation of the levelling-off of performance growth with N in 3D. The recommended maximum network size in 6D is on the order of 10^5.

We now consider the calculated performance of the $\{2, 3, 6\}$-dimensional networks, as a function of the number of neurons per unit surface area (Fig. 6.6c, right panel). We see that the 2D and 3D networks have the same level of performance as their respective coverage gets closer to the recommended ratio of 17.4 neurons per unit surface area.

The 6D network was simulated only up to $N = 900$, which is far below the recommended value. We try to get a sense of what would happen if we had more neurons in 6D, and so we extend the results to 10,000 neurons. The four green stars in the right panel of Figure (6.6c) show simulations with $N = \{2 \times 10^3, 4 \times 10^3, 6 \times 10^3, 10^4\}$ neurons. These are estimated performance values, as the test function used to obtain them was only 5 seconds long (as opposed to 100 seconds for smaller network sizes). This means that only a subset of neurons are active, as much of the space is not explored by the test function. Furthermore, we train the optimal decoder on this test function, and use the mean decoder error as a proxy for the mean error. We estimate the mean error by multiplying the decoder error by 1.04. This factor is the ratio of the decoding and test errors in smaller size networks, for which we trained the decoder properly on a long input that was different from the test input. This should give us a fair estimate, because the behaviour on a small part of the 6D surface will not be very different from the behaviour on the entire surface.

We see then that the estimated performance of a 6D network also starts to level off, as the coverage approaches the recommended value. Furthermore, it looks like the performance is converging to the same level as in the 2D and 3D networks. To make this a definitive conclusion, we would have to either do full simulations of larger networks, or derive analytic bounds. If we are correct, then given a K-dimensional input of fixed gain, there is a network size, N_K, for which the performance of the network starts to improve only marginally. Furthermore, the performance for this N_K is the same for all K.

We postulate that the error and firing rate are the same in the regime of good coverage, N_K, for all K. This is the case in 2D and 3D. We checked the error and rate for extended results in 6D, and we found that they approach those from 2D and 3D. Thus, when comparing the performance trade-off with increased sparseness, we could hypothetically pick this N_K, and see how fast the performance decays for different K, as we remove connections.

However, N_K grows exponentially with K. In 6D, the recommended number of neurons is far too large to encode only 6 different signals. It amounts to having $\mathcal{O}(10^4)$ neurons per dimension. So we suggest that real neural networks do not operate in this regime. From an efficiency point of view, it would be better to have a few smaller sub-networks, which code for 2 or 3 signals at a time. For example, we could have two networks with a 1,000 neurons each, which encode three signals at a time. This is two orders of magnitude smaller than requiring a single network to encode 6 signals, in the regime of good coverage.

Alternatively, it might be that biological networks do not operate at this 'edge' of performance. It might be that they operate in a regime where performance could still improve with more neurons added, but what matters is that a particular level of precision is ensured. If we pick a level of 0.1 mean relative error, then this level of accuracy is achieved with 0.2×10^3, 1×10^3, and 2×10^3 neurons in 2, 3, and 6 dimensional networks, respectively. The mean error is, as always, measured relative to signal strength. We could then use these network sizes to compare the performance trade-off for different K. In that case, the performance decays with increasing K (just like it does for fixed N, $p = 1$, in Fig. 6.4), but the decay is much smaller: the mean error is 0.1 in all three networks, while the population fires about 140 spikes per second in 2D, and about 150 spikes per second in 6D. The performance is only slightly worse in 6D.

6.4.2 | Varying network size in independent networks

We now address the influence of network size on the performance of networks with no recurrent connections. This is the case of independent leaky integrate-and-fire neurons. Given the fact that we consider signals without any noise, adding more neurons to these networks will decrease their performance. This is in contrast to optimal fully connected networks, where the performance grows with N.

As more neurons are added to the independent network, the firing rate of each neuron stays the same, as they are driven entirely by their feedforward input (Fig. 6.7a, middle panel). This means that the population firing rate grows linearly with N. Simultaneously, the mean error decays as N gets larger (Fig. 6.7a, left panel). The higher precision is a consequence of having more spikes per unit of time, which can be used to fit the signal, $\mathbf{x}(t)$. The optimal decoder for each neuron becomes smaller in length. Thus, as the corrections of the estimated signal, $\hat{\mathbf{x}}(t)$, become more frequent, these corrections become smaller, leading overall to a better tracking of the signal.

The decrease in error however is not as fast as the growth in the population rate, and so the performance of an independent network decays with N (Fig. 6.7a, right panel). Given a fixed network size, the higher the dimensionality of the input, the slower the performance decay with N. Furthermore, for a certain network size, the error starts to decay a lot more slowly (around $N = 200$ in 2D, and $N = 600$ in 3D), and then the performance starts to

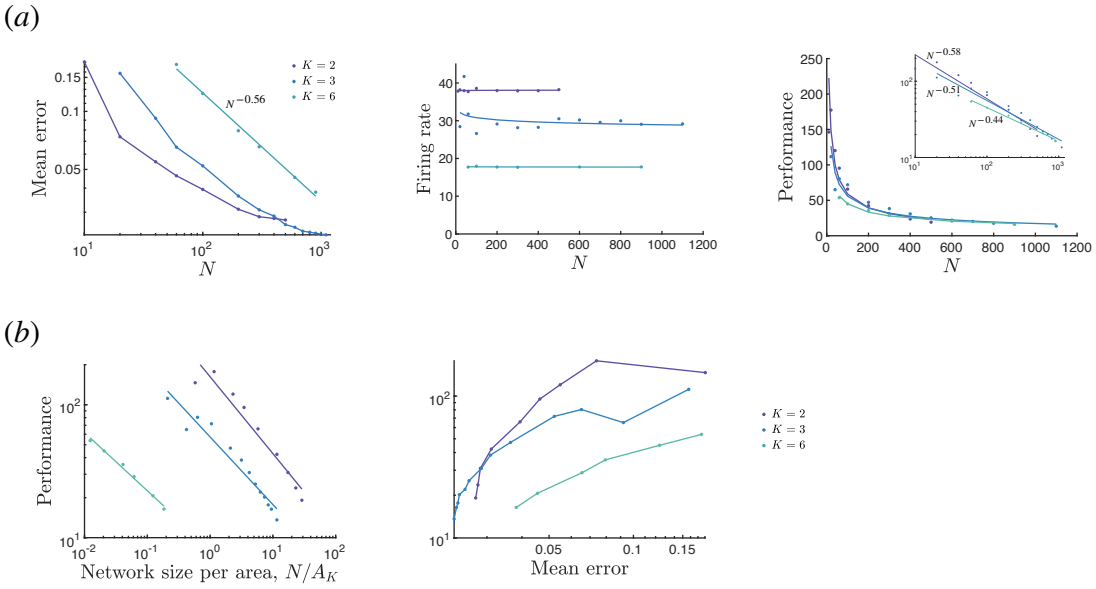


Figure (6.7) Performance of the independent network as a function of its size. We consider networks of independent leaky integrate-and-fire neurons, of increasing size N. Their performance is assessed for $\{2, 3, 6\}$-dimensional inputs of fixed gain ($g = 5$). **(a)** Mean error, firing rate, and performance, as a function of network size, for three different input dimensionalities. The performance is calculated as the inverse of the product of the mean population spike count and the mean relative reconstruction error. The insert shows the log scale. The means are taken over all time steps. The test function has a fixed gain ($g = 5$) and radial speed (500 steps per unit length covered) for all K, and is 100 s long. **(b)** Performance values from (a) plotted versus the number of neurons per unit surface area of the K-sphere, N/A_K (left), and versus the mean error (right).

decay more quickly. In this regime, the performance decay is driven by the linear growth of the population rate with N. We will comment on the levelling off of error decay for both independent and optimal networks, in the section on the preferred stimulus.

Lastly, we consider how performance depends on coverage of the input space, N/A_K, and on precision (Fig. 6.7b). For a set coverage, the higher dimensional networks perform worse than the lower-dimensional ones. This generally holds true for a set precision level as well. However, as the error continues to decrease, it might be true that higher-dimensional networks start to outperform the lower-dimensional ones. Comparing the performance of 2D and 3D networks for an error level smaller than 0.03 (Fig. 6.7b, right panel), shows that the 3D network starts to perform better at this point. Furthermore, if we extend the results for even smaller error levels, the 6D network might eventually outperform the other two networks.

6.4.3 | Performance trade-off between independent and optimal networks as a function of network size

We conclude this section by comparing the performance of fully connected and non-connected networks, in the context of variable network size and input dimensionality.

The performance of independent networks decays with network size, and it does so more slowly for higher dimensional inputs. In contrast, the performance of fully-connected optimal networks improves with increasing network size, and it does so faster for higher dimensional inputs (Fig. 6.8a). These two networks give us the performance at the extremes of connection probability: $p = 0$ and $p = 1$. We can thus estimate how much would sparsification of a network impact its performance, based on these edge cases.

We calculate the performance trade-off, $\triangle P$, between these two extremes of connection probability, as the difference between their performance values, P, for the same N and K:

$$\triangle P(N, K) = P(N, K|p = 1) - P(N, K|p = 0). \tag{6.5}$$

This gives us an estimate of the slope of the performance decay, from $p = 1$ to $p = 0$.

We find that, given a fixed network size, the performance trade-off is always smaller for a higher dimensional network (Fig. 6.8b, left panel). This means that the more signals a fixed size network is representing, the smaller the performance impact of removing synaptic connections. Also, for a fixed K, increasing the network size will always increase the performance impact of sparser connectivity, as $\triangle P$ grows with N. This means that the performance of larger networks will be impacted more by having fewer synaptic connections, than smaller ones.

Considering the performance trade-off for a given level of coverage shows that the trade-off will in fact be smaller in lower-dimensional networks (Fig. 6.8b, middle panel). This is also the case for the trade-off given a particular level of precision (Fig. 6.8b, right panel). Here, for a fixed mean error, the difference in the performance of optimal and independent networks is larger for higher dimensional networks, and it grows with decreasing error.

If biological neural networks prioritize a particular level of precision, then having fewer synaptic connections will make a smaller performance impact in lower dimensional networks. If on the other hand, there is a fixed number of neurons dedicated to a task,

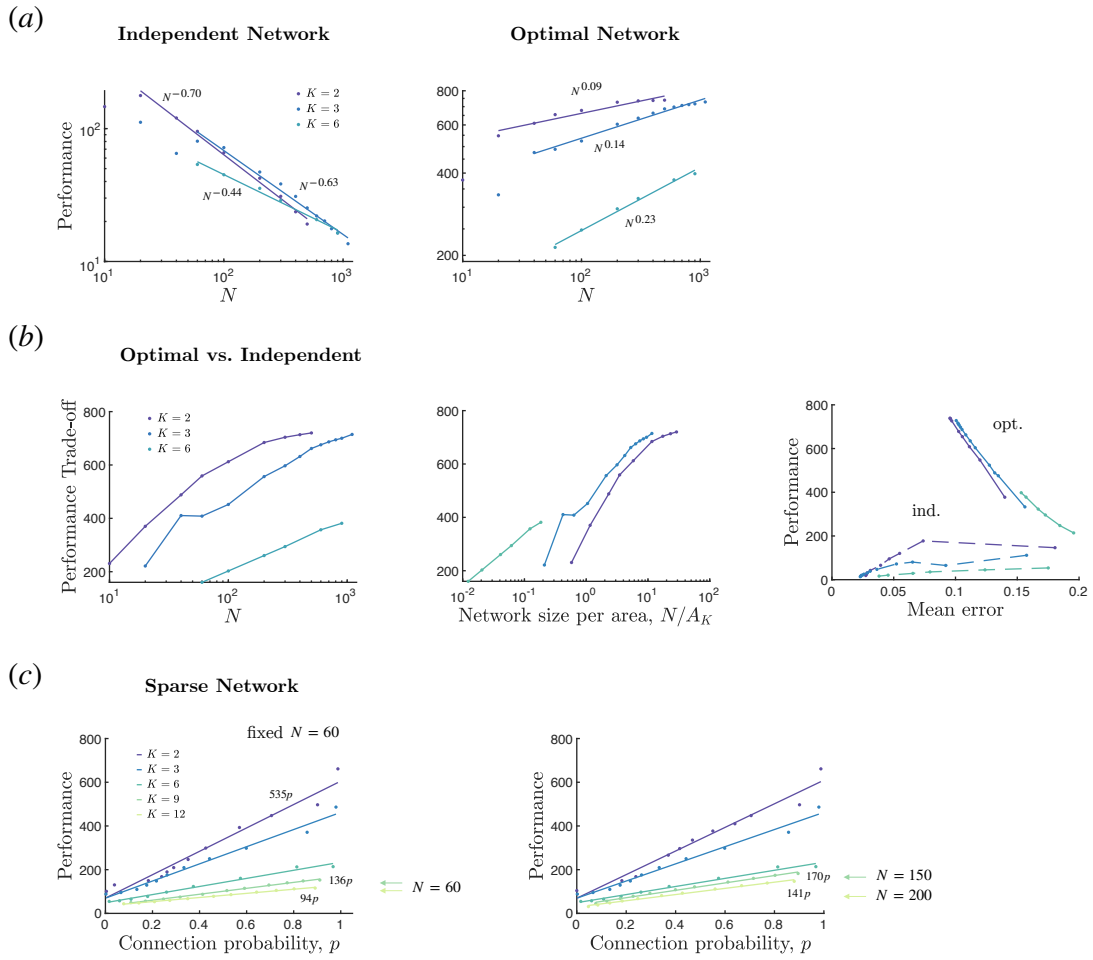


Figure (6.8) Performance trade-off between fully connected and non-connected networks, as a function of network size. **(a)** Performance, as a function of network size, for three different input dimensionalities. Left: Values for networks of independent neurons. Right: Values for the fully connected optimal networks. The performance is calculated as the inverse of the product of the mean population spike count and the mean relative reconstruction error. These plots are reproduced from Figures 6.6a and 6.7b, for convenience. **(b)** Performance trade-off between optimal and non-connected networks, as a function of N (left panel), and as a function of input space coverage (middle panel). The trade-off is calculated as the difference between the performance value of the optimal network and that for the independent network, for the same N and K. This gives an estimate of the slope of the performance decay, from $p = 1$ (optimal) to $p = 0$ (independent) networks. Right: Performance values as a function of the mean error for the optimal (full lines) and independent (dashed lines) networks. **(c)** Slopes of the linear performance increase with increasing connection probability, p, for networks of varying dimensionality. Left: Trained networks of fixed size, $N = 60$, trained to have different connection probabilities. This plot is reproduced from Figure 6.4c. Right: Same as left, except the 9D network has $N = 150$ neurons, and the 12D network has $N = 200$ neurons.

then it is the higher-dimensional networks that will be least impacted by the pruning of connections.

Lastly, in the earlier section on the orthogonality of high-dimensional spaces, we hypothesized that trained sparse networks do not perform much worse than fully connected ones, when K is large. We said that we would expect the performance to decay more slowly with decreasing connection probability, p, because the all-to-all optimal connections are already highly concentrated around zero. This agreed with the simulations of fixed size networks with varying K (Fig. 6.4c), which we reproduce in this section for convenience, (Fig. 6.8c, left panel).

However, the results in this section showed that as we increase the coverage of the input space, and enter a regime of more precise coding, the performance hit in going from full to zero connectivity is, in fact, worse for higher dimensional networks. The smaller trade-off in the fixed network size simulations, was in fact due to the higher dimensional network being in a higher error coding regime, and not due to its higher dimensionality. So in fact, when considering the impact of removing synaptic connections in general, it is the networks that encode signals with a larger relative error that are least impacted. This agrees with our consideration of the performance trade-off given a fixed precision level (Fig. 6.8b, right panel).

Might the performance decay with decreasing p not be linear for higher K, in the more precise regime? Preliminary analysis indicates that this is not the case. A more extensive analysis would be required, but we train networks for 9D and 12D inputs, with a somewhat greater network size, 150 and 200 neurons respectively (Fig. 6.8c, right panel). The decay in performance is greater, as we are in a somewhat more precise coding regime. But, the decay is still linear. This supports the idea that the many small synaptic weights are just as important for performance as the few strong ones.

6.5 | Input scale and the performance trade-off

We have so far considered the trade-off for fixed input gain, g. However, input gain can impact the performance trade-off between more and less connected networks.

We do a short analysis of its impact by considering 15-dimensional networks, with connection probabilities, $p = 1$ and $p = 0$, on the two ends of the connection probability spectrum. We consider first the optimal networks, with $p = 1$, and three choices of network size. Not surprisingly, the mean (relative) error of optimal networks decays with $1/g$ (Fig. 6.9a). This is due to the fact that the mean error is measured relative to the input strength, which grows linearly with g, and to the fact that the absolute error remains generally unchanged, as g grows. The error does start to decay more slowly, however, for the 2D simulations, where it starts to decay as $g^{-0.7}$, for $g > 20$. We will comment on this in the next section, where we briefly discuss the preferred stimulus. The mean error of independent networks decays with g as well, but more slowly than in the optimal networks.

The population firing rate grows with g, and it does so faster for independent networks, due to the missing inhibition of similarly tuned neurons that optimal networks benefit from. Overall, the performance of optimal networks is always better than that of independent ones, with increasing gain. The performance decays for both types of networks, and it

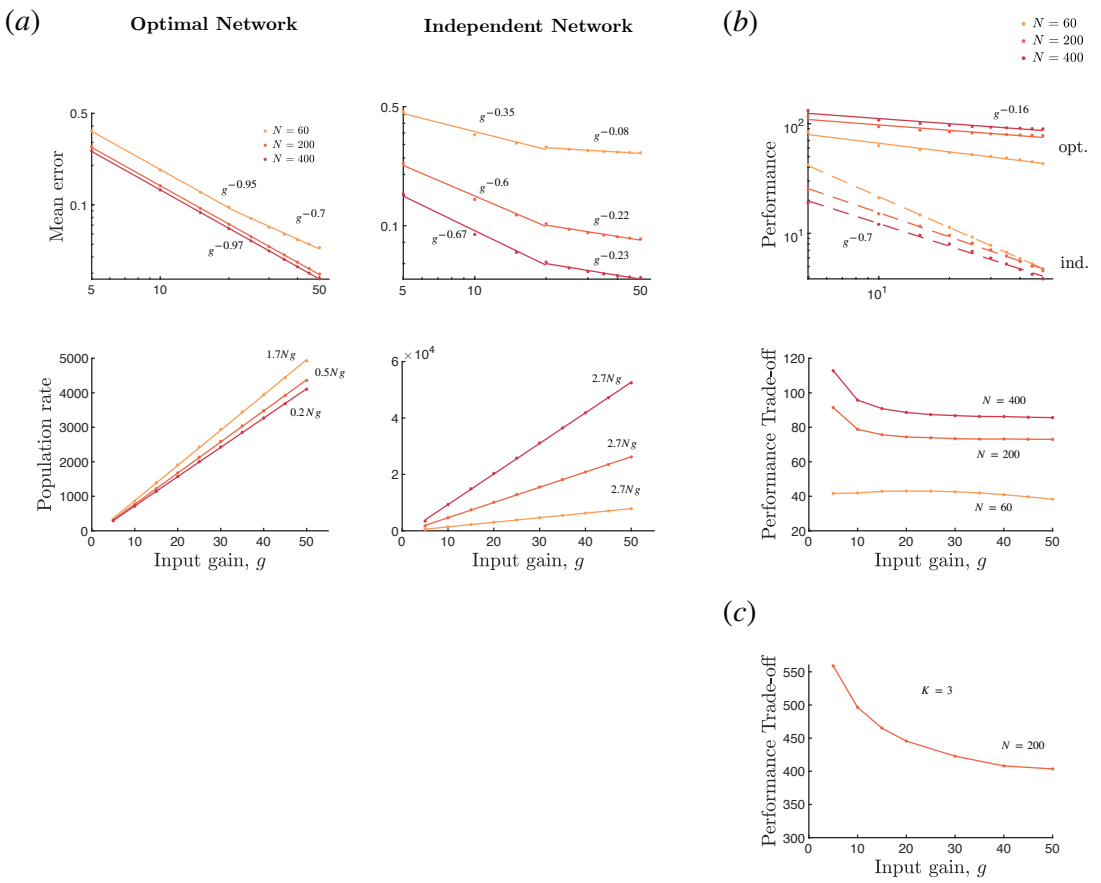


Figure (6.9) Performance trade-off between fully connected and non-connected networks, as a function of input gain. We consider optimal and independent networks, of increasing size N, which receive 15-dimensional inputs. The input has fixed gain, g, ranging from low (5) to high (50). **(a)** Mean error and population firing rate, as a function of input gain. Three different network sizes are considered. Left: Results from the fully connected optimal networks. Right: Results from the independent networks. **(b)** Performance of optimal and independent networks, as a function of input gain. The performance is calculated as the inverse of the product of the mean population spike count and the mean relative reconstruction error. Top: Performance of optimal networks (full lines) and independent networks (dashed lines), for three different network sizes. Middle: Performance trade-off, as a function of network size, for low ($g = 5$) and high ($g = 50$) input gains. The performance trade-off is calculated as the difference of the optimal and independent performance values, for fixed N and g. Bottom: Same as the middle plot, but on a logarithmic scale. **(c)** Performance trade-off for a network which receives a low-dimensional input (K=3), with network size, $N = 200$.

does so faster for independent networks (Fig. 6.9b, top panel). The performance trade-off generally becomes smaller as the input gain grows (Fig. 6.9b, bottom panel). The exception is the smallest considered network consisting of only 60 neurons, which represents the 15-dimensional input. There, the trade-off increases slightly with g, and then starts to decay.

These results indicate that the higher performance trade-off for higher dimensional networks might be ameliorated at least somewhat in networks that represent higher gain inputs.

Lastly, we also show the performance trade-off for an example lower dimensional network, with K=3 (Fig. 6.9c). Its trade-off is higher in general, because of the higher coverage (higher precision regime), but the percentage decrease in trade-off is larger than in the 15D network. So we expect that had we considered larger 15D networks, the increased gain would have decreased the trade-off more.

6.6 | Relating the performance trade-off to the preferred stimulus

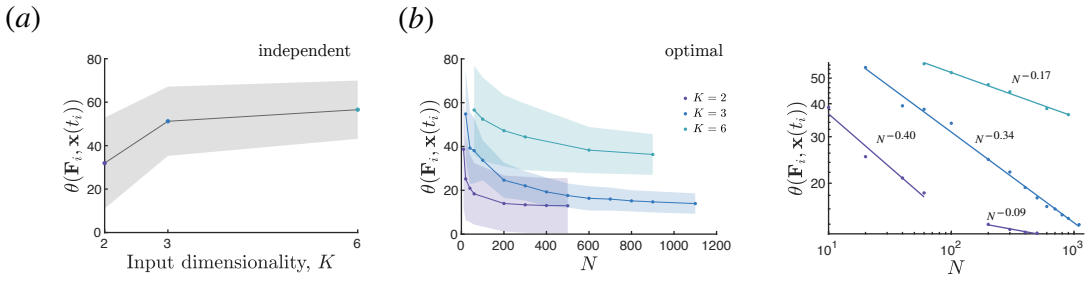


Figure (6.10) Preferred stimulus of neurons in optimal and independent networks, as a function of network size and input dimensionality. The input gain is $g = 5$. For each neuron i, we calculated the angles between its encoding vector, $\mathbf{F}_i$, and the inputs, $\mathbf{x}(t_i)$, at the times, t_i, of neuron i's spikes. The mean of these values over spike times gives us the preferred angle, θ_i, for each neuron, i, in reference to its encoding vector. We will refer to it as the preferred relative angle. We finally calculated the mean of these angles, $\theta = \langle\theta_i\rangle_i$, over all neurons, $i = \{1, ..., N\}$, and their standard deviation. These two values give us a measure of the difference between the preferred angle and the encoding vector, for all neurons in the network. **(a)** The mean angle between the preferred stimulus and encoding vector, for the independent networks, as a function of dimensionality, K. The standard deviation is shown by the shaded region. There was no dependence on the network size, and we considered $N \in [20, 500]$, $N \in [20, 1100]$, and $N \in [60, 900]$, in 2-, 3-, and 6-dimensional networks, respectively. **(b)** Left: The mean angle between the preferred stimulus and encoding vector, for the optimal networks, as a function of networks size, N. The three curves show results for input dimensionalities, $K = \{2, 3, 6\}$. The standard deviation is shown by the shaded region. Right: Same results on a logarithmic scale. Only the mean values are shown.

Throughout this chapter, we examined how varying recurrent connection probability impacts the performance of higher dimensional networks. We explored how this trade-off

depends on variables, such as the network size, input dimensionality, and input gain. In fact, most of the changes in the trade-off can be summarized by how these variables impact the preferred stimulus of neurons in the network.

We will define the preferred stimulus in reference to the encoding vectors, as follows. Given a network, which collectively represents the variable $\mathbf{x}(t)$, we will consider the angle between the encoding vector of each neuron and the variable, $\mathbf{x}(t)$, at the time of the neuron's spikes. More precisely, for each neuron i, we calculate the angles, $\theta_i(t_i)$, between its encoding vector, $\mathbf{F}_i$, and the inputs, $\mathbf{x}(t_i)$, at the times, t_i, of neuron i's spikes:

$$\theta_i\left(\mathbf{F}_i, \mathbf{x}(t_i)\right) = \cos^{-1}\left(\mathbf{F}_i\mathbf{x}(t_i)/\left\|\mathbf{x}(t_i)\right\|_2\right). \tag{6.6}$$

We then calculate the preferred angle, θ_i, in reference to the encoding vector, as the mean over all t_i, for each neuron, i. We call this angle the preferred relative angle, as it is measured relative to the encoding vector. We finally calculate the mean of these angles, $\theta = \langle\theta_i\rangle_i$, over all neurons, $i = \{1, ..., N\}$, as well as their standard deviation. These two values give us a measure of the difference between the preferred angle and the encoding vector, for all neurons in the network.

We would naively expect that the angle θ, should in general be zero or close to zero. However, this is generally not the case. The independent network, for example, has the preferred relative angle around 30° for 2D inputs, and 50° for 6D inputs, when the input gain is $g = 5$ (Fig. 6.10a). The preferred relative angle does not depend on the network size in the absence of recurrent connectivity. The optimal network, on the other hand, does show dependence of the preferred relative angle on the network size (Fig. 6.10b). This angle gets closer to zero for increasing coverage, although it is still at best around 20° for good coverage in 2D and 3D, and 40° in 6D. The preferred relative angle equals that in the independent networks for low coverage, and then decays toward zero, as N increases. The variance around the mean also decreases with N, meaning that the neural tuning curves are becoming more narrow.

The largest possible angle between a neuron's spike-triggered input and its encoding vector

To understand why this happens and how it relates to our earlier discussion of the trade-off, we need to estimate the largest angle between the input signal and an encoding vector, that could still lead to that neuron spiking. When we consider the independent network, a neuron i spikes when $\mathbf{x}(t)\mathbf{F}_i \geq T$, as long as it had not spiked recently. Their scalar product can be expressed in terms of the cosine of the angle between them:

$$\mathbf{x}(t)\mathbf{F}_i = gT\cos(\phi). \tag{6.7}$$

Here, we have used the fact that the length of the input vector is given by gT, and the length of the encoding vector is unity.

We are interested in the limit case, when the scalar product is equal to the threshold. This is the case when $gT\cos(\phi) = T$, that is, when $\cos(\phi) = 1/g$.

So we get that the angle ϕ is given by:

$$\phi = \cos^{-1}(1/g). \tag{6.8}$$

Recalling that $\cos(0) = 1$ and $\cos(\pi/2) = 0$, we see that $\phi = 0$ when $g = 1$, and $\phi = 90°$ as $g \to \infty$. For unit gain inputs, neurons will spike only when the input alines fully with their encoding vector. As the input gain grows, the largest angle between $\mathbf{F}_i$ and the spike-triggered input will keep growing toward 90°.

In the simulation results presented in Fig. 6.10, the input gain was 5, meaning that $\phi = 78°$. The preferred relative angle for the independent 2D network is approximately equal to $\phi/2$, which agrees with the uniform distribution of angles in 2D (Fig. 6.5a, left panel). In 3D and 6D, the preferred relative angle is greater than $\phi/2$ because of the bell-shaped distribution of angles in these spaces (there are more orthogonal vectors).

In the optimal network, on the other hand, the increased coverage brings the preferred relative angle, θ, closer to zero, because of recurrent inhibition of similarly tuned neurons. Here, in addition to being impacted by the gain, the angle θ is also affected by N and by the radial speed of the input.

For example, if the coverage is small, and the input speed is somewhat fast, then θ can be a bit larger than in the independent network. One can see this around $N = 20$ in the optimal network, where the 2D θ is actually 40°, as opposed to about 35° in the independent network (Fig. 6.5a, and b, left panel). The effect is not large, and can be explained as follows. Once a neuron has spiked and inhibited all neurons which encoding vectors are in its hemisphere, then, as the input vector continues to move, it will not be able to cause a spike in the neurons that were inhibited the most. It will instead cause a spike in a neuron that was inhibited less, which encoding vector can be quite misaligned with the input vector. As soon as the coverage improves, θ quickly becomes better than in the independent network. What causes the preferred relative angle to deviate more from zero is large gain, large input speed, and small coverage.

Intuitive explanation of the impact of input speed

For example, when considering how input speed impacts the preferred relative angle, we can consider a network with the neuron i which just fired a spike at the center, and the remaining neurons shown as a function of their tuning distance from i (Fig. 6.11a). The distance between neuron i and neuron j is inferred from the scalar product of their encoding weights, $\mathbf{F}_i^T\mathbf{F}_j$. The filtered input $\mathbf{x}(t)$ can be imagined to be parallel to $\mathbf{F}_i$, and lead to a spike of neuron i (although this is not necessary for it to cause the neuron i to spike).

After neuron i spikes, the filtered recurrent input, V_r, that neurons j receive depends on the distance of their encoding vectors from $\mathbf{F}_i$ (Fig. 6.11b, left panel). It is given by:

$$V_r = -\frac{\beta T}{2}\mathbf{F}_i^T\mathbf{F}_j. \tag{6.9}$$

The closer the neuron j is to neuron i in tuning space, the more inhibited it will be.

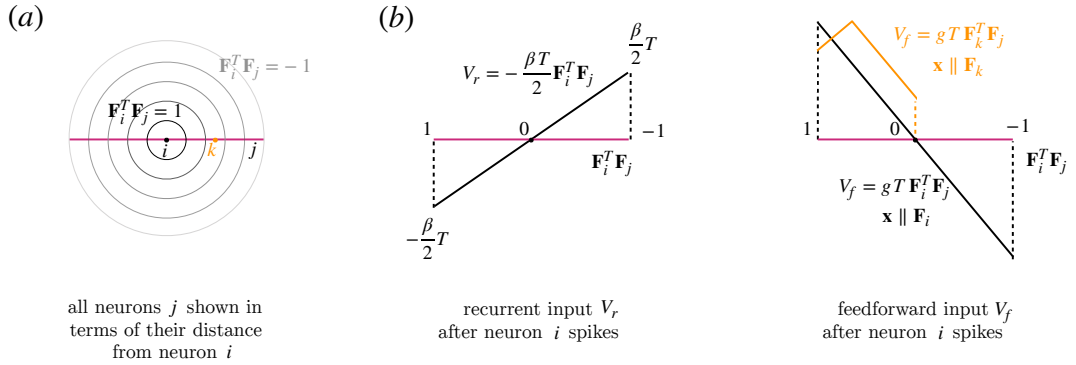


Figure (6.11) Schematic of the influence of input speed on which neuron spikes next in an optimal network. **(a)** Schematic of all neurons in a network, with neuron i at the center, and the furthest neurons away from it on the outside circle. The distance between neuron i and neuron j is inferred from the scalar product of their encoding weights, $\mathbf{F}_i^T\mathbf{F}_j$. Thus, distance is not necessarily a physical distance, but rather a distance in tuning space. The filtered input $\mathbf{x}(t)$ can be imagined to be parallel to $\mathbf{F}_i$, and lead to a spike of neuron i (although this is not necessary for it to cause the neuron i to spike). **(b)** Left: After neuron i spikes, the filtered recurrent input, V_r, that neurons j receive depends on the distance of their encoding vectors from $\mathbf{F}_i$. It is given by, $V_r = -\frac{\beta T}{2}\mathbf{F}_i^T\mathbf{F}_j$. The closer the neuron j is to neuron i in tuning space, the more inhibited it will be. Right: The feedforward input some short time after the spike of neuron i depends on the relationships between encoding vectors, as well as on the present filtered input $\mathbf{x}$. We illustrate two situations: the input is still parallel to $\mathbf{F}_i$ (black line), or it has moved radially and is parallel to some $\mathbf{F}_k$ (orange line). We only show the positive values in the second case. In the first case, the feedforward input for all neurons is given by, $V_f = gT\mathbf{F}_i^T\mathbf{F}_j$. Similarly, the same formula holds in the second case, with the index i replaced by index k. How far away $\mathbf{F}_k$ is from $\mathbf{F}_i$ will influence which neuron spikes next. Details in the text.

The feedforward input some short time after the spike of neuron i depends on the relationships between encoding vectors, as well as on the present filtered input $\mathbf{x}$ (Fig. 6.11b, right panel). We illustrate two situations: the input is still parallel to $\mathbf{F}_i$ (black line), or it has moved radially and is parallel to some $\mathbf{F}_k$ (orange line). In the first case, the feedforward input for all neurons is given by:

$$V_f = gT\mathbf{F}_i^T\mathbf{F}_j. \tag{6.10}$$

Similarly, the same formula holds in the second case, with the index i replaced by index k. How far away $\mathbf{F}_k$ is from $\mathbf{F}_i$ will influence which neuron spikes next. This will depend on the sum of V_r and V_f. The filtered recurrent input, V_r, is greater for neurons further away from i, and V_f is the greatest for neurons close to k, so the next neuron could be the one that was less inhibited and also closer alined with $\mathbf{F}_k$. This could be somewhere half way between i and k, but the exact neuron will depend on the decay of the filtered recurrent input (time constant λ), and the gain and speed of the input.

An easier way to visualize the influence of input speed is to imagine an extreme case in which the input moves really fast. Then, any neuron which forms an angle smaller than $\cos^{-1}(1/g)$ with $\mathbf{x}$ could potentially spike, as long as the inhibitory recurrent input has decayed. If $\mathbf{x}$ moves really fast, it could move outside of the influence of recurrent inhibition of the previous spike, and then the first neuron that spikes will be exactly some neuron k, for which the angle between $\mathbf{F}_k$ and $\mathbf{x}$ is the largest possible, $\cos^{-1}(1/g)$. This could be close to 90°, depending on g.

This gives some intuition behind how input speed affects the preferred relative angle of neurons in an optimal network. Similar analysis could explain the influence of coverage and input gain.

6.6.1 | Relationship between the preferred angle and network performance

The preferred relative angle is closely related to the precision with which the signal is represented. The smaller it is, the more alined the decoder is with the spike triggered input. This makes the decoder point more in the direction of the global error of signal representation, thus making the difference between the signal and the network estimate smaller. For inputs which do not move very fast, increased high coverage should lead to an absolute error which approaches a minimum. This minimum absolute error is approximately given by $T\triangle t_{isi}/2$, where the average population inter-spike interval, $\triangle t_{isi}$, depends on λ and input gain. $\triangle t_{isi}$ could be solved for:

$$\left(gT + \frac{T}{2}\right) e^{-\lambda \triangle t_{isi}} = gT - \frac{T}{2}. \tag{6.11}$$

Here, the LHS is the magnitude of the signal estimate after an optimal correction, and the RHS is its value after an exponential decay, which elicits a second correcting spike.

Solving for $\triangle t_{isi}$:

$$\triangle t_{isi} = \frac{1}{\lambda} \log\left(\frac{2g+1}{2g-1}\right). \tag{6.12}$$

So the minimum mean error, weighted by the input strength, is approximately given by:

$$\text{Mean error} = \frac{T}{2g\lambda} \log\left(\frac{2g+1}{2g-1}\right). \tag{6.13}$$

Similarly, the associated population firing rate is inversely proportional to $\triangle t_{isi}$:

$$\text{Population Firing Rate} = \lambda \log^{-1}\left(\frac{2g+1}{2g-1}\right). \tag{6.14}$$

This gives us an expression for the performance of optimal networks in the limit of infinite coverage, and slow input speeds:

$$\text{Performance} = \frac{2g}{T}. \tag{6.15}$$

It is important to note that these expressions are meaningful only for small inputs which move slowly, and networks with large coverage. For example, the expression for the population inter-spike interval, (6.12), relies on the assumption that the gain and speed are small. Otherwise, another neuron could spike much sooner than estimated. Furthermore, the positive relationship between performance and the gain can easily be misconstrued and is misleading, as it should only apply to small g. The performance in our simulation of a 3D network with modest coverage and input gain, $g = 5$, was found to actually decrease with gain (Fig. 6.9c). Even with these fallbacks, the brief analysis above gives us an idea of how we could go about deriving analytical results that apply to conditions that networks actually operate in.

We could use similar reasoning to estimate the error in the regime of smaller performance, when the decoder is not exactly alined with the input signal, due to higher input speed or lower coverage. This case corresponds to the greater preferred relative angle. This would give us an analytic bound on the performance of the optimal network in the regime of lesser performance.

Similarly, we could try to get analytic results for the performance of the independent network. There we would estimate the mean error and firing rate based on the percentage of neurons, which receive enough feedforward input to spike. The area of a spherical cap, which contains these neurons, is: $A = 2\pi^2(1 - \cos(\phi)) = 2\pi^2(1 - 1/g)$, for a 2D network. The ratio of the area, A, to the total surface of the 2-sphere of radius gT, is an estimate of the proportion of neurons that will be activated by the signal, $\mathbf{x}(t)$.

This analysis would give us an analytic prediction of the performance trade-off between the edge cases, the optimal and independent networks, and is part of future work.

6.7 | Conclusion

In this chapter, we showed that learning sparse recurrent connectivity translates naturally to networks which receive higher-dimensional inputs. The decrease in the performance of the network, as measured by the increase in its error and firing rate, was gradual with decreasing connection probability. As expected, the most performant network was always the optimal one with all-to-all connections, and least performant network was the independent one, with no recurrent connectivity. Comparing the decrease in performance from one extreme of connection probability to the other, allowed us to characterize the performance trade-off, as a function of various variables that influence it. We studied the trade-off as a function of the number of input dimensions, K, the network size, N, the scale of the input, g, and the derivative of the input, as measured by its radial speed.

Comparing networks of various input dimensionalities, K, showed that the performance trade-off was the lowest in higher dimensional networks, when we considered a fixed number of neurons, N. However, when comparing the trade-off in networks with an appropriate coverage of the high-dimensional input space, we found that the trade-off was in fact worse in higher-dimensional networks. Given that the input space becomes larger with increasing K, we considered N that was proportional to the size of this space. This then also ensured that the precision with which a signal was represented by a network was of the same order with increasing K. We considered high coverage, which allowed high precision coding, and in this regime the performance trade-off was smaller in lower-dimensional networks.

Furthermore, we found that the high precision requirement led to an exponential growth of the required network size, with increasing dimensionality K. We propose that a more optimal coding of high-dimensional variables is achieved by a number of smaller circuits, which code for fewer variables at a time. For example, instead of representing six variables with a single large network, which network size is of the order 10^5, we could have three small networks which represent two variables at a time, and achieve the same level of accuracy with two orders of magnitude fewer neurons.

Keeping K fixed and considering two networks of different network sizes, the larger network will be more sensitive to the removal of synaptic connections. This is because, as the network size increases, the performance of the optimal network increases, while as that of the independent one decreases. The performance decrease of the independent network is faster than the performance increase of the optimal one, and so the performance trade-off increases with network size. Thus, a network with more neurons, which also necessarily represents a variable more accurately, will be impacted more by a reduced connection probability. Networks, which operate in the large error regime, are impacted less by sparse connectivity than those, which operate in the lower error regime.

Increasing the gain of the input lessens the performance trade-off. It leads to the worsening of performance of both optimal and independent networks, with the population rate growing with input gain, and the mean relative error decaying. In the optimal network, the error decay is inversely proportional to the input gain, so that doubling the gain halves the error. A high-dimensional network with poor coverage and a high coding error, could thus benefit from this. It could turn up the input gain and enter the high precision coding regime, without needing to ensure sufficient coverage. When comparing the performance

trade-off in different networks of the same precision level, we could thus also consider increasing the input gain for networks with lower coverage. This would make the networks with lower coverage and higher input scale be the least impacted by reduced connection probability. However, this would be at the expense of lower over-all performance, as the population firing rate would be high. It is interesting to consider nevertheless, especially for networks which only sometimes need to ensure high precision coding. In contrast, those that need to ensure high precision most or all of the time would probably best rely on a larger network size. In this case, the network would minimize the operational cost of firing, at the expense of recruiting more neurons into a functional unit.

Lastly, we also considered how performance depends on connection probability, in networks which were trained with the optimal sparse learning rule. We expected that the decrease in performance with increased sparseness would not be monotonic, because of the distribution of optimal weights in higher dimensions. However, we observed a monotonic relationship, suggesting that the many small recurrent connections are as important as the few strong ones.

7 | Conclusion and Perspectives

Local cortical networks are dominated by recurrent connections with neurons within the same layer. The connection probability is variable, but is always somewhere at an intermediate level. It is neither highly sparse, as in random balanced networks (Brunel, 2000), nor dense and all-to-all, as in idealized coordinated spike-coding networks (Boerlin et al., 2013). The connection probability and the structure of connections have a big impact on the feature selectivity of individual neurons (Harris and Mrsic-Flogel, 2013), and on the efficiency with which they represent stimuli (Deneve and Machens, 2016).

7.1 | Learning efficient representation in networks with limited connectivity

In this thesis, we showed that networks with structured recurrent connections at an *intermediate/sparse* connection probability can represent signals *efficiently*. We developed a local synaptic rule which optimizes the number and efficacy of recurrent connections, so that the resulting network can represent its feedforward inputs with a minimal number of spikes for a given precision level (Chapter 3). The learned connectivity structure is such that the connections that impact the efficiency the least are pruned, and the strength of the remaining ones is optimized for efficient signal representation.

The learning rule can train efficient networks with any connection probability, by optimizing the divergence of subthreshold voltage from the rest state, distributing the spiking across all neurons, and penalizing the total sum of the recurrent inputs into a neuron (Chapter 4). The penalization is implemented by a Lasso regularization of synaptic weights, which encourages sparseness. When the constraint is not active, the synaptic rule recovers the E/I balance restoring rule (Brendel et al., 2017), which converges to the optimal all-to-all connections for idealized coordinated spike-coding networks with instantaneous synaptic connections. The self-organizing principle, which optimizes the level of E/I balance, also optimizes the divergence of the coding error from zero.

Here, we go further and derive additional results for the balance restoring rule. We provide a proof for its convergence to a fixed point for particular inputs, and explain why the fixed point in practice looks like a noisy version of the analytical fixed point. We also derive the learning rule which implements synaptic delays and dynamics, in fully connected networks. With the addition of the Lasso constraint on recurrent weights, this rule leads to optimal connections in networks with intermediate connection probabilities, and realistic

synaptic transmission dynamics.

The learned recurrent connections can be interpreted as a maximum a posteriori estimate of a linear Bayesian regression with a Laplace prior over the recurrent weights. The likelihood function is a normal distribution over the difference between the projected feedforward inputs and recurrent inputs, in the case of uncorrelated inputs. The mean of this distribution is zero, and the standard deviation scales with the firing threshold and with one over the number of input dimensions. The Laplace prior favours sparse solutions in the recurrent weights. By sparse, we simply mean a solution where some recurrent weights are set to zero. We could have also tried other priors, such as a Dirichlet-Laplace prior, which improves the optimality of parameter shrinkage (Bhattacharya et al., 2015). However, we do not expect the results to change significantly.

Following the Lasso regularized synaptic rule, the network learns to cancel the shared feedforward inputs with the intrinsic recurrent inputs, while also pruning those lateral connections that contribute the least to the feedforward input cancelation. The lateral connections that get pruned are those between neurons, which are selective for close-to orthogonal input orientations. There is almost no overlap between the tuning curves of these neurons, and so they do not benefit much from sharing information about their respective activity levels. On the other hand, the lateral connections that are grown are those between neurons, which are selective for input orientations with a non-zero overlap (non-zero scalar product). For example, two neurons which are selective for almost the same input orientations, i.e., orientations with a big overlap, will have almost overlapping tuning curves. The network's efficiency will be maximized when these neurons share strong negative connections, which prevent unnecessary error-correcting spikes immediately after one of them has fired an error-correcting spike.

Our learning rule can train networks of any connection probability. On the other hand, the E/I balance learning rule in randomly connected networks (Vogels et al., 2011) applies to very sparse connectivities, on the order of 0.02, which is much sparser than observed in the cortex, and much more restrictive than our learning rule.

7.1.1 | The structure and activity of networks with limited connectivity

As the sparseness constraint on the recurrent weights becomes more important, as in networks which occupy a very small volume, for example, the number of connections that are grown will decrease. The network will progressively prune more of its lateral connections, starting from the connections between the neurons with orthogonal input selectivities, and then moving on to connections between neurons with increasingly more overlapping tuning curves. The last lateral connections that will be pruned, in the limit of zero recurrent connectivity, will be between the most similarly tuned neurons, which have the most similar input selectivity. These connections contribute the most to the efficiency of the networks, as they are the strongest and suppress unnecessary network activity the most.

The connections are structured and symmetric, with almost all connections being reciprocal. The synaptic connectivity matrix can be decomposed into a sum of a low rank matrix and additional rank one matrices, which become more dominant, as the connectivity becomes more sparse. For fully connected networks, the synaptic connectivity has

the dimensionality of the input space, which is much smaller than the dimensionality of the neural space. As the connections become more rare, the low-dimensional signature is still there, but the dimensionality of the recurrent matrix starts to get larger. In very sparse networks, the dimensionality grows to the maximum level, and equals the number of neurons in the network.

For a neural network without synaptic delays, the dimensionality of the recurrent connections can be used to assess the optimality with which the network represents the feedforward inputs. The closer this dimensionality is to the dimensionality of the input, the more efficient is the representation. This greater efficiency is achieved, however, at the cost of growing and maintaining more synaptic connections.

7.1.2 | Coding strategies and the trade-off between firing and wiring

The networks fire more spikes when they have fewer recurrent connections, and their efficiency decays. However, they still perform better than a network of independent leaky integrate-and-fire neurons. There is a trade-off between the operational cost, associated with the firing activity of the network, and the developmental cost, associated with the growth of synapses. Our synaptic rule provides a way to minimize the firing cost in a network that can only grow a certain number of connections. Similarly, it provides a way to minimize the number of synapses that are developed in a network that can only afford a certain level of firing activity. This could potentially explain why there is an observed variability in the connection profiles of different neural circuits (Harris and Mrsic-Flogel, 2013).

Our analysis might also potentially explain the variability in connection profiles across different species. Different environmental constraints have different optimal trade-offs between an investment in growth and the operational cost during the life of an organism (Bullmore and Sporns, 2012). Long lived systems might invest more in the growth and maintenance of synapses, while short-lived systems might benefit less from this investment.

Different trade-off set points might also provide a way for a local circuit to optimize different functions, such as to transform inputs efficiently, or to transmit them reliably. For example, the numerous L2/3 principal cells can be interpreted as implementing a sparse code, with an efficient storage in the local synapses (Harris and Mrsic-Flogel, 2013). In contrast, the smaller number of long-range projection L5 cells in the output layer, have higher firing rates, which could subserve a reliable broadcasting of the results of local computations. Given this view of the functionality of the inner and outer layers, the optimal trade-off between wiring and firing would require denser connectivity of the inner layers, compared to that of the outer layers. A more sparsely connected network, with increased firing, could potentially be associated with improved broadcasting of signals.

7.1.3 | Efficient representation of high-dimensional inputs

The Lasso regularized synaptic rule extends naturally to networks, which represent higher-dimensional inputs (Chapter 6). For a fixed network size, an increase in the number of shared inputs decreases the amount of trade-off between firing and wiring. This result depends on the ratio of the network size and the surface area of the higher-dimensional

input space. When the network size and the firing thresholds are fixed, increasing the input dimensionality leads to a higher representation error and lower population firing rate.

However, if we scale the network size with the area of the inputs space, so as to ensure high coverage and low representation error, we find that the performance trade-off is smaller in lower-dimensional networks. This holds true for fixed firing thresholds, which lead to very sparse activity of neurons in high-dimensional networks. It should not hold true in networks with adaptive thresholds, where the neural firing rates are fixed regardless of input dimensionality.

The higher-dimensional framework allows us to compare different coding regimes, such as low error (high precision) and low firing rate (small operational cost) regimes. The optimal trade-offs between these two criteria are obtained at different network sizes and connection probabilities depending on the total number of feedforward inputs.

7.2 | Learning efficient signal representation with synaptic delays

Synaptic transmission delays are known to cause synchronization and mediate oscillations of the activity in coordinated spike-coding networks (Schwemmer et al., 2015; Chalk et al., 2016). The synchronous regular and synchronous irregular regimes are also well studied regimes in balanced random networks with synaptic delays (Brunel, 2000; Brunel and Wang, 2003).

Previous work has found that adding an optimal level of synaptic noise desynchronizes the coordinated-spike coding network, and improves its performance (Chalk et al., 2016). In the present work, we show for the first time that optimal recurrent connections can be learned in a network, which aims to represent its inputs efficiently with synaptic delays (Chapter 5). Furthermore, we show that the network can learn efficient signal representation with the additional biological constraint of limited recurrent connection probability.

Here we used a fixed synaptic delay for all neurons in the network. The longer this fixed delay is, the smaller the strength of the learned recurrent connections. Thus, the strength of the learned synaptic connections scales inversely with the length of the synaptic delay.

The network performance should increase when shorter delays are implemented between neurons with similar tuning properties. This is biologically motivated, as the length of the synaptic delay scales with the distance between neurons (Levy and Reyes, 2012). In networks with continuous tuning maps, such as retinotopic and tonotopic maps, the length of the delay then also scales with the distance between the neural preferred stimuli. Accounting for this variability in delay lengths would increase the over-all performance of the trained coordinated-spike coding networks.

The performance might further be improved by modeling delays with wide distributions. Sharply distributed synaptic delays lead to synchronous network activity, when excitatory inputs dominate inhibitory inputs in balanced random networks (Brunel, 2000). However, the activity becomes asynchronous when the distribution of delays becomes wide. This might lead to an increased desynchronization in our model as well.

Lastly, we point out that the scaling of the synaptic weights can be related to the

strength of the feedforward input. The synaptic efficacy is smaller for larger mean numbers of population spikes, which are fired during the synaptic delay. For fixed synaptic delays, larger feedforward inputs can thus be related to weaker recurrent synaptic connections.

7.2.1 | Optimal connection probability with synaptic dynamics

Our work points to an existence of an optimal connectivity level, for which the performance of the network is maximized in the presence of transmission delays.

With the added biological constraint of more realistic synapses, the optimal network connectivity is at an intermediate level. The smaller recurrent connection probability acts like a source of noise, which desynchronizes the network and optimizes its performance. This is contrasted by the idealized network with instantaneous synapses, for which the fully connected network is the most efficient.

A sparse number of feedforward inputs has been related to maximizing the dimensionality of the signal representation in the cerebellum (Litwin-Kumar et al., 2017). The authors of this study found that the anatomically observed sparse feedforward connectivity maximizes the expansion ratio of the output dimensionality vs. the number of feedforward connections. This analysis was done in purely feedforward networks. We could do a similar analysis for recurrent networks and find the optimal number of connections in different scenarios: with synaptic delays, as already discussed, as well as by systematically varying input dimensionality and network size.

7.2.2 | Robustness of the trained networks

The trained networks are robust to changing network size, input gain (Chapter 4), and presence of synaptic delays (Chapter 5). Here we comment briefly on the network's adaptation to changing input scales in the presence of synaptic delays.

If the scale of the feedforward input changes over time, learning will allow the network to adapt its connectivity structure to the new input scale, so that its connections are optimized for accurate signal representation. However, transient variability in input scale could lead to a decreased performance of the network, as spurious spikes which do not correct the error could be fired by the population. A positive feedback loop between the neurons which are activated by positive and negative values of the input along a particular direction in input space, would cause a sudden increase in the population firing rate. In this case, most of the spikes are a consequence of an internally generated instability.

This kind of activity has been linked to mental disorders, such as schizophrenia (Lewis et al., 2005; Tamminga, 1998; Jackson et al., 2004) and autism (Rubenstein and Merzenich, 2003; Tabuchi et al., 2007), and studied theoretically in spiking balanced networks with random connections (Vogels and Abbott, 2009). The formation of hallucinations could possibly be linked to the mistaken interpretation of internally generated activity, as the external sensory signal. The inability to differentiate between externally and internally generated activity might be mediated by anomalies in the NMDA current (Tamminga, 1998), which could lead to a high excitability of the local circuit (Jackson et al., 2004).

In our network, this kind of an instability could be stabilized with homeostatic thresholds, which would adaptively increase to control the firing rate (Benucci et al., 2013).

7.3 | The scaling and distribution of synaptic efficacies

7.3.1 | The scaling with the number of lateral connections

In our work, we have primarily focused on examining the effect of learning in neurons with fixed firing thresholds, for all connection levels and all population firing rates. With this constraint, the strength of the recurrent connections decays with decreasing connection probability. This is a consequence of the stronger L_1 sparseness constraint, which normalizes the total synaptic input, and can be related to the stabilizing effects of heterosynaptic plasticity (Turrigiano et al., 1998; Lissin et al., 1998; O'Brien et al., 1998; Harvey and Svoboda, 2007; Yger and Gilson, 2015; Bachatene et al., 2015). However, this scaling changes drastically when we introduce adaptive thresholds to offset the increased firing rate in less connected networks.

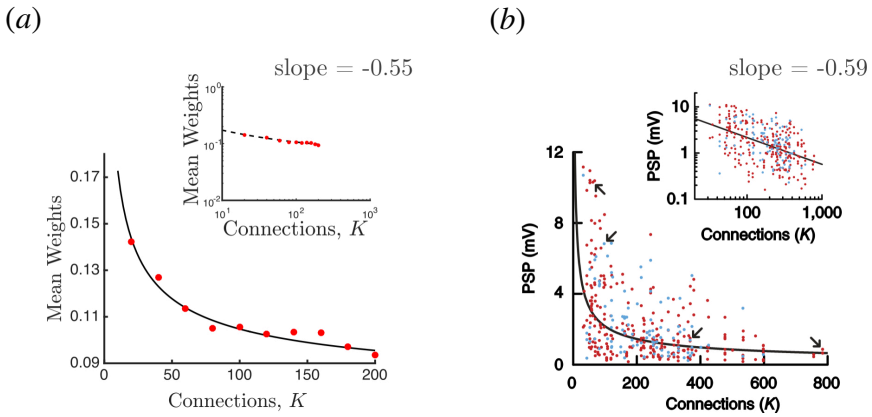


Figure (7.1) The scaling of trained synaptic weights with the number of recurrent inputs. When homeostatic threshold adaptation is added to maintain the firing of individual neurons at a fixed level, the scaling of the weights agrees with experimental observations. **(a)** The mean of the trained recurrent weights (red) as a function of the number of recurrent inputs, pN. Here we temporarily assume the notation K for this number of connections, as in (Barral and Reyes, 2016). If the firing threshold was kept fixed for all connection levels, the mean synaptic weight would increase with the number of recurrent connections. With adaptive thresholds, which keep the neural firing rate constant, this relationship changes dramatically: the strength of the recurrent weights is proportional to one over the square root of the number of connections. **(b)** Scaling of the recurrent weights as a function of the number of recurrent inputs, K, in cultured circuits. Reproduced with the exact caption from (Barral and Reyes, 2016). Amplitudes of unitary EPSPs (red, n = 261) and IPSPs (blue, n = 99) vs. K. Inset: data in log-log scales. Slope of linear fit is -0.59. Individual fits for unitary EPSPs and IPSPs have exponents of -0.60 and -0.52, respectively.

Biological neurons keep their firing rates at a near constant average level, by continuously adapting their firing thresholds. When homeostatic threshold adaptation is added to maintain the firing of individual neurons at a fixed level, the scaling of the trained weights

agrees with experimental observations (Fig. 7.1). For a fixed feedforward input strength, the threshold has to be adaptively increased to keep the firing rate constant, in less connected networks. The mean of the trained recurrent weights then grows with one over the square root of the number of recurrent inputs, as observed in experiments with cultured networks (Barral and Reyes, 2016). In this experiment, the scaling of the synaptic efficacy is mediated by a changing number of boutons from a single presynaptic cell, and not by the changing amplitude of the unitary PSPs.

The experimentally observed scaling is also required for the stability of the asynchronous irregular regime in sparsely connected balanced random networks (Brunel, 2000), and maximizes the number of stored activity patterns (Brunel, 2016).

7.3.2 | The distribution as a function of input dimensionality

The observed distribution of synaptic weights in local circuits is lognormal (Britten et al., 1993; Teramae and Fukai, 2014; Buzsaki and Mizuseki, 2014). Our synaptic rule does not exactly lead to lognormally distributed weights, but the difference between the learned distributions and a lognormal fit decreases with increased input dimensionality (Fig. 7.2). As the dimensionality of the input increases, the weight distributions start to resemble lognormal distributions more, and are in fact better described by lognormal than by normal distributions. This fit might improve further with the following two changes, which are consistent with biology.

First of all, we could change the hard thresholding to soft thresholding, when determining which connections have effectively been pruned. In the numerical simulations of our learning rule, no weights are ever exactly equal to zero. This is a consequence of the numerical discretization, with the weakest weights oscillating in value around zero, but always staying small. Analytically, these weights would be precisely zero. To determine which connections have effectively been pruned in numerical simulations, we implement a hard thresholding rule. All connections with a strength, which is smaller than a small proportion of the firing threshold (we use the cut-off value $T/25$), are considered to have been pruned. If we instead implement a more biologically plausible soft thresholding, we might get a closer match between the distribution of the weakest existing connections and the lognormal fit. This will depend on the exact nature of the noise distribution, which affects the removal of weak connections via long-term depression (Wiegert and Oertner, 2013; Sakuragi et al., 2015).

Secondly, if the firing thresholds of all neurons were not identical, or if there was variability in the feedforward input strength, the distribution of the strongest connections might better resemble a lognormal distribution. In general, a greater heterogeneity in the network might result in a closer match of the learned weight distributions with experimentally observed ones.

We also note that the distribution of the network activity in coordinated spike-coding networks with intermediate connection probability is long-tailed, but not lognormal.

The lognormal distribution is usually associated with a multiplicative effect. Following from the central limit theorem, the product of independent and identically distributed random variables can be approximated by a lognormal distribution (Beaulieu and Xie,

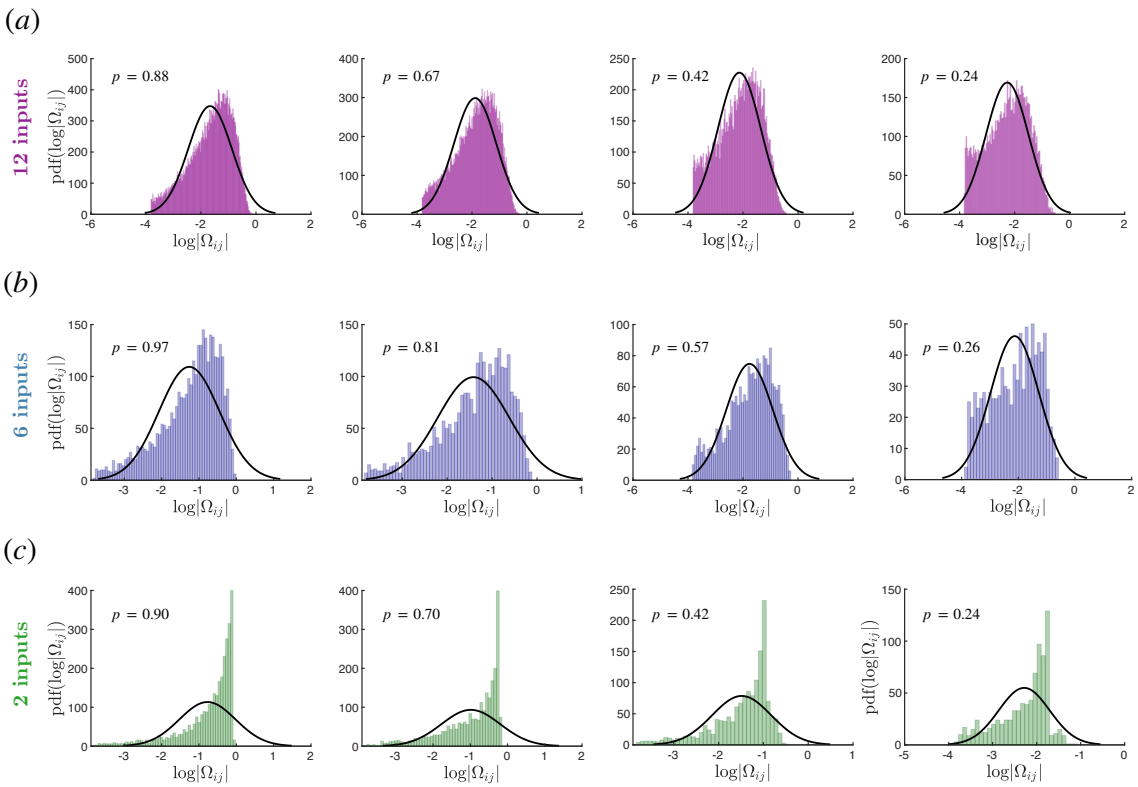


Figure (7.2) The distribution of trained sparse synaptic weights is closer to a lognormal distribution for higher dimensional inputs. The panels from left to right show the distributions of trained recurrent weights in the absence of delays, for decreasing connection probabilities p. The histogram of trained recurrent weights is shown for networks of 200 neurons, which represent 12-dimensional inputs (purple) in **(a)**, and for networks of 60 neurons, which represent 6-dimensional inputs (blue) in **(b)**, and 2-dimensional inputs (green) in **(c)**. The histograms are shown for the logarithms of the absolute values of the recurrent weights, Ω_{ij}, with the pruned connections not shown. The fit to a lognormal distribution is shown in black. Small weights are overrepresented, and the average absolute weight is higher than what would be expected if the weights were lognormally distributed. However, this effect decreases with increasing input dimensionality.

2004; Barakat, 1976; Schwartzman, 2016). We could investigate how this relates to CSC networks and our synaptic rule. Potentially, if the stimulus was high dimensional and usually subsampled just a small portion of the input space, e.g. certain feature combinations are much more common, then the pairwise scalar product of encoding vectors could be approximated with a lognormal distribution. The learning rule would then naturally lead to a sparser connectivity, with weights which are approximately lognormally distributed.

A true comparison of the distribution of the learned synaptic weights with those observed in experiments, would have to be based on anatomical connections. Throughout this thesis, we considered functional connections in networks, which do not respect Dale's law. These functional connections could anatomically be implemented in a network of excitatory neurons, which inhibit similarly tuned pyramidal cells via local interneurons. We address the extension of the learning rule to this case in the following section. An example of anatomically correct synaptic weight distributions is proposed in (Brunel et al., 2004; Brunel, 2016). These studies investigate the learning of feedforward excitatory synapses, which optimize the number of stored patterns in a perceptron model of the Purkinje cell. The optimal synaptic distributions have many silent synapses, and this high level of sparseness is a direct consequence of maximizing storage capacity. These studies link the level of sparseness to the amount of information that the network can store. In our work, we link the level of sparseness to the efficiency with which the network represents information.

7.4 | Extension to E/I networks

The results presented in this thesis extend to networks with separate excitatory (E) and inhibitory (I) neurons, which can be used to fit even more precise models of cortical networks. The impact of smaller recurrent connection probabilities in E-I networks is similar to the one observed in non-Dale networks. The E-I networks represent an input signal with an increased firing rate, and the learned connections are structured and reciprocal, with emerging E-I assemblies between similarly tuned neurons.

The E-I network is arranged so that excitatory neurons receive direct feedforward inputs. Inhibitory neurons receive information about the feedforward inputs indirectly, via E-I lateral connections, which are trained with a Hebbian synaptic rule. This rule is described in (Brendel et al., 2017), and we modify it to account for synaptic delays. All the remaining connections, E-E, I-I, and I-E, are trained using the Lasso regularized synaptic rule with synaptic delays and dynamics. Dale's law is respected, and consequently the E-E and I-I connections naturally have an intermediate connection probability, which is at most 0.5 when the sparseness constraint is not active (Fig. 7.3a,b). Over the course of learning, each inhibitory neuron starts to receive input connections from a subset of similarly tuned excitatory neurons. In this way, inhibitory cells inherit their tuning properties from their presynaptic excitatory neurons, and then learn to cancel the activity in the same excitatory neurons. The network learns a structure in which strongly connected E-I subassemblies efficiently represent specific directions in the input space. With an active sparseness constraint, these subassemblies become more specific and the size of the subassembly decreases (Fig. 7.3c).

The connection probability between neurons depends on their distance in the tuning space. The probability is close to unity between neurons, which have similar tuning prop-

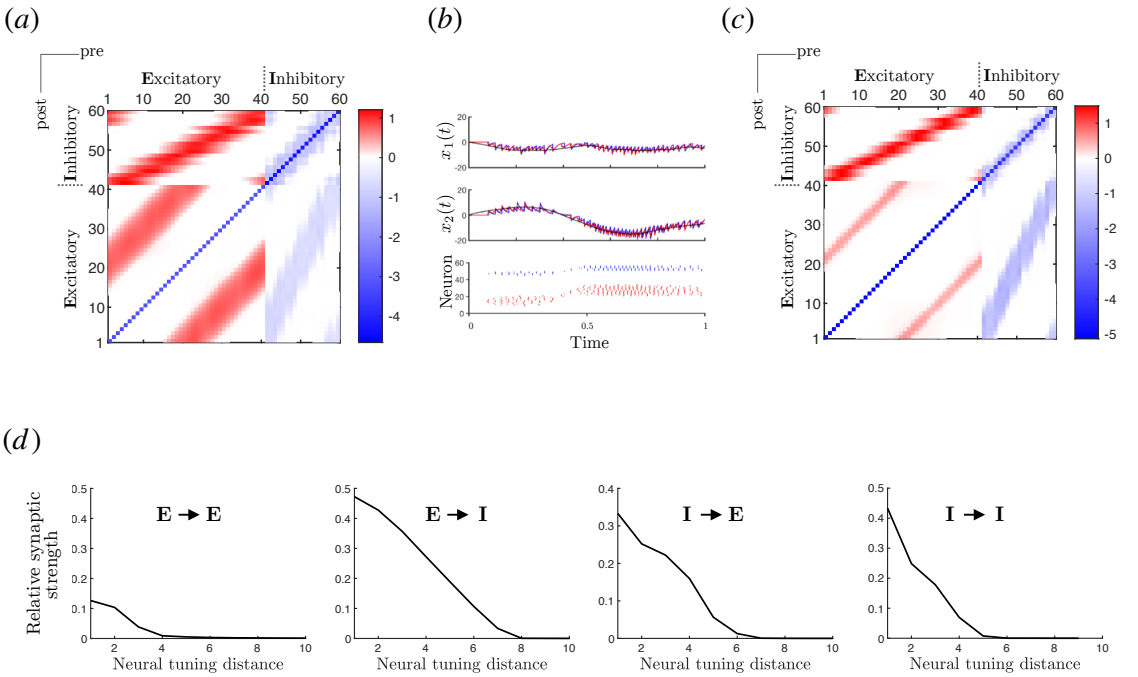


Figure (7.3) Learning sparse networks of separate excitatory and inhibitory populations with synaptic delays. Networks of 40 excitatory (E) and 20 inhibitory (I) neurons were trained to represent 2D inputs. Only the excitatory neurons receive feedforward inputs, and the inhibitory neurons receive information about these inputs indirectly, via E-I lateral connections. A Hebbian feedforward learning rule is used to train the E-I connections, while the E-E, I-I, and I-E connections are learned using the Lasso regularized synaptic rule. Dale's law is respected, so that outgoing E connections are positive or zero, and outgoing I connections are negative or zero. **(a)** Trained recurrent connections with 0.4 ms synaptic delays and no pruning of recurrent connections (the sparseness parameter was set to zero). Only 50% of E-E and I-I connections get learned, because of Dale's law. Over the course of learning, each inhibitory neuron inherits tuning properties from a subset of similarly tuned excitatory neurons, from which it receives recurrent connections. The inhibitory neurons also learn to cancel the activity in these same excitatory neurons. **(b)** Network dynamics of the network shown in (a). Excitatory (red) and inhibitory (blue) reconstruction of the input, and the raster plot. **(c)** Same as (a), but with a non-zero Lasso/sparseness constraint. **(d)** The results from (c) shown as synaptic efficacy over neural tuning difference (difference in the preferred stimulus). The mean of the absolute synaptic weights is normalized by the mean of the firing threshold of the target population. We used adaptive thresholds, and the firing threshold for inhibitory neurons was 20% higher than that of the excitatory neurons. The horizontal axis is ordered so that the postsynaptic neurons, which have the most similar tuning properties to those of the presynaptic neuron, are shown as having the smallest neural tuning distance.

erties. The connection probability then falls off sharply to zero, when the tuning difference increases beyond a level set by the sparseness constraint. On the other hand, the synaptic strength between the connected neurons falls off smoothly with the distance in tuning properties. Overall, the synaptic efficacy resembles the connection probability profiles between the different cell groups in (Levy and Reyes, 2012).

Comparing (Fig. 7.3d) and (Fig. 1.1), shows qualitative agreement. The influence of the presynaptic neuron falls off with distance of the postsynaptic target, with the E-E connections having the smallest efficacy relative to the postsynaptic firing thresholds.

We also comment on the existence of excitatory connections between neurons which represent the same direction in input space, but spike for positive or negative values, respectively. Throughout this thesis, we referred to these neurons as 'oppositely tuned' neurons, as their encoding vectors are 180° apart. However, this name might be misleading, as these neurons represent the same part of the input space. One might think of them as the ON and OFF cells, which code for the increase and decrease of an input variable. In the coordinated spike-coding network, these neurons share strong positive connections.

The behaviour of our synaptic rule is such that it leads to suboptimal solutions if a network is not allowed to learn these excitatory connections. In this case, the network learns a subset of sub-optimal excitatory connections, instead of optimal inhibitory connections, between neurons with overlapping tuning curves. This contradicts the efficiency principle, according to which these neurons should suppress each other's activity. Since the so-called 'oppositely tuned' neurons code for the same direction in the input space, the optimal excitatory connections between them are necessary for efficient signal representation.

7.5 | Future outlook

In this thesis, we started from the hypothesis that biological networks are tuned to achieve maximal efficiency and robustness, while respecting certain biological constraints. We showed that efficient representation can be learned in spiking networks with experimentally observed local connection probabilities and synaptic dynamics. The learned recurrent connections are structured and reciprocal, and the optimal connectivity is somewhere at an intermediate level, neither too sparse nor too dense.

The intermediate optimal connection probability in the presence of synaptic delays echoes the experimentally observed connection probabilities. With a further biological constraint of threshold adaptation, the synapses have an experimentally observed scaling, which depends on the local connection probability. These results collectively support the point of view that neural circuits are highly organized and principled. The local connectivity is structured to sharpen the tuning curves of individual neurons, and spikes are coordinated to increase the efficiency of the network's function. Our work focused on synapses, which mediate accurate signal representation, and could be extended to synapses which mediate other network functions, involving more complex computations.

Recordings from large numbers of neurons are starting to show that neural activity in response to high-dimensional inputs is more complex than anticipated. A detailed examination of the local circuit organization can help us uncover the finer aspects of the principles which govern sensory representation.

A | The Fixed Point of the Learning Rule, when β=1

We give an alternative derivation of the fixed point, (3.18), in section 3.2 of Chapter 3. We assume that the voltage and $\mu r(t)$ term in the synaptic rule, (3.16), are scaled by $\beta = 1$, rather than $\beta = 2$. We then re-examine the fixed point, (3.18), and express it in terms of the linear encoders and linear decoders.

We substitute the expression for $V_i(t)$, given by (3.2), into (3.18). Then, the fixed point equation, (3.18), can be re-expressed as follows:

$$\Big\langle \sum_{k=1}^{K} F_{ik} x_k(t^-) + \sum_{m=1}^{N} \Omega_{im}\big(r_m(t^-) + \delta_{jm}\big) + \mu r_i(t^-) + \kappa \operatorname{sgn}\big(\Omega_{ij}\big) \Big\rangle_j = 0. \tag{A.1}$$

We now use the fact that a spike of neuron j leads to a sudden jump by one of the filtered spike r_j, while as the other filtered spikes are basically unchanged, as they have a small leak constant:

$$r_m(t) = r_m(t^-) + \delta_{jm}. \tag{A.2}$$

We can also assume that the neuron j has not fired recently before its present spike, so that $r_j(t^-) = 0$. This is not a strong assumption in a network with many neurons with distributed spiking. Similarly, the input variable $x_k(t)$ is a leaky integral of the input current, and so $x_k(t^-)$ can be approximated with $x_k(t)$.

We can then rewrite the fixed point condition more succinctly:

$$\Big\langle \sum_{k=1}^{K} F_{ik} x_k(t) + \sum_{m=1}^{N} \big(\Omega_{im} + \mu\delta_{im}\big) r_m(t) + \kappa \operatorname{sgn}\big(\Omega_{ij}\big) \Big\rangle_j = 0. \tag{A.3}$$

We now introduce $\mathbf{r}(t)$ as an $N \times 1$ vector of population filtered spike trains, $r_j(t)$, and $\tilde{\mathbf{o}}(t)$ as an $N \times 1$ vector, which j^{th} element equals one if neuron j spiked at time t, and is zero otherwise. This allows us to rewrite the fixed point condition in matrix form:

$$\Big\langle \mathbf{F}\mathbf{x}(t)\mathbf{r}^T(t) + \big(\boldsymbol{\Omega} + \mu\mathbf{I}\big)\mathbf{r}(t)\mathbf{r}^T(t) + \kappa \operatorname{sgn}(\boldsymbol{\Omega})\, \tilde{\mathbf{o}}(t)\mathbf{r}^T(t) \Big\rangle_{\text{spikes}} = 0, \tag{A.4}$$

where we have multiplied all terms by $\mathbf{r}^T(t)$. We can then write the implicit steady state of the recurrent weights as:

$$\mathbf{\Omega} + \mu\mathbf{I} = -\mathbf{F}\big\langle \mathbf{x}(t)\mathbf{r}^T(t)\big\rangle\big\langle \mathbf{r}(t)\mathbf{r}^T(t)\big\rangle^{-1} - \kappa\big\langle \mathrm{sgn}(\mathbf{\Omega})\tilde{\mathbf{o}}(t)\mathbf{r}^T(t)\big\rangle\big\langle \mathbf{r}(t)\mathbf{r}^T(t)\big\rangle^{-1}, \tag{A.5}$$

where the averaging is always done over spike times. This fixed point expression contains the least squares linear regression solution for the linear decoder, $\mathbf{D}$:

$$\mathbf{D} = \big\langle \mathbf{x}(t)\mathbf{r}^T(t)\big\rangle_{\text{sp.}}\big\langle \mathbf{r}(t)\mathbf{r}^T(t)\big\rangle_{\text{sp.}}^{-1}, \tag{A.6}$$

which we introduced in (3.7), as the optimal decoder.

In contrast to the derivation of the fixed point in the main text in Chapter 3, and of the local balance restoring rule in (Brendel et al., 2017), the derivation here contains the optimal linear decoder, without an additional length regularization (2.36).

We can now write the implicit expression for the fixed point, (A.5), as the negative product of the linear encoder and optimal linear decoder, with an added sparseness regularizer:

$$\mathbf{\Omega} + \mu\mathbf{I} = -\mathbf{F}\mathbf{D} - \kappa\big\langle \mathrm{sgn}(\mathbf{\Omega})\tilde{\mathbf{o}}(t)\mathbf{r}^T(t)\big\rangle_{\text{sp.}}\big\langle \mathbf{r}(t)\mathbf{r}^T(t)\big\rangle_{\text{sp.}}^{-1}. \tag{A.7}$$

The first term on the RHS of (A.7) is a low rank matrix, its rank being given by the dimensionality of the input space. The second term, on the other hand, is a full rank matrix. The greater the sparseness parameter, κ, the further away are the recurrent weights from the low-dimensional structure, which is a signature of optimality in CSC networks.

Bibliography

L. F. Abbott and S. B. Nelson. Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3:1178–1183, 2000.

M. Abeles. Role of the cortical neuron: integrator or coincidence detector? *Isr. J. Med. Sci.*, 18:83–92, 1982.

E. D. Adrian and Y. Zotterman. The impulses produced by sensory nerve-endings: Part II. The response of a single end-organ. *Journal of Physiology*, 61:151–171, 1926. doi: 10.1113/jphysiol.1926.sp002281.

A. Alemi, R. Bourdoukan, and S. Deneve. The brain as an efficient and robust adaptive learner. *Neuron*, 94:969–977, 2017. doi: 10.1016/j.neuron.2017.05.016.

A. Alemi, U. C. Davis, C. K. Machens, and S. Deneve. Learning nonlinear dynamics in efficient, balanced spiking networks using local plasticity rules. *The Thirty-Second AAAI Conference on Artificial Intelligence*, pages 588–595, 2018.

D. J. Amit and N. Brunel. A model of spontaneous activity and local delay activity during delay periods in the cerebral cortex. *Cerebral Cortex*, 7(3):237–252, 1997. doi: 10.1093/cercor/7.3.237.

E. Archer, K. Urs, J. Pillow, and J. H. Macke. Low-dimensional models of neural population activity in sensory cortical circuits. *Advances in Neural Information Processing Systems*, 27:343–351, 2014.

A. Artola and W. Singer. Long-term potentiation and NMDA receptors in rat visual cortex. *Nature*, 330:649–652, 1987. doi: 10.1038/330649a0.

D. Attwell and S. B. Laughlin. An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow and Metabolism*, 21:1133–1145, 2001.

R. Azouz and C. Gray. Cellular mechanisms contributing to response variability of cortical neurons in vivo. *Journal of Neuroscience*, 19:2209–2223, 1999.

L. Bachatene, V. Bharmauria, S. Cattan, N. Chanauria, and J. Rouat. Summation of connectivity strengths in the visual cortex reveals stability of neuronal microcircuits after plasticity. *BMC Neuroscience*, 16:1–11, 2015. doi: 10.1186/s12868-015-0203-1.

R. Barakat. Sums of independent lognormally distributed random variables. *Journal of the Optical Society of America*, 66:211–216, 1976.

J. Barral and A. D. Reyes. Synaptic scaling rule preserves excitatory-inhibitory balance and salient neuronal network dynamics. *Nature Neuroscience*, 19(12):1690–1696, 2016. doi: 10.1038/nn.4415.

D. G. T. Barrett, S. Deneve, and C. K. Machens. Optimal compensation for neuron loss. *Elife*, 5:1–36, 2016. doi: 10.7554/eLife.12454.

B. Bathellier, D. L. Buhl, R. Accolla, and A. Carleton. Dynamic ensemble odor coding in the mammalian olfactory bulb: Sensory information at different timescales. *Neuron*, 57: 586–598, 2008. doi: 10.1016/j.neuron.2008.02.011.

N. Beaulieu and Q. Xie. An optimal lognormal approximation to lognormal sum distributions. *IEEE Trans Veh Technol*, 53:479–489, 2004.

C. C. Bell, V. Z. Han, Y. Sugawara, and K. Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387:278–281, 1997.

A. Benucci, A. B. Saleem, and M. Carandini. Adaptation maintains population homeostasis in primary visual cortex. *Nature Neuroscience*, 16(6):724–729, 2013. doi: 10.1038/nn.3382.

A. Bhattacharya, D. Pati, N. S. Pillai, and D. B. Dunson. Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110(512):1479–1490, 2015. doi: 10.1080/01621459.2014.960967.

G. Bi and M. Poo. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472, 1998.

W. Bialek, R. R. de Ruyter van Steveninck, and D. Warland. Reading a neural code. *Science*, 252:1854–1857, 1991.

E. L. Bienenstock, L. N. Cooper, and P. W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006. ISBN 978-0-387-31073-2.

T. V. P. Bliss and A. R. Gardner-Medwin. Long-lasting potentiation of synaptic transmission in the dentate area of the unanaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232:357–374, 1973.

T. V. P. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232:331–356, 1973.

D. D. Bock, W. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid. Network anatomy and *in vivo* physiology of visual cortical neurons. *Nature*, 471(7337):177–182, 2011. doi: 10.1038/nature09802.

M. Boerlin and S. Deneve. Spike-based population coding and working memory. *PLOS Computational Biology*, 7(2):1–18, 2011. doi: 10.1371/journal.pcbi.1001080.

M. Boerlin, C. K. Machens, and S. Deneve. Predictive coding of dynamical variables in balanced spiking networks. *PLOS Computational Biology*, 9(11), 2013. doi: 10.1371/journal.pcbi.1003258.

R. Bourdoukan and S. Deneve. Enforcing balance allows local supervised learning in spiking recurrent networks. *Advances in Neural Information Processing Systems*, pages 1–9, 2015.

M. Brecht, M. Schneider, B. Sakmann, and T. Margrie. Whisker movements evoked by stimulation of single pyramidal cells in rat motor cortex. *Nature*, 427:704–710, 2004.

W. Brendel, R. Bourdoukan, P. Vertechi, C. K. Machens, and S. Deneve. Learning to represent signals spike by spike. arXiv:1703.03777v2, 2017.

R. Brette. Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in Systems Neuroscience*, 9(151):1–14, 2015. doi: 10.3389/fnsys.2015.00151.

K. Britten, M. N. Shadlen, W. T. Newsome, and A. J. Movshon. Responses of neurons in macaque MT to stochastic motion signals. *Visual Neuroscience*, 10:1157–1169, 1993.

N. Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory neurons. *Computational Neuroscience*, 8:183–208, 2000. doi: 10.1016/S0928-4257(00)01084-6.

N. Brunel. Is cortical connectivity optimized for storing information? *Nature Neuroscience*, 19(5):749–755, 2016. doi: 10.1038/nn.4286.

N. Brunel and X. Wang. What determines the frequency of fast network oscillations with irregular neural discharges? I. Synaptic dynamics and excitation-inhibition balance. *Journal of Neurophysiology*, 90:415–430, 2003. doi: 10.1152/jn.01095.2002.

N. Brunel, V. Hakim, P. Isope, J. P. Nadal, and B. Barbour. Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell. *Neuron*, 43 (5):745–757, 2004.

E. Bullmore and O. Sporns. The economy of brain network organization. *Nature Reviews Neuroscience*, 13(April):336–349, 2012. doi: 10.1038/nrn3214.

A. N. Burkitt. A review of the integrate-and-fire neuron model : I . Homogeneous synaptic input. *Biol Cybern*, 2006. doi: 10.1007/s00422-006-0068-6.

G. Buzsaki and K. Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014. doi: 10.1038/nrn3687.

J. Cafaro and F. Rieke. Noise correlations improve response fidelity and stimulus encoding. *Nature*, 468:964–968, 2010. doi: 10.1038/nature09570.

N. Calaim, F. A. Dehmelt, P. J. Goncalves, and C. K. Machens. Spike coding networks of neurons are robust to perturbations of synapses, cells and populations. in preparation, 2019.

N. Caporale and Y. Dan. Spike timing-dependent plasticity: A Hebbian learning rule. *Annual Reviews Neuroscience*, 31:25–46, 2008. doi: 10.1146/annurev.neuro.31.060407.125639.

S. Celebrini and W. T. Newsome. Neuronal and psychophysical sensitivity to motion signals in extrastriate area MST of the macaque monkey. *Journal of Neuroscience*, 14 (7):4109–4124, 1994.

M. Chalk, B. Gutkin, and S. Deneve. Neural oscillations as a signature of efficient coding in the presence of synaptic delays. *eLife*, 5:1–8, 2016. doi: 10.7554/eLife.13824.

J. K. Chapin and M. A. L. Nicolelis. Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. *J Neurosci Methods.*, 94:121–140, 1999.

S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

S. N. Chettih and C. D. Harvey. Single-neuron perturbations reveal feature-specific competition in V1. *Nature*, 567:334–340, 2019. doi: 10.1038/s41586-019-0997-6.

C. Clopath, L. Busing, E. Vasilaki, and W. Gerstner. Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature Neuroscience*, 13(3):344–352, 2010. doi: 10.1038/nn.2479.

S. F. Cooke and M. F. Bear. How the mechanisms of long-term synaptic potentiation and depression serve experience-dependent plasticity in primary visual cortex. *Phil. Trans. R. Soc. B*, 369, 2014. doi: 10.1098/rstb.2013.0284.

S. Crochet and C. C. H. Petersen. Correlating whisker behavior with membrane potential in barrel cortex of awake mice. *Nature Neuroscience*, 9:608–610, 2006.

S. Crochet, J. F. A. Poulet, Y. Kremer, and C. C. H. Petersen. Synaptic mechanisms underlying sparse coding of active touch. *Neuron*, 69(6):1160–1175, 2011. doi: 10.1016/j.neuron.2011.02.022.

J. A. D. Damour and R. C. Froemke. Inhibitory and excitatory spike-timing-dependent plasticity in the auditory cortex. *Neuron*, 86(2):514–528, 2015. doi: 10.1016/j.neuron.2015.03.014.

Y. Dan and M. Poo. Spike timing-dependent plasticity of neural circuits. *Neuron*, 44: 23–30, 2004.

P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*. The MIT Press, 2001. ISBN 9780262541855.

D. Debanne, B. H. Gahwiler, and S. M. Thompson. Asynchronous pre- and postsynaptic activity induces associative long-term depression in area CA1 of the rat hippocampus in vitro. *Proceedings of the National Academy of Sciences USA*, 91:1148–1152, 1994.

D. Debanne, B. Gahwiler, and S. Thompson. Bidirectional associative plasticity of unitary CA3-CA1 EPSPs in the rat hippocampus in vitro. *J. Neurophysiol.*, 77:2851–2855, 1997.

D. Debanne, H. G. Beat, and S. M. Thompson. Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures. *Journal of Physiology*, 507(1):237–248, 1998.

S. Deneve and C. K. Machens. Efficient codes and balanced networks. *Nature Neuroscience*, 19(3):375–82, 2016. doi: 10.1038/nn.4243.

M. DeWeese and A. Zador. Non-Gaussian membrane potential dynamics imply sparse, synchronous activity in auditory cortex. *Journal of Neuroscience*, 26:12206–12218, 2006.

M. R. Deweese, M. Wehr, and A. M. Zador. Binary spiking in auditory cortex. *Journal of Neuroscience*, 23(21):7940–7949, 2003.

A. L. Dorrn, K. Yuan, A. J. Barker, C. E. Schreiner, and R. C. Froemke. Developmental sensory experience balances cortical excitation and inhibition. *Nature*, 465:932–937, 2010. doi: 10.1038/nature09119.

R. J. Douglas, C. Koch, M. Mahowald, K. A. Martin, and H. H. Suarez. Recurrent excitation in neocortical circuits. *Science*, 269:981–985, 1995.

V. Egger, D. Feldmeyer, and B. Sakmann. Coincidence detection and efficacy changes in synaptic connections between spiny stellate neurons of the rat barrel cortex. *Nature Neuroscience*, 2:1098–1105, 1999.

M. Fagiolini, T. Pizzorusso, N. Berardi, L. Domenici, and L. Maffei. Functional postnatal development of the rat primary visual cortex and the role of visual experience: dark rearing and monocular deprivation. *Vision Res.*, 34:709–720, 1994.

D. E. Feldman. Timing-based LTP and LTD at vertical inputs to layer II/III pyramidal cells in rat barrel cortex. *Neuron*, 27:45–56, 2000.

D. E. Feldman. Synaptic mechanisms for plasticity in neocortex. *Annual Review of Neuroscience*, 32:33–55, 2009. doi: 10.1146/annurev.neuro.051508.135516.

D. E. Feldman. The spike-timing dependence of plasticity. *Neuron*, 75:556–571, 2012. doi: 10.1016/j.neuron.2012.08.001.

E. Fino, J. Glowinski, and L. Venance. Bidirectional activity-dependent plasticity at corticostriatal synapses. *Journal of Neuroscience*, 25(49):11279–11287, 2005. doi: 10.1523/JNEUROSCI.4476-05.2005.

E. Fino, J. Deniau, and L. Venance. Cell-specific spike-timing-dependent plasticity in GABAergic and cholinergic interneurons in corticostriatal rat brain slices. *Journal of Physiology*, 586(1):265–282, 2008. doi: 10.1113/jphysiol.2007.144501.

R. C. Froemke. Plasticity of cortical excitatory-inhibitory balance. *Annual Review of Neuroscience*, 38(1):195–219, 2015. doi: 10.1146/annurev-neuro-071714-034002.

R. C. Froemke, M. M. Merzenich, and C. E. Schreiner. A synaptic memory trace for cortical receptive field plasticity. *Nature*, 450:425–429, 2007. doi: 10.1038/nature06289.

E. Gal, M. London, A. Globerson, S. Ramaswamy, M. W. Reimann, E. Muller, H. Markram, and I. Segev. Rich cell-type-specific network topology in neocortical microcircuitry. *Nature Neuroscience*, 20(7):1004–1013, 2017. doi: 10.1038/nn.4576.

P. Gao and S. Ganguli. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current Opinion in Neurobiology*, 32:148–155, 2015. doi: 10.1016/j.conb.2015.04.003.

P. Gao, E. Trautmann, B. Yu, G. Santhanam, and S. Ryu. A theory of multineuronal dimensionality, dynamics and measurement. *Biorxiv*, 2017. doi: 10.1101/214262.

L. J. Gentet, M. Avermann, F. Matyas, J. F. Staiger, and C. C. H. Petersen. Membrane potential dynamics of GABAergic neurons in the barrel cortex of behaving mice. *Neuron*, 65(3):422–435, 2010. doi: 10.1016/j.neuron.2010.01.006.

W. Gerstner and W. Kistler. *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

W. Gerstner, R. Kempter, J. L. V. Hemmen, and H. Wagnert. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78, 1996.

A. Gilra and W. Gerstner. Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *eLife*, 6:1–37, 2017. doi: 10.7554/eLife.28295.

Y. Goto and P. O'Donnell. Network synchrony in the nucleus accumbens *in vivo*. *Journal of Neuroscience*, 21:4498–4504, 2001.

M. Graupner and A. D. Reyes. Synaptic input correlations leading to membrane potential decorrelation of spontaneous activity in cortex. *Journal of Neuroscience*, 33(38):15075–15085, 2013. doi: 10.1523/JNEUROSCI.0347-13.2013.

B. Gustafsson, H. Wigstrom, W. C. Abraham, and Y. Huang. Long-term potentiation in the hippocampus using depolarizing current pulses as the conditioning stimulus to single volley synaptic potentials. *Journal of Neuroscience*, 7:774–780, 1987.

B. Haider, A. Duque, A. R. Hasenstaub, and D. A. McCormick. Neocortical network activity *in vivo* is generated through a dynamic balance of excitation and inhibition. *Journal of Neuroscience*, 26(17):4535–4545, 2006. doi: 10.1523/JNEUROSCI.5297-05.2006.

V. Z. Han, K. Grant, and C. C. Bell. Reversible associative depression and nonassociative potentiation at a parallel fiber synapse. *Neuron*, 27:611–622, 2000.

A. Harauzov, M. Spolidoro, G. Dicristo, R. D. Pasquale, L. Cancedda, T. Pizzorusso, A. Viegi, N. Berardi, and L. Maffei. Reducing intracortical inhibition in the adult visual cortex promotes ocular dominance plasticity. *Journal of Neuroscience*, 30(1):361–371, 2010. doi: 10.1523/JNEUROSCI.2233-09.2010.

K. D. Harris and T. D. Mrsic-Flogel. Cortical connectivity and sensory coding. *Nature*, 503(7474):51–58, 2013. doi: 10.1038/nature12654.

C. Harvey, F. Collman, D. Dombeck, and D. Tank. Intracellular dynamics of hippocampal place cells during virtual navigation. *Nature*, 461:941–946, 2009.

C. D. Harvey and K. Svoboda. Locally dynamic synaptic learning rules in pyramidal neuron dendrites. *Nature*, 450:1195–1202, 2007. doi: 10.1038/nature06416.

D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, 1949.

G. Hennequin, E. J. Agnes, and T. P. Vogels. Inhibitory plasticity: Balance, control, and codependence. *Annual Review of Neuroscience*, 40:557–579, 2017.

A. V. Hill. Excitation and accommodation in nerve. *Proc R Soc B*, 119:305–355, 1936.

G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and Helmholtz free energy. *Advances in neural information processing systems*, 6:3–10, 1994.

J. C. Hirsch and F. Crepel. Postsynaptic calcium is necessary for the induction of LTP and LTD of monosynaptic EPSPs in prefrontal neurons: An *in vitro* study in the rat. *Synapse*, 10(2):173–175, 1992.

S. B. Hofer, H. Ko, B. Pichler, J. Vogelstein, H. Ros, H. Zeng, E. Lein, N. A. Lesica, and T. D. Mrsic-Flogel. Differential connectivity and response dynamics of excitatory and inhibitory neurons in visual cortex. *Nature Neuroscience*, 14(8):1045–52, 2011. doi: 10.1038/nn.2876.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, 79:2554–2558, 1982. doi: 10.1073/pnas.79.8.2554.

J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA*, 81: 3088–3092, 1984. doi: 10.1073/pnas.81.10.3088.

J. J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(2):33–36, 1995.

T. Hromadka, M. R. Deweese, and A. M. Zador. Sparse representation of sounds in the unanesthetized auditory cortex. *PLOS Biology*, 6(1):124–137, 2008. doi: 10.1371/journal.pbio.0060016.

D. H. Hubel and T. N. Wiesel. The period of susceptibility to the physiological effects of unilateral eye closure in kittens. *Journal of Physiology*, 206:419–436, 1970.

D. Huh and T. J. Sejnowski. Gradient descent for spiking neural networks. *Advances in Neural Information Processing Systems*, 31:1–11, 2018.

J. Iglesias, J. Eriksson, F. Grize, M. Tomassini, and A. Villa. Dynamics of pruning in simulated large-scale spiking neural networks. *Biosystems*, 79:11–20, 2005.

A. Iriki, C. Pavlides, A. Keller, and H. Asanuma. Long-term potentiation in the motor cortex. *Science*, 245:1385–1387, 1989. doi: 10.1126/science.2551038.

J. S. Isaacson and M. Scanziani. How inhibition shapes cortical activity. *Neuron*, 72(2): 231–243, 2011. doi: 10.1016/j.neuron.2011.09.027.

E. M. Izhikevich, J. A. Gally, and G. M. Edelman. Spike-timing dynamics of neuronal groups. *Cereb. Cortex*, 14:933–944, 2004.

M. Jackson, H. Homayoun, and B. Moghaddam. NMDA receptor hypofunction produces concomitant firing rate potentiation and burst activity reduction in the prefrontal cortex. *Proceedings of the National Academy of Sciences USA*, 101:8467–8472, 2004.

W. James. *The Principles of Psychology*. New York: Henry Holt & Co., 1890.

X. Jiang, S. Shen, C. R. Cadwell, P. Berens, F. Sinz, A. S. Ecker, S. Patel, and A. S. Tolias. Principles of connectivity among morphologically defined cell types in adult neocortex. *Science*, 350(6264):1–10, 2015. doi: 10.1126/science.aac9462.

A. M. Kerlin, M. L. Andermann, V. K. Berezovskii, and R. C. Reid. Broadly tuned response properties of diverse inhibitory neuron subtypes in mouse visual cortex. *Neuron*, 67(5): 858–871, 2010. doi: 10.1016/j.neuron.2010.08.002.

A. G. Khan, J. Poort, A. Chadwick, A. Blot, M. Sahani, T. D. Mrsic-Flogel, and S. B. Hofer. Distinct learning-induced changes in stimulus selectivity and interactions of GABAergic interneuron classes in visual cortex. *Nature Neuroscience*, 21:851–859, 2018. doi: 10.1038/s41593-018-0143-z.

H. Ko, L. Cossell, C. Baragli, J. Antolik, C. Clopath, S. B. Hofer, and T. D. Mrsic-Flogel. The emergence of functional microcircuits in visual cortex. *Nature*, 496(7443):96–100, 2013. doi: 10.1038/nature12015.

A. Kumar, S. Rotter, and A. Aertsen. Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding. *Nature Reviews Neuroscience*, 11: 615–627, 2010. doi: 10.1038/nrn2886.

G. Lajoie, K. K. Lin, and E. Shea-Brown. Chaos and reliability in balanced spiking networks with temporal drive. *Phys Rev E Stat Nonlin Soft Matter Phys*, 87(5):1–28, 2013. doi: 10.1103/PhysRevE.87.052901.

L. Lapicque. Recherches quantitatives sur l'excitation electrique des nerfs traitée comme une polarization. *J Physiol Pathol Gen (Paris)*, 9:620–635, 1907.

J. Leger, E. Stern, A. Aertsen, and D. Heck. Synaptic integration in rat frontal cortex shaped by network activity. *J Neurophysiol*, 93:281–293, 2005.

S. LeVay. Synaptic organization of claustral and geniculate afferents to the visual cortex of the cat. *Journal of Neuroscience*, 6:3564–3575, 1986.

R. B. Levy and A. D. Reyes. Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex. *Journal of Neuroscience*, 32(16):5609–5619, 2012. doi: 10.1523/JNEUROSCI.5158-11.2012.

W. B. Levy and D. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8:791–797, 1983.

D. Lewis, T. Hashimoto, and D. Volk. Cortical inhibitory neurons and schizophrenia. *Nature Reviews Neuroscience*, 6:312–324, 2005.

J. Lisman and N. Spruston. Postsynaptic depolarization requirements for LTP and LTD: a critique of spike timing-dependent plasticity. *Nature Neuroscience*, 8(7):839–841, 2005.

J. Lisman and N. Spruston. Questions about STDP as a general model of synaptic plasticity. *Frontiers in Synaptic Neuroscience*, 2:1–5, 2010. doi: 10.3389/fnsyn.2010.00140.

D. V. Lissin, S. N. Gomperts, R. C. Carroll, C. W. Christine, D. Kalman, M. Kitamura, S. Hardy, R. A. Nicoll, R. C. Malenka, and M. von Zastrow. Activity differentially regulates the surface expression of synaptic AMPA and NMDA glutamate receptors. *Proceedings of the National Academy of Sciences USA*, 95:7097–7102, 1998. doi: 10.1073/pnas.95.12.7097.

A. Litwin-Kumar, K. D. Harris, R. Axel, H. Sompolinsky, and L. Abbott. Optimal degrees of synaptic connectivity. *Neuron*, 93(5):1153–1164, 2017. doi: 10.1016/j.neuron.2017.01.030.

J. Lu, C. Li, J. Zhao, M. Poo, and X. Zhang. Spike-timing-dependent plasticity of neocortical excitatory synapses on inhibitory interneurons depends on target cell type. *Journal of Neuroscience*, 27(36):9711–9720, 2007. doi: 10.1523/JNEUROSCI.2513-07.2007.

S. Luccioli, S. Olmi, A. Politi, and A. Torcini. Collective dynamics in sparse networks. *Phys. Rev. Lett.*, 109:1–5, 2012. doi: 10.1103/PhysRevLett.109.138103.

J. C. Magee and D. Johnston. A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons. *Science*, 275:209–213, 1997.

H. Markram, J. Lu, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–216, 1997.

B. J. Marlin, M. Mitre, J. A. D'amour, M. V. Chao, and R. C. Froemke. Oxytocin enables maternal behaviour by balancing cortical inhibition. *Nature*, 520(7548):499–504, 2015. doi: 10.1038/nature14402.

D. Marr. *Vision*. New York, NY: W.H.Freeman & Co Ltd., first edition, 1982.

F. Mastrogiuseppe and S. Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99:609–623, 2018. doi: 10.1016/j.neuron.2018.07.003.

G. Mongillo, S. Rumpel, and Y. Loewenstein. Inhibitory connectivity defines the realm of excitatory plasticity. *Nature Neuroscience*, 21(10):1463–1470, 2018. doi: 10.1038/s41593-018-0226-x.

A. Morrison, A. Aertsen, and M. Diesmann. Spike-timing-dependent plasticity in balanced random networks. *Neural Computation*, 19:1437–1467, 2007.

B. K. Murphy and K. D. Miller. Balanced amplification: A new mechanism of selective amplification of neural activity patterns. *Neuron*, 61(4):635–648, 2009. doi: 10.1016/j.neuron.2009.02.005.

W. Murray, P. Gill, and M. Wright. *Practical Optimization*. Academic Press, New York, 1981.

W. T. Newsome, M. N. Shadlen, E. Zohary, K. H. Britten, and J. Movshon. Visual motion: Linking neuronal activity to psychophysical performance. In M. S. Gazzaniga, editor, *The cognitive neurosciences*. MIT Press, 1995.

R. J. O'Brien, S. Kamboj, M. D. Ehlers, K. R. Rosen, G. D. Fischbach, R. L. Huganir, and H. Hughes. Activity-dependent modulation of synaptic AMPA receptor accumulation. *Neuron*, 21:1067–1078, 1998.

E. Oja. A simplified neuron model as a principal component analyzer. *J. Math. Biol.*, 15: 267–273, 1982.

M. Okun and I. Lampl. Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities. *Nature Neuroscience*, 11(5):535–7, 2008. doi: 10.1038/nn.2105.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. doi: 10.1038/381607a0.

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

S. Ostojic. Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons. *Nature Neuroscience*, 17(4):594–600, 2014. doi: 10.1038/nn.3658.

R. Perin, T. K. Berger, and H. Markram. A synaptic organizing principle for cortical neuronal groups. *Proceedings of the National Academy of Sciences USA*, 108(13):5419–5424, 2011. doi: 10.1073/pnas.1016051108.

D. H. Perkel, G. L. Gerstein, and G. P. Moore. Neuronal spike trains and stochastic point processes. *Biophysical Journal*, 7(4):391–418, 1967. doi: 10.1016/S0006-3495(67)86596-2.

A. Peters and B. R. Payne. Numerical relationships between geniculocortical afferents and pyramidal cell modules in cat primary visual cortex. *Cereb Cortex*, 3:69–78, 1993.

A. Peters, B. R. Payne, and J. Budd. A numerical analysis of the geniculocortical input to striate cortex in the monkey. *Cereb Cortex*, 4:215–229, 1994.

C. K. Pfeffer, M. Xue, M. He, Z. J. Huang, and M. Scanziani. Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nature Neuroscience*, 16(8):1068–1076, 2013. doi: 10.1038/nn.3446.

J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454:995–1000, 2008. doi: 10.1038/nature07140.

C. Poo and J. S. Isaacson. Odor representations in olfactory cortex: "sparse" coding, global inhibition, and oscillations. *Neuron*, 62(6):850–861, 2009. doi: 10.1016/j.neuron.2009.05.022.

R. P. N. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1): 79–87, 1999. doi: 10.1038/4580.

A. Renart, J. de la Rocha, P. Bartho, L. Hollender, N. Parga, A. D. Reyes, and K. D. Harris. The asynchronous state in cortical circuits. *Science*, 327(5965):587–90, 2010. doi: 10.1126/science.1179850.

T. Requarth and N. B. Sawtell. Neural mechanisms for filtering self-generated sensory signals in cerebellum-like circuits. *Current Opinion in Neurobiology*, 21(4):602–608, 2011. doi: 10.1016/j.conb.2011.05.031.

F. Rieke, D. Warland, R. R. de Ruyter van Steveninck, and W. Bialek. *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press, 1997.

P. D. Roberts, T. K. Leen, B. A. Carlson, and J. Pfister. Anti-Hebbian spike-timing-dependent plasticity and adaptive sensory processing. *Frontiers in Computational Neuroscience*, 4:1–11, 2010. doi: 10.3389/fncom.2010.00156.

J. D. Rocha, C. Marchetti, M. Schiff, and A. D. Reyes. Linking the response properties of cells in auditory cortex with network architecture: Cotuning versus lateral inhibition. *Journal of Neuroscience*, 28(37):9151–9163, 2008. doi: 10.1523/JNEUROSCI.1789-08.2008.

C. Rossant, S. Leijon, A. K. Magnusson, and R. Brette. Sensitivity of noisy neurons to coincident inputs. *Journal of Neuroscience*, 31(47):17193–17206, 2011. doi: 10.1523/JNEUROSCI.2482-11.2011.

A. Roth and M. van Rossum. *Computational Modeling Methods for Neuroscientists*. The MIT Press, 2009.

J. Rubenstein and M. Merzenich. Model of autism: increased ratio of excitation/inhibition in key neural systems. *Genes Brain Behav.*, 2:255–267, 2003.

P. Rupprecht and R. W. Friedrich. Precise synaptic balance in the zebrafish homolog of olfactory cortex. *Neuron*, 100(3):669–683, 2018. doi: 10.1016/j.neuron.2018.09.013.

S. Sakuragi, K. Tominaga-Yoshino, and A. Ogura. Dendritic spine dynamics leading to spine elimination after repeated inductions of LTD. *Scientific Reports*, 5(7707):1–6, 2015. doi: 10.1038/srep07707.

A. Sale, N. Berardi, M. Spolidoro, L. Baroncelli, and L. Maffei. GABAergic inhibition in visual cortical plasticity. *Frontiers in Cellular Neuroscience*, 4:1–6, 2010. doi: 10.3389/fncel.2010.00010.

D. B. Salkoff, E. Zagha, and D. A. McCormick. Synaptic mechanisms of tight spike synchrony at Gamma frequency in cerebral cortex. *Journal of Neuroscience*, 35(28): 10236–10251, 2015. doi: 10.1523/JNEUROSCI.0828-15.2015.

C. Savin and S. Deneve. Spatio-temporal representations of uncertainty in spiking neural networks. *Advances in Neural Information Processing Systems*, 27:1–9, 2014.

B. Scholl, J. J. Pattadkal, G. A. Dilly, N. J. Priebe, and B. V. Zemelman. Local integration accounts for weak selectivity of mouse neocortical parvalbumin interneurons. *Neuron*, 87(2):424–436, 2015. doi: 10.1016/j.neuron.2015.06.030.

A. Schwartzman. Lognormal distributions and geometric averages of symmetric positive definite matrices. *Int Stat Rev*, 84(3):456–486, 2016. doi: 10.1111/insr.12113.

M. A. Schwemmer, A. L. Fairhall, S. Deneve, and E. T. Shea-Brown. Constructing precisely computing networks with biophysical spiking neurons. *Journal of Neuroscience*, 35(28): 10112–10134, 2015. doi: 10.1523/JNEUROSCI.4951-14.2015.

H. S. Seung and H. Sompolinsky. Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences USA*, 90:10749–10753, 1993.

M. N. Shadlen and W. T. Newsome. Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, 4(4):569–579, 1994. doi: 10.1016/0959-4388(94)90059-0.

M. N. Shadlen and W. T. Newsome. The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *Journal of Neuroscience*, 18(10):3870–3896, 1998.

Y. R. Shao, B. R. Isett, T. Miyashita, J. Chung, O. Pourzia, and R. J. Gasperini. Plasticity of recurrent L2/3 inhibition and Gamma oscillations by whisker experience. *Neuron*, 80: 210–222, 2013. doi: 10.1016/j.neuron.2013.07.026.

R. Shapley, M. Hawken, and D. Xing. The dynamics of visual responses in the primary visual cortex. *Progress in Brain Research*, 165:21–32, 2007.

Y. Shu, A. Hasenstaub, and D. A. Mccormick. Turning on and off recurrent balanced cortical activity. *Nature*, 423(6937):288–293, 2003. doi: 10.1038/nature01614.1.

J. M. Sjostrom, G. G. Turrigiano, and S. B. Nelson. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32:1149–1164, 2001.

W. R. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *Journal of Neuroscience*, 13(1):334–50, 1993.

H. Sompolinsky and R. Shapley. New perspectives on the mechanisms for orientation selectivity. *Curr Opin Neurobiol*, 7:514–522, 1997.

S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature*, 3(9):919–926, 2000.

S. Song, J. Sjostrom, M. Reigl, S. Nelson, and D. B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLOS Biology*, 3(3):507–519, 2005. doi: 10.1371/journal.pbio.0030068.

E. Stern, A. Kincaid, and C. Wilson. Spontaneous subthreshold membrane potential fluctuations and action potential variability of rat corticostriatal and striatal neurons *in vivo*. *J Neurophysiol*, 77:1697–1715, 1997.

C. Stringer, M. Pachitariu, N. Steinmetz, M. Carandini, and K. D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019. doi: 10.1038/s41586-019-1346-5.

K. Tabuchi, J. Blundell, M. R. Etherton, R. E. Hammer, X. Liu, C. M. Powell, and T. C. Sudhof. A neuroligin-3 mutation implicated in autism increases inhibitory synaptic transmission in mice. *Science*, 318(5847):71–76, 2007.

C. Tamminga. Schizophrenia and glutamatergic transmission. *Crit. Rev. Neurobiol.*, 12: 21–36, 1998.

A. Y. Tan, S. Andoni, and N. J. Priebe. A spontaneous state of weakly correlated synaptic excitation and inhibition in visual cortex. *Neuroscience*, 247:364–375, 2013. doi: 10.1016/j.neuroscience.2013.05.037.

J. Teramae and T. Fukai. Computational implications of lognormally distributed synaptic weights. *Proceedings of the IEEE*, 102(4):500–512, 2014.

D. Thalmeier, M. Uhlmann, and H. J. Kappen. Learning universal computations with spikes. *PLOS Computational Biology*, 12(16):1–29, 2016. doi: 10.1371/journal.pcbi.1004895.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58: 267–288, 1996.

D. J. Tolhurst, J. A. Movshon, and A. F. Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res.*, 23(8):775–785, 1982.

G. G. Turrigiano, K. R. Leslie, N. S. Desai, L. C. Rutherford, and S. B. Nelson. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391:892–896, 1998.

C. van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274:1724–1726, 1996. doi: 10.1126/science.274.5293.1724.

P. Vertechi, W. Brendel, and C. K. Machens. Unsupervised learning of an efficient short-term memory network. *Advances in Neural Information Processing Systems*, 27:3653–3661, 2014.

T. P. Vogels and L. F. Abbott. Signal propagation and logic gating in networks of integrate-and-fire neurons. *Journal of Neuroscience*, 25(46):10786–10795, 2005. doi: 10.1523/JNEUROSCI.3508-05.2005.

T. P. Vogels and L. F. Abbott. Gating multiple signals through detailed balance of excitation and inhibition in spiking networks. *Nature Neuroscience*, 12(4):483–491, 2009. doi: 10.1038/nn.2276.

T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–73, 2011. doi: 10.1126/science.1211095.

D. K. Warland, P. Reinagel, and M. Meister. Decoding visual information from a population of retinal ganglion cells. *J Neurophysiol.*, 78:2336–2350, 1997.

M. Wehr and A. M. Zador. Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature*, 426(6965):860–863, 2003.

J. S. Wiegert and T. G. Oertner. Long-term depression triggers the selective elimination of weakly integrated synapses. *Proceedings of the National Academy of Sciences USA*, 2013:4510–4519, 2013. doi: 10.1073/pnas.1315926110.

C. Wilson and Y. Kawaguchi. The origins of two-state spontaneous membrane potential fluctuations of neostriatal spiny neurons. *Journal of Neuroscience*, 16:2397–2410, 1996.

G. K. Wu, R. Arbuckle, B. Liu, H. W. Tao, and L. I. Zhang. Lateral sharpening of cortical frequency tuning by approximately balanced inhibition. *Neuron*, 58:132–143, 2008. doi: 10.1016/j.neuron.2008.01.035.

M. Xue, B. V. Atallah, and M. Scanziani. Equalizing excitation-inhibition ratios across visual cortical neurons. *Nature*, 511(7511):596–600, 2014. doi: 10.1038/nature13321.

H. Yao, L. Shi, F. Han, H. Gao, and Y. Dan. Rapid learning in cortical coding of visual scenes. *Nature Neuroscience*, 10(6):772–778, 2007. doi: 10.1038/nn1895.

P. Yger and M. Gilson. Models of metaplasticity: A review of concepts. *Frontiers in Computational Neuroscience*, 9:1–14, 2015. doi: 10.3389/fncom.2015.00138.

Y. Yoshimura, J. Dantzker, and E. Callaway. Excitatory cortical neurons form fine-scale functional networks. *Nature*, 433:868–873, 2005.

J. Yu and D. Ferster. Membrane potential synchrony in primary visual cortex during sensory stimulation. *Neuron*, 68(6):1187–1201, 2010. doi: 10.1016/j.neuron.2010.11.027.

F. Zenke, W. Gerstner, and S. Ganguli. The temporal paradox of Hebbian learning and homeostatic plasticity. *Current Opinion in Neurobiology*, 43:166–176, 2017. doi: 10.1016/j.conb.2017.03.015.

L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris, and M. Poo. A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395:37–44, 1998.

Y. Zhao and T. Tzounopoulos. Physiological activation of cholinergic inputs controls associative synaptic plasticity via modulation of endocannabinoid signaling. *Journal of Neuroscience*, 31(9):3158–3168, 2011. doi: 10.1523/JNEUROSCI.5303-10.2011.

RÉSUMÉ

La complexité de l'entrée sensorielle est parallèle à la complexité de sa représentation dans l'activité neurale des systèmes biologiques. Partant de l'hypothèse que les réseaux biologiques sont réglés pour atteindre une efficacité et une robustesse maximales, nous étudions comment une représentation efficace peut être réalisée dans des réseaux avec des probabilités de connexion locale et une dynamique synaptique observée de manière expérimentale. Nous développons une règle synaptique locale régularisée de type Lasso, qui optimise le nombre et l'efficacité des connexions récurrentes. Les connexions qui affectent le moins le rendement sont élaguées, et la force des connexions restantes est optimisée pour une meilleure représentation du signal. Notre théorie prédit que la probabilité de connexion locale détermine le compromis entre le nombre de potentiels d'action de la population et le nombre de connexions synaptiques qui sont développées et maintenues dans le réseau. Les réseaux plus faiblement connectés représentent des signaux avec des fréquences de déclenchement plus élevées que ceux avec une connectivité plus dense. La variabilité des probabilités de connexion observées dans les réseaux biologiques pourrait alors être considérée comme une conséquence de ce compromis et serait liée à différentes conditions de fonctionnement des circuits. Les connexions récurrentes apprises sont structurées et la plupart des connexions sont réciproques. La dimensionnalité des poids synaptiques récurrents peut être déduite de la probabilité de connexion du réseau et de la dimensionnalité du stimulus. La connectivité optimale d'un réseau avec des délais synaptiques se situe quelque part à un niveau intermédiaire, ni trop faible ni trop dense. De plus, lorsque nous ajoutons une autre contrainte biologique comme la régulation des taux de décharge par adaptation, notre règle d'apprentissage conduit à une mise à l'échelle observée de manière expérimentale des poids synaptiques. Nos travaux soutiennent l'idée que les micro-circuits biologiques sont hautement organisés et qu'une étude détaillée de leur organisation nous aidera à découvrir les principes de la représentation sensorielle.

MOTS CLÉS

réseau de potentiels d'action; codage efficace; apprentisage

ABSTRACT

The complexity of sensory input is paralleled by the complexity of its representation in the neural activity of biological systems. Starting from the hypothesis that biological networks are tuned to achieve maximal efficiency and robustness, we investigate how efficient representation can be accomplished in networks with experimentally observed local connection probabilities and synaptic dynamics. We develop a Lasso regularized local synaptic rule, which optimizes the number and efficacy of recurrent connections. The connections that impact the efficiency the least are pruned, and the strength of the remaining ones is optimized for efficient signal representation. Our theory predicts that the local connection probability determines the trade-off between the number of population spikes and the number of recurrent synapses, which are developed and maintained in the network. The more sparsely connected networks represent signals with higher firing rates than those with denser connectivity. The variability of observed connection probabilities in biological networks could then be seen as a consequence of this trade-off, and related to different operating conditions of the circuits. The learned recurrent connections are structured, with most connections being reciprocal. The dimensionality of the recurrent weights can be inferred from the network's connection probability and the dimensionality of the feedforward input. The optimal connectivity of a network with synaptic delays is somewhere at an intermediate level, neither too sparse nor too dense. Furthermore, when we add another biological constraint, adaptive regulation of firing rates, our learning rule leads to an experimentally observed scaling of the recurrent weights. Our work supports the notion that biological micro-circuits are highly organized and principled. A detailed examination of the local circuit organization can help us uncover the finer aspects of the principles which govern sensory representation.

KEYWORDS

spiking networks; efficient coding; learning