Learning and coding in biological neural networks

A thesis presented

by

Ila Rani Fiete

to

The Department of Physics

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Physics

Harvard University

Cambridge, Massachusetts

December 2003

for my parents

Mummy–Babuji

Abstract

Learning and coding in biological neural networks

Ila Rani Fiete

Thesis Advisor: Professor H. S. Seung

Committee Members: Professors D. S, Fisher (chair), C. M. Marcus and V. N. Murthy

How can large groups of neurons that locally modify their activities learn to collectively perform a desired task? Do studies of learning in small networks tell us anything about learning in the fantastically large collection of neurons that make up a vertebrate brain? What factors do neurons optimize by encoding sensory inputs or motor commands in the way they do? In this thesis I present a collection of four theoretical works: each of the projects was motivated by specific constraints and complexities of biological neural networks, as revealed by experimental studies; together, they aim to partially address some of the central questions of neuroscience posed above.

We first study the role of sparse neural activity, as seen in the coding of sequential commands in a premotor area responsible for birdsong. We show that the sparse coding of temporal sequences in the songbird brain can, in a network where the feedforward plastic weights must translate the sparse sequential code into a time-varying muscle code, facilitate learning by minimizing synaptic interference.

Next, we propose a biologically plausible synaptic plasticity rule that can perform goal-directed learning in recurrent networks of voltage-based spiking neurons that interact through conductances. Learning is based on the correlation of noisy local activity with a global reward signal; we prove that this rule performs stochastic gradient ascent on the reward. Thus, if the reward signal quantifies network performance on some desired task, the plasticity rule provably drives goal-directed learning in the network.

To assess the convergence properties of the learning rule, we compare it with a known example of learning in the brain. Song-learning in finches is a clear example of a learned behavior, with detailed available neurophysiological data. With our learning rule, we train an anatomically accurate model birdsong network that drives a sound source to mimic an actual zebrafinch song. Simulation and theoretical results on the scalability of this rule show that learning with stochastic gradient ascent may be adequately fast to explain learning in the bird.

Finally, we address the more general issue of the scalability of stochastic gradient learning on quadratic cost surfaces in linear systems, as a function of system size and task characteristics, by deriving analytical expressions for the learning curves.

Contents

List of Figures

Acknowledgments

For giving me the chance to enter into the vast world in neuroscience, and for being my superb and incisive guide, I am deeply indebted to my advisor Sebastian Seung. His openness to new directions, uncanny ability to simultaneously and easily grasp both minute details and panoramic vistas of any issue, and his remarkable clarity of thought and exposition, are qualities I can only aspire to. I am grateful to him for allowing me much independence in work and travel, yet also providing me with guidance and advice whenever I needed it. Above all else, I thank Sebastian for being not just an inspiring and generous advisor, but a wonderful friend.

I am very lucky to have Daniel Fisher as my Harvard advisor, and am grateful to him for taking an active interest in my work, and for many helpful and interesting discussions. He made it possible for me to explore new directions: without his interest and involvement, I am sure I wouldn't be able to present this (gasp!) blatantly neuroscience thesis for a degree in Physics. Daniel has persistently kept me on track, and given me excellent and timely advice: to be the recipient of his very sincere concern for my scientific development, I consider myself privileged.

I am fortunate to have had several fantastic collaborators: many thanks go to Richard Hahnloser, Michale Fee, and Alexay Kozhevnikov for exciting conversations and for introducing me to the study of birdsong. I have benefited greatly from working with Richard, Michale, and Alex, and learning first-hand about their technically beautiful experiments. Their enthusiasm about their own work, and about neuroscience in general, is infectious. I won't forget the many light moments we shared over meals at Bell Labs, the Socity for Neuroscience meetings, and at MIT. I thank David Tank, Guy Major, and Emre

Aksay: extensive discussions and the opportunity to watch Guy and Emre perform some of the oculomotor experiments made my stay at Bell Labs in the summer of 2001 most educational and enjoyable.

I am grateful to Venkatesh Murthy and Charles Marcus for being on my thesis committee. I thank Markus Meister for letting me participate in his group meetings, and for insightful comments and discussions. Through his thoughtful approach to science, and his openness and friendliness with junior colleagues, he sets an example.

I thank Xiaohui Xie, Justin Werfel, Russ Tedrake, Dezhe Jin, Richard Hahnloser, and Mark Goldman for great discussions and companionship in the Seung group; Jen Wang thanks for the steady supply of adventure books; Naveen Agnihotri, Jen, Ming-fai Fong, Artem Starovoytov, Neville Sanjana, Sawyer Fuller, and John Choi made the Seung lab a very friendly and fun place to be. I am grateful to Amy Dunn, Alan Chen, Ben Pearre, and Mary van den Ijssel for their assistance with many matters, big and small. It has been a privilege and a pleasure to work with you all.

I thank Larry Abbott, Carlos Brody, Allison Doupe, Peter Latham, Anthony Leonardo, Mayank Mehta, Matt Tresch, and Steven Lisberger for thought-provoking conversations and helpful interactions.

I thank the many excellent teachers I had at the University of Michigan and in preceding years of schooling in Bombay, Berkeley, Princeton, and Ann Arbor. I thank Mr. Collins of Huron High School, ye olde perfect calculus teacher, and Mrs. Strang, one of my favorite teachers of all time. For being shining examples of wonderful people doing wonderful science, I thank my undergraduate research advisors Meigan Aronson, Chuck

Bloch, Stefan Süllow, and Andrea Heldsinger. Their guidance in my undergraduate years instilled in me, as I am sure in several others, an abiding love for physics.

One of the best parts of graduate school has been the chance to enjoy time with old and new friends. I thank the TIFR gang, and William Ratcliff, Bob Michniak, Mike Bassik, Marc Humphrey, Mathew Abraham, Irving Johnson, Betsy Catalano, and all the friends Greg and I have from the graduate dorms and the physics department.

Finally, without my family, I could have done so little. Mummy-Babuji have made everything possible. I am deeply grateful to my mother who has selflessly and lovingly given us her all, and to my father, who always set a steady example of discipline, rectitude, hard work, and generosity. Anoop is the finest elder brother a sister could have, and is my constant source of humor and good cheer; it is even better, and twice as much fun now with Sangeeta Bhabhiji and adorable Amit. Mummy, Bhabhiji, and Anoop have kept Greg and me very well fed with frequent treats and feasts throughout graduate school!

I am also extremely glad to be in the same family as Mom and Brian. Mom's courage is an inspiration, and her support has meant a lot. We're very proud of Brian: both in his choice to blaze his own path, and with the remarkable experiences and achievements he has had at such a young age.

Greg, you are the sunshine of my life. May we together have many journeys as wonderful as this one.

Chapter 1

Introduction

Cars can easily outcompete cheetahs in a test of speed; modern computers perform precise numerical computations, in breathtakingly small amounts of time. Animals appear at a great disadvantage when their performance is compared to that of machines in such tasks. In what way then can we think of animals and ourselves as more capable; that is, in what way do lobsters, birds, rats, cheetahs, or humans outperform machines?

Arbuably, the greatest advantages that biological systems possess are *flexibility* and *adaptability,* or the ability to respond to varied and novel situations, and adjust to a wide range of different and changing external stimuli. For example, cheetahs can, in principle, run on many terrains, including narrow tangled forests and rocky plains (as well as on novel surfaces not previously encountered) and climb trees, while cars cannot; different forms of life populate very diverse ecological niches, and can thrive in conditions that are harsh when viewed from another niche; crows, apes, and humans can build tools to extract food; animals and human can rapidly perform image segmentation, decomposing a complex picture of multiple overlapping objects into its constituents, even when the objects are novel, while computer programs either take large amounts of time, or fail utterly in the segmentation problem. Clearly, these are regimes where biology excels, but machines fail; it is of great interest, for both science and engineering, to understand what rules and characteristics allow biological systems to be changeable in an adaptive way.

Depending on the time-scale we choose to scrutinize, biological adaptability may take the form of genetic evolution (slow changes, on the order of several generations or the lifetime of the species), learning and intelligent behavior (on intermediate time-scales, of the order of hours to the lifespan of an individual), or intrinsic sub-cellular and cellular dynamics (rapid changes, in direct response to quickly varying inputs). Of course, these adaptations are not independent: slowly evolving genetic changes are likely to be involved in selecting which cellular or chemical actors play a role and how they interact in the faster time-scale adaptations of learning and memory or sub-cellular dynamics, while the short-time effects of sub-cellular chemical dynamics set fundamental limits on possibilities for change in the dynamics of neural networks and genetics. However, the time-scales separating these dynamics are large enough that we can hope to dissociate evolution from learning and sub-cellular processing.

Of all the biologically relevant adaptations, the focus of this thesis is on learning in neural systems. Animals are capable of using experience to alter their behavior in an adaptive manner: an example from common experience is of squirrels that cleverly manage to raid ever-more-elaborately rigged bird feeders and eat the forbidden birdseed. There are many other instances where the learning of specific behaviors is critical to the long-term success of an individual: in order to attract mates, a young songbird must learn to imitate the song of a tutor, and finally be able to produce a good version of the song. Similarly, polar bear pups must learn to hunt successfully, a task which involves the acquisition and synthesis of diverse skills such as observation, strategic planning, and motor control. Since all these examples of learning are dependent on past activity and experience, we are interested in forms of neural plasticity, or long-lasting neural modification, that depend on past neural activity (in contrast with neural modifications arising from a developmental program, such as age-related cell growth or death).

More specifically, we aim to link learning and activity on the broad, systems level to the plasticity of single neurons. The core of this thesis is devoted to the following question: What is the *learning rule*, or map from neural activity to neural change, that underlies the ability of animals to learn goal-directed tasks?

Our line of reasoning in formulating a neural plasticity rule for goal-directed learningis as follows: Behaviorally, animals can learn from experience. In the brain, too, there is extensive evidence of local activity-dependent neural change. In addition, there are global signals in the brain, that signal the existence or expectation of external reinforcements (such as reward or punishment). These signals are known to be involved in modulating neural change and are crucial for certain forms of behavioral learning. However, it is not understood how these two mechanisms interact: i.e., how do global signals affect activity-dependent mechanisms of synaptic plasticity between local neurons, to drive learning? Moreover, it is known that behaviorally animals are capable of learning complex tasks with the help of simple external rewards. Once again, however, it is not understood what neural mechanisms underlie this ability, and how large groups of neurons cooperatively modify their behavior to learn specific tasks.

Interestingly, these two problems may not be distinct. Goal-directed learning can be viewed as a problem of reward optimization: If performance on a task is quantified by an internal reward signal, then optimization of the reward will lead to improved performance on the task, and vice versa. Based on this premise, we propose a synaptic learning rule, showing how neurons could use synaptic plasticity mechanisms dependent on local activity (which are known to operate in the brain), and modulate this learning with a global reward signal (known to be broadcast in the brain, and to be involved in learning), to optimize the internal reward. Since the internal reward is correlated with external rewards or the expectation of external rewards (a measure of performance on the task), this learning rule can explain the ability of animals to learn goal-directed behaviors.

In the following, we review experimental support for our line of reasoning, as described above: First, we describe neurophysiological evidence of activity-dependent synaptic change. We discuss briefly some of the theoretical successes that resulted from applying such observed learning rules to neural network models of the brain. Next, we describe data on reward signals in the brain, their relationship with external rewards, and their involvement in neural learning. Then, we describe a set of psychology experiments on goal-directed learning in animals, and use principles of behavioral learning to motivate and explain our formulation of rules for goal-directed learning on the neural level. Finally, we briefly describe efforts, in the machine learning literature, of learning goal-directed tasks by reward optimization, and contrast the achievements of that field with our work, and point to some future directions in the study of reward-based learning the brain.

The last part of this introduction consists of individual chapter summaries.

1.1 Local activity-dependent synaptic plasticity

1.1.1 Hebb's proposal

The most influential idea on activity-dependent neural plasticity (in the neurophysiology community), also happened to be the first concrete proposal attempting to connect neural activity with synaptic plasticity, and was suggested by Donald Hebb [1]. According to his proposal, repetitive neural activities are transferred into longer-lasting changes in or between those neurons, according to the following rule:

> When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased. (Hebb, 1949.)

Although largely speculative when it was first formulated, Hebb's proposal for learning started to accumulate support from neurophysiological data, begining with an important work in 1973, which showed evidence of synaptic change based on activity in the adjoining pre- and post-synaptic neurons [2]. As summarized below, numerous and diverse forms of local activity-induced plasticity of synapses have since been found.

1.1.2 Experimental observations of local activity-induced plasticity

Activity-driven synaptic plasticity has been induced and observed between pairs of neurons, with use of varied activity-dependent induction paradigms, in slice and culture preparations, of different regions of the brain, including the hippocampus, the cerebellum, and the neocortex, to name a few examples.

The systematic study of synaptic learning involves a large space of parameters: Do all neuron types in the brain show the same type of plasticity under identical induction paradigms? No; some preparations show synaptic strengthening, or long term potentiation (LTP) in regimes where other preparations show signs of synaptic weakening, or long term depression (LTD). What stimulation paradigms, or patterns of neural activiation, lead to plasticity? Some systems show plasticity in response to stimulation of pre-synaptic afferents alone [2], [3], where the sign of change (LTP/LTD) is a function of stimulation frequency; others respond only when the presynaptic stimulation is paired, simultaneously, with postsynaptic depolarization. In some preparations, excitatory postsynaptic potentials (EPSPs) must be paired, following a brief delay, with postsynaptic action potentials, to increase the amplitude of future EPSPs [4], but actual spiking of the presynaptic neuron per se is not necessary for plasticity; if the relative timing between the two is reversed, future EPSP amplitudes will be weaker (this phenomenon is known as spike timing dependent plasticity, or STDP); influential recent works clearly demonstrated how the relative timing of single spikes in the pre- and postsynaptic neurons determines the sign of change in the synaptic strength [5], [6].

In this tangle of experimental data, the main principle that seems to emerge is that regardless of detail, the phenomenon of local, activity-dependent synaptic plasticity is widespread in the brain.

1.1.3 Theoretical successes of local activity-based learning rules

Hebbian learning can be summarized in the following way: if neurons A and B are active simultaneously or within a narrow time window of each other, the synapse between them is strengthened. STDP provides a refinement of this idea: if neuron B consistently fires shortly after neuron A, the synapse from A to B is strengthened. If the firing of B is not caused in part by the firing of A, then the synapse between them is weakened. Both these learning rules directly translate coincident, or nearly coincident causal inputs into neural synaptic connectivity in a way that the temporal structure of the inputs is reflected in the activity of the network [7]: that is, if two inputs typically occur together, then through Hebbian learning, neurons that individually respond to one or the other of those inputs will wire together, and in the future will tend to fire in synchrony, even in the absence of the driving inputs.

This is the conceptual basis for the formation of orientation columns, ocular dominance stripes, and other types of spatio-temporal maps in the cortex. A similar concept can explain the formation of associative memories in the brain: two temporally contiguous events become associated with the help of Hebb-like learning rules. Modeling and theoretical studies have used STDP to explain how the brain may perform predictive sensory coding, and how it can learn to recognize or produce temporal sequences [8], [9].

In general, sensory neurons repeatedly driven by external inputs can, with the help of local learning rules alone, come to predictively encode the temporal or spatio-temporal correlations present in the external world. The examples cited above illustrate that local activity-based learning paradigms are powerful, since they produce complex and interesting patterns of neural connectivity, resembling structures found in sensory brain areas.

However, these theoretical formulations of neural plasticity do not take into account a major factor that is known to be involved in the modulation of many forms of experience-dependent learning and neural plasticity: global reinforcement (reward or punishment) signals in the brain. Below, we describe experimental evidence on the crucial involvement of internal reinforcing signals in learning.

1.2 Reinforcement signals in the brain

In this section, we briefly discuss several studies, each of which deals only with fragments of the story of learning with reward, but which together reveal that behavioral reward learning is mediated by the interaction of local neural activity with more globally broadcast reward signals in the brain. The studies involve a wide range of techniques, from psychology, to molecular biology genetic biology, to electrophysiology. Taken together, results from the physiological study of reinforcement-based learning (described in the following sub-sections) imply that

- There are clear neural signals that convey information about external rewarding stimuli (such as sucrose in insects) to the brain.

- The outputs of these neural reinforcers can be used in place of the external reinforcing stimuli to induce learning.

- The broadcasting of the reinforcement signal is relatively global (in the spatial sense), and reinforcing neurons project widely to several functionally distinct brain areas; nevertheless, the induced learning is local to the specific subregions coactivated by the conditioned and unconditioned stimuli.

- It is not clear how specific or non-specific (in the sense of coding for different classes or types of external reinforcers) the mechanism of reward-signaling is: for example, the sucrose-related insect reward signal octopamine may or may not generalize to fructose or more distantly related rewarding stimuli. The separable effects of octopamine and dopamine in the learning of sugar rewards and shock punishments, respectively, in the insect brain, indicate that at least one basic form of specificity is present in learning from external reinforcement: aversive and appetitive reinforcements are mediated by different signals in the insect brain. There is currently no clear evidence of such opponency in the coding of rewards and punishments in the mammal brain, but dopamine neurons preferentially signal the presence of external rewards and not punishments.
- Internal reward signals seem to code not only for external rewards, but for the expectation of future external rewards, where expectations are derived from past experiences of reward.

These results, described in more detail below, show that reinforcement signals are widespread in the brain, code for rewards or expected rewards, and are intricately involved in learning. However, there are no clear experimental or theoretical principles that explain how these reward signals are used to direct synaptic plasticity between different recurrently connected non-reward neurons in the brain.

1.2.1 Behavioral paradigm for reward-based learning

A standard paradigm for reward-based behavioral learning is classical conditioning: animals are trained to associate a neutral stimulus (the conditioned stimulus, or CS) with a rewarding or punishing stimulus, by repeatedly pairing the neutral stimulus with presentations of the reward or punishment. The probe for whether the animal has formed a mental association of the formerly neutral stimulus with reward or punishment is to study whether the CS, presented alone, now evokes similar behavioral responses as the original evocative rewarding or punishing stimulus. In Pavlov's famous example, salivation is the probe (response) to see if the animal has come to associate a bell ring (the neutral stimulus, or CS) with food (the reward).

1.2.2 The physiology of reward and punishment in insects

Insects face a potentially large number choices that they must select between based on positive reinforcements from the outside world; their ability to learn complex discrimination and matching tasks on the basis of reinforcement has been demonstrated in several behavioral studies [10], [11].

Honeybees can learn to associate different odors with sucrose rewards when trained with a classical conditioning procedure; they respond to applications of sucrose on their antennae by proboscis extension, and this response can be used as an assay of appetitive conditioning. In the honeybee brain, the firing of the identified VUMmx1 octopamine neuron signals the delivery of sucrose rewards to the antennae and proboscis [12], [13]. The VUMmx1 neuron projects extensively to various olfactory brain areas. Moreover, if octopamine neurons are silenced, the temporally coincident injection of octopamine into a subset of the olfactory target areas of VUMmx1 together with the presentation of the conditioned odor stimulus can substitute for the presentation of sucrose, the unconditioned stimulus, in an appetitive odor-association conditioning experiment [14].

The fruit-fly drosophila can also readily learn associations between neutral odors and either aversive electric shock punishment or appetitive sucrose food rewards. The probe for conditioning in drosophila is whether flies tend preferentially toward or away from the previously neutral odor in a T-maze task compared to pre-conditioning behavior, where one arm of the T has a low concentration of the odor, and the other has plain air.

In drosophila as in bees, appetitive olfactory learning is crucially dependent on octopamine: octopamine-deficient mutants are still able to learn aversive odor–shock associations, but are unable to learn appetitive odor–sucrose associtations; however, if octopamine expression is rescued by heat-shock or with the help of orally administered octopamine, then the potential for appetitive conditioning [15] is restored. Odor-shock aversive conditioning in drosophila depends on dopamine and not octopamine, in the same way as appetitive conditioning relies on octopamine and not dopamine citeSchwaerzel03. Genetic manipulations have made it clear that although both appetitive and aversive conditioning are mediated by globally broadcast octopamine or dopamine signals, the site of odor-related learning is very small, and both forms of associative memory are located in the Kenyon cells of the mushroom bodies; these cells are deeply involved in the sensory processing of odors.

1.2.3 The physiology of reward in mammals

The neurons in the deeply imbedded substantia nigra and ventral tagmental areas (VTA) of the brain release the neuromodulator dopamine, and their axons project extensively throughout the neocortex (and especially to the frontal cortex), the nucleus accumbens, and the striatum [16], [17]. Dopamine neurons show strong, phasic responses to both food and drink rewards [17], and are inhibited by aversive stimuli [18]. Moreover, deep brain stimulation studies show that direct excitation of the dopamine pathway can be used very successfully in place of external food or drink rewards to train animals to depress levers using an instrumental conditioning paradigm [19], [20].

Dopamine release is observed in rat primary auditory cotex (A1) during auditory learning [21]. Repeated bouts, of auditory stimulation with a pure-tone pip followed by VTA stimulation, led to a greater representation of and selectivity for that tone in A1 [22]; if the order was reversed, so that auditory stimulation preceded VTA stimulation, then the effects on the cortical representation of that frequency were also reversed. That is, the cortical area devoted to that tone was reduced [23]. The observed changes were long-lived. Rats that were subject to either auditory or VTA stimulation, but not both, showed no noticeable changes in their A1 representations. Thus, dopamine-pairing can lead to temporally non-trivial, bi-directional persistent changes in neural connectivity.

There are many aminergic or cholinergic candidates for the role of chemicals that signal behavioral state and modulate plasticity; on the whole, their contributions have not been as well-characterized as dopamine. One study implicates acetylcholine, as secreted by a fraction of nucleus basalis (NB) neurons, in playing a role similar to dopamine in the learning of new auditory representations in rat A1. Nucleus basalis projects widely to cortex. If NB afferent stimulation is paired with the presentation of auditory tones, the A1 cortical area devoted to representing the paired tone is increased [24]. Other commonly mentioned candidates include serotonin, norepinephrine, and nitrous oxide; there is also no reason currently to rule out the possibility that reward signals are encoded in and transmitted by regular, glutamatergic or GABA-ergic neurons. In fact, the inferior olive neurons of the cerebellum are GABA-ergic, and are thought to carry a motor-error signal for use in motor learning.

There is no clear evidence of a separate mechanism that codes for punishments in a phasic manner in mammals. Some argue that the neuromodulator serotonin may be an opponent system to the dopamine reward signal, and may act as a slowly-changing subtractive baseline for the assessment of reward [25]

In the following, we briefly discuss some experimental results demonstrating that reward coding is modulated by experience. We also address the issue of whether opponent mechanisms are vital to learning, from a theoretical perspective, and whether greater or less stimulus specificity in the coding of rewards would be ”better” for goal-directed learning.

1.2.4 Experience-dependent reward coding

Interestingly, the VUMmx1 neuron in the honeybee appears to code not only for direct sucrose rewards, but also fires in response to odors predictive of sucrose reward. For example, some reward-related VUM neurons start to fire in response to formerly neutral odors that did not previously evoke VUM responses, once those odors are conditioned to sucrose in an appetitive conditioning paradigm [13]. This result in honeybees parallels the coding of rewards by dopamine in mammals, described below.

In the mammal dopamine reward system, as in the insect octopamine system, there are signs of experience-dependent coding of current rewards, based on rewards received in the past. Specifically, if an external reward is expected, based on past experience, and if the actual reward matches expectations, then the activity of dopamine neurons remains at baseline levels; if the external reward exceeds expectations, dopaminergic activity is high; finally, if reward is lower than expected, dopamine neuron activity is depressed to lower than baseline [17]. Thus, midbrain dopamine neurons are most responsive to the appearance of unexpected external rewards, and appear to primarily code for the receipt of rewards that deviate from a baseline set by past experiences.

1.3 Learning complex tasks with simple rewards: insights from psychology

Instrumental conditioning is another behavioral paradigm for reward-based learning. In instrumental conditioning, naïve animals are trained to perform specific tasks, but without instruction; in fact, even the goal of the task is unknown to the test subject. Instead, the animal is trained, by selective administration of external reinforcements (rewards or punishments) depending on its actions, to perform some desired task. Instrumental conditioning can be contrasted with classical conditioning, because in the former scheme reinforcement is contingent on specific actions of the animal, while in the latter, reinforcement is paired with the appearance of another sensory cue, but is independent of the animal's response.

Although classical conditioning may be interpreted as a form of associational Hebb-like learning acting on synapses between global reward neurons and local sensory neurons coding for the conditioned stimulus, insrumental conditioning presents another challenge: it involoves the generation of a behavior and the subsequent shaping of that behavior with a reinforcer. The reinforcer plays a third-party role, in modulating plasticity between local synapses responsible for generating the behaviors. How does the presence or absence of reward affect learning within the circuits that generate behaviors, in a way that allows animals to improve on the task?

Thorndike, an originator of the paradigm of instrumental conditioning, placed cats inside 'puzzle' boxes, to escape from which a cat had to depress a platform, pull a string, and turn a latch. If it escaped from the confines of the box, it received a food reward, and after a delay, was once again placed inside the box, to be rewarded again if it managed to escape; this was done repeatedly for each subject. Initially, the cat tried various combinations of lever presses; after many tries, it hit upon the correct combination, and could escape. On subsequent escape trials, the cat again tried random approaches, getting better on average, but with large variations in latency (escape time) from trial to trial (Thorndike, 1898); eventually, it learned to consistently engage the correct combination of levers and escape quickly. In his famous ”Law of Effect”, Thorndike surmised that the animal correlates each trial with its respective outcome, to gradually shape a successful strategy:

> Of several responses made to the same situation those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will

> be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections to the situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond. [26]

Thorndike's work, and numerous subsequent experiments, show that animals can be trained to learn complex, goal-directed behavioral or motor tasks, with the help of simple rewards. When confronted with a novel task and little or no instruction, animals appear to randomly try different strategies, and specifically alter their actions or strategies to increase their chance of obtaining reward. Behaviorally, therefore, the learning of goal-directed tasks may be driven by an approximate process of reward maximization [27].

Since the brain contains extensive circuitry for the delivery of internal rewards, we surmise that such goal-directed learning may be also be thought of, on the neural level, as a problem in reward maximization. However, even if goal-directed learning proceeds by the optimization of a global reward signal in the brain, and if all neurons involved in learning have access to information about the reward, individual synapses still do not receive information on which direction they should change to increase the future probability of reward.

In the highly recurrent networks of inhibitory and excitatory neurons found in the brain, a single reward signal indicating average performance on a task does not alone carry enough information to instruct synapses on how to change in a way that will improve the network output. How do neurons then "know," or "decide," in which way to change, to increase the average reward?

In analogy with animal behavior in Thorndike's experiment, and with his law of effect, we propose that learning on the neural level also depends on the generation (by neurons) of variable outputs; and on the selective strengthening of those actions (via modification of active synaptic connections) that lead to rewards.

Our proposed learning rule acts in a system consisting of a recurrent network of spiking neurons with plastic synapses, a separate noise injector, which perturbs the activity of neurons in the network, and a reinforcer, which sends information about network performance back to the network. According to the rule, if a neuron receives greater-than-average excitatory input from a noisy neuron, and if this activity contributes to (is followed by) positive reinforcement, then all recently active non-noise (and non-reward) excitatory synaptic inputs to that neuron should be strengthened. Conversely, if lower-than-expected noise input to the postsynaptic neuron is followed by positive reinforcement, then the recently active non-noise inputs to the neuron should be weakened. We prove in this thesis that the learning rule is guaranteed, even when applied to realistic, voltage and conductance based spiking model neurons, to lead towards increasing reward, through stochastic gradient ascent on the reward function.

1.4 Contributions from the field of machine learning

Much of the mathematical framework for our work on learning in biological neural networks was laid in studies of goal-directed learning in artificial neural networks. Goal-directed learning has long been a focus of the machine learning and artificial intelligence communities, because many problems of interest (e.g., handwriting and speech recognition) involve training networks to perform specific tasks with the help of a set of known examples.

In software implementations, where the artificial neural units comprising the network are simple and well-behaved, and where all the transformations from input to network to output are known, differentiable functions, a typical machine learning approach has been to define and optimize an objective or error function that quantifies network performance on the task. Optimization is carried out by directly computing derivatives of the network output with respect to the tunable parameters, to assign credit or blame to individual units for their role in producing the error. The individual neural contributions to the error, thus computed, are then used to adjust the network connection strengths, so that the total change in the objective function is along the gradient. A specific algorithm to compute the appropriate derivatives, known as backpropagation, is effective and popular, and though originally formulated for feedforward multilayered networks, has been generalized to work in recurrent networks as well.

In hardware implementations, where transistors may be noisy and temperature-dependent, and where the transformation from network to output involves motors and servos whose transfer functions are complex, state-dependent, sometimes not well-characterized, and possibly not differentiable, backpropagation cannot be used. Biological neural networks of the brain are faced with similar problems: complex, state-dependent dynamics, noisy dynamics, unknown and changeable transformations from neural activity to muscle output, etc. An alternative strategy that has been successful for machine-learning in such situations, is learning by stochastic search and reinforcement. Stochastic learning schemes such as weight perturbation and node perturbation [28], [29] compute stochastic estimates of local gradients, and move along them. This and related strategies circumvent many of the problems faced by learning prescriptions like backpropagation that depend on the explicit computation of gradients, and have been used convincingly to train hardware-based artificial neural circuits.

Our learning rule follows essentially the same strategy as node-perturbation or weight-perturbation: stochastic estimation of local gradients, followed by selective movement in the direction of improved outputs. Our work goes further, from the biological viewpoint, because we provide a rule that is capable of performing stochastic gradient learning for networks of nonlinear, voltage and conductance based spiking neurons, whereas this has not been shown before. Another issue that is more relevant to the study of goal-directed learning in biological systems than machine learning, is the question of the scalability of stochastic gradient algorithms to realistically large networks. We analyze the scalability of stochastic gradient rules as a function of network size and the characteristics of the task, in feedforward networks.

Specifically, in Chapter 4 we introduce the learning rule and describe its mathematical underpinnings and guarantees of stochastic gradient following convergence, when applied to recurrent networks of voltage and conductance based spiking neurons. In Chapter 5, we treat the specific example of goal-directed learning in songbirds, and show that the proposed learning rule can give rise to song acquisition, and moreover, that learning can take place fast enough to be biologically plausible. In Chapter 6, we deal more generally with the scalability of stochastic gradient learning rules, such as this one, as a function of network size and task difficulty.

1.5 Learning many tasks simultaneously: compartmentalization?

What are natural future steps in the study of reinforcement learning in the brain? We already mentioned one important question, that of the scalability of stochastic reward optimization to large networks. A closely related experimental and theoretical question is whether aniamls learn all reinforced behaviors with just a single internal reward signal, or whether learning compartmentalized in some way, so that different external rewards resulting from different behaviors are represented by separate internal signals.

On the experimental end, are insect octopamine signals specific to sugar, or do they code more generally for ”reward”? Similarly, does dopamine mediate other aversive stimuli besides electric shock in the insect brain? These questions are unanswered by current experimental studies. However, it would be surprising if, for example, dopamine signaling in insects is related only to shock-stimuli, since electric shock is not a common enough natural stimulus to expect a specialized neural mediator for that stimulus alone. In mammals, moreover, dopamine generalizes at least to several different food and drink rewards. Finally, because associations between neutral and rewarding stimuli can be learned by the reward system though classical conditioning, it follows logically that the reward system must come, through experience, to code for stimuli that are closely related to reward even if not directly rewarding. Hence, we expect internal reward and punishment systems to respond at least somewhat generally to external rewarding and punishing stimuli.

From a theoretical perspective, the existence of multiple reward signals, that separately evaluate performance on *separable* tasks, would help greatly with reducing the total learning time in models learning, though variance reduction. (If two tasks can be optimized independently, without leading to sub-optimal performance when combined, we call them *separable*.) In other words, the separate quantification of performance on separable tasks would allow for more compartmentalized learning in the brain, so that an animal would not collectively have to optimize ”life,” for example, to optimize ”chewing,” or ”song.” An alternative mechanism for compartmentalizing learning without requiring an explosive growth in the number of reward-signaling neuromodulators, could be to gate learning by attention in addition to reward.

It is possible that the brain compartmentalizes tasks, whether with the use of separate reward signals or by attention, even when they are not truly separable according to our definition. This possibility is theoretically interesting, since there may be fundamental tradeoffs involved in compartmentalization. On the one hand, increasing compartmentalization would decrease variance, and speed up learning; on the other hand, decreasing compartmentalization would allow for the existence and search of better, more globally optimal, solutions to problems that involve the cooperative action of several motor and behavioral outputs.

An alternative to the compartmentalization picture is that instead of breaking down the learning of different tasks into separate compartments, the brain has a mechanism for heirarchically arranging the assesment and delivery of rewards to different parts of the brain. In this picture, brain areas responsible for some higher-level aspect of learning, directly receive an internal reward that reflects the presence of external reinforcers; these high-level brain areas decide how to distribute reward to lower areas involved in more detailed aspects of behavioral processing, and so on. This field, of hierarchical reinforcement learning, is relatively unexplored, and may provide answers to some of the larger questions of learning in the brain.

1.6 Quick overview

This thesis begins, in Chapter 2, with a primer on neurons and networks, so that readers outside the field of neuroscience can familiarize themselves with some basic terminology; the second half of the chapter contains a brief sketch of past and current areas of interest in the theoretical study of neuroscience and neural networks. Readers who are already somewhat familiar with neuroscience may prefer to skip Chapter 2. The following pages contain a summary of the chapters in this thesis.

Neural activities are widely observed to be ‘sparse’: for example, sensory neurons fire in response to a very small fraction of all presented stimuli, and higher cortical neurons tend to fire relatively little in time. In Chapter 3, we explore the question of sparse premotor coding in the context of birdsong production. We find that sparse premotor neural codes may be advantageous for song acquisition, in terms of the amount of time taken to learn a song. In a more general context, our results imply that sparse coding in perceptron-like feedforward neural networks may facilitate the learning of output sequences from abstract input sequences.

In Chapter 4, we propose a novel synaptic learning rule for recurrent networks of biologically realistic spiking neurons, and prove that it performs stochastic optimization of a reward. The learning rule relies on local exploratory noise and a single globally broadcast reward signal. It is capable of dealing with delayed reward, and allows the training of complex biological networks to perform goal-directed tasks with a simple and plausible plasticity rule for synapses.

The test of a learning rule comes from applying it to actual biological systems, and studying whether learning can converge within a reasonable amount of time in a biologically large and realistic system. Stochastic learning rules, in particular, have been criticized and cast aside as being too slow to be plausible for learning in neural systems; however, these claims have been based largely on heuristic arguments. In Chapter 5 we examine the issue by applying the learning rule of Chapter 4 to the experimentally well-characterized example of birdsong. We show, with a combination of simulations and theoretical arguments, that learning actually converges rapidly in our model of birdsong learning, and can scale up fast enough in large networks for stochastic gradient to indeed be applicable to certain forms of learning in the brain.

We were motivated, based on the studies and results of Chapter 5 to analyze, in a more general setting, the scaling of stochastic gradient rules in systems with anisotropic quadratic cost functions. In Chapter 6, we study the convergence behavior resulting from stochastic gradient descent on such cost functions in a linear quadratic system. In the linear system, we are able to derive explicit learning curves and study the dependence of learning time on the size of the system, as well as on the precise shape of the cost surface. We show how this analysis applies to learning by stochastic methods such as weight and node perturbation in neural networks, and discuss what the results imply for the learning of real-world problems in neural networks.

1.7 Chapter summaries

Chapter 2: Background and perspective. This chapter starts with a commonly-used description of neurons as capacitors, and goes on to briefly describe single-neuron dynamics and primary modes of inter-neural communication. The goal of the next part of the chapter is to provide a description, in words, of the mathematical framework used in the study of neural networks, and to use this description to sketch a history of neural network research: the study of neural networks as trainable systems for artificial intelligence, and the study of pattern-forming neural networks as physical spins. The chapter concludes with a summary of some important modern questions in neuroscience; the aim in the last part is to provide a perspective for the work contained in this thesis.

Chapter 3: Sparse neural codes and learning. Sparse neural codes have been widely observed in cortical sensory and motor areas. Birdsong is an example of a learned complex motor behavior of great interest since it is generated by neural circuits whose anatomy and physiology have been well-characterized. The motor neurons that innervate avian vocal muscles are driven by premotor nucleus RA, which in turn is driven by nucleus HVC. Recent experiments have revealed that RA-projecting HVC neurons fire just one burst per song motif [30], but the function of this remarkable temporal sparseness has remained unclear. Here we explore the role of sparse codes in motor systems through the specific example of birdsong. We show that the sparseness of HVC activity facilitates song learning in a neural network model: in numerical simulations with non-linear neurons, as HVC activity is made progressively less sparse, the learning time increases significantly. This slowdown arises because of increasing interference in the weight updates for different synapses. If activity in HVC is sparse, synaptic interference is reduced, and is minimized if each synapse from HVC to RA is used during only one instant of the motif, which is the situation observed experimentally. Our numerical results are corroborated by a theoretical analysis of learning in linear networks, for which we derive a relationship between sparse activity, synaptic interference and learning time. We discuss implications of this study for juvenile HVC activity and coding in other motor systems.

Chapter 4: A biologically plausible learning rule for reward optimization in networks of conductance-based spiking neurons. It is not known how animals learn complex goal-directed tasks. In machine learning, since several techniques exist to perform optimization, it has proved fruitful to transform the goal-directed learning problem into one of reward maximization by equating closeness to the goal with reward. However, such direct-gradient techniques cannot be applied to neural networks with conductance-based spiking model neurons, where, because of their rich temporal dynamics, the relevant gradients cannot be expressed explicitly. Stochastic gradient rules also fail to incorporate many important biological features, and thus cannot explain learning in biological neural networks. In this work, we propose a biologically plausible learning rule at the synaptic level that is able, with the help of a simple global reward signal, to produce goal-directed learning by optimizing reward in networks of conductance and voltage based spiking neurons. We prove that under certain conditions the proposed learning rule performs stochastic gradient ascent on the reward; when these conditions are relaxed, we provide bounds within which learning follows hill-climbing or performs approximate gradient following on the reward.

Chapter 5: Synapses to song – *rapid* learning by reinforcement of variation due to injected noise in spiking networks. The learning of complex motor skills through practice depends on the generation of variable behavior, and a means of reinforcing favorable variations so as to improve average performance. While many computational models have attempted to relate such learning to synaptic plasticity, they have generally only been applicable to vastly simplified model neurons. Even so, an important criticism of noise-based learning in general is that it may be too slow to be applicable to biologically large networks. In this work we propose a synaptic learning rule for biophysically realistic, spiking model neurons. Variability is generated by noisy synaptic drive from an external source, and plasticity is instructed by the correlation of the postsynaptic injected noise with a global reward signal. The learning rule can be shown mathematically to perform stochastic gradient ascent on the reward, and converges rapidly when applied to a network model of birdsong acquisition. Numerical simulations and theoretical arguments indicate that learning time has little dependence on network size, when the network is larger than necessary to perform the task. Consequently, stochastic gradient learning may be fast enough to be taken seriously as a biological mechanism, at least for practice-driven motor behaviors such as birdsong, and other relatively “simple tasks.

Chapter 6: Stochastic gradient descent on anisotropic cost surfaces: scaling of learning time with system size and task difficulty. Stochastic exploration has been widely used as a tool for optimization in the fields of biological neural learning [31], the study of evolutionary dynamics of populations, and machine learning. Optimization, or learning, by stochastic means, is a convenient and often necessary approach for many problems because it is possible to optimize system outputs without needing to explicitly compute gradients that cannot be expressed in explicit form, or in systems where sufficient information about the underlying variables is not available to compute the true gradient. One of the chief criticisms, however, is that perturbative or stochastic learning methods are slow and scale poorly with increasing network size, and hence are not plausible candidates for learning in biologically large neural networks, which are typically large. Such claims, however, are largely based on heuristic estimates. In this work, we analytically compute the dependence of learning time on system size, and on the shape of the cost surface. We discuss the applicability of the analysis to learning by weight and node perturbation in neural networks, and to the learning of real-world examples.

Chapter 2

Background and perspective

The chapters of this thesis are written so that anyone outside the field of neuroscience but curious about it, with some mathematical training, may get a sense of questions and approaches in the field. Several of the topics discussed are currently undergoing vigorous growth, because they occupy the important niche of being biologically relevant but nevertheless amenable to theoretical treatments. Only a brief sketch of the necessary background material is provided, in Chapter 2; the motivated reader will hopefully find that the recommended books [32, 33, 34, 35], which I found to be indispensible when entering the field, are helpful in filling in the many omissions I have made. [1]

The first half of this chapter provides a brief primer on the essential computational properties of neurons and a basic introduction to the language of neural networks, and is meant for readers untrained in neuroscience. The second half is a rudimentary historical perspective on the field of theoretical neuroscience and an introduction to topics and questions of recent interest in the field.

[1] A very good and comprehensive database for neuroscience articles, with links to the relevant journal sites, is *PubMed* (accessed from http://www.ncbi.nlm.nih.gov/), a part of the National Center for Biotechnology Information (NCBI), which is run by the National Institutes of Health.

2.1 What is a neuron?

Neurons, or brain cells, are fluid-filled sacs bound by a lipid bilayer that separates the intracellular contents from the extracellular space. Neurons maintain a negative internal voltage relative to the extracellular space; this potential difference is maintained by ion channels and pumps. In most neurons of the central nervous system, neural activity is signaled by a spike, or rapid intracellular depolarization followed by repolarization; information about a neuron's activity is communicated to adjacent neurons by one of two means. Some neurons communicate with simple resistive coupling, via channels that allow direct ion flow. Most neurons in the central nervous system (CNS) of higher animals, however, communicate through chemical synapses: a neural spike triggers the release of chemicals called neurotransmitters into the extracellular space. These neurotransmitters bind to ion channels in adjacent neurons, causing a brief ionic current to flow into the neuron. Depending on whether the neurotransmitter is excitatory or inhibitory, the resulting current flow in the recepient neuron will be depoloraizing, or hyperpolarizing, respectively.

A volley of excitatory inputs from adjacent neurons will depolarize a neuron, and once sufficiently depolarized, will result in a spike. The dynamics of spike generation are very stereotyped; when a neuron reaches a threshold depolarization, a non-linear cascade of voltage-gated ion channel openings and closings is unleashed which produces a rapid and large depolarization of the neuron, followed by an almost equally rapid repolarization. Spikes typically last 1 ms.

Since chemical synapses are asymmetric by their nature (one neuron *releases* chemicals into the extracellular space, the other *is affected by* these chemicals), neural communication is directional; viewed locally from a single synapse, the neuron on the releasing side of the synapse is termed ‘pre-synaptic’, while the other one is ‘post-synaptic’.

Neurons in an individual vertebrate brain come in amazing varieties, with regard to size, morphology, electrical properties, etc. The functional roles for the dramatic variations in the branching structures and voltage properties of different neural types is not well understood; in fact, it is a matter of active debate whether neural diversity is superfluous and accidental, or if it is essential for neural computation.

The dynamics of individual idealized point-like neurons (with no spatial extent) have been modeled at various levels of detail. On a very basic level, neurons may be viewed as simple *leaky* or *forgetful* capacitors; an internal battery maintains their negative holding potential relative to the outside, but excitatory and inhibitory synpatic inputs act to temporarily connect the capacitor to depolarizing or hyperpolarizing batteries. Well below the spiking threshold, neurons display approximately linear charging behavior, justifying simple models that treat them as capacitors. Famous examples of model neurons, in rough order of decreasing biological detail and increasing abstraction include the Hodgkin–Huxley, Morris-LeCar, FitzHugh-Nagumo, quadratic integrate-and-fire, and integrate-and-fire model equations. Numerous other models describe the detailed firing properties of specific sub-types of neurons, or are simplified to resemble familiar nonlinear oscillators and allow rigorous phase-space analysis of the dynamics.

2.2 Neural networks

One abstract description of a neural network is that of a directed graph, whose nodes represent neurons, and whose edges represent the asymmetric connections between neurons. Each node sums the outputs of adjacent (connected) nodes, weighted by the strengths of the connecting edges, and applies a nonlinear transformation to the sum to produce an output. Recurrently connected networks can have rich intrinsic dynamics, even when edges are viewed as fixed and static. An additional layer of dynamics can be introduced by also allowing edge weights to change, usually on a much slower time-scale.

The approach to the study of neural networks in the literature has been two-pronged: (1) Construction of general, 'trainable' intelligent networks and learning rules so that weights can be changed to enable the network to perform goal-directed tasks. (2) Investigation of various emergent phenomena that arise from the interactions of large assemblies of simpler entities, using simple models of interacting neurons and pair-wise, possibly biologically motivated, neural learning rules.

Early successes in approach (1) included learnable pattern classification with linear single-layer perceptrons. A perceptron is a feedforward network that consists of a set of inputs connected to a set of output units through layers of "hidden" units. Patterns presented as inputs drive activity in the output units; sets of inputs producing the same output are classified as belonging in the same category. The network is trained, by appropriately changing weights, to produce a *desired* classification on a set of 'training' inputs. It is then used to classify previously unseen 'test' inputs. Interesting theoretical work on this front included studies on the computational capacity and limitations of perceptrons with linear neural units; the categorization of different nonlinear networks as capable of universal computation or not; and the search for learning rules for multilayer feedforward and recurrent networks with nonlinear units.

One of the most important developments, from the point of view of both theoretical neuroscience and artificial intelligence, was the backpropogation algorithm, a method for training multilayer, feedforward networks with non-linear units to perform gradient descent on a suitably defined error function. Multilayer non-linear networks can perform complex tasks and some such networks are capable of universal computation. The goal of learning can be quantified by an error function, for example, the mean-squared distance of the outputs of the network from the desired outputs. The success of backpropogation is that it allows 'credit' or 'blame' for the output error to be assigned correctly to all neurons in a feedforward network, even those in the hidden layers. Mathematically, backpropogation is like a chain-rule that computes and uses partial derivatives of the error with respect to the network edge weights to drive learning along the error gradient. Since its introduction, the backpropogation algorithm has been extended to learning in recurrent networks and to trajectory learning; in all, backpropogation has been applied to many problems in machine learning with considerable success, and has brought the field of artificial intelligence closer to the grail of general trainable networks.

Approach (2), with roots in statistical physics, brought the study of neural networks to the wider attention of physicists. If neural activities are binarized, and interactions between neurons are assumed to be symmetric, they may be viewed as particles interacting through spins. Excitatory-excitatory pairings correspond to ferromagnetic interactions, while inhibitory-inhibitory pairings would be antiferromagnetic. Thus, an interacting network of neurons could be treated as a magnetic solid, and techniques from statistical physics could be imported in bulk and used to study neural network dynamics: approaches and topics include mean-field solutions of network dynamics, analysis of networks as attractors settling to fixed points (associative memories), or to limit cycles (oscillators or sequence-producing networks), phase-space analysis with replica-symmetric and replica-symmetry breaking studies at various effective temperatures, characterization of a spin-glass phase, calculation of network information storage capacity, studies on the formation of self-organizing spatial maps, etc.

These studies set the foundation for and trajectory of much current work in theoretical neuroscience. One major gap in all these works has been that they did not consider many fundamental details about biological neurons, such as the fact that they are extremely nonlinear, have complex temporal dynamics and communicate with spikes not rates, etc.

2.3 Modern topics in theoretical neuroscience

New directions in theoretical neuroscience have emerged from the re-examination of old questions (some of which are mentioned above) in light of recent experimental results and with an emphasis on dealing with biological complexity. In particular, thinking of neurons as biological entities with energy and space constraints, susceptibility to damage and replacement, and as dynamic and adapting elements, has led to the opening of whole new fields of inquiry. Briefly, examples of current research areas in neuroscience are:

- SPIKES Why do neurons spike? Information conveyed in the form of spike trains of neurons is digital, and therefore more noise resistant than analog signals. However, in terms of energy considerations, analog signals are more efficient. Therefore, does the brain use spike codes for better loss-free communication over long distances at the expense of energy? Or are spikes used in a fundamentally different way than analog signals might be, such as for nonlinear coincidence detection?

- NOISE AND REDUNDANCY Neurons in the vertebrate cortex often fire extremely noisy spike trains, with a coefficient of variance close to 1. Thus, the study of noise in the brain is of central importance. Searching for optimal, noise-resistant coding and detection schemes, analyzing whether the brain performs at a close to optimal level, are interesting related questions.

- NEURAL STATISTICS A more basic question in the study of noise is how neurons get to be so noisy in the first place, i.e., how do neurons generate spike trains with a large variance given that each neuron receives $10^3 - 10^4$ convergent inputs. Are common inputs or recurrent feedback connectivity responsible for correlations between different neurons?

- CODING Why do certain brain areas represent sensory inputs or motor commands in way they do – for example, what features of the external world are conveyed in the spike trains of retinal neurons, or how are premotor commands related to the motorneuron and muscle outputs they produce? Among other reasons, specific forms of neural coding may evolve as a result of learning time-pressure, or out of a need for robustness or efficiency of information transmission.

- ROBUSTNESS Animals and brains of animals can function reasonably well even when subject to environmental abuses such as temperature variations, drug injections, etc., but corresponding models seem to depend very sensitively on the exact values of the underlying parameters, and are not robust to perturbations. What are the general mechanisms that allow large assemblies of complex interacting elements to be robust against random and systematic perturbations in external or internal parameters?

- LEARNING Learning rules that mimic pair-wise neural activity-dependent plasticity in spiking model neurons give rise to some interesting emergent phenomena such as map formation, but have not been demonstrated to be capable of solving specific goal-directed problems. Direct and stochastic gradient based learning rules can perform goal-directed learning, but only in vastly simplified networks of rate-based units. Much work is focused on applying and better understanding existing learning rules, and attention is now being given to new learning rules that are both biologically realistic and capable of performing goal-directed learning.

Chapter 3

Temporal sparseness of the premotor drive is important for rapid learning in a model of birdsong

3.1 Abstract

Birdsong is an example of a complex motor behavior generated by neural circuits that have been characterized both anatomically and physiologically. The motor neurons that innervate avian vocal muscles are driven by premotor nucleus RA, which in turn is driven by nucleus HVC. Recent experiments have revealed that RA-projecting HVC neurons fire just one burst per song motif [Hahnloser, Kozhevnikov, & Fee (2002) *Nature* **419**, 65-70], but the function of this remarkable temporal sparseness has remained unclear. Here we show that the sparseness of HVC activity facilitates song learning in a neural network model. If HVC activity is made less sparse in the

model, the learning time increases in numerical simulations. This slowdown arises because of increasing interference between synapses. Interference is minimized if each synapse from HVC to RA is used during only one instant of the motif, which is the situation observed experimentally. Our numerical results concerning nonlinear networks are corroborated by an analysis of learning in linear networks, for which the relationship between sparse activity, synaptic interference, and learning time can be treated mathematically.

3.2 Introduction

Birdsong is a complex, learned motor behavior driven by a discrete set of premotor brain nuclei with well-studied anatomy. Neural activity in these areas has been characterized through recordings in awake, singing birds, making the birdsong circuit a uniquely rich and accessible system for the study of motor coding and learning.

Syringeal and respiratory motoneurons are driven by precisely executed sequences of neural activity in the premotor nucleus robustus archistriatalis (RA) [36]. Activity in RA is strongly driven by excitatory feedforward inputs from the forebrain nucleus HVC [37, 38, 39, 40], whose RA-projecting neural population displays temporally sparse, stereotyped, temporally precise sequential activity: Individual RA-projecting HVC neurons burst just once in an entire ≈ 1 s song motif, for a typical duration of 6 ms and at a high firing rate (600-700 Hz), and fire almost no spikes elsewhere in the motif [30]. Burst onset times for different RA-projecting HVC neurons are distributed across the motif. Each HVC neuron bursts reliably at precisely the same time-point (referenced to some acoustic landmark in the motif) in repeated renditions of the song.

The temporal sparseness of HVC activity implies that synapses from HVC to RA are used in a very special manner during song. Namely, each synapse is used during only one instant in the motif. Is there any functional significance to this way of utilizing synapses? Here we investigate the possibility that it facilitates song learning.

Our hypothesis is motivated by two considerations. First, plasticity of synapses from HVC to RA is likely to be involved in song learning, given anatomical evidence of extensive outgrowth followed by partial retraction during the song-learning critical period. Second, in artificial neural networks, it has been observed that interference between synapses can hinder learning [41, 42, 43, 44].

Intuitively, the situation where each synapse participates in the production of just one part of the motif seems ideally suited for minimizing interference between different synapses during learning. In this paper we make the intuitive argument more concrete through both computer simulations and mathematical analysis of a simple neural network model of birdsong learning.

We model the song circuit as a feedforward network (consisting of HVC, RA, and an output motor layer) with an existing HVC sequence that drives sequential output activity, typically different from a specified sequence of desired output activity. Experiments indicate that with practice, the songbird iteratively minimizes the mismatch (error) between its own vocalizations and an internally stored template of a tutor song [45, 46]. Accordingly, the goal of learning in the model network is to make the output activity match the desired sequence by gradual adjustment of the HVC-to-RA weights [47, 48], and we study the dependence of learning speed on the sparseness of HVC activity.

It is not known how the brain performs goal-directed learning, although it is thought [47] that by appropriate modification of underlying synaptic weights the brain does hill-climbing on an objective (e.g., reward) function that quantifies how close the goal is. A common computational approach in modeling this phenomenon is to define an objective function and then alter the synaptic weights to move along the direction of steepest ascent (gradient) toward the optimum of the objective function. Backpropagation is a direct method of computing the gradient and updating synaptic weights. Several other, more biologically plausible learning algorithms have more recently been proven to optimize an objective function, which they also do by implicitly computing and moving along stochastic estimates of the gradient [49, 50]. Since various gradient-based learning rules are in a mathematically similar class, we expect sparseness arguments made in the context of one learning rule to generalize to others in the same class. Thus, for simplicity we use learning by direct gradient computation, or backpropagation.

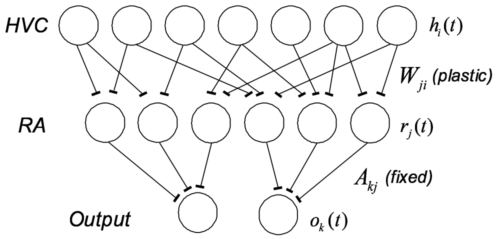


Figure 3.1: Model network structure: HVC, RA, and the output layer are arranged with feed-forward plastic weights W from HVC to RA, and fixed weights A from RA to the output. HVC activities provide sequential inputs to the network, and the output units are the read-outs. The RA neurons form a “hidden” layer.

3.3 Methods

General framework. We study a multilayer perceptron (Fig. 1) with an HVC layer that provides inputs to the network and drives activity in the hidden layer RA; the output layer of motor units is driven by activity in RA. HVC activities are written as $h_i(t)$, RA activities as $r_j(t)$, and output activities as $o_k(t)$, with

$$r_j(t) = f\left(\sum_{i=1}^{N_h} W_{ji} h_i(t) - \theta_j\right), \tag{3.1}$$

and

$$o_k(t) = \sum_{j=1}^{N_r} A_{kj} r_j(t). \tag{3.2}$$

N_h, N_r, and N_o are the numbers of units in the HVC, RA, and motor layers, respectively; f is the activation function of RA neurons, and θ_j is the threshold for the j^{th} RA neuron. The plastic weights from HVC to RA are given by the matrix W, and the fixed weights from RA to the outputs are represented by A.

HVC activities and desired output activities are externally imposed (see below for numerical details); the goal of the network is to learn to match the actual outputs $o_k(t)$ of the network, driven by HVC activity, with the desired outputs $d_k(t)$, through adjustment of the plastic weights W. In one pass through the learning process, called an epoch, the network outputs are computed for the entire song motif from Equations (3.1) and (3.2). The total network error for that epoch is determined from the objective function

$$C = \int_0^T dt \sum_{k=1}^{N_o} (d_k(t) - o_k(t))^2. \tag{3.3}$$

Network weights W are adjusted to minimize this cost function according to:

$$\Delta W_{ji} = -\eta \frac{\partial C}{\partial W_{ji}} = \eta \int_0^T dt \sum_{k=1}^{N_o} 2(d_k(t) - o_k(t)) A_{kj} f_j' h_i, \tag{3.4}$$

where f_j' is the derivative of the activation function of RA neuron j, and the parameter η scales the overall size of the weight update.

Numerical details of non-linear network simulations. We simulate learning in the network described above, with $N_h = 500$ HVC neurons, $N_r = 800$ RA neurons, and $N_o = 2$ output units. Assuming that each HVC neuron bursts B times per motif, activity for the i^{th} HVC neuron is fixed by choosing B onset times $t_i^{\{1,..,B\}}$ at random from the entire time-interval, T. A burst is then modeled as a simple binary pulse of duration τ_b, so that $h_i(t) = 1$ for $t_i^{\{1..B\}} \le t \le t_i^{\{1..B\}} + \tau_b$, and $h_i(t) = 0$ otherwise (Fig. 2(a)). We use values of $B = 1, 2, 4, 8$, and $\tau_b = 6$ ms. We assume a non-linear form for the RA activation function, given by the sigmoid $f(x) = r_{\max}/(1+e^{-2x/s})$, so $f'(x) = f(x)(r_{\max} - f(x))(2/sr_{\max})$, with $r_{\max} = 600$

Hz and $s = 5$ (s is a parameter that stretches the analog part of the response; the $s \to 0$ limit produces binary responses). In all simulations, the total duration of the simulated song motif is T = 150 ms, and time is discretized with dt = 0.1 ms. The initial HVC-to-RA weights W_{ij} are picked randomly from the interval $[0, 1/B]$, with P=40% of them randomly diluted to zero. The threshold for RA neurons is given by $\theta = 1.2(1 - 0.01P)N_h\tau_b/T$. Each RA neuron projects to one output neuron (i.e., the RA-to-output weight matrix A is block-diagonal), and equal numbers of RA neurons project to each output. The non-zero entries of A are chosen from a Gaussian distribution with mean 1 and standard deviation 1/4. Desired sequences $d_k(t)$ for the output units are fixed by choosing a sequence of steps of 12 ms duration and random heights chosen from the interval $[0, N_r/(8N_o)]$, and are smoothed with a 2 ms linear low-pass filter. The gradient-following rule, Eq.(3.4), is used to update the weights W after each epoch. Most detailed numerical quantities described above were arbitrarily chosen, and simulations did not depend sensitively on the values used.

To study the effects of sparse HVC activity on learning speed, we performed 4 groups of simulations where B, the number of bursts per HVC neuron per song motif, was fixed at $B = 1, 2, 4$, or 8, respectively. In each group, we performed sets of 15 learning trials each: within a set, the overall learning step-size η was fixed, but A and W were drawn randomly, as described above. η was varied systematically across sets. All other parameters, including the desired outputs $d_k(t)$, were kept fixed for all B and all η. By this process, a value of $\eta = \eta^*(B)$ was found for each B that resulted in the fastest learning (defined below) when averaged over the 15 trials; values of η less than η^*_B produced slower learning, while values greater than η^*_B resulted in slower convergence or divergent behavior.

We consider the network to have learned the task when it reaches an error of 0.02 or better (corresponding to $\int dt \sum_k (d_k - o_k)^2 < 1\% \int dt \sum_k d_k^2$, thin horizontal line in Fig. 3; for an example of the output performance in what we consider to be a well-learned task, see Fig. 2(c) where $\int dt \sum_k (d_k - o_k)^2 = 0.15\% \int dt \sum_k d_k^2$).

3.4 Results

Simulations.

We simulated learning by gradient following (as described in Eqs.(3.1) – (3.4) and Methods) in a feedforward network consisting of an HVC, an RA, and a motor output layer, Fig. 1. Sample input (HVC activity) and the initial and desired outputs (for one of two output units) are shown in Figs. 2(a) and 2(b), respectively. In the simulation of Fig. 2, each HVC neuron is active exactly once in the song motif. After several epochs of learning (gradient descent on the mismatch between actual and desired outputs), activity in the output units closely matches the desired outputs, Fig. 2(c). Note that in our model, the RA neurons act as hidden units and their patterns of activity are not explicitly constrained. The activities of three randomly selected RA neurons from the model network after learning is complete are shown in Figs. 2(d)–(f). It is interesting to note that with sigmoid RA activation functions, if initial

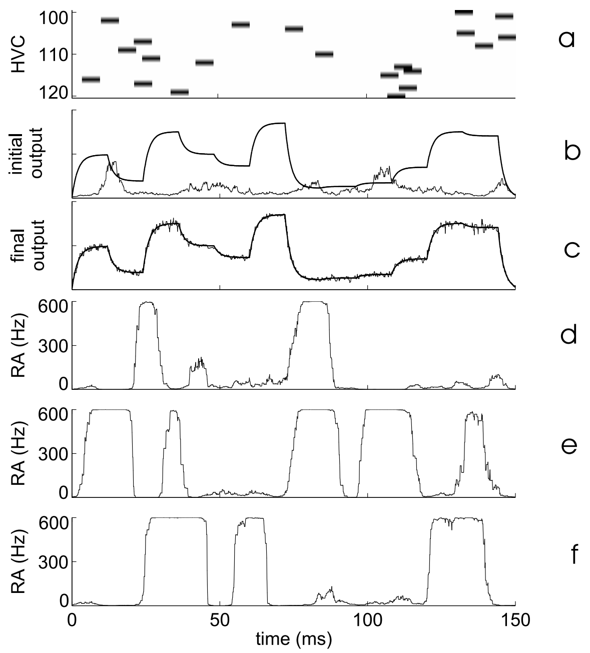


Figure 3.2: An example of the inputs and outputs of the network before and after learning, with one burst per HVC neuron over the entire simulated motif. (a) Activity of RA-projecting HVC neurons as a function of time, shown for 20 of the 500 neurons in the simulation. Black bars indicate that the neuron is bursting at that time, while otherwise the neuron is silent. (b) Desired (*thick line*) and actual (*thin line*) output activity for one of the two output units, before learning begins. (c) Desired (*thick line*) and actual (*thin line*) activity of the same output unit after learning. The second output behaves similarly. (d–f) An example of the activities of three RA units, after learning (see text for further discussion).

connections between HVC and RA are weak and random and if initial RA activity is low, the emergent activity patterns of RA neurons in the trained network qualitatively resemble the behavior of real RA neurons recorded *in vivo* during singing [51] (also A. Leonardo & M.S. Fee, unpublished observations): for example, individual RA unit activity is not well-correlated with the outputs, and similar output features may be driven by rather different patterns of activity in RA.

Our goal is to examine the effects of the sparseness of HVC drive on the learning speed of the network. We repeated the learning simulations, as pictured in Fig. 2, with fixed song length, single-burst length (for HVC neurons), and network size, but varied B, the number of bursts fired per HVC neuron per song motif (see Methods). Fig. 3 shows four learning curves, corresponding to simulations where the number of bursts per HVC neuron is varied. Each learning curve in Fig. 3 is an average over fifteen trials that start with different random intial weights W and A but with a single fixed B $(= 1, 2, 4$, or 8, respectively). The network was considered to have learned the task when the error dropped below a pre-specified error tolerance, signified by the thin horizontal line. For each value of B, the task of learning was realizable, i.e., the network could successfully learn the desired outputs. Also for each B, the overall coefficient controlling the weight-update step-size was optimized to give the fastest learning possible (see Methods). In going from $B = 1$ burst per HVC neuron per motif to 2 bursts, we see in Fig. 3 that the learning time (number of iterations for the learning curve to intersect the learning criterion line) nearly *doubles;* the same happens in going from 2 bursts to 4, or 4 to 8. This apparently strong dependence of learning time on the number of HVC bursts, even for small B, is surprising considering that in all cases (all B) the learning task was realizable, and that the premotor HVC drive in going from $B = 1$ to $B = 2$, for example, was still relatively sparse. To better understand the process by which an increase in the number of HVC bursts per motif leads to slower learning, we turn to an analysis of learning in a linear network.

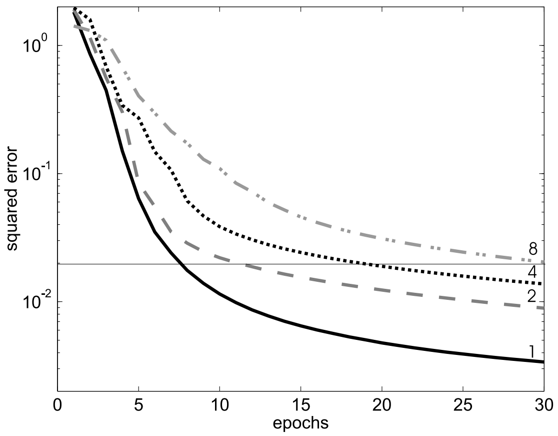


Figure 3.3: Averaged learning curves from repeated learning simulations, as in Fig. 2. The four curves track error as a function of epoch while learning with $B = 1, 2, 4$, and 8 bursts per HVC neuron per simulated song segment. For each B, the overall weight update step size was optimized to give the fastest possible monotonic convergence toward zero error. The number of epochs taken to reach a pre-specified learning criterion (thin horizontal line) grows sharply with B, nearly doubling each time B doubles.

Linear analysis.

We found the basic effect of the slow-down of learning with temporally denser HVC codes to be robust across many changes in network properties, such as network size, length of simulated motif, and choice of RA activation function. To isolate critical factors involved in the learning slow-down, we study the learning curves of a network with the same architecture and learning rule as above, but with linear RA activation functions, $f(y) = y$. Although this is a simplification, a linear network permits us to analytically derive the dependence of the learning curves on B, the number of times each HVC neuron bursts during a song motif. Moreover, linear analysis lends itself to a convenient geometric interpretation of the learning process.

Learning speed and eigenvalues. With linear RA units, the error function C in Eq. (3.3) is a quadratic surface over the multi-dimensional space of HVC-to-RA weights

$\{W\}$ (see Appendix):

$$C = \mathrm{Tr}\{AWQW^TA^T\}, \tag{3.5}$$

where the matrix Q reflects zero-time-lag correlations between HVC neurons, with element Q_{ij} reflecting the cross-correlation in the activity of neurons i and j, summed over all times in the motif. For example, if the two neurons are always co-active, $Q_{ij} = 1$, and if they are never co-active, $Q_{ij} = 0$.

Learning corresponds to moving on the quadratic surface toward minimum error by adjustment of the underlying plastic network weights W. Learning by gradient descent, Eq. (3.4), means that on average the path to the minimum is along the direction of steepest descent: for a linear system, it is well known that components of the total error along different directions in the weight space $\{W\}$ decrease as decaying exponentials with different decay rates. With certain assumptions on the distribution of the fixed weights A (see Appendix), these decay rates are given by ratios of the eigenvalues of Q, $\{\lambda_1, \lambda_2, ..., \lambda_N\}$, with the largest eigenvalue, λ_1. The importance of Q in shaping the error surface emerges from the fact that HVC activity determines which synapses W are active in driving the output, how often they are used, and thus whether and when they must be modified to reduce error. Thus, the learning speed along mode α can be defined as

$$\nu_\alpha = \lambda_\alpha / \lambda_1. \tag{3.6}$$

The larger the learning speeds ν_α, the larger is the decrease in total error per epoch.

It is instructive to consider two cases: (i) all eigenvalues are essentially equal. (ii) all eigenvalues are equal but one, which is very much larger. Generally, the maximal learning-step size must be small enough that a gradient-based step in any direction in weight-space does not cause the error to greatly increase. In (i), the learning speeds along all α will be equal and close to 1; the geometric interpretation is that the error surface is isotropic, Fig. 4(a), and learning can proceed equally rapidly in all directions of the error surface. In (ii), the error surface is strongly anisotropic, Fig. 4(b). Learning will still be fast in the direction corresponding to λ_1; however, the maximum weight-update step size is constrained by this direction, since the error surface is very steep (small changes in weight-space lead to large changes in error) and can quickly lead to divergent error. However, the remaining directions are much shallower, and the small step size constraint leads to much slower learning along all other directions (learning speeds $\lambda_\alpha/\lambda_1 \ll 1$), resulting in a sharp slow-down in the decrease of the cumulative error. Hence, a narrowly distributed range of eigenvalues leads to faster learning, while singularly large eigenvalues that stand out from the rest broaden the range and cause a slow-down.

Mean-field eigenvalue analysis. With Eq. (3.6), the problem of deriving learning curves is reduced to the problem of computing the eigenvalues of the correlation matrix Q. Certain important features of the eigenvalue distribution can be derived from a mean-field matrix $\langle Q \rangle$, obtained by replacing each element of the correlation matrix with its ensemble-averaged expectation value (see Appendix); moreover, $\langle Q \rangle$ elucidates the relationship between features of HVC activity and features of the eigenvalue

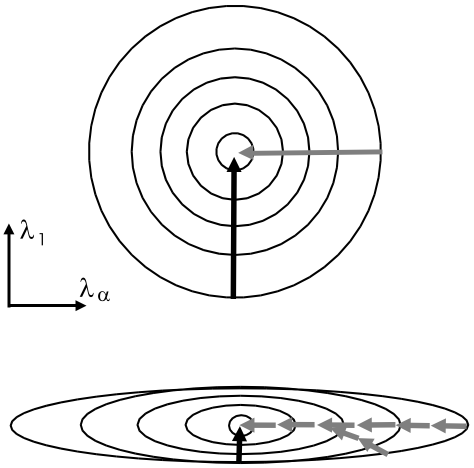


Figure 3.4: A geometric interpretation of learning on a quadratic error function: projection of the error surface onto the underlying weight-space. The ellipses are contours of equal-error, and a varying density of these contours corresponds to varying steepness on the error surface (higher density = steeper). (a) Starting from a given error, the maximum allowable step-size in weight-space is the same regardless of the direction from which the minimum is approached. (b) On an anisotropic surface, the steepest direction (corresponding to the eigenvector with largest eigenvalue, and designated here by λ_1) dictates the maximum allowable step-size in weight space, and constrains learning in all other directions (λ_α) as well.

spectrum. As B is increased, the size of HVC auto-correlations (diagonal elements of $\langle Q \rangle$) increase as B, while the cross-correlations (off-diagonal) increase as a small factor times B^2: these cross-correlations contribute only to the largest eigenvalue of $\langle Q \rangle$. Let $\nu_\alpha(B)$ designate the learning speed along the α^{th} eigenvector of Q as a function of B. The mean-field eigenvalue calculation yields (see Appendix):

$$\nu_1(B) = \nu_1(1) \qquad \text{in steepest direction} \tag{3.7}$$

$$\nu_\alpha(B) = \frac{1}{B}\,\nu_\alpha(1) \qquad \text{all other directions} \tag{3.8}$$

In other words, as B is increased, the learning speeds decrease as $1/B$ along *all* directions in weight space except along the direction corresponding to λ_1. The cumulative initial error generically has significant components in several directions, so the learning time with $B = 2$ will be approximately *twice as long* as for $B = 1$. It is important to note, also, that the effects of increasing B on learning speed should be visible soon after learning has begun and the first transients have passed (learning of the first eigenvector), and not just toward the end of learning, where fine features remain to be learned. This is in good agreement with the overall decrease in learning speeds observed in the simulations of the last section. Moreover, in the linear analysis, the scaling of network speed with B is an essential one (see Appendix): given a fixed network size, motif length, and HVC single-burst duration, increasing the number of bursts per HVC neuron per motif necessarily leads to a reduction in the optimal learning speed for the network (no adjustable parameters to remove this dependence).

The mean-field analysis also sheds light on the identity of the eigenvector with the largest eigenvalue λ_1: it is the *common mode* eigenvector, with all positive entries, that corresponds to a simultaneous increase or decrease, for all parts of the motif, in the summed drive from HVC to the motor outputs. It is intuitive that this is the most “volatile” mode, leading to explosive growth of network activity. The remaining modes are *differential*, allowing rearrangements of the motor drive from moment to moment in the song without a large net change in the mean strength of the drive.

Numerical eigenvalue calculation. The vastly simplified mean-field derivation of the scaling of learning speed with B (from the eigenvalues of $\langle Q \rangle$) neglected variance and other higher-order statistics of Q. To check the results of the analysis, therefore, we numerically compute the eigenvalues of Q from randomly generated HVC activity matrices (see Methods), with N_h=3000, T=300 ms, τ_b = 6 ms, and dt = 0.1 ms; B was varied to be $1, 2, 4$, or 8. The results are shown in Fig. 5, and agree well with the mean-field analysis. In Fig. 5(a), we plot the top 300 $B = 1$ eigenvalues, together with the top 300 $B = 8$ eigenvalues scaled by 1/8. All the eigenvalues for $B = 1$ form a continuum, and the scaled $B = 8$ eigenvalues sit on the same continuum, except for the top eigenvalue, which is much larger than the rest. The gap between the topmost eigenvalue and all the rest for $B > 1$ is better seen in the inset of Fig. 5(a), where the largest eigenvalue scales as B^2, while the second-largest scales as B. This causes learning speeds to scale as $1/B$ (Fig. 6), as derived in Eq. (3.8).

The numerical computation shows that there is some spread in the eigenvalue continuum even with $B = 1$: When HVC neurons burst once per song motif, their activities partially overlap due to their small but non-negligible burst durations, leading to non-zero cross-correlations between neurons and a concomitant spread in the eigenvalue distribution; this would lead to slower learning than if bursts were completely non-overlapping. However, increasing the number of bursts per HVC neuron leads to further correlations in HVC activity and a considerably larger spread of eigenvalues and thus a slower descent on the error surface.

3.5 Discussion.

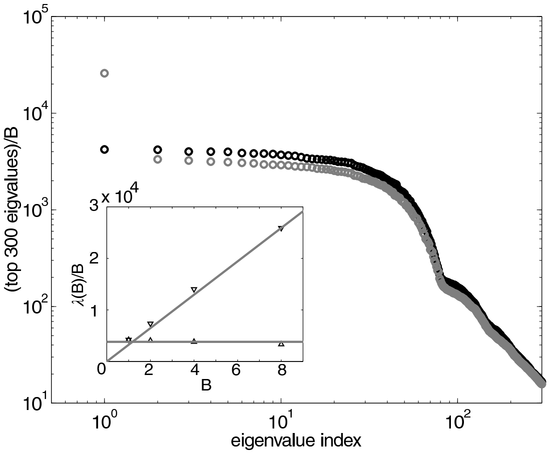


Figure 3.5: Eigenvalues: numerical calculation to check mean-field results. The top 300 eigenvalues of the correlation matrix Q, divided by B, for $B = 1$ bursts per HVC neuron per song segment (*Black* ○), and for $B = 8$ (*Grey* ○). *Inset:* The scaling of λ_1 ($\bigtriangledown$) and λ_2 ($\bigtriangleup$) with B, from numerical calculations. We see that $\lambda_1/B \sim B$, while $\lambda_2/B \sim$ const. Solid lines show the same scaling, derived from $\langle Q \rangle$.

Summary. We have built a simplified framework to analyze the learning of premotor representations in the songbird premotor circuit, given a sparse premotor drive from HVC, a set of plastic connections between HVC and RA, and a gradient learning rule that minimizes the mismatch between the tutor and pupil songs. Within this framework, we have demonstrated how temporally sparse activity allows the fast learning of premotor representations, and quantified, in a network of linear neurons, the dependence of learning rate on the number of times an HVC neuron is active during a motif. Sparsely active HVC neurons have small cross-correlations: increasing the

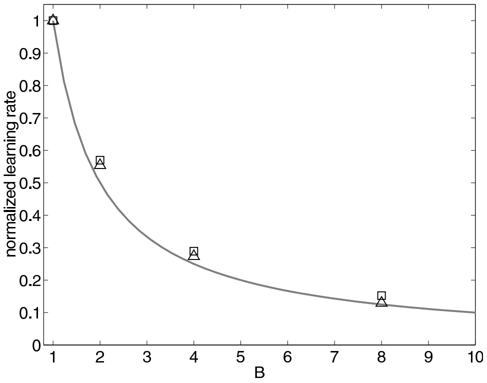


Figure 3.6: Linear analysis: learning speed as a function of B, numerical and analytical results. The scaling of the 1-burst-normalized learning speeds, $\nu_\alpha(B)/\nu_\alpha(1)$ vs B, is shown for modes $\alpha = 2$ ($\triangle$) and $\alpha = 200$ (). These values are obtained from the numerical calculation of the eigenvalues of Q. *Solid line*: the predicted scaling of learning speed with B for all $\alpha > 1$, from the mean-field correlation matrix $\langle Q \rangle$.

number of HVC bursts per motif increases cross-correlations in HVC activity, which leads to correlated changes of synaptic weights. To keep network activity from diverging due to the correlated weight changes, the maximum allowable weight-update size must be constrained; this normalization decreases the step-size for other, uncorrelated weight changes that are required for learning. Hence, the overall learning speed decreases with increasing numbers of HVC bursts per motif. Although our analytical description is based on linear units, the simulations (Fig. 3) of learning in nonlinear units and the heuristic explanation of increased synaptic interference with increased numbers of bursts point to a broader relevance of this analysis to networks with more realistic neurons.

For a similar reason, we expect these sparseness arguments to pertain to other, more biologically plausible learning rules that act to minimize a cost function or error signal by stochastically approximating the gradient, for example, various reinforcement learning type algorithms. In fact, the impetus to increase learning speed through sparse coding may be considerably greater if learning proceeds through reinforcement of random trials because such stochastic gradient algorithms are typically slow.

Relation to past work. Questions about training time in perceptrons have been studied in the machine learning community, resulting in prescriptions to speed up learning by rescaling the learning rate parameter (weight update step size) differently along the different eigenvectors, or by re-parameterizing neuronal activities to make the error surface more isotropic. In a closely related work to this one, LeCun *et al.* [52] in particular recommend that the eigenvector associated with the largest eigenvalue be subtracted from the learning updates, or that symmetrically active $\{-1, 1\}$ units be used in the input layer instead of $\{0, 1\}$ units, to reduce the mean of the off-diagonal entries of the input correlation matrix and hence reduce the anisotropy of the error surface by reducing the distance between the largest and remaining eigenvalues. Since neural firing rates are zero or positive, units in biological networks are necessarily asymmetric. Furthermore, although the learning rate parameter (overall step size) may easily be set at the individual synaptic level, it is not obvious how to apply separate learning rates to separate eigenvectors, since individual synapses participate in multiple eigenvectors. However, we suggest that with the use of unary HVC activity, biology may have found its own solution to these very problems in the case of birdsong learning.

Juvenile HVC activity. Single-unit HVC recordings have only been made in adult birds, where the coding is seen to be unary. We wondered what the role of such sparseness in HVC might be, and found that it could confer a great advantage in terms of learning speed if present during the learning process. On this basis, we predict that the sparseness of HVC activity may be integral to the learning process and should thus already be present in the HVC of juvenile birds instead of arising as an emergent property late in song learning.

Other implications of sparse coding. The arguments presented here add to an existing body of theoretical work arguing for the importance of sparse codes for learning in various neural systems. It is well established that for sequence generation in recurrent networks, relatively sparse activity allows for the reliable storage of longer sequences (higher capacity) [41, 42, 43, 44]; however, the role of HVC in the generation of sequences, and hence the relevance of these findings to the birdsong motor circuit, remains unclear. Temporally sparse coding could also help with the problem of temporal credit assignment in learning, encountered when feedback about a performance arrives significantly later than the neural activities that generated it. Sparse HVC activity could reduce the potential for incorrect associations of multiple bursts of neural activity with an error signal generated by just one of the bursts. Finally, HVC may play an important role not just in motor aspects of birdsong learning and production but in song recognition as well [53, 54, 55, 56], and sparse coding could be an asset for such capabilities in ways not discussed in this work.

3.6 Appendix

Learning curve With linear RA neurons, we define the network equations to be $r = Wh$, $o = AWh \equiv Xh$, where h is the $N_h \times N_s$ matrix of HVC activity, r is the $N_r \times N_s$ matrix of RA activity, o is a $N_o \times N_s$ matrix of output activity, A is the matrix of fixed weights from RA to the outputs, and W is the matrix of plastic weights from HVC to RA. $N_s = T/dt$ is the number of discrete time bins in the motif, where T is the motif length and dt is the grain size. With a change of variables $Q \equiv hh^T$ and $x = X - X^*$, where X^* is defined by $X^*h = d$ (X^* exists if a solution exists, i.e., if the learning task is realizable), the cost-function is $C = \frac{1}{2}\text{Tr}\{xQx^T\}$. Applying a gradient descent update, Eq.(3.4), on C, we have that $x \to (x - \eta AA^TxQ)$, where η is a positive scalar that scales the overall learning step size. If each RA neuron projects to one output unit, and if the summed synaptic weights to each output are approximately equal, AA^T becomes a scalar matrix that can be absorbed into η. Thus, the multilayer perceptron problem with two layers of weights becomes effectively a single-layer perceptron, and we have that after n iterations $x^{(n)} = x^{(0)}(1 - \eta Q)^n$, so

$$C^{(n)} = \frac{1}{2}\text{Tr}\{x^{(0)}(1 - \eta Q)^n Q(1 - \eta Q^T)^n x^{(0)T}\}. \tag{3.9}$$

In the eigenvector basis of the Hessian matrix Q (eigenvalues $\{\lambda_{\alpha=1,\ldots,N_h}\}$ arranged in non-increasing order, $\lambda_1 \geq \ldots \geq \lambda_{N_h}$), with projection of the k^{th} row of $x^{(0)}$ along the α^{th} eigenvector given by $\chi_{k\alpha}$, the error after n learning iterations is given by

$$C^{(n)} = \frac{1}{2}\sum_{\alpha,k}(1-\eta\lambda_\alpha)^{2n}\lambda_\alpha|\chi_{k\alpha}|^2. \tag{3.10}$$

Let $c_\alpha \equiv \lambda_\alpha \sum_k |\chi_{k\alpha}|^2$ represent the initial error along the αth eigenvector. The total error evolves iteratively via multiplication of the initial errors c_α by a factor $(1-\eta\lambda_\alpha)^2$ per iteration; η must be chosen small enough so that $|1-\eta\lambda_\alpha| < 1$ for each α, to allow error to decrease and for the learning curve of Equation (3.10) to converge. Hence, η must be less than $2/\lambda_1$, and it is easy to see that the optimal choice for η is $\eta^* = 1/\lambda_1$ ($\eta < 1/\lambda_1$ leads to over-damped convergence, while $1/\lambda_1 < \eta < 2/\lambda_1$ displays under-damped oscillatory convergence).

Analysis of eigenvalues The mean-field matrix $\langle Q\rangle$ is formed by replacing all elements of Q by their ensemble-averaged expectation values (i.e., generate Q and average, element by element, over several trials). Therefore, $\langle Q\rangle = BN_b\mathrm{I}+(B^2N_b^2/N_s)\mathbf{1}\mathbf{1}^T$, where $N_b \equiv \tau_b/dt$. There are only two distinct eigenvalues, $\lambda_1 = BN_b+B^2N_b^2(N_h/N_s) \approx B^2N_hN_b(\tau_b/T)$ (provided $N_h\tau_b/T \gg 1$) corresponding to the common mode eigenvector, and $\lambda_2 = BN_b - B^2N_b^2/N_s \approx BN_b$ (provided $T \gg B\tau_b$) corresponding to the N_h-1 differential modes. (This effect, of a continuous eigenvalue spectrum with a single large eigenvalue, is generic for N × N matrices with random entries of mean a on the diagonal and b on the off-diagonal, if $b \gg a/N$ [58].) Hence the learning rate for all modes $\alpha > 1$ is given by $\nu_\alpha = (1/B)(T/N_h\tau_b) \sim 1/B$, as in Eq. (3.8), Figure 6.

Chapter 4

A biologically plausible learning rule for reward optimization in networks of conductance-based spiking neurons

It is not known how animals learn to perform complex goal-oriented tasks. Behavioral and neurophysiology studies as well as common experience indicate that animals can be taught to perform such tasks with the help of simple reinforcing rewards. Numerous learning rules exist in the machine learning literature that perform reward optimization; however, they cannot be applied to networks with biologically realistic neurons interacting through voltages and conductances. In this paper we present a biologically plausible rule for synaptic learning in recurrent networks of spiking, voltage and conductance based neurons, that can provably optimize reward under fairly general conditions.

4.1 Introduction

In machine learning, since several techniques exist to perform optimization, it has proved fruitful to transform the goal-directed learning problem into one of reward maximization by equating closeness to the goal with reward. Some of these approaches (e.g., backpropogation) rely on the direct computation of gradients to maximize reward. Such direct-gradient techniques cannot be applied to neural networks with conductance-based spiking model neurons, where, because of their rich temporal dynamics, the relevant gradients cannot be expressed explicitly. If the learning task involves any form of motor control, reward would depend on the neural–motor transformation; thus, learning depends on knowledge of many gradients that are impossible to compute explicitly. An alternative approach to this problem is stochastic gradient ascent, where the appropriate gradients are estimated by measuring the impact on the reward of perturbations in the system parameters; the learning rule moves the system parameters along the stochastically measured local gradient. While some concrete rules for learning by stochastic gradient computation have been formulated and shown to perform gradient ascent, few rules have been articulated on the synaptic or single-neuron level that are both biologically realistic and capable of provably performing reward optimization.

Our goal in this paper is to provide a biologically plausible and demonstrably workable learning rule that can optimize reward. The learning rule proposed here can be placed in the general category of REINFORCE algorithms [49], and it applies to arbitrary networks of conductance-based spiking neurons. It relies on the correlation of reward with synaptic noise and presynaptic inputs. Heuristically, the rule works in the following way: if greater-than-average excitatory noisy synaptic input to a neuron at some time correlates positively with reward, increase the excitatory synaptic weights of all inputs to that neuron which were active at that time. Conversely, if all lower-than-average excitatory input to a neuron at one time correlates with reward, reduce the activity of that neuron for that time by lowering the strength of all participating inputs to it. Thus, noisy inputs act as a source of exploration, to probe the effects of changing neural activity on reward; the correlation information is used by the learning rule to make changes in the network weights that reinforce patterns of neural activity that lead to high reward, and to weaken patterns of neural activity that lead to low reward.

In the next section, we present the formal learning rule; in sections two and three, we prove in a network of recurrently connected excitatory conductance based neurons that learning follows the gradient of the expected reward, and in section four we generalize the learning rule to apply in networks with excitatory and inhibitory neurons, as well as in networks with noisy currents instead of noisy conductance inputs.

4.2 Learning rule

We consider learning in a system with three central elements: a recurrent network of spiking neurons, a noise generator that perturbs the activities of the neurons in the recurrent network, and an evaluator that provides a reward signal R based on the network output.

The prescribed learning rule is given by

$$\Delta W_{ij} = \eta R \int_0^T dt\, \left(\xi_i(t) - \langle \xi_i(t) \rangle\right) s_j(t). \tag{4.1}$$

W_{ij} is the weight of the $i \leftarrow j$ synapse from neuron j to neuron i, and η is a positive scalar learning rate parameter. Learning updates are made following each episode of network activity that lasts from time 0 to T. The reward R is some functional of the neural activities that occurred in this interval. Each neuron in the network receives, in addition to feedforward or recurrent synaptic inputs from the rest of the network, a time-dependent noise input, $\xi_i(t)$. Neural activity is characterized by neural voltages, $V_i(t)$, synaptic activations $s_i(t)$ resulting from presynaptic spikes, and synaptic conductances, $g_i(t)$. The dynamics of these variables are summarized in the following equations:

$$\begin{aligned}
s_i(t) &= f_s\big(V(t'), s(t') \;\; \forall t' < t\big) && (4.2)\\
g_i(t) &= \sum_j W_{ij} s_j(t) + b_i(t) + \xi_i(t) \;\equiv g_{i,det}(t) + \xi_i(t) && (4.3)\\
\frac{dV_i(t)}{dt} &= -g_L\big(V_i(t) - V_L\big) - g_i(t)\big(V_i(t) - V_E\big) - \sum_\alpha g_i^\alpha(t)\big(V_i(t) - V_\alpha\big) && (4.4)\\
&\equiv V_{i,det} - \xi_i(t)\big(V_i(t) - V_E\big) && (4.5)
\end{aligned}$$

Dynamic variables without a subscript are vectors that represent the state of all neurons in the network at time t: for example, $V(t) = \{V_1(t), ..., V_{\mathrm{N}}(t)\}$ is the vector of neural voltages at t. Variables with no subscript and no time index are matrices that represent the states of all neurons for all times in an episode, so that

$\xi = \{\xi_i(t) | i = 1N, t \in [0, T]\}$.

Equations (4.2– 4.4) are quite general and can be used to describe the dynamics of various conductance-based neurons, including Hodgkin–Huxley and integrate-and-fire models. The voltage equation includes contributions from an intrinsic neural leak conductance, g_L, synaptic input conductances, g_i, and fast non-linear ion-channel conductances, g_i^α, responsible for spiking dynamics.

The goal of learning is to maximize reward; the learning rule proposed above does this by collecting information, via the correlation of stochastic inputs with reward, on how reward depends on network parameters. More simply, if greater than average excitatory noise input to a neuron leads to a large reward, increase the weights of all recently active synaptic inputs to that neuron. On the other hand, if lower than average excitatory noise input to a neuron leads to a large reward, decrease the weights of all recently active inputs to than neuron. While the weight updates improve the average performance of the network, the weight change of an individual synapse in one trial is stochastic, though slightly biased in the direction of improving performance. From this description, it seems like the learning rule is hill-climbing on the reward surface, but to guarantee this property in arbitrary networks would require a proof.

Here, we prove that the learning rule of Equation (4.1) performs stochastic gradient ascent on the reward in arbitrary networks of spiking neurons, under the following conditions:

- If the noise ξ is Gaussian, we prove, for arbitrary networks, that the learning rule performs stochastic gradient ascent on the reward.

- More generally, for arbitrary, spatially and temporally uncorrelated noise $\xi_i(t)$, we prove within a linear approximation that the learning rule performs stochastic gradient ascent on the reward. We show that this is still true if the injected noise has some temporal correlation.

4.3 Arbitrary noise

Our strategy in this section is to start with the stated learning rule, Equation (4.1), and average over noise to derive the expected weight change. With some linearization and algebraic manipulation, we see that the expected weight change is proportional to the gradient of the noise-averaged reward.

Let the network dynamics be given by Equations (4.2–4.4). We assume that the noise inputs $\xi_i(t)$ are random variables drawn from an *arbitrary* distribution or process, but are spatially uncorrelated; that is, $(\xi_i(t) - \langle\xi_i(t)\rangle)(\xi_j(t) - \langle\xi_j(t)\rangle) = \delta_{ij}K(t - t)$, where K is a normalized correlation function, $\int K(\tau)d\tau = 1$. For example, $\xi_i(t)$ could be the synaptic activation of a neuron firing a poisson spike train. In this case, since ξ is derived by temporally filtering a neural spike train, it would have clear temporal correlations.

Let $\langle\,\rangle$ represent noise-averages over ξ. The expected weight update is

$$\langle\Delta W_{ij}\rangle = \int d\xi\, p(\xi) \left(R \int_0^T dt \big(\xi_i^t - \langle\xi_i^t\rangle\big)\, s_{ij}^t \right) \tag{4.6}$$

$$= \int_0^T dt \int d\xi\, p(\xi)\, C_{ij}^t\, \big(\xi_i^t - \langle\xi_i^t\rangle\big), \tag{4.7}$$

where we have defined $C^t_{ij} \equiv Rs^t_{ij}$. Note that C^t_{ij} is a functional of ξ. Taylor expanding $C^t_{ij}[\xi]$ to first order around $\langle\xi\rangle$, the average weight update becomes

$$\begin{aligned} \langle \Delta W_{ij} \rangle &= \int_0^T dt \int d\xi \, p(\xi) \left[C^t_{ij}[\langle\xi\rangle] + \sum_k \int dt' \frac{\delta C^t_{ij}[\langle\xi\rangle]}{\delta b^{t'}_k} (\xi^{t'}_k - \langle \xi^{t'}_k \rangle) \right] (\xi^t_i - \langle \xi^t_i \rangle) \\ &= \int_0^T \int_0^T dt dt' \frac{\delta C^t_{ij}[\langle\xi\rangle]}{\delta b^{t'}_i} K(t-t') \qquad (4.8) \\ &\approx \int_0^T \int_0^T dt dt' \frac{\delta \langle C^t_{ij} \rangle}{\delta b^{t'}_i} K(t-t'). \qquad (4.9) \end{aligned}$$

The last line follows from the expression above it only within a linear approximation, i.e., if C^t_{ij} depends linearly on ξ.

Temporally uncorrelated noise. If ξ is temporally uncorrelated so that $K(t-t') = \delta(t-t')$, we see that

$$\begin{aligned} \langle \Delta W_{ij} \rangle &= \int_0^T dt \frac{\delta}{\delta b^t_i} \langle R \, s^t_{ij} \rangle \qquad (4.10) \\ &= \frac{\partial \langle R \rangle}{\partial W_{ij}}. \qquad (4.11) \end{aligned}$$

Equation (4.11) follows from direct application of the sensitivity lemma (Lemma 2). So, in the special case of spatially and temporally uncorrelated noise injections, we have proved that the learning rule on average performs gradient ascent on the expected reward.

Temporally correlated noise. If ξ is temporally correlated, $K(t-t') \neq \delta(t-t')$; defining $\langle C^t_{ij} \rangle \equiv f(t)$ and making the change of variable $\tau = t - t'$,

$$\langle \Delta W_{ij} \rangle = \int_0^T dt' \frac{\delta}{\delta b_i^{t'}} \int d\tau \; f(t'+\tau) \; K(\tau). \tag{4.12}$$

K is an autocorrelation function, so it peaks at $\tau = 0$ and is symmetric. Thus, K windows f, the function in the integrand, around $\tau = 0$; if correlations are short, K decays rapidly away from $\tau = 0$, and the window is narrow. We Taylor expand $f(t' + \tau)$ about $\tau = 0$, and change notation so that $t' \to t$, to get

$$\begin{aligned} \langle \Delta W_{ij} \rangle &= \int_0^T dt \frac{\delta}{\delta b_i^t} \int d\tau \left\{ f(t) + \left. \frac{\partial f}{\partial \tau} \right|_{\tau=0} \tau + \frac{1}{2} \left. \frac{\partial^2 f}{\partial \tau^2} \right|_{\tau=0} \tau^2 \right\} K(\tau) &(4.13) \\ &= \int_0^T dt \frac{\delta f(t)}{\delta b_i^t} + \frac{\overline{\tau^2}}{2} \int_0^T dt \frac{\delta f''(t)}{\delta b_i^t}. &(4.14) \end{aligned}$$

We assume K is normalized, $\int d\tau \; K(\tau) = 1$; the term involving $\int d\tau K(\tau)\tau$ vanishes because K is symmetric. The constants $\overline{\tau^n} \equiv \int d\tau \; \tau^n \; K(\tau)$ reflect the temporal characteristics of the noise ξ (say for example that ξ has correlations on the time-scale of T_ξ, with autocorrelation function $K(\tau) = \frac{1}{T_\xi} e^{-|\tau|/T_\xi}$; then $\overline{\tau^2}$ reflects this information, because $\overline{\tau^2} = 2T_\xi^2$). Substituting $f(t) = \langle C_{ij}^t \rangle = \langle R s_j^t \rangle$, we see at once that, like Equation (4.10), the first term of equation (4.14) is the gradient of the expected reward:

$$\langle \Delta W_{ij} \rangle = \frac{\partial \langle R \rangle}{\partial W_{ij}} + \frac{\overline{\tau^2}}{2} \int_0^T dt \frac{\delta}{\delta b_i^t} \frac{\partial^2 \langle R s_j^t \rangle}{\partial t^2} \tag{4.15}$$

The second term reflects deviations, due to temporal correlations in the injected noise, from the true gradient. Thus, for the weight updates to approximately follow the gradient of the reward, the second (non-gradient) term should be much smaller than the first (gradient) term. We see from Equation (4.14) that for this to be true, it is sufficient (but not necessary) to have $f''(t)\frac{\overline{\tau^2}}{2} \ll f(t)\ \forall t$, or in other words,

$$\frac{1}{\overline{\tau^2}} \gg \frac{\frac{\partial^2 \langle Rs_j^t \rangle}{\partial t^2}}{2\langle Rs_j^t \rangle}. \tag{4.16}$$

This expression provides bounds within which, even if correlations are present in the injected noise, deviations of the weight update away from the gradient of the expected reward can be neglected. If the condition in Equation (4.16) is satisfied, we have once again proved that the learning rule on average performs gradient ascent on the expected reward.

4.4 Gaussian noise

In this section we show, without a linear approximation, that if the noise ξ is gaussian, learning proceeds by stochastic gradient ascent on the expected reward. Our strategy is to compute the expected, or noise-averaged reward, and use the gradient of this expression to derive a weight update rule, which we show is proportional to our proposed learning rule of Equation (4.1).

To perform gradient following, a stochastic learning rule must obey

$$\langle \Delta W_{ij} \rangle = \frac{\partial \langle R \rangle}{\partial W_{ij}} = \frac{\partial}{\partial W_{ij}} \sum_{g,V} R[V] P(g,V) \tag{4.17}$$

$$= \sum_{g,V} R[V] \frac{\partial}{\partial W_{ij}} P(g,V) \tag{4.18}$$

$$= \langle R \frac{\partial \log P}{\partial W_{ij}} \rangle. \tag{4.19}$$

Learning rules of the form $\Delta W_{ij} = R \frac{\partial \log P}{\partial W_{ij}}$ will clearly satisfy the condition above, and belong to the general category of REINFORCE [49] stochastic gradient algorithms. To explicitly formulate such a learning rule, we must compute $\partial \log P / \partial W_{ij}$ for our network.

We assume that the network dynamics are as given in Eqs. (4.2 –4.4). Since $\xi_i(t)$ are spatially and temporally uncorrelated, and are uncorrelated with the network state, we can decompose $\log P(g,V)$ into a sum of the conditional probabilities over different neurons and past times:

$$\log P(g,V) = \sum_{i=1}^{N} \sum_{t=0}^{T} \log P(g_i(t); V_i(t) | V(t'), g(t'), t' < t). \tag{4.20}$$

Using the definitions of $g_{i,det}(t)$, $V_{i,det}(t)$ in Equations (4.3) and (4.5), we may rewrite the above as $\log P(g,V) = \sum_{i=1}^{N} \sum_{t=0}^{T} \log P(g_i(t), V_i(t) | g_{i,det}(t), V_{i,det}(t))$. Since the difference between $V_i(t)$ and $V_{i,det}(t)$ is $\xi_i(t)\big(V_i(t) - V_E\big)$, the log-probability becomes $\log P(g,V) = \sum_{i=1}^{N} \sum_{t=0}^{T} \log P\big(\xi_i(t) = -\frac{V_i(t) - V_{i,det}(t)}{V_i(t) - V_E} \big| V_{i,det}(t)\big)$. Therefore,

$$\begin{aligned}
\frac{\partial \log P}{\partial W_{ij}} &= \int dt \frac{\delta}{\delta W_{ij}^t}\Big(\sum_{k,u} \log P\big(\xi_k(u) = -\frac{V_k(u) - V_{k,det}(u)}{V_k(u) - V_E} | V_{k,det}(u)\big)\Big) \\
&= \int dt\ \big(\log P(\xi_k(u))\big)'\ \frac{1}{V_k(u) - V_E} \frac{\delta V_{k,det}(u)}{\delta W_{ij}^t} \qquad (4.21) \\
&= \frac{1}{\sigma_\xi^2} \int dt\ \big(\xi_i(t) - \langle \xi_i(t) \rangle\big)\ s_j(t) \qquad (4.22)
\end{aligned}$$

where the last line follows because ξ is Gaussian distributed, and because $\delta V_{k,det}(u)/\delta W_{ij}^t = \delta_{ik}\delta_{tu}\big(V_i(t) - V_E\big)\ s_j(t)$. Therefore, learning will follow the gradient of the expected reward if the update rule is

$$\Delta W_{ij} = R \frac{\partial \log P}{\partial W_{ij}} = R \int_0^T dt\ \big(\xi_i(t) - \langle \xi_i(t) \rangle\big)\ s_j(t). \qquad (4.23)$$

This the stated learning rule of Equation (4.1), completing the proof that with Gaussian noise inputs in an arbitrary recurrent network of conductance based neurons, the learning rule performs stochastic gradient ascent on the reward.

4.5 Generalization of the learning rule

Here we describe how learning is modified if the network contains both excitatory and inhibitory neurons, or if noise enters through currents instead of through conductances. The proofs are straightforward, as they closely follow the derivations provided above.

4.5.1 Learning with excitatory and inhibitory conductances

In general, neural networks contain both excitatory and inhibitory neurons, and it is thought that inhibitory connections may also be plastic. Furthermore, the synaptic noise inputs could also be excitatory or inhibitory. We generalize the learning rule of Equation (4.1) so it applies to networks with excitatory and inhibitory neurons.

The new learning rule is essentially the same as Equation (4.1), but modified by a ± 1 constant prefactor, that dictates the sign of synaptic change based on the reversal potentials of the presynaptic input and the noise:

$$\Delta W_{ij} = k_{\xi i} k_j R \int_0^T dt \; (\xi_i(t) - \langle \xi_i \rangle) s_j(t). \tag{4.24}$$

The constant $k_{\xi i} = 1$ if the noise synapse to neuron i is excitatory, and is -1 if it is inhibitory; similarly, $k_j = 1$ if the presynaptic input from neuron j is excitatory, and is -1 if it is inhibitory. This learning rule performs hill-climbing on the reward function. In other words, the prescribed weight updates drive the network toward nondecreasing reward, in a direction that is always within 90 degrees of the true gradient of the expected reward.

To prove this, we rewrite Equations (4.3–4.4) to include both types of conductances:

$$
\begin{aligned}
s_i(t) &= f_s\big(V(t'), s(t')\ \forall t' < t\big) && (4.25)\\
g^E_{i,det}(t) &= \sum_{j \in E} W_{ij} s_j(t) + b^E_i(t) && (4.26)\\
g^I_{i,det}(t) &= \sum_{j \in I} W_{ij} s_j(t) + b^I_i(t) && (4.27)\\
\frac{dV_i(t)}{dt} &= -g_L\big(V_i(t) - V_L\big) - g^E_i(t)\big(V_i(t) - V_E\big) - g^I_i(t)\big(V_i(t) - V_I\big) - \sum_{\alpha} g^{\alpha}_i(t)\big(V_i(t) - V_{\alpha}\big)
\end{aligned}
$$

where $g^E_i(t) = g^E_{i,det}(t) + \xi_i(t)$ and $g^I_i(t) = g^I_{i,det}(t)$ if the noise injected into neuron i is excitatory ($V_{\xi,rev} = V_I$); conversely, $g^E_i(t) = g^E_{i,det}(t)$ and $g^I_i(t) = g^I_{i,det}(t) + \xi_i(t)$ if the noise injected into neuron i is excitatory ($V_{\xi,rev} = V_E$). Closely following the derivation of Section 4.4, the learning rule for stochastic gradient ascent on the reward becomes:

$$
\Delta W_{ij} = R \int_0^T dt\, \frac{\big(V_{j,rev} - V_i(t - \Delta t)\big)}{\big(V_{\xi,rev} - V_i(t - \Delta t)\big)} (\xi_i(t) - \langle \xi_i \rangle) s_j(t). \tag{4.28}
$$

where $V_{j,rev} \in \{V_I, V_E\}$, depending on whether the presynaptic neuron j is inhibitory or excitatory. We notice that the signs of $\big(V_{\xi,rev} - V_i(t - \Delta t)\big)$ and $\big(V_{j,rev} - V_i(t - \Delta t)\big)$ stay the same at all times, because $V_I \leq V(t) \leq V_E$ at all times. Thus, Equation (4.28) differs from the learning rule of Equation (4.24) by a factor of $\frac{\left|V_{j,rev} - V_i(t - \Delta t)\right|}{\left|V_{\xi,rev} - V_i(t - \Delta t)\right|}$ multiplied into the eligibility at each time step. Since the stated learning rule differs from the expression for stochastic gradient ascent on the reward only by multiplication of a positive vector, it follows that the learning rule of Equation (4.24), for networks with excitatory and inhibitory conductances, stays within 90 degrees of the gradient, and consequently performs hill-climbing on the reward.

4.5.2 Noise in currents

If the postsynaptic neuron is perturbed by noisy currents instead of conductances, the expression to move along the reward gradient is slightly modified to include a postsynaptic voltage dependence,

$$\Delta W_{ij} = R \int_0^T dt\ \big(V_{j,rev} - V_i(t - \Delta t)\big)\big(\xi_i(t) - \langle \xi_i \rangle\big) s_j(t). \tag{4.29}$$

In analogy with previous section, we define the actual learning rule to be:

$$\Delta W_{ij} = k_j R \int_0^T dt\ \ \big(\xi_i(t) - \langle \xi_i \rangle\big) s_j(t). \tag{4.30}$$

Once again, this learning rule drives the network uphill on the reward surface, always within 90 degrees of the gradient.

4.6 Discussion

In summary, we have proposed a synaptic learning rule for recurrent networks of voltage and conductance based spiking neurons that demonstrably performs stochastic gradient ascent on a reward function. Synapses are only required to maintain a local eligibility trace of recent past local activity; plasticity in these synapses is governed by the product of this local eligibility trace with a global reward signal.

Another quality of this learning rule is that it is able to weather complex, short time-scale dynamics such as facilitation and depression in synapses; learning still provably follows the gradient of the reward.

The form of the proposed learning rule makes it possible to generate specific predictions for neurophysiology. There are two conditions for synaptic change to take place: (1) the presynaptic neuron must have been recently active; (2) presynaptic neural activity must be followed by reward. The sign of synaptic change is instructed by the noise synapse onto the postsynaptic neuron: if the noise input to the postsynaptic neuron was active at the time of the presynaptic input, then the synaptic weight would increase. On the other hand, quiescence of the postsynaptic noise input at the time of the presynaptic input, if followed by reward, would lead to a decrease in the synaptic weight. Thus, to gain a direct handle on plasticity, the synaptic physiologist would need to identify the reward signal, and be able to independently control activity in the presynaptic neuron and the postsynaptic noise input.

This learning rule requires the confluence of three signals at the postsynaptic neuron: presynaptic input, a unique noise input, and reward. The cerebellum, with convergent circuitry from climbing fibers and parallel fibers onto Purkinje cells, is suggestive of such a confluence. Climbing fibers are traditionally thought to convey error signals from the inferior olive to the Purkinje cells, while parallel fibers undergo plasticity; however, it is possible the role of climbing fibers is noise injection, not error transmission. Similarly convergent circuitry is found in the song-related premotor area RA of songbirds. RA receives inputs from two brain areas, HVC and LMAN. HVC–RA synapses are crucial for song production and thought to be plastic, while the LMAN–RA synapse is necessary for songlearning, but not for song production. It would be interesting to reexamine data on cerebellar and songbird area RA plasticity from the point of view of this learning rule.

Perturbative network learning depends on the correlation of noise with reward. Typically, stochastic gradient rules extract this correlation from the product of reward with an eligibility trace that has zero mean. If the estimation of the average noise is biased, the eligibility ceases to have zero mean, and the product of reward with eligibility drives learning in a biased direction, away from the gradient of the reward. If the bias is large compared to the size of the correlations between synaptic noise and reward, the network can go seriously astray. The learning rule proposed in this paper is subject to the same requirement as other stochastic gradient rules, that the subtraction of the mean noise be accurate. In this regard, the learning rule proposed here has a distinct advantage over two recently proposed noise-based gradient rules for biological neural networks [31, **?**], because the noise inputs in our model are distinct from the network being trained, and hence are uninfluenced by the dynamics of the network or by the changing network parameters. For this reason, it is possible for an individual neuron to maintain a stochastic but unbiased estimate of its input noise by locally time-averaging the noise input. In contrast, the noise variable is the plastic variable in the model of [31] and thus its mean could be changing systematically with learning; in the model of [**?**], the noisy variable is the postsynaptic neural spike train, which can vary on fast time scales due to ongoing network dynamics or time-varying non-noise inputs; thus, attempts to compute the noise by time-averaging in these models could be problematic and lead to biased estimates of the noise.

Mathematically, this learning rule is analogous to node perturbation. In fact, if adapted to linear, rate based units, it becomes exactly node perturbation. A common criticism leveled at learning rules based on the correlation of noise with reward is that since these correlations are small in large networks, the learning time increases with increasing network size. Potentially, the learning time for a demanding task whose complexity scales up with network size, could grow as N, the number of neurons receiving independent noise inputs. In comparison, learning rules similar to weight-perturbation would scale up with NM, where NM is the total number of plastic synapses in the network [59]. Not all tasks scale up in complexity with network size, however. In fact, many artificial neural network problems have been shown with principal components analysis to depend on the learning of only a few modes, and we show in certain biological problems, such as birdsong acquisition, that learning time with stochastic gradient rules can be quite fast, and independent of the size of hidden layers in the network [Fiete and Seung, unpublished].

We showed conditions under which temporal correlations in the injected noise will and will not affect the ability of the learning rule to perform gradient ascent on the reward. Intuitively, temporal correlations in the noise will not adversely affect learning if the task itself has broad, or slowly varying, temporal characteristics, compared to the correlation time of the noise. For the same reasons, we expect that patterns of spatial correlation in the noise should not adversely affect learning if the task can be learned if the groups of neurons receiving correlated input become correlated in activity. For example, if a group of neurons sends convergent projections to a single output unit, it would not affect the learning task adversely if all neurons in the group received correlated noise. In general, however, it is possible for the network task to be demanding enough so that individual neural activities have to be learned precisely, and in this case spatially correlated noise would slow down or even prevent learning.

Finally, there is no reason for the noise statistics to be constant: it is possible to imagine that the particular noise statistics and spatial and temporal noise correlations could themselves be learned and could be adapted to increase the learning speed, depending on the task that needs to be learned. If noise adapts on a slow time scale, that would still allow for baseline noise estimation by time-averaging, and could also speed up learning. What tasks would benefit from the adaptation of noise statistics, and from the introduction of noise correlations, remains an open question for future study. In general, much work remains to be done in the study of ‘meta-learning, the designing of networks that can learn how to learn.

Another set of open questions for future work has to do with exploring the mathematical relationship between seemingly different learning rules, for example, might there be a regime where noise-based reinforcement learning is analogous to a more traditional Hebbian or spike-time dependent learning algorithm? Hebbian learning rules, even when gated by reward, have not been shown to provably perform reward optimization. If some Hebbian-like rules are empirically found to be able to perform such reward-driven learning in particular networks, then the mathematical gateway for proving reward optimization might be through proving equivalence with this or other existing stochastic gradient descent rules.

Chapter 5

Rapid learning of birdsong by reinforcement of variation in spiking networks

5.1 Abstract

The adage that "practice makes perfect" refers to the gradual learning of complex skills through an iterative process of trial, error, and improvement. Here we hypothesize that such behavioral learning is based on blind variation, in the form of noisy neural inputs, and selective reward-based modification, guided by the correlation of the noise inputs with a global reinforcement signal. The proposed rule is an example of stochastic gradient learning, which is often criticized as too slow to be biologically plausible. We demonstrate that, contrary to conventional wisdom, the learning time in a test case of birdsong acquisition in a spiking network model with delayed reward is fast enough to be compatible with that of real zebra finches. This is because learning time scales up with the complexity of the task to be learned, and not with the size

of the network; since the information content of song is reasonably small, learning proceeds rapidly. Our results suggest that stochastic gradient learning can, in at least some contexts, be fast enough to be taken seriously as a biological mechanism.

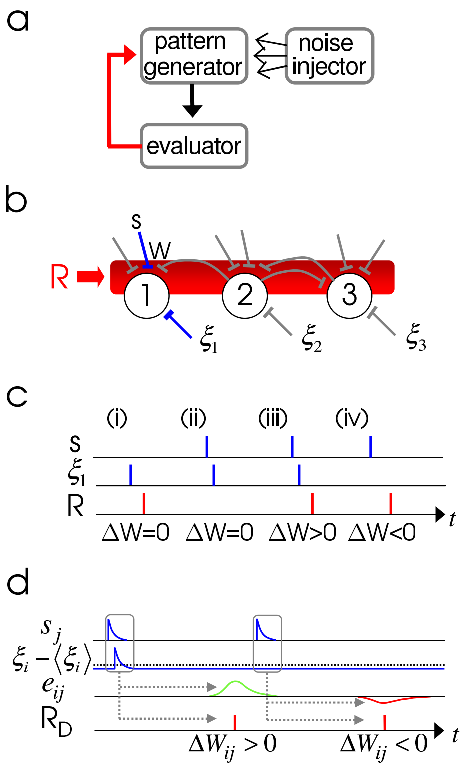
At the abstract level, our model of motor learning involves interactions between three component subsystems: a pattern generator, an evaluator, and a noise injector (Figure 1a). The pattern generator produces motor output that, due to input from the noise injector, fluctuates from trial to trial. When the evaluator rates the output as better than average, the pattern generator receives positive reinforcement, making that output more likely to occur in the future. By this process of trial and error, the average performance, as measured by the evaluator, improves over time. The goal of this paper is to make the preceding abstract model, standard in the field of reinforcement learning [60, 28, 49], relevant to neurobiology.

At the neural level, our model contains a pattern generator that is a network of integrate-and-fire neurons. Each neuron receives a single noise synapse, in addition to regular synapses from other neurons in the pattern generator (Figure 1b). Each noise synapse is driven by a random spike train from a corresponding neuron in the noise injector. All regular synapses are assumed to receive a global reinforcement signal from the evaluator. The present work takes the noise and reinforcement signals as given, and focuses on the question of how the pattern generator uses these two signals to iteratively improve its performance over repeated trials.

We propose that this is accomplished by a synaptic learning rule that is driven by the correlation between presynaptic spiking, local noise, and the global reinforcement signal (Figure 1c). When followed by reinforcement, the noise is treated as a teaching signal that instructs active synapses to change. Positive (negative) noise, if followed by reinforcement leads to the strengthening (weakening) of active synapses. Put another way, if greater than expected activity of a neuron leads to reward, the learning rule raises the activity of that neuron and hence increases the expected reward by strengthening inputs to that neuron. Similarly, if lower-than-expected neural activity correlates with reward, this would lead to the weakening of synapses to that neuron.

Due to the rudimentary, correlation-based nature of this synaptic learning rule, individual neurons do not require or receive information, except what is conveyed in the single global reward signal, about downstream network activity and connectivity or about the transformations from neural activity to physical motor output to reward. This rule is similar in concept to learning by "node perturbation" [61, 60], and more distantly related to "weight perturbation" and other stochastic gradient learning rules for artificial neural networks [28]. In detail, the present rule is novel because it is applicable to biophysically realistic voltage and conductance based model neurons.

Although noise followed by reinforcement is treated as an instructive signal, the instruction is in the correct direction only on average; often the prescribed change is in the "wrong" direction for individual synapses. In fact, in a network that contains hundreds or thousands (N_ξ) of noisy variables, the correlation of reward with any one independently fluctuating variable is small ($\sim 1/N_\xi$); to measure this correlation accurately would require averaging over a large number ($\sim N_\xi$) of iterations.



$$\Delta W_{ij} = R_D \left(\left(\xi_i^t - \left\langle \xi_i^t \right\rangle \right) s_j^t \right)_D$$

Figure 5.1: The learning rule. **a** A schematic view of learning by blind variation and selective reward-based modification: a pattern generator produces a sequence of outputs, and a noise injector perturbs activity in the pattern generator from moment to moment and iteration to iteration. Based on the noisy outputs, an evaluator generates a reward signal, which it broadcasts back to the pattern generator. This reinforcement signal is used by the pattern generator to modify its behavior in a way that increases the expected reward. **b** In a neural implementation, the pattern generator consists of a network of spiking neurons. Each neuron in the pattern generator receives, in addition to the regular network inputs, a synaptic noise input that perturbs the activity of the neuron and thus affects the outputs of the entire network. The outputs of the pattern generator may be subject to additional (e.g., neural to motor) transformations before reaching the evaluator. **c** A synaptic weight can change only if the synapse was recently active, and reward followed. Otherwise, the weight change is zero (columns 1 and 2, $\Delta W = 0$). If synaptic activity is accompanied by above-average excitatory noise input to the postsynaptic target, and is followed by reward, the weight W is increased (column 3). If synaptic input is paired with below-average excitatory noise input to the postsynaptic neuron, and this leads to reward, W is decreased (column 4). **d** The learning rule is capable of dealing with delayed reward, as long as each neuron can maintain a local *eligibility* trace of its recent noise inputs and regular synaptic inputs.

Since the learning process is based on such correlations, convergence in large networks may be very slow. This is a potential weakness of all stochastic gradient learning rules.

To test biological plausibility and see whether learning can take place in a time consistent with that observed experimentally, therefore, it is essential to apply these learning rules to specific biological examples. Imitative song learning in birds is an easily quantified but ethologically rich behavior [62, 63] that has been amply cataloged throughout the learning process, which includes periods of gradual, practice-driven improvement; in addition, detailed results on the anatomy and physiology of song-related premotor areas are available in the literature [46]. This makes birdsong acquisition an excellent candidate system for the analysis of motor learning.

We implement a model birdsong network to study the speed with which a noise-based rule can drive learning in a biological system; to this end, we incorporate important biological details, such as the architecture of and neural activity in the songbird premotor circuit. However, we do not attempt to provide an exact model of the entire song system, since important details about acoustic production, song preference, comparison, and reward evaluation, are unknown. It is likely that, due to evolutionary adaptation, the bird brain uses a convenient perceptual metric for the assessment of reward; this and other adaptive features would serve to make learning easier. For example, two sounds with a factor of two difference in pitch may be perceptually closer than two sounds separated in pitch by a factor of just 1.5. This, together with the period-doubling ability of the syrinx [64], could give rise to some of the interesting features seen in the song-learning trajectory of actual zebra finches [65]. Because finch songs are not of the pure-tone variety, but contain multiple harmonics, such a perceptual metric could be an adaptive feature to aid learning. Since our aim is to show that learning can be fast enough with a simple stochastic gradient rule, omitting details that could facilitate learning ought not to detract from the demonstration, but bolster it.

The work of Doya and Sejnowski provides an ideal starting point for this portion of our study [47]. They have suggested that the abstract model of Figure 1a can be mapped onto avian brain anatomy. The pattern generator corresponds to the premotor nuclei HVC and RA, the motor neurons, and the vocal and respiratory organs. Avian brain nucleus RA is a candidate area in which our synaptic learning rule might operate. RA is important for song generation, because it drives the motor neurons that control vocalization. It receives input from two prominent sources, nuclei HVC and LMAN. We hypothesize that HVC sends regular synapses to RA, while LMAN sends noise synapses. Our reasoning is as follows. HVC lesions cause loss of song, while lesion of LMAN in adults has little immediate effect, and lesion of LMAN in juveniles leads to premature crystallization of song [66, 67]. This suggests that HVC is important for song generation, while LMAN is only necessary for song learning. RA-projecting HVC neurons produce spike trains that are repeatable from trial to trial [30], while LMAN neurons produce highly random spike trains during undirected song [68]. Each RA neuron receives just one or a few synaptic inputs from LMAN [69]. The synapses from LMAN, which are morphologically and physiologically distinct from the synapses from HVC, could correspond to the noise synapses of our model. This is consistent with the idea that HVC enables RA to produce a stereotyped song, while LMAN helps it generate variability. Finally, it is believed that song acquisition involves plasticity at the synapses from HVC to RA [70, 71, 39, 40], although direct evidence of activity-dependent, long-term plasticity is still lacking.

It is widely believed that an evaluator exists in the avian brain: Juvenile males can learn to reproduce the song of an adult tutor over a period of roughly 50 days [72], even if deprived of social interaction with or feedback from the tutor and other birds [65]. Exposure to a tutor song is only necessary in the earliest stages of learning, suggesting that the bird acquires and uses a mental template of the tutor's song [45, 73]. Finally, auditory feedback of the bird's own song is necessary for learning, since song acquisition is impaired in deafened juveniles [45], while deafening of adults that have already learned song has little short-term impact on song generation [74]. Consequently, it is thought that the evaluator matches auditory input with a stored template of the tutor's song.

To implement the model, HVC and RA were modeled as sets of integrate-and-fire model neurons; HVC neurons received current injections that caused brief bursts of action potential firing, in accord with experimental observations [30]. RA neurons were driven by regular synapses from HVC, as well as inputs from the noise injector; activity in the noise injector was modeled as Poisson spike trains, and synaptic currents to RA due to the noise spikes were excitatory. To mimic RA connectivity in the actual bird, we included weak global feedback inhibition within RA. The synaptic output activities of the RA neurons were summed to produce two control signals, the period and amplitude, for a pulse train. The pulse train was sent through a filter to produce the vocal output. Such a source-filter model is simpler than the nonlinear dynamical systems that have been proposed for modeling the bird syrinx, but captures much of their functionality. Since little is known about the neural evaluator, we constructed a simple model evaluator using signal processing techniques. The evaluator extracted the pitch and amplitude of the vocal output, which were compared with the pitch and amplitude of the template to produce a score. Whenever the score exceeded a threshold, the evaluator produced a reinforcement signal. The threshold was adaptively increased as performance improved. Evaluator output was assessed and administered to the network continuously (online) throughout the song, but with a 45 ms delay relative to the network activity that produced it. Learning at each synapse was driven by the correlation of this globally broadcast delayed reward with a temporally smeared and decaying trace of past synaptic activity and postsynaptic noise.

5.2 Results

An actual zebra finch song was used as the tutor template (Figure 3b). Before learning, the model output fluctuates about a constant frequency (Figure 3a). After learning, the model reproduces the harmonic and amplitude structure of the template (Figure 3c). Thus, an HVC sequence, a source of noisy inputs and a single positive global reward signal are sufficient for this simple learning rule to learn a seemingly complex task in a recurrent network of conductance-based spiking neurons with delayed reward. We also see from the learning curve (Figure 3d) that song-learning is nearly complete after just 1000 iterations, which seems rather fast. The overall network size, however, is relatively small: The pictured simulation has 800 HVC neurons, 200 RA neurons and as many independent noise inputs, 1.6×10^5 plastic HVC–RA synapses, and two output units. The actual zebra finch is estimated to have 20000 HVC neurons, 8000 RA neurons, 1.6×10^8 synapses, and eight outputs. How should learning time scale-up with network size?

To answer this question, we analytically derive the full learning curves for a linearized version of the birdsong premotor network, using node and weight perturbation. We illustrate the theoretical results together with numerical simulations, in three examples (Figure 4a): First, vary the hidden layer (RA) size while keeping the number of input (HVC) and output units fixed. Next, vary the number of output units, while keeping the hidden and input layers fixed. The learning rule used in both these examples is node perturbation, which, for a network of linear, rate-based neurons, is mathematically identical to the learning rule proposed in this paper. Finally, for a fixed number of input, hidden, and output units, apply the weight perturbation learning rule (for a biologically plausible implementation of this, cf. [31]). In each case, we plot the best possible (fastest) learning curves. It is clear that the learning speed with 2000 independent noises and hidden units can be as fast as the best-case learning speed with 20 or 200 independent noises and hidden units; this is also true when the number of noises is 40000, as is the case in the example with weight perturbation.

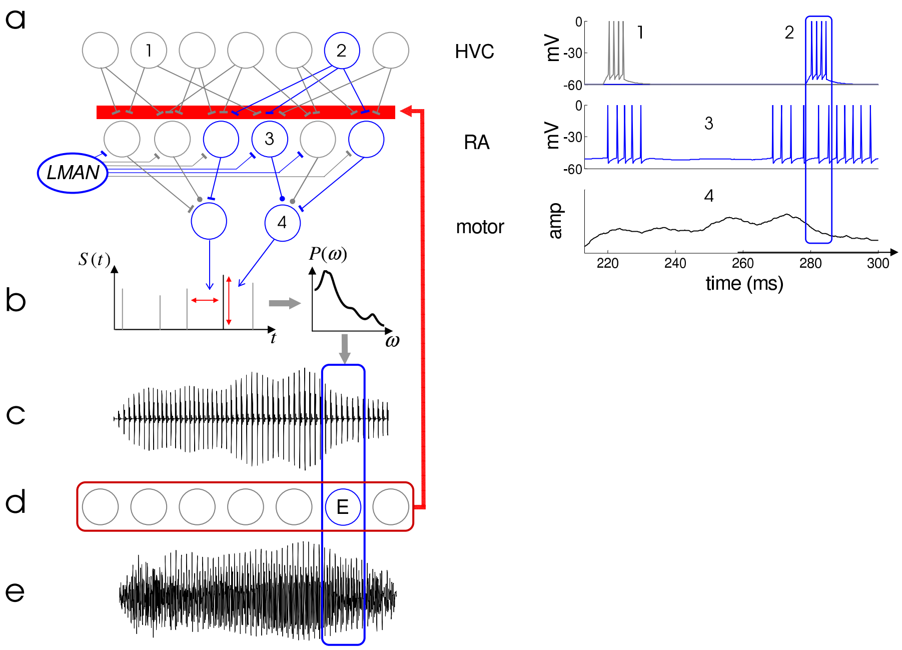


Figure 5.2: Details of the model song-learning network. (a) The time-varying activities of a model network of HVC and RA premotor neurons and motor outputs control the parameters of (b) a simple sound-producing system. Acoustic features of the output 'student' song, shown as a spectrogram in (c), are matched by (d) an evaluator, by comparison with features from (e) a recording of an actual zebra finch 'tutor' song. A reward signal, generated by the evaluator from a good match, is broadcast to all synapses in the premotor network. Reward, correlated with stochastic synaptic inputs to RA (possibly from LMAN), drives learning in recently active HVC–RA synapses. (f) Activity of two typical model HVC neurons, driven by brief current pulses to mimic experimental data, shown for a 150 ms segment of the simulated song motif. (g) Activity of a typical model RA neuron, driven by HVC and LMAN inputs, after 1000 iterations of learning.

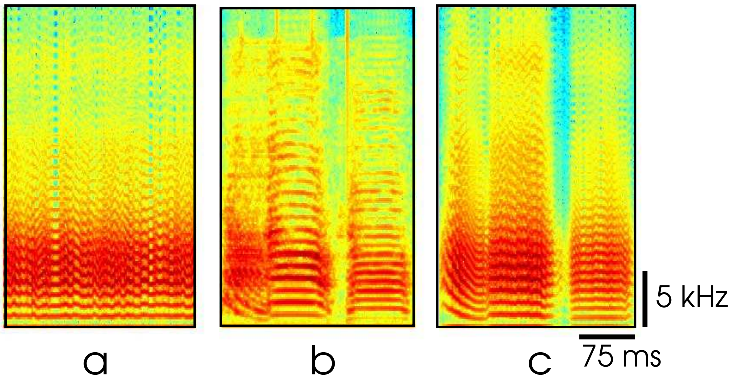


Figure 5.3: Results from the model song-learning network. (a) Spectrogram of “student” song before learning: the song consists of a harmonic stack with no systematic pitch or amplitude modulation, but with small noise-induced fluctuations in both. (b) The “tutor” song used to train the network, is a recording of an actual zebra finch song. (c) The student song, after 2000 iterations, closely resembles the tutor song (sound files of both tutor and student songs are available as supplementary information).

The theoretical calculation, verified by numerical simulation, shows that in this network the learning time scales not with the number of independently injected noises, but with the number of output units. These results imply that if the model is scaled to the size of the actual finch song network, there should be a learning slow-down proportional to the number of independent output controls the bird uses to produce song (the bird uses eight muscle groups to produce song, while the model network of Figure 3 included only two – thus, learning should take four times longer); more importantly, there should be no additional slow-down from the vast increase in the number of RA neurons, HVC–RA synapses, and independent noises in going from the current model to a realistically large network.

Since learning curves cannot be derived with non-linear spiking neurons, we run the full spiking neuron song learning simulation of Figure 3 on two networks with 200 and 800 RA neurons respectively. Learning parameters are empirically optimized in the 200 RA neuron network to produce the fastest learning curve possible. The same parameter values are used in the 800 RA neuron network, but rescaled according to the linear theory. As predicted above, we see in Figure 4b that learning with 800 independent noises and RA neurons is as fast as the best-speed learning with 200 independent noises and RA neurons.

Another important feature of the model song network, evident on comparison of Figures 3d and 4b, is that learning time does not depend on the song duration. There are two central reasons for this: (1) non-overlapping sets of HVC neurons and HVC–RA synapses generate non-overlapping parts of the song, minimizing synaptic interference in the learning of separate parts of the song, and (2) the reward signal is assumed to carry time-varying, albeit delayed, information about song performance over the course of a single song.

That stochastic gradient learning in large networks can be tolerably fast and does not scale-up poorly with network size is surprising, and seems to go against the prevailing orthodoxy which is based on both heuristic arguments and supported recently by more principled calculations [59, 75]. Actually, both our observation, that learning can be fast, and the orthodox view, that learning is too slow, are correct, but in different regimes. Learning in our model is fast for an essentially simple reason: since there is a massive convergence from RA to the outputs, RA neural activity does not have to be learned individually. Rather, it is enough that the convergent RA activities deliver the right summed drive to the output units; for the same reason, individual HVC–RA weights may take any value, so long as the summed drive to the RA population is correct. In terms of information theory, the total information encoded in the trained song network is proportional to the number of independently controlled output units. Since the number of outputs is small but the number of HVC and RA units is large, the total information that needs to be encoded for song is small compared to the capacity of the network, where potentially every HVC–RA weight can be learned individually.

In terms of learning, we observe that while it is true that the correlation of a single noise with the reward is very small, this does not matter, precisely because single synaptic weights do not need to be learned exactly. The whole distributed pattern of thousands of individual noises, if it results in higher than average reward, is reinforced; this strategy is effective because learning effectively depends on correlations between reward and a much smaller number of noises, equal to the number of output units, whose activities fluctuate due to the summed inputs from the individually noisy RA neurons. If, in an extreme case, the number of independent degrees of freedom necessary for controlling song (and reward) was equal to the number of RA neurons, learning would involve accurately determining the correlations of individual RA neuron inputs with reward, and would indeed be very slow. Learning with gradient estimation is akin to hill climbing on the reward surface, which is a function of the underlying network parameters; stochastic gradient estimation is like climbing the hill by performing a biased random walk in the parameter space. This process is generally very slow in a multi-dimensional space, but in this problem converges in a reasonable amount of time since the effective dimensionality of the parameter space, which corresponds not to the number of independently injected noises, but to the number of output units, is small.

In summary, if this learning rule were applied to a model network of the same size as the zebra finch song system, with as many neurons and independently controlled vocal muscles, we would expect learning to converge to reproduce the tutor song within 8000 iterations. For comparison, an actual bird performs more than 100000 practices before song crystallization. The margin in the learning time of the model network and the actual zebra finch means that additional biological complexity can be added to the model without compromising the ability of the learning rule to converge in a realistic number of practice iterations.

Massive neural to motor convergence is characteristic of the birdsong system in particular, and of biological motor control systems in general. Thus, we expect lessons on the speed and related scale-up of stochastic gradient learning in birdsong to carry over to other problems in motor learning. This rule also has a significant advantage over other stochastic gradient rules based on weight perturbation [31]: learning would be faster, by a factor of N, when applied to networks of size N, for problems that are more "difficult," or information-rich, than birdsong [59]. Although for simplicity we have ignored dynamic aspects of synaptic communication such as short term facilitation and depression, we show elsewhere that the learning rule performs stochastic gradient optimization even with this additional complexity. We have shown here that this learning rule is consistent with details of biological neural systems, can learn goal-directed behaviors in networks of conductance based spiking neurons, can deal with delayed reward, and can be fast enough to be taken seriously as a biological mechanism of learning.

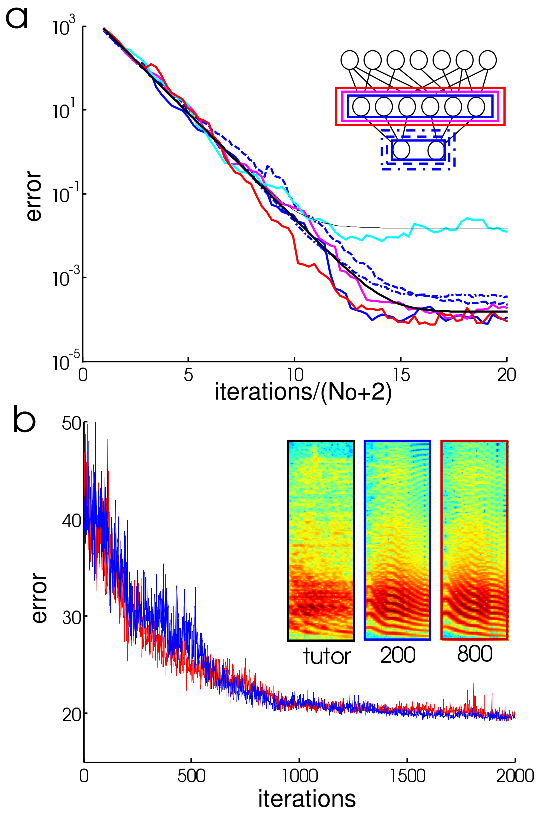


Figure 5.4: Scaling of learning speed with network size. (a) Theory and numerical simulations of learning in the linear 3-layer network pictured in the inset. The best-case learning curves are plotted as a function of iteration number divided by $(N_o + 2)$, where N_o is the number of output units. Solid lines: $N_o = 2$ is fixed, and the number of hidden units, and thus noises, is scaled to be 20 *(magenta)*, 200 *(blue)*, or 2000 *(red)*; learning time does not depend on the number of noises or hidden units, as predicted by theory *(black)*. Blue lines: fix the number of hidden units to be 200, and scale the number of output units, to be 2 *(solid blue)*, 5 *(dashed blue)*, or 10 *(dot-dashed blue)*; we see that the learning time scales linearly with the number of output units, as $N_o + 2$, once again agreeing with the theory curve *(black)*. These curves were for learning by node perturbation. We applied weight perturbation to the network, with 200 hidden units and 2 output units; the best learning curve with weight perturbation (theory, *black*; simulation *cyan*) is identical to the node-perturbation learning curves with $N_o = 2$ units, except for a higher residual or final error, because the variance of each injected noise is the same, but there are many more noises, leading to a higher total variance. (b) Simulation of learning in the full birdsong network, with 200 (blue) and 800 (red) RA neurons. The learning rate was optimized for the 200 RA network. Scaling up the network size and the number of noises does not change the best learning time for the network. *Inset:* tutor, 200 RA network, and 800 RA network spectrograms for the 75 ms long segment of song used for training in this simulation.

5.3 Methods

The model song-learning network consists of HVC, RA, and an output motor layer, as well as a set of ‘noise-injection LMAN neurons that fire irregular spike trains; HVC projects to RA which projects to the output layer, while the noisy neurons project to RA. The output motor layer controls the parameters of a simple sound-producing model (Figure 2a).

5.3.1 Premotor network

The premotor network consists of HVC, RA, and LMAN, all of which are made up of spiking neurons. The HVC, RA, and the noise injection area consist of spiking neurons, while the output motor layer reflects the summed synaptic currents from the RA layer. The LMAN neurons fire poisson spike trains at a rate R_ξ. All neurons in HVC and RA are modeled as spiking neurons with integrate-and-fire dynamics, with membrane potentials V_i governed by

$$C_m \frac{dV_i}{dt} = -g_L(V_i - V_L) - g_{E,i}(V_i - V_E) - g_{I,i}(V_i - V_I) \tag{5.1}$$

with reset condition $V_i \to V_{reset}$ when V_i crosses the threshold voltage V_θ; this threshold-reset event represents a spike followed by repolarization.

Every time neuron i (from HVC, RA, or the irregular spike source) spikes, the synaptic activation variable $s_i(t)$ (or rather, $s_i^h(t), s_i^r(t)$, or $s_i^\xi(t)$, respectively) jumps up, and between spikes decays, according to

$$\frac{ds_i(t)}{dt} = \frac{-s_i(t)}{\tau_s} + \sum_a \delta(t - t_i^a) \tag{5.2}$$

where t_i^a is the list of times when neuron i spiked.

Parameters are $C_m = 0.5$ μF/cm^2, $V_L = -60$ mV, $V_E = 0$ mV, and $V_I = -70$ mV. The leak conductance is $g_L = 0.15$ mS/cm^2 and $g_L = 0.22$ mS/cm^2 for HVC and RA neurons, respectively. When the the membrane potential reaches $V_\theta = -50$ mV, it is reset to $V_{reset} = -55$ mV. The synaptic time constant is $\tau_s = 5$ ms for HVC–RA, irregular spike source–RA, and RA–motor connections. In all simulations, time is discretized with $dt = 0.2$ ms, and the total length of the simulated song is $T = 300$ ms. Equations (**??**) and (5.2) are discretized, and $\delta(t - t_i^a) \to \delta_{t,t_i^a}$. At any time step $[t, t + dt)$, LMAN neurons fire a spike with probability $p = R_\xi dt$, where the poisson spike rate is $R_\xi = 80Hz$. LMAN spike trains are regenerated and thus vary from epoch to epoch. In the simulations, there are $N_h = 720$, $N_r = 200$, and $N_\xi = 200$ HVC, RA, and LMAN neurons, and $N_o = 2$ output units.

- **Inputs to the network** HVC neurons receive short excitatory synaptic pulses that cause them to fire bursts of action potentials; these inputs are chosen so that activity in the model HVC neurons mimics that seen experimentally in the awake singing bird [30]. The excitatory synaptic pulses have duration $T_b = 6$ ms, and size $g_{E,i} = 0.065$ mS/cm^2. Each HVC neuron receives once such pulse in the course of the song motif; the pulse onset times are distributed evenly across the neurons. The pattern of HVC activities stays fixed throughout learning. No inhibitory conductances are used in HVC.

- **RA layer inputs** RA neurons receive feedforward excitatory synaptic excitation from HVC and from the irregularly firing source: the RA excitatory conductances are given by $g_{E,i}(t) = .0012 \sum_i W_{ij} s^h_j(t) + 0.05 s^\xi_i(t)$. RA neurons also receive global feedback inhibition from within RA, $g_I = .0005 \sum_i s^r_i(t)$. The synaptic weights W from HVC are initialized randomly on the uniform interval $[0, 1.5]$. There is a separate noise input for each RA neuron.

- **Motor ouputs** Two non-spiking output units sum the synaptic activations from RA, with the matrix A of fixed RA–output weights:

  $$\tau_m \frac{dm_i(t)}{dt} + m_i(t) = \sum_j A_{ij} s^r_j(t) + b_i$$

  Half of all RA neurons project to m_1; the other half project to m_2. Of the set projecting to m_1, half the weights are uniformly -2.2 and half are uniformly 2.2. Similarly, of the set projecting to m_2, half the weights are uniformly -3.2, and the other half are uniformly 3.2. In words, the model RA neurons have myotopic connections to the outputs and have push-pull control over each output. The baseline activities of the output units are $b_1 = 60$ and $b_2 = 40$.

- **Eligibility** The eligibility trace tags HVC–RA synapses as eligible for plasticity based on whether they were recently active. The instantaneous eligibility,

  $$e^{(0)}_{ij}(t) = \xi_i(t) s^h_j(t),$$

  reflects ongoing activity in the synaptic HVC and noise inputs to RA. The instantaneous eligibility is delayed by passing it successively through

  $$\tau_e \frac{de^{(n)}_{ij}(t)}{dt} + e^{(n)}_{ij}(t) = e^{(n-1)}_{ij}(t)$$

so that $e_{ij}^{(n)}$ is delayed by $(n-1)\tau_e$ relative to e_{ij}^0, but is also smeared so that temporal information about activity is spread over a width of order $n\tau_e$. We use $\tau_e = 5$ ms and $n = 10$, or in other words, we define the eligibility to be $e_{ij}(t) \equiv e_{ij}^{(10)}(t)$, so that the eligibility is delayed by 45 ms and has a full-width at half-max of 35 ms.

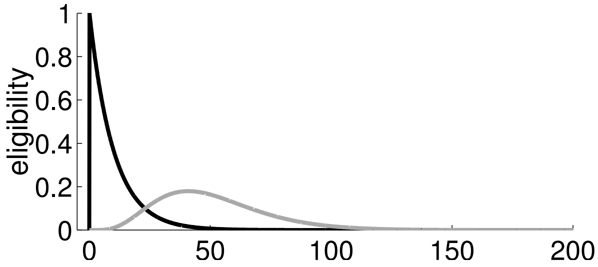


Figure 5.5: The instantaneous eligibility (black) is delayed by 45 ms to produce the eligibility trace (grey); typically, delaying a signal also leads to dispersion, so we model the delay by passing the instantaneous eligibility through a cascade of linear filters that also smear and damp the trace.

We chose this method to delay the eligibility because it resembles a signal transduction cascade, and does not assume that eligibility can be delayed without distortion. Inclusion of key nonlinearities in a transduction cascade could reduce the spread of the eligibility trace as it is delayed, and thus speed up learning. It is also possible to choose other forms for the time-course of synaptic activation, without adversely affecting the proof of reward optimization.

5.3.2 Sound production

The actual songbird syrinx is a non-linear oscillator that vibrates as a fundamental frequency (pitch), producing a range of harmonics of the fundamental; pitch is modulated by muscles that control the tension of the syringeal fold [**?**]. Sound amplitude is controlled in part by air flow through the syrinx. The vocal tract and beak filter the broad spectral content of the syringeal output and possibly also directly affect the syringeal oscillations [76, 77].

The source-filter model is a simple sound generation method commonly used in the field of signal processing. A harmonic source, analogous to the bird syrinx, produces a train of delta pulses of varying heights and spacing: this results in a sound train of varying volume and pitch. To mimic the effects of the bird vocal tract and beak, the train of delta pulses is passed through a filter that modulates the relative strengths of the harmonics in the signal to capture broad features of the spectral envelope typical of zebra finch songs. In our implementation, the spacing and heights of the delta pulses in the harmonic source are controlled dynamically by the two outputs, $m_1(t)$ and $m_2(t)$, respectively. We use a simple linear filter whose parameters are derived from finch song recordings; the parameters are static and do not change over the course of the song and over the course of song learning.

Numerical details: Due to the 0.2 time-discretization used to integrate the network dynamics, the outputs $m(t)$ are sampled at 5 kHz only. We linearly interpolate these output trains to generate a pair of output command signals, $\bar{m}(t)$, sampled at 44 kHz. $\bar{m}_1(t)$ specifies the delta-pulse spacing (pitch period); for period to pulse conversion, a counter sums $\frac{1}{\bar{m}_1(t)}$ until it crosses 1, which triggers a pulse, and the counter is reset to 0. The height of each pulse is specified by the value of $\bar{m}_2(t) \times 10^{-3}$ at the time of the pulse. We use a fixed 10-parameter linear predictive coding (lpc) filter derived from a concatenated sample of 3 arbitrarily selected zebra finch song recordings. The real part of the filtered pulse train is the student song.

5.3.3 Sound comparison and reward

The reward is computed from a comparison of the sound files of the teacher (actual zebra finch song recording) and student songs. Pitch and sound amplitude are two good quantifiers of the songs; we extract these features from the sound files and compare them to compute the reward.

Pitch extraction: The songs are windowed into overlapping segments by multiplication with a 300-sample (6.8 ms) hanning window that shifts by 10 samples (0.23 ms) at a time until the entire song is covered. We compute the autocorrelation of each windowed song segment; the number of samples between the highest peak (at zero time-lag) and the second-highest peak is the pitch period, so long as this value is between 12 and 80 samples; if outside this range, the distance to the next-highest peak is computed, until a value is found that falls in the allowed range. The middle ten samples of the current windowed segment are assigned to have this value of estimated pitch. This procedure is repeated for each segment. The beginning of the first windowed segment and the end of the last windowed segment of the song are assigned the same pitch values as of their closest assigned neighbors.

Amplitude extraction: The songs are windowed into 100-sample (2.3 ms) non-overlapping segments. For all 100 samples of each song-segment, the amplitude is assigned to be $0.3 \times \max|\text{song segment}|$.

Reward: The online reward is a function of the squared difference between teacher and student pitch and amplitude; moreover, it is delayed relative to the premotor activity that produced latexthe song, by an amount T_{delay}. Let $p(t), a(t)$ represent the student song pitch and amplitude, and let $\bar{p}(t), \bar{a}(t)$ represent the tutor song pitch and amplitude. Then

$$\begin{aligned} E(t + T_{delay}) &= \frac{(\bar{p}(t) - p(t))^2}{c_p^2} + \frac{(\bar{a}(t) - a(t))^2}{c_a^2} \quad \text{if } \bar{a}(t) > a_\theta & (5.3) \\ E(t + T_{delay}) &= 2\frac{(\bar{a}(t) - a(t))^2}{c_a^2} \qquad \text{otherwise.} & (5.4) \end{aligned}$$

Note that pitch only enters into the error at times when the song is supposed to be non-silent, $a(t) > a_\theta$. Different reward neurons assess reward for separate times in the song, by thresholding the error, so that

$$R(t) = \Theta\big(\overline{E}(t) - E(t)\big) \tag{5.5}$$

In other words, if the song-error $E(t)$ is lower than a threshold $\overline{E}(t)$, the reward is $R(t) = 1$, but is zero otherwise. The adapting reward thresholds $\overline{E}(t)$ of the different reward neurons reflect the past averaged song-quality, and are obtained by linearly low-pass filtering $E(t)$ over the past 5 song iterations. Parameters are: $c_p = 60, c_a = 80 \times 10^{-3}, T_{delay} = 45ms, a_\theta = 0.005$.

Reward is assumed to arrive at RA continuously in time, but delayed by 45 ms relative to the premotor activities that generated the song that lead to that reward; hence, the reward and delayed eligibility are on-register. In fact, learning proceeds in the direction of the gradient of the reward for any non-zero overlap of the reward and eligibility traces.

5.3.4 Learning

Online weight updates are made to the HVC–RA connections within each song iteration:

$$\Delta W_{ij}(t) = \eta R(t) e_{ij}(t).$$

Since $R(t)$ contains information about song that occured at $t - T_{delay}$, these updates start from $t > T_{delay}$, and continue until $t = T + T_{delay}$. We use $\eta = 0.0022$.

5.3.5 Scaling simulations

We simulated the learning of a single pattern in a linear three-layer HVC–RA–motor network with activities h_i, r_i, and m_i respectively:

$$h_i = 1 \tag{5.6}$$

$$r_j^{\xi} = \sum_i W_{ji} h_i + \xi_j \tag{5.7}$$

$$r_j^{0} = \sum_i W_{ji} h_i \tag{5.8}$$

$$m_k^{\xi,0} = \sum_j A_{kj} r_j^{\xi,0} \tag{5.9}$$

$$C^{\xi,0} = \sum_k (d_k - m_k^{\xi,0})^2 \tag{5.10}$$

$$\Delta W_{ji} = \eta (C^0 - C^{\xi}) \xi_j h_i \tag{5.11}$$

C is the error function, which can be thought of as negative reward. In all linear network simulations, we used $N_h = 200$ HVC neurons; since we considered only a single pattern, they were all considered to be active, with $h_i = 1\ \forall i$. W was initialized randomly from the uniform interval [0,1]. A is a block diagonal with N_o blocks of size (1,N_r/N_o) each: half the entries in each block are 1 and half are -1. The noise injections into RA are Gaussian distributed, with $\langle \xi_i \rangle = 0$, $\langle \xi_i \xi_j \rangle = \sigma^2 \delta_{ij}$, and $\sigma = 0.001$. We used $\eta = N_o/((\sigma^2) N_r (N_o + 2)(h^T h))$, determined by our theoretical calculation, and verified by simulation, to be optimal for learning speed: that is, this value of η results in the fastest convergence of the learning curves. First, we varied the number of RA units to be $N_r = 20, 200$, or 2000 RA units and as many independent noises, while keeping the number of output units fixed at $N_o = 2$. Next, we varied the number of output units to be $N_o = 2, 5$, or 10, keeping the number of RA units fixed at 200. We set the desired outputs to be $d_k = 0$ for all N_o outputs.

5.4 Appendix: calculation of the learning curve in a linear model birdsong network

5.4.1 The network and learning rule

We assume the network has an input layer, a hidden layer, and an output layer, with plastic weights between the input and hidden layers, and fixed weights between the hidden and output layers. Inputs are denoted by x (size $N_x \times P$); hidden units by z (size $N_z \times P$); and output units by Az (size $N_y \times P$). The plastic weight matrix W

from the input to the hidden layer is of size $N_z \times N_x$, and the fixed weight matrix A from the hidden to the output layer is of size $N_y \times N_z$.

In node perturbation, noises are injected into the postsynaptic neurons; if a particular pattern of noise leads to better network performance than without noise, weights to the postsynaptic neurons are updated in a direction that makes that postsynaptic pattern of activity more likely. For example, if positive noise input to neuron z_i leads to better output than without noise, increase the weights of synapses to neuron z_i that were active when the positive noise was injected; if performance is worse with the positive noise, make the opposite weight change to those synapses. In weight perturbation, network weights are independently perturbed; if that pattern of noise leads to better performance than without noise, the weight perturbations are made permanent. Otherwise, a permanent weight change is made in the opposite direction.

Since the feedforward network under consideration has one layer of plastic weights W, only neurons postsynaptic to those receive noise injections in node perturbation, and only those weights are varied in weight perturbation. Here we consider the learning of one pattern, $P = 1$, in a network with several input, output, and hidden layer units. We assume that the injected noise ξ is Gaussian, with mean zero and variance σ_ξ^2. The network activity and learning equations are:

$$\bar{z} = Wx \tag{5.12}$$

$$z = Wx + \xi_{\text{N}}, \text{ or } z = (W + \xi_{\text{W}})\, x \tag{5.13}$$

$$\overline{C} = (d - A\bar{z})^T (d - A\bar{z}) \tag{5.14}$$

$$C = (d - Az)^T (d - Az) \tag{5.15}$$

$$\Delta W_{\text{N}} = \eta(\overline{C} - C)\, \xi_{\text{N}}\, x^T, \text{ or } \Delta W_{\text{W}} = \eta(\overline{C} - C)\, \xi_{\text{W}} \tag{5.16}$$

5.4.2 Learning curves

The learning curves for both weight perturbation and node perturbation are given by the following linear iteration equations

$$\langle Z^{(n)} \rangle = \Gamma^n Z^{(0)} + B \tag{5.17}$$

$$\langle C^{(n)\rangle} \rangle = \sum_i \langle Z_i^{(n)} \rangle \lambda_{Ai} \tag{5.18}$$

where $Z_i \equiv \langle z_i^2 \rangle$, and C is the summed square output error; the superscript n on Z and C designates the nth iteration of learning. The recursion matrix Γ is given by

$$\Gamma = I - 4\eta\sigma^2\Lambda_A + 4\eta^2\sigma^4(2\Lambda_A^2) + \mathbf{1}{\lambda_A^2}^T \tag{5.19}$$

and the residual error B is

$$B_i = \eta^2\sigma^6(8\lambda_{Ai}^2 + 4\lambda_{Ai}\text{Tr}(Q_A) + 2\text{Tr}(Q_A^2) + (\text{Tr}(Q_A))^2) \tag{5.20}$$

Λ_A is a diagonal matrix of the eigenvalues of $Q_A \equiv A^T A$; λ_A is a vector of the eigenvalues of Q_A. Information about the scaling of learning speed on the network parameters and the learning rule is contained in Γ.

5.4.3 Scaling

It is easy to analytically solve for the eigenvalues of Γ if all the non-zero eigenvalues of Q_A are identical. In the next section, we will show that this is a case of interest to us in the study of birdsong learning. If there are N such identical non-zero eigenvalues, of size c, the eigenvalue spectrum of Γ is

$$\gamma_c = 1 - 2\eta' c + (N+2)\eta'^2 c^2 \quad \text{(single common mode)} \tag{5.21}$$

$$\gamma_d = 1 - 2\eta' c + 2\eta'^2 c^2 \quad (N-1 \text{ differential modes}) \tag{5.22}$$

For notational convenience, we have defined $\eta' \equiv \eta\alpha\sigma_\zeta^2$. For learning to occur successfully, the error must converge toward zero or in the case of stochastic learning with noise of finite (non-zero) variance, toward the residual. Since all the eigenvalues λ_i of Q_A are positive, this will happen only if $Z \to 0$. This in turn is only possible if the real parts of all the eigenvalues of Γ lie in $(-1, 1)$. The eigenvalues of Γ are guaranteed to lie in $[0, \infty)$, since expression (5.22) has form $(1 - 2k + 2k^2) > 0 \,\forall k$, and $\gamma_c > \gamma_d$. We need only ensure therefore that $\gamma_c, \gamma_d < 1$. For this to be true, $0 < \eta' < \frac{2}{c(N+2)}$, and the value that leads to fastest convergence of the error to zero is obtained at $\eta'^* = \frac{1}{N+2}$. The eigenvalues of Γ with this η' are given by

$$\gamma_c = 1 - \frac{1}{N+2} \tag{5.23}$$

$$\gamma_d = 1 - \frac{2}{N+2} + \frac{2}{(N+2)^2} \tag{5.24}$$

We can solve for n_ϵ, the number of iterations required for Z_i to reach a value of ϵ times the original, as a function of N:

$$n_\epsilon \approx (N+2)\log(\frac{1}{\sqrt{\epsilon}}). \tag{5.25}$$

Note that n_ϵ does not in any way depend on η', c or any parameters besides N. In any problem, $N \leq \min(N_z, N_y)$, since N is the rank of the projection matrix A. Thus, for fastest learning (best η), weight perturbation and node perturbation both scale with the number of non-zero eigenvalues of $Q_A \equiv A^T A$. The residual error, B, does depend on certain parameters in the problem, including the variance of the injected noise:

$$B_i = \left(\frac{\sigma_\zeta}{\alpha}\right)^2 \frac{8+6N+N^2}{(N+2)^2}. \tag{5.26}$$

Thus, we have proved that for learning by node- and weight-perturbation in a 3-layer network, the learning time scales not with the potentially large number of independently injected noises in the hidden layer, but with the complexity of the learning task, which is specified by the number of independent output units whose outputs need to be learned for each time step.

5.4.4 Application to birdsong learning

Let us now consider how these results apply to our model of birdsong learning. The 3-layer structure represents the HVC, RA, and output layers. The matrices W of plastic weights and A of fixed weights correspond directly to the HVC–RA connections and the RA–output connections. $N_z = N_{\text{ra}}$, and $N_y = N_{\text{output}}$, and since we are considering a single pattern, N_x is the typical number of RA-projecting HVC neurons active at one moment in song, call it $N_{\text{hvc,t}}$. The number of outputs that must be learned is a small number, corresponding to the number of muscles controlled by the song premotor circuit. If RA projects myotopically to the outputs, A is a block diagonal matrix and Q_A will be diagonal with N_{output} non-zero eigenvalues; further, if the summed connection strengths to each muscle are the same, the N_{output} non-zero eigenvalues will be identical. This is exactly the case covered in the section above. Other parameters of the problem map in the following way: $\sigma_\zeta = \sigma_\xi$ and $\alpha = N_{\text{hvc,t}}$ for node perturbation; $\sigma_\zeta = \sqrt{N_{\text{hvc,t}}}\sigma_\xi$ and $\alpha = 1$ for weight perturbation. Finally, $\eta^* = 1/(\sigma_\zeta^2 \alpha c(N_{\text{output}} + 2))$. Summarizing results from the last section and putting them in context of the birdsong model, learning time scales as

$$n_\epsilon \approx (N_{\text{output}} + 2)\log(\frac{1}{\sqrt{\epsilon}}). \tag{5.27}$$

Note that our calculations dealt with learning $P = 1$ patterns. Since in birdsong any two moments in song spaced apart by more than approximately 10 ms are driven by independent and nonoverlapping sets of HVC neurons, learning a stretch of song is like learning multiple independent patterns in parallel; so, these results apply to the acquisition of an entire song. In summary, the time needed to learn song by stochastic node or weight perturbation scales as the number of outputs that need to be learned, and not with the number of independently injected noises or the total number of plastic synapses or hidden units in the network.

Chapter 6

Learning by stochastic gradient descent on anisotropic cost surfaces: scaling of learning time with system size and task difficulty

Stochastic optimization methods are used in the fields of biological neural learning [31], the study of evolutionary dynamics of populations, and machine learning. Optimization, or learning, by stochastic means, is a convenient and often necessary approach to many problems. Stochastic gradient estimation eliminates the need for direct gradient calculations, which is a difficult or impossible task in systems where such gradients cannot be expressed in explicit form, or where sufficient information about the underlying variables is not available for direct gradient computation. One of the chief criticisms of perturbative or stochastic learning methods, however, is that they are slow and scale poorly with increasing network size, and hence are impracticable in biologically large neural networks, which are typically large. Such claims, however, are largely based on heuristic estimates.

A few systematic studies of the scaling of learning time with stochastic gradient algorithms have been made: recent analytical work in single-layer feedforward linear neural networks showed that best-case learning with weight perturbation is a factor of N slower than node perturbation, which is a factor of M slower than backpropogation [59], where N and M are the number of inputs and outputs, respectively,, in the perceptron. However, this study implicitly dealt with learning only on an isotropic error surface. In other works, it has been asserted that learning with node perturbation is slower by a factor of $\sqrt{N}$ slower than backpropogation [75]. The current work seeks to provide a more comprehensive description of learning time as a function task characteristics in addition to network size.

Consider a learning problem in the parameters x, with a general quadratic cost function, defined as:

$$C_0 = \frac{1}{2} x Q x^T \tag{6.1}$$

The cost function is a scalar, and x is a $1 \times N$ dimensional vector (for generalization to the case where x is $M \times N$ dimensional, see Appendix and Discussion). Q, the Hessian of the cost, is symmetric and assumed to be positive semidefinite, so that the cost is always nonnegative. The eigenvalue spectrum of Q determines the shape of the cost surface: large eigenvalues correspond to steep directions on the learning surface, while small eigenvalues correspond to directions along which the error changes slowly.

We study the learning curves of the cost function described above with the help of a simple stochastic learning rule: perturb all parameters independently by adding a $1 \times N$-dimensional Gaussian noise ξ of mean zero and variance σ^2, so that $C = (x+\xi)^T Q(x+\xi)$. Then make parameter update

$$x^{(1)} = x^{(0)} - \frac{\eta}{\sigma^2}(C - C_0)\xi \tag{6.2}$$

where C_0 is the unperturbed cost, and η is a positive scalar learning rate parameter. It is easy to show that this learning algorithm minimizes the cost by moving along a stochastic approximation of the gradient of the cost function. Descent on the cost function proceeds like a biased random walk in the multiple dimensional space of the parameters x.

It is clear that learning should be relatively slow if the system size, i.e., the dimensionality of x, is large. Generally, learning by random search in multiple dimensions scales as the number of dimensions in the search space. However, it is not merely the dimensionality of x that enters into the learning time, but the effective number of dimensions in which the cost has significant components and along which the cost varies significantly. In other words, learning time depends not on the actual size of x, but on the *effective* dimensionality of x as determined by the shape of the cost surface in these dimensions. We illustrate this point in Figure (6.1), which shows the results of three simulations, each in the same sized system, with identical initial values of x and identical initial error. The learning curves, however, are very different. This is because although the initial error is chosen to be the same for the three examples, the distribution of eigenvalues is very different. In the first case, all the eigenvalues are uniform; in the second and third cases, the eigenvalue spectrum is of the form $(a, ..., a, b, ..., b)$. In the second case, a quarter of all eigenvalues have size $b > a$, and b is chosen so that 90% of the total eigenvalue sum is contributed by the eigenvalues of size b. The third case is identical to the second, except that only 5% of the eigenvalues are bs, but they contribute 90% of the total eigenvalue sum. In all three cases, the eigenvalues are scaled so that the initial errors, given the initial values of x, are equal. We see that the difference in learning times between the three curves (for the error to reach 0.12 times the original error) spans a factor of 10, clearly demonstrating that there are more factors in the determination of learning time than system size alone.

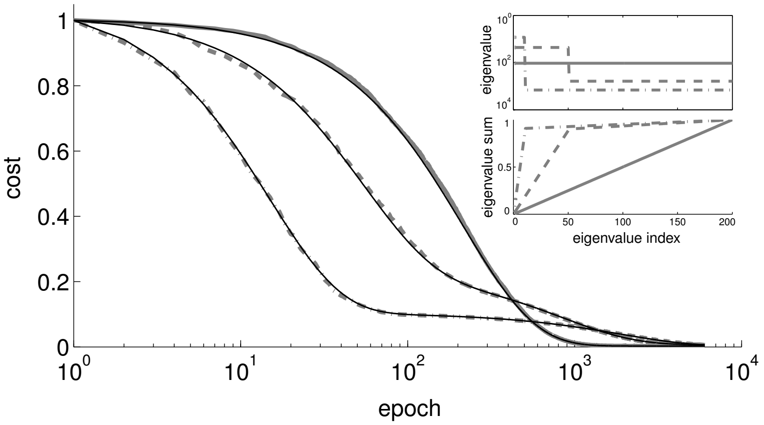


Figure 6.1: Simulation of learning with N=200 parameters and as many independent noises, with three different cost surfaces. Solid line – all eigenvalues of Q are assumed to be equal; dashed line – N/4 large eigenvalues add up to 90% of the total eigenvalue sum; dot-dashed line – N/20 large eigenvalues add up to 90 % of the total eigenvalue sum. In all cases, the eigenvalue spectra are normalized so that the initial cost is equal to one, and the injected noise has a variance of $\sigma = 0.01$. For each set of curves, we use $\eta = \frac{1}{2}\eta_{crit}$, where η_{crit} is the maximum value of η that would allow convergence; η_{crit} is derived theoretically in paper, and checked numerically. Each curve is the average of 20 learning trials with the same initial conditions.

The concept that learning will depend on the shape of the cost function is not itself new. The dependence of learning speed on the cost surface shape, as determined by the eigenvalues of the Hessian matrix, has been well studied in the literature on *direct* gradient learning rules [52, **?**]. However, the effect of anisotropic cost surfaces on the speed of learning with stochastic gradient descent has not been dealt with. Results on the scalability of stochastic gradient learning have tended to focus solely on system size. We expand the discussion to include the eigenvalue spectrum of Q, which gives more detailed information about the specific problem to be learned. We show that learning on an isotropic surface [59] provides the worst-case picture of learning with stochastic gradient descent. Intuitively, this is because in the case of learning on an isotropic surface, all directions in parameter space are equally important, and so the number of effective dimensions is equal to the actual size of the system, and equal also to the number of independent noises in the system. If learning depends predominantly on the decrease of error along a few eigenvectors, the effective dimensionality of the learning problem is much smaller than the actual system size, and learning can proceed relatively rapidly.

Despite the intuitive argument made above, the precise way in which the cost function valley shape affects learning by stochastic gradient is not clear, because of the different nature of learning by random walk in a multidimensional space than by direct gradient descent. One could imagine that most learning slow-downs resulting from anisotropies in the valley shape would be completely overshadowed by the slowdowns inherent in learning by random search in a large dimensional space; perhaps only anisotropies larger than the system size could visibly affect convergence if learning is by stochastic gradient. To address these issues, we analytically compute the ensemble-averaged learning curves for stochastic gradient descent on an anisotropic cost function.

6.1 Learning curve

Since the cost function is quadratic in x and since there are N noises injected into the system, driving independent exploration in as many dimensions, we must track the learning process by following the evolution of the N variables x_i^2. If we write x and ξ in the eigenvector basis of Q, then after one iteration the prescribed change in x_i is given by $\Delta x_i = -\frac{\eta}{\sigma^2}\left(\xi Q x^T + \frac{1}{2}\xi Q \xi^T\right)\xi_i$; we use this to obtain an expression for $(x_i^{(1)})^2$ as a function of $(x_i^{(0)})^2$, and compute the expectation over the distribution of ξ to get:

$$\begin{aligned}\langle (x_i^{(1)})^2\rangle_\xi &= \left[(x_i^{(0)})^2 - 2\eta\lambda_i(x_i^{(0)})^2 + \eta^2\left(2\lambda_i^2(x_i^{(0)})^2 + \sum_j \lambda_j^2 (x_j^{(0)})^2\right)\right] \\ &\quad + \frac{1}{4}\eta^2\sigma^2\left(8\lambda_i^2 + 4\lambda_i\sum_j \lambda_j + 2\sum_j \lambda_j^2 + (\sum_j \lambda_j)^2\right) \qquad (6.3)\end{aligned}$$

If we define $y_i \equiv \langle x_i^2\rangle$, we can express the error as $\langle C^{(1)}\rangle = \frac{1}{2}\sum_i \lambda_i y_i^{(1)}$. Moreover, Equation (6.38) is linear in y_i; this allows us to write the evolution of y_i as a function of the number of learning iterations, t, in the form of a recursion relation: $y^{(t+1)} = \Gamma y^{(t)} + B$. The recursion matrix Γ is a function of the eigenvalues of the original cost matrix Q, and of the learning rate η; it is given by $\Gamma_{ij} = \delta_{ij}(1 - 2\eta\lambda_i + 2\eta^2\lambda_i^2) +$

$\eta^2\lambda_j^2$, and $B_i = \frac{1}{4}\eta^2\sigma^2\big(8\lambda_i^2 + 4\lambda_i\sum_j\lambda_j + 2\sum_j\lambda_j^2 + (\sum_j\lambda_j)^2\big)$ is a residual term. The central advantage of being able to write down the evolution of y in terms of a recursion relation governed by a recursion matrix, is that the entire learning process depends on just the eigenvalues of Γ, and not on the specific intermediate values of x or y. Thus, we can write down $y^{(t)} = \Gamma^t y^{(0)} + \sum_{r=0}^{t-1}\Gamma^r B$. If v is the column matrix of the normalized eigenvectors v_α of Γ, and γ_α are the associated eigenvalues, then the learning curve for stochastic gradient descent on the anisotropic cost surface $C = \frac{1}{2}xQx^T$ is given by:

$$C^{(t)} = \sum_{i,\alpha}\lambda_i v_{i\alpha}\hat{y}_\alpha\gamma_\alpha^t + \text{residual} \tag{6.4}$$

$$= \sum_{i,\alpha,j}\lambda_i v_{i\alpha}\gamma_\alpha^t(v^{-1})_{\alpha j}(x_j^{(0)})^2 + \text{residual} \tag{6.5}$$

where $\hat{y}_\alpha = \sum_j(v^{-1})_{\alpha j}y_j$ is the projection of y onto v_α in the eigenvector basis of Γ, and the residual term is $\sum_{r=0}^{t-1}\sum_\alpha\gamma_\alpha^r(v^{-1})_{\alpha j}B_j$. Equation (6.5) is therefore the full learning curve, and the problem of learning with stochastic gradient descent on anisotropic learning surfaces is solved, if we can compute the eigenvalues and eigenvectors of Γ. In fact, it is not possible to derive the eigenvalue spectrum of Γ for general Q, because of the non-trivial dependence of Γ on the learning rate parameter η as well as on the eigenvalues of Q.

For learning to converge, the eigenvalues γ_i of the recursion matrix must lie in $[-1, 1]$. The learning rate η must be chosen to ensure that this condition is satisfied. To do so we need to explicitly compute $\gamma_i(\eta)$.

6.2 Bounds on the learning rate

While we do not know how to solve for the eigenvalues of Γ in general, we can derive expressions for them in special cases that are nevertheless of considerable interest.

Case 1 The simplest case to consider is if the $N \times N$ matrix Q has $m \leq N$ uniform eigenvalues, $\lambda_i = a \quad \forall i \leq m$, and the remaining $N - m$ eigenvalues are 0. It is easy to verify that the eigenvalues of Γ are:

$$\begin{aligned} \gamma_c &= 1 - 2\eta a + (m+2)\eta^2 a^2 && \text{one common mode} && (6.6) \\ \gamma_d &= 1 - 2\eta a + 2\eta^2 a^2 && \text{m} - 1 \text{ differential modes} && (6.7) \\ \gamma_0 &= 0 && \text{multiplicity N} - \text{m} && (6.8) \end{aligned}$$

These eigenvalues are all real and nonnegative, $\forall a, \eta > 0$. For convergence, both γ_c and γ_d must be in $[-1, 1]$; since we know they are positive, the only non-trivial bound is from above; thus, we see from the union of constraints for both eigenvalues that for convergence of the error, we must have $0 \leq \eta \leq \eta_{crit} = \frac{2}{a(m+2)}$. The special case of $m = N$ is the case where all the eigenvalues of Q are equal and nonzero, and corresponds to learning on an isotropic surface, as considered by Werfel et al in [59]. Under the isotropic eigenvalue assumption, our conditions for convergence and solution for the learning curve agree with the results of [59].

Case 2 Next consider the case where Q has two distinct non-zero eigenvalues, a (of multiplicity m), and $b > a$ (of multiplicity n); $m + n = N$. We solve for the eigenvalues of Γ to get (see Appendix):

$$\gamma_{dm} = \quad 1 - 2\eta a + 2\eta^2 a^2 \qquad v_{dm} = \begin{bmatrix} d_m \\ \\ 0 \end{bmatrix} \quad \text{(m-1 differential modes)}$$

$$\gamma_{dn} = \quad 1 - 2\eta b + 2\eta^2 b^2 \qquad v_{dn} = \begin{bmatrix} 0 \\ \\ d_n \end{bmatrix} \quad \text{(n-1 differential modes)}$$

$$\gamma_{\pm} \qquad\qquad\qquad v_{\pm} = \tfrac{1}{(m+n\delta^2_{(\pm)})} \begin{bmatrix} 1_m \\ \\ \delta_{(\pm)} 1_n \end{bmatrix} \quad \text{(2 modes)}$$

The eigenvalues $\gamma_{\pm}$ are obtained from equating two equivalent expressions for the numbers $\delta_{(\pm)}$, $\delta_{(\pm)} = \frac{\eta^2 a^2 m}{\gamma_{\pm} - (1-2\eta b+\eta^2 b^2 (n+2))} = \frac{\gamma_{\pm} - (1-2\eta a+\eta^2 a^2 (m+2))}{\eta^2 b^2 n}$. The $m-1$ m-dimensional vectors labeled by d_m are normalized orthogonal differential modes, named because $\sum_i (d_m)_i = 0$ for each of the modes; v_{dm} are N-dimensional zero-padded versions of the vectors d_m. The same is true for the $n-1$ differential modes d_n, and the corresponding vectors v_{dn}.

We must once again solve for convergence criteria in terms of the allowed learning rates η, by imposing that $\gamma \in [-1, 1]$. The differential mode eigenvalues are clearly real and positive, and provide necessary bounds on η: $\gamma_{dm} < 1 \implies 0 \leq \eta \leq \frac{1}{a}$; $\gamma_{dn} < 1 \implies 0 \leq \eta \leq \frac{1}{b}$. Since $b > a$, the combined constraint from the two sets of differential modes becomes $0 \leq \eta \leq \frac{1}{b}$. The remaining constraints come from the two remaining modes. With a little work, is possible to show that the eigenvalues $\gamma_{\pm}$ are also real and positive (see Appendix), and that they provide constraints on η so that either $\eta \leq \eta^*_-$, or $\eta \geq \eta^*_+$, where

$$\eta^*_\pm = \frac{\big(a(m+2)+b(n+2)\big) \pm \sqrt{\big(a(m+2)+b(n+2)\big)^2 - 8ab(n+m+2)}}{2ab(n+m+2)} \tag{6.9}$$

One can show, from Equation (6.9), that $\eta^*_\pm$ is real, and that $\eta^*_\pm > 0$. The allowed range for η lies in the intersection of all the bounds, and will depend on where $\frac{1}{b}$ lies relative to $\eta^*_\pm$. We can expand the expression for $\eta^*_\pm$ to first order in $\frac{8\frac{b}{a}(n+m+2)}{((m+2)+\frac{b}{a}(n+2))^2}$ to see that

$$\eta^*_- \approx \frac{2}{a(m+2)+b(n+2)} < \frac{1}{b}, \tag{6.10}$$

$$\eta^*_+ \approx \frac{a(m+2)+b(n+2)}{ab(n+m+2)} > \frac{1}{b} \tag{6.11}$$

where the latter inequality for η^*_+ holds because $b > a$.

Thus, the intersection of constraints is that $\eta < \eta_{crit}$, where

$$\eta_{crit} = \eta^*_- \approx \frac{2}{a(m+2)+b(n+2)} \tag{6.12}$$

6.3 Learning time and scaling

The iteration matrix Γ is a function of both η and the eigenvalues of Q. Since we now have values of η_{crit} for both the isotropic and anisotropic cases, we can use the them to compute the eigenvalues of Γ, and derive the dependence of learning time on a, b, n, and m.

Case 1 (Isotropic learning) For ease of comparison with Case 2 below, we assume that Q has $n+m$ non-zero eigenvalues of size a. We put $\eta = \frac{\eta_{crit}}{2} = \frac{1}{a(m+n+2)}$, and find that

$$\gamma_c^* = 1 - \frac{1}{(n+m+2)} \tag{6.13}$$

$$\gamma_d^* = 1 - \frac{2}{(n+m+2)} + \frac{2}{(n+m+2)^2} \tag{6.14}$$

For each mode, the learning time can be defined to be the number of iterations requried for the error along that mode to decrease to some fraction of the initial error. If this fraction is defined to be ϵ, and if $\gamma_x^* \approx 1-2x$, then

$$T_x(\epsilon) = \frac{\log(\epsilon)}{\log(1-2x)} \approx \frac{\log(\frac{1}{\epsilon})}{2x} \tag{6.15}$$

where $T_x(\epsilon)$ is the number of iterations needed for mode x to reach ϵ times the initial error along that direction. The larger the value of x, the faster that mode will learn. We see that

$$T_c(\epsilon) \approx 2(n+m+2)\log(\frac{1}{\epsilon}) \tag{6.16}$$

$$T_d(\epsilon) \approx (n+m+2)\log(\frac{1}{\epsilon}) \tag{6.17}$$

Case 2 (Anisotropic learning) Recall that Q is assumed to have m eigenvalues of size a, and n eigenvalues of size b. Using the allowed bounds for η, we put $\eta = \frac{\eta_{crit}}{2} = \frac{1}{a(m+2)+b(n+2)}$; now we may solve for the resulting eigenvalues γ_i of Γ. The expressions for γ_{dm}, γ_{dn} give

$$\gamma^*_{dm} = 1 - \frac{2a}{(m+2)a+(n+2)b} + \frac{2a^2}{((m+2)a+(n+2)b)^2} \tag{6.18}$$

$$\gamma^*_{dn} = 1 - \frac{2b}{(m+2)a+(n+2)b} + \frac{2b^2}{((m+2)a+(n+2)b)^2} \tag{6.19}$$

Since $b/a > 1$ and $\frac{1}{(m+2)+(n+2)(b/a)} < \frac{1}{2}$, it is clear that $\gamma^*_{dn} < \gamma^*_{dm}$. The expressions for $\gamma^*_{\pm}$, which are obtained by substituting $\eta = \eta_{crit}/2$ into Equation (6.31), are more complicated:

$$\gamma^*_{\pm} = \left(1 - \frac{a+b}{p} + \frac{q}{p^2}\right) \pm \frac{b-a}{p}\sqrt{\left(1+\frac{\bar{q}}{2p(b-a)}\right)^2 + \frac{mna^2b^2}{(b-a)^2p^2}} \tag{6.20}$$

where $p \equiv (a(m+2)+b(n+2))$, $q \equiv (a^2(m+2)+b^2(n+2))$, and $\bar{q} \equiv (a^2(m+2)-b^2(n+2))$. However, to leading order (a reasonable approximation, especially for γ_+, as confirmed by numerics: see Figure 6.2), we find that

$$\gamma^*_{+} \approx 1 - \frac{2a}{(m+2)a+(n+2)b} \tag{6.21}$$

$$\gamma^*_{-} \approx 1 - \frac{2b}{(m+2)a+(n+2)b} \tag{6.22}$$

and comparing these to the expressions of Equations (6.18), we see that $\gamma^*_- \approx \gamma^*_{dn} < \gamma^*_{dm} \approx \gamma^*_+$. In other words, the $n-1$ differential modes $(0, d_n)$ and the mode $(1_m, \delta_{(-)}1_n)$ are learned roughly equally fast, and faster than the $m-1$ differential modes $(d_m, 0)$ and $(1_m, \delta_{(+)}1_n)$.

Applying Equation (6.15), we find that

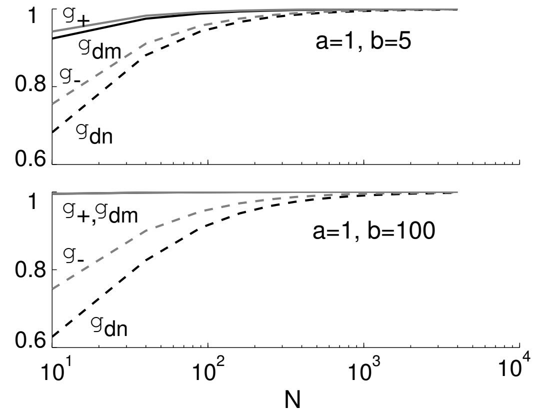


Figure 6.2: Plotted are the four distinct eigenvalues of the recursion matrix Γ. *Top* The eigenvalues of the cost matrix Q are $a = 1$ and $b = 5$. *Bottom* The eigenvalues of Q are $a = 1$ and $b = 100$. In both plots, the system size, N, is varied between 10 and 4000; $N/4$ of the eigenvalues are b, while the remaining are a; finally, $\eta = \eta_{crit}/2$. It is clear from the figure that $\gamma^*_{dm} \approx \gamma^*_+$, and that γ^*_{dn} and γ^*_- bunch together.

$$T_{dm}(\epsilon),\ T_+(\epsilon) \approx \left((m+2) + (n+2)\frac{b}{a}\right)\log(\frac{1}{\epsilon}) \tag{6.23}$$

$$T_{dn}(\epsilon),\ T_-(\epsilon) \approx \frac{(m+2) + (n+2)\frac{b}{a}}{(b/a)}\log(\frac{1}{\epsilon}) \tag{6.24}$$

We can further simplify the expression of Equation (6.5) for the learning curve. Since the components of a differential mode sum to zero, $\sum_i \lambda_i v_{i\alpha} = 0$ if v_α is a differential mode and $\lambda_i = a \ \ \forall i$, which is the case in the isotropic case. For the differential modes α in the anisotropic case also, $\sum_i \lambda_i v_{i\alpha} = 0$. Hence, the only contribution to the cost, and to the learning process, comes from the remaining, non-differential modes.

$$C^{(t)}_{\text{iso}} = (\gamma^*_c)^t\, C_0 \tag{6.25}$$

$$C^{(t)}_{\text{aniso}} = (\gamma^*_+)^t\, C_+ + (\gamma^*_-)^t\, C_- \tag{6.26}$$

where the initial cost in the isotropic learning problem is $C_0 \equiv a\sum_j (x^{(0)}_j)^2$, and in the anisotropic learning problem is $C_0 = C_+ + C_-$, with $C_+ \equiv \frac{ma+\delta_+ nb}{m+\delta_+^2 n} \sum_j (v^{-1})_{N-1,j} (x_j^{(0)})^2$, and $C_- \equiv \frac{ma+\delta_- nb}{m+\delta_-^2 n} \sum_j (v^{-1})_{N,j} (x_j^{(0)})^2$. The expression for v^{-1} is given in the appendix.

The final learning curve for stochastic gradient descent on an isotropic cost surface, in Equation (6.25), is a single exponential, as in the case of direct gradient descent (see below), but with a different learning speed. The final learning curve, Equation (6.26), for learning on a cost surface with eigenvalue spectrum $(a, ..., a, b, ..., b)$, is bi-exponential, with two time-constants of decay, once again in agreement with the direct gradient case, but with different learning speeds. There is an important difference, however, which we will discuss below.

6.3.1 Comparison of anisotropic and isotropic stochastic learning

As we saw in Figure 6.1, if a subset of the eigenvalues of Q provide the bulk of the error, then learning can be much faster than if all the eigenvalues are equally involved. That is, it is possible, if we are not concerned about error very close to zero, for learning with anisotropic cost surfaces to be *much* faster than with an isotropic cost surface. With the help of the analysis of the past few sections, and results of the analysis presented in the last section, we can quantify this statement.

From Equations (6.25)–(6.26), we see that if $\gamma_- \ll \gamma_c \ll \gamma_+$ then the total error in the anisotropic case will decrease much more rapidly than the total error in the isotropic case, until $C^{(t)} = C_+/(C_+ + C_-)$, when learning on the isotropic cost surface will overtake anisotropic learning. If C_- forms a major part of the total error in the anisotropic case, i.e., if $C_- \gg C_+$, then the overall learning will appear faster in the anisotropic case than in the isotropic case. Below, we consider the actual relative sizes of γ_+^*, γ_-^*, and γ_c.

Assuming that $n + m = N$ is large, we write down the approximate convergence times with isotropic and anisotropic learning. In the isotropic case, all modes are equally important and learning time scales like $T \sim (n + m + 2) \sim N$. In the anisotropic case, let us consider the following case: the n modes with eigenvalue b contribute to a fraction f of the summed total of the eigenvalues of Q. Then,

$$T^{\text{iso}} \sim N \tag{6.27}$$

$$T_b^{\text{aniso}} \sim n + \frac{n(1-f)}{f} \tag{6.28}$$

$$T_a^{\text{aniso}} \sim N + \frac{Nf - n}{(1-f)} \tag{6.29}$$

If f is close to one, and $n \ll N$, then $(T_b^{\text{aniso}} \sim n) \ll (T^{\text{iso}} \sim N) \ll T_a^{\text{aniso}}$; thus, learning along the fast modes of the anisotropic surface would be much faster than learning along the isotropic surface; if n is independent of N, that is, if the number of important modes does not scale with N, then learning could be rather fast, and not scale as N, but as n.

Let us compare these predictions with the example of Figure 6.1: in one case, $N/20$ of the eigenvalues were assumed to comprise 90% of the total eigenvalue sum, with $b = 171a$. With these numbers, $T^{\text{iso}} \sim N, T_b^{\text{aniso}} \sim N/20$, while $T_a^{\text{aniso}} \sim 9N$. Thus we expect that for the anisotropic cost surface (dot-dashed grey curve), learning along the n fast modes will more rapid, by a factor of 20, than learning along an isotropic surface (solid grey curve); the anisotropic learning curve should moreover display a 'knee', marking the transition from learning along the fast modes to learning along the slower modes. Finally, the full learning curve in the anisotropic case should be faster than the isotropic learning curve until an error is reached where learning is dominated by the slow mode; at this point, learning will be faster, by a factor of 9, for the isotropic surface. Comparing the dot-dashed and solid grey curves, we see that the difference in learning time to reach down to an error of 10 % of the initial error, which marks the begining of the knee, is a factor of 10–20, in agreement with our expectations. The knee leads to a phase of learning of the slow mode; when plotted on a log-linear scale, the two anisotropic learning curves look linear with a transition between two separate slopes, while the isotropic learning curve is linear.

6.3.2 Comparison with direct gradient learning

The analysis for direct gradient descent on this class of quadratic cost functions for linear networks is much simpler. The recursion matrix for y in this case is $\Gamma = \delta_{ij}(1-\eta\lambda_i)^2$; this matrix is purely diagonal, and unlike the stochastic case where the different learning modes are entangled, here learning along each mode can progress independently. There is no residual error. The eigenvalues $\{\gamma_i\}$ of Γ are trivially $\{(1-\eta\lambda_i)^2\}$, and the convergence condition $|\gamma_i| < 1\,\forall\, i$ gives $\eta < 2/\lambda_i$ for each mode. The intersection of all the bounds is

$$\eta < \eta_{crit} = \frac{2}{\lambda_{\max}} \tag{6.30}$$

where $\lambda_{\max}$ is the largest eigenvalue in the spectrum of Q. The optimal η for fastest convergence is $\eta^* = \frac{1}{2}\eta_{crit} = \frac{1}{\lambda_{\max}}$. For modes with $\lambda < \lambda_{\max}$, the number of steps needed for this mode to go below the fraction ϵ of the initial error will be

$$T(\epsilon) = \frac{1}{2}\frac{\log\epsilon}{\log(1-\lambda/\lambda_{\text{max}})}.$$

These exact results for arbitrary eigenvalue spectra $\{\lambda_i\}$ can be applied to the two special cases considered for stochastic learning above. In Case 1 ($\lambda_i = a \ \forall\, i$), $\eta^* = 1/a$, and so $\gamma_i = 0 \ \forall\, i$, and the error goes to 0 in one step. In Case 2 (m eigenvalues of size a and n of size $b > a$), $\eta^* = 1/b$; thus, the top n eigenmodes converge to 0 error after one step, and error along the remaining m modes decreases by $(1-a/b)^{2t}$, so that $C^{(t)}$ likewise goes as $(1-a/b)^{2t}$ independent of m and n. $T_m(\epsilon) = \frac{1}{2}\frac{\log\epsilon}{\log(1-a/b)}$; $T_n(\epsilon) = 1 \ \forall\,\epsilon$. The ratio of the learning speed for direct as compared to stochastic gradient descent scales linearly with m and n, paralleling the result derived in [59].

Interestingly, in contrast to the stochastic case however, direct gradient learning on an isotropic cost surface is *always* faster or as fast as any of the modes in anisotropic learning. To summarize the difference, we found that with stochastic gradient learning, some modes on an anisotropic cost surface can be learnt much faster than the modes in isotropic learning. With direct gradient, however, there are no modes in the anisotropic case that can learn faster than in the isotropic case; in the most favorable scenario, some of the anisotropic modes can be equally fast, but not faster than, isotropic learning. Thus, in more realistic scenarios where learning is by stochastic gradient, an anisotropic learning problem need not be automatically slower than an isotropic one; in fact, the reverse may be true.

6.4 Applications to neural network learning

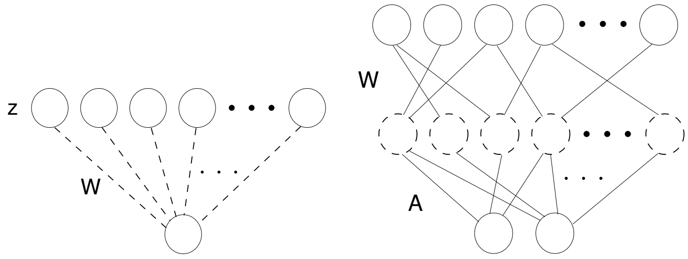
The preceeding analysis, of learning with stochastic gradient descent on a general quadratic cost function, can be applied directly to learning with weight or node perturbation in two different linear neural networks.

- Learning by weight perturbation in a single layer network (Figure 6.3).

Assume the network has N input units, 1 output unit, and is being trained to learn P patterns, and define $x \equiv W$ to be the layer of plastic network weights. Then, the cost is $C = WQW^T$, where $Q = zz^T$ is the covariance matrix of the input patterns. Learning by weight perturbation gives exactly the same update rule for W as learning by stochastic gradient descent, as described in the past sections.

- Learning in a network with one hidden layer and a single plastic layer of weights (Figure 6.3).

We assume the network has N_i input units, N_h hidden units, N_o output units, and is being trained to learn 1 pattern, with weight or node perturbation. The weights W from the input to the hidden layer are assumed to be plastic, and the weights A from the hidden layer to the outputs are assumed fixed. In weight perturbation, only the plastic weights are perturbed, while in node perturbation only the units postsynaptic to the plastic weights are perturbed. Then, the cost is $C = (1^T W^T Q W 1)$, where $Q = A^T A$. Learning with either rule, on one pattern, follows learning as described in the past sections. The results could also apply to the learning of $P < N$ patterns if the P patterns are orthogonal, such that $zz^T = I$. For more details, cf. the section on scaling in Chapter ch:birdsong.



(a) 1-layer network, weight perturbation

(b) 2-layer network, node perturbation

Figure 6.3: The analysis of stochastic gradient descent on an anisotropic quadratic cost function can be applied directly to problems in neural network learning. The solutions can be mapped directly to the learning, with weight perturbation, of P patterns in a single-layer linear perceptron, where $x \equiv W$ is the set weights from N inputs to a single output, and $Q = zz^T$ is the $N \times N$ covariance matrix of the P inputs. Alternatively, the analysis applies to the learning of one pattern in a 2-layer network with one hidden layer and multiple output units.

Next, we compute the eigenvalue spectrum of the cost function of a real-world handwritten digit data set, used to train neural networks to perform digit recognition, Figure (**??**).

We see from the cumulative sum of the eigenvalues that the bulk (>80 %) of the eigenvalue sum comes from the top 40 eigenvalues. In other words, it is only a few eigenvectors (40 out of 784 possible modes), corresponding to the top few eigenvalues, that really need to be learned. Thus, learning time should depend not on $1/N = 1/784$, but rather as $1/40$, or the number of vectors with large eigenvalue, the principal components.

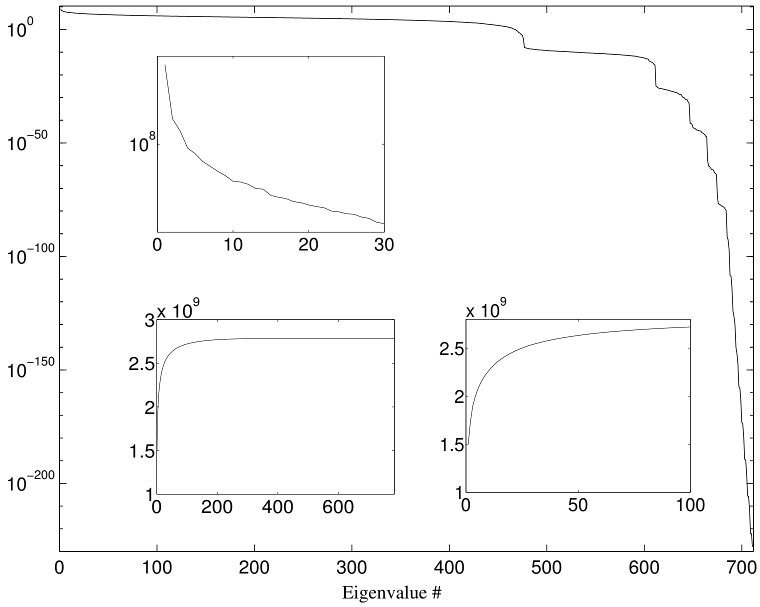


Figure 6.4: Eigenvalue spectrum and derived quantities for a data set of 550 exemplars of the digit ‘7’ from the NIST handwritten digit database []. Main figure, eigenvalue spectrum of the covariance matrix of these inputs, $Q = zz^T$. Insets: top left, closeup of largest eigenvalues; bottom left, cumulative sum of eigenvalues; bottom right, closeup of first 100 terms of the cumulative sum. The first 30-40 eigenvectors make up the bulk of the variation in the data set, by principal components analysis.

We can compute the eigenvalues of the combined covariance matrix of 5000 samples of all 10 digits, from 0 to 9, and find once again that the top 40 eigenvalues contribute ¿80% of the total eigenvalue sum. Examining by eye the reconstruction of the digits based on the first P principal components, we found that $P \approx 40$ is sufficient to identify almost all digits; in other words, 40 eigenvectors make up the bulk of the variation in the data set.

Studies of scaling behavior with network size frequently rely on the unwritten assumption that the difficulty of the task the network is trained on grows along with the size of the network. While this is sometimes true, it is not so in many real-world cases. Figure 6.5 shows the cumulative sums of the eigenvalue spectra for the dataset of Figure 6.4, and a variant thereof obtained by reducing the image to half its original resolution in both linear dimensions. The total number of pixels, and thus number of eigenvalues, thus decreases by a factor of four; however the number of eigenvalues at which the cumulative eigenvalue sum levels off—that is, the number of eigenvalues contributing to the bulk of the variance, and hence the effective dimensionality of the problem—is nearly unchanged.

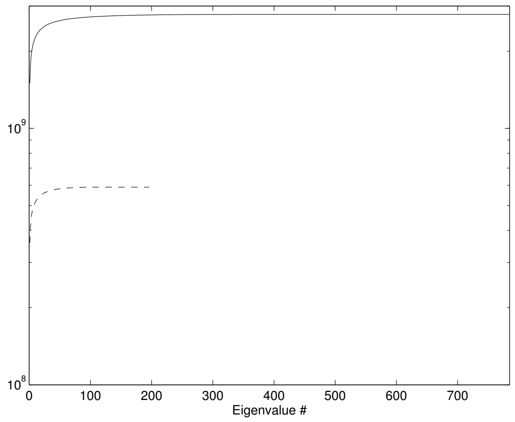


Figure 6.5: Demonstration of the extent of important information in the digit data set. Solid line: cumulative sum of the eigenvalues of the covariance matrix for the digit 7, as in Figure 6.4. Dotted line: cumulative sum of all the eigenvalues for a data set generated from the first by downsampling the pixel resolution by a factor of two in both linear dimensions of the image. The latter has only 1/4 as many eigenvalues as the former, but number of eigenvectors making up the bulk of the eigenvalue sum is about the same in both cases. The first 30 eigenvectors This is an

Note that these results imply that if the difficulty of the task (i.e., the number of important eigenvectors, or effective dimension of the problem) is kept the same, then learning time does not increase even as the network size is increased.

6.5 Appendix

6.5.1 Derivation of γ_i and bounds on η

Γ is a block matrix,

$$\Gamma = \begin{bmatrix} (1-2a+2a^2)I + a^2 11^T & b^2 11^T \\ a^2 11^T & (1-2b+2b^2)I + b^2 11^T \end{bmatrix}$$

Since any matrix of form $\alpha I + \beta 11^T$ has one common mode vector, 1, and $N-1$ differential modes d (defined as $\sum_i d_i = 0$), we guess a similar form for the eigenvectors of Γ: $m-1$ differential modes $(0, d_m)$, where $\sum_i d_{mi} = 0$, $n-1$ differential modes $(d_n, 0)$, where $\sum_i d_{ni} = 0$, and two common modes, $(1_m, \delta_{(\pm)} 1_n)$. The differential mode eigenvalues, γ_m and γ_n are very easy to compute. Here we compute the two common mode eigenvalues, $\gamma_\pm$, and obtain bounds on η from the condition $0 \leq Re(\gamma_\pm) \leq 1$.

Get a quadratic equation for $\gamma_\pm$ from solving the eigenvalue equation,

$$\gamma_\pm = \frac{1}{2}\left[(A+D) \pm \sqrt{(A-D)^2 + 4BC}\right] \qquad (6.31)$$

where $A \equiv (1 - 2\eta a + (m+2)\eta^2 a^2)$, $D \equiv (1 - 2\eta b + (n+2)\eta^2 b^2)$, $B = n\eta^2 b^2$, and $C = m\eta^2 a^2$. Then,

$$\gamma^2 - \gamma(A+D) + AD - BC = 0 \qquad (6.32)$$

where $A \equiv (1-2a+(m+2)a^2)$, $D \equiv (1-2b+(n+2)b^2)$, $B = nb^2$, and $C = ma^2$.

From this we have that

$$\begin{aligned}\gamma_\pm &= \frac{1}{2}\big[(A+D) \pm \sqrt{(A+D)^2 - 4(AD-BC)}\big] &(6.33)\\ &= \frac{1}{2}\big[(A+D) \pm \sqrt{(A-D)^2 + 4BC}\big] &(6.34)\end{aligned}$$

Notice from their definitions that A, B, C and D are positive. From Equation (6.34) we see therefore that the argument of the square root is positive, so $\gamma_\pm$ must be real. Moreover, γ_+, γ_- are both positive, with $\gamma_+ < \gamma_-$. The only non-trivial bound on $\gamma_\pm$ is therefore $\gamma_+ < 1$, which gives

$$(AD - BC) - (A + D) + 1 > 0. \tag{6.35}$$

Since A, B, C and D each contain η^2, this inequality appears to be quartic in η; when fully expanded, however, the two lowest powers of η drop out, and we get that

$$\eta^2(ab(m+n+2)) - \eta(a(m+2) + b(n+2)) + 2 > 0 \tag{6.36}$$

This is an upward-facing parabola in η, giving allowed values of η to be $\eta < \eta^*_-, \eta > \eta^*_+$, where

$$\eta^*_\pm = \frac{\big(a(m+2) + b(n+2)\big) \pm \sqrt{\big(a(m+2)+b(n+2)\big)^2 - 8ab(m+n+2)}}{2ab(m+n+2)} \tag{6.37}$$

6.5.2 Inverse of the Γ eigenvector matrix

If the cost surface is anisotropic, so that Q has an eigenvalue spectrum consisting of m as and $n = N - m$ bs, the eigenvector matrix v of Γ is as described in Section

??; since the differential modes are orthogonal to each other as well as to the two remaining modes, the inverse matrix is almost the transpose of v. However, the two non-differential modes couple, and the inverse matrix is

$$v^{-1} = \left(v_1, ..., v_{N-2}, \frac{v_+ - (v_+ \cdot v_-)v_-}{1-(v_+ \cdot v_-)^2}, \frac{v_- - (v_+ \cdot v_-)v_+}{1-(v_+ \cdot v_-)^2} \right)^T .$$

6.5.3 Generalization to $M > 1$ output units

We proceed in analogy to the 1-output case; now, x is a matrix of size $M \times N$, and $C = \mathrm{Tr}(xQx^T)$. Since there are NM noises injected into the system, driving independent exploration in as many dimensions, we need to track all NM variables, x_ij^2, to follow the learning curve. If we write the columns of x and ξ in the eigenvector basis of Q, then after one iteration the prescribed change in x_{ij} is given by $\Delta x_{ij} = -\frac{\eta}{\sigma^2 a}\big(\mathrm{Tr}(\xi Q x^T) + \frac{1}{2}\mathrm{Tr}(\xi Q \xi^T)\big)\xi_ij$; we use this to obtain an expression for $(x_{ij}^{(1)})^2$ as a function of $(x_{ij}^{(0)})^2$, and compute the expectation over the distribution of ξ to get:

$$\begin{aligned}\langle (x_{ij}^{(1)})^2 \rangle_\xi \quad &= \quad \left[(x_{ij}^{(0)})^2 - 2\eta\lambda_j (x_{ij}^{(0)})^2 + \eta^2 \left(2\lambda_j^2 (x_{ij}^{(0)})^2 + \sum_{ab} \lambda_b^2 (x_{ab}^{(0)})^2 \right) \right] \\ &\quad + \frac{1}{4}\eta^2\sigma^2 \left(8\lambda_j^2 + 4M\lambda_j \sum_k \lambda_k + 2M \sum_k \lambda_k^2 + M^2 (\sum_k \lambda_k)^2 \right)\end{aligned}$$

At this point, however, we notice that it is not possible to write down a linear recursion relation for x_{ij}^2, because of the presence of the $\sum_{ab} \lambda_b^2 x_{ij}^2$ term in the equation above. Thus, although it is possible to compute the noise-averaged 1-iteration learning update, it is not possible to derive a general recursion relation, since the next step in the recursion depends in detail on the values of x in the preceding step. This is an obstacle that prevents further analysis of the multiple-output stochastic gradient learning problem.

Bibliography

[1] D. O. Hebb, *Organization of Behavior: A Neuropsychological Theory* (Wiley, Inc., ADDRESS, 1949).

[2] T. Bliss and T. Lomo, J Physiol. **232**, 331 (1973).

[3] G. Tong, R. Malenka, and R. Nicoll, Neuron **16**, 1147 (1996).

[4] H. Markram, J. Lbke, M. Frotscher, and B. Sakmann, Science **275**, 213 (1997).

[5] L. Zhang *et al.*, Nature **395**, 37 (1998).

[6] G. Bi and M. Poo, J Neurosci **18**, 10464 (1998).

[7] D. Amit, N. Brunel, and M. Tsodyks, J Neurosci. **14**, 6435 (1994).

[8] S. Dehaene, J. Changeux, and J. Nadal, Proc Natl Acad Sci U S A. **84**, 2727 (1987).

[9] T. Nowotny, M. Rabinovich, and H. Abarbanel, Phys Rev E Stat Nonlin Soft Matter Phys. **68**, 011908. Epub 2003 Jul 18. (2003).

[10] S. Loo and M. Bitterman, J Comp Psychol. **106**, 29 (1992).

[11] M. Giurfa *et al.*, Nature. **410**, 930 (2001).

[12] M. Hammer, Nature **366**, 59 (1993).

[13] M. Hammer, Trends Neurosci. **20**, 245 (1997).

[14] M. Hammer and R. Menzel, Learn Mem. **5**, 146 (1998).

[15] M. Schwaerzel *et al.*, J Neurosci. **23**, 10495 (2003).

[16] J. Mirenowicz and W. Schultz, Nature. **379**, 449 (1996).

[17] W. Schultz, Neuron. **36**, 241 (2002).

[18] M. Ungless, P. Magill, and J. Bolam, Science. **303**, 2040 (2004).

[19] R. Wise, Neuron. **36**, 229 (2002).

[20] L. Stein and J. Belluzzi, Neurosci Biobehav Rev. **13**, 69 (1989).

[21] H. Stark and H. Scheich, J Neurochem. **68**, 691 (1997).

[22] S. Bao, V. Chan, and M. Merzenich, Nature. **412**, 79 (2001).

[23] S. Bao, V. Chan, L. Zhang, and M. Merzenich, Proc Natl Acad Sci U S A. **100**, 1405 (2003).

[24] M. Kilgard and M. Merzenich, Science. **279**, 1714 (1998).

[25] N. Daw, S. Kakade, and P. Dayan, Neural Netw. **15**, 603 (2002).

[26] E. Thorndike, *Animal Intelligence* (Macmillan, New York, ADDRESS, 1911), p. 244.

[27] C. Gallistel, T. Mark, A. King, and P. Latham, J Exp Psychol Anim Behav Process. **27**, 354 (2001).

[28] M. Jabri and B. Flower, IEEE Transactions on Neural Networks **3**, 154 (1992).

[29] B. Widrow and M. Lehr, Proceedings of the IEEE **78**, 1415 (1990).

[30] R. Hahnloser, A. Kozhevnikov, and M. Fee, Nature **419**, 65 (2002).

[31] H. Seung, Neuron. **40**, 1063 (2003).

[32] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, CA, 1991).

[33] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons* (Oxford University Press, New York, New York 10016, 1999).

[34] S. Haykin, *Neural Networks: A Comprehensive Foundation* (Prentice Hall, Upper Saddle River, New Jersey 07458, 1999).

[35] R. Horn and C. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, UK, 1999).

[36] H. Simpson and D. Vicario, J Neurosci **10**, 1541 (1990).

[37] F. Nottebohm, D. Kelley, and J. Paton, J Comp Neurol **207**, 344 (1982).

[38] F. Nottebohm, T. Stokes, and C. Leonard, J Comp Neurol **165**, 457 (1976).

[39] R. Mooney, J Neurosci **12**, 2464 (1992).

[40] L. Stark and D. Perkel, J Neurosci **19**, 9107 (1999).

[41] D. Willshaw, O. Buneman, and H. Longuet-Higgins, Nature **222**, 960 (1969).

[42] M. Tsodyks and M. Feigelman, Europhys Lett **6**, 101 (1988).

[43] C. Meunier, H. Yanai, and S. Amari, Network: Comput. Neural Syst. **2**, 469 (1991).

[44] M. Hermann, J. Hertz, and A. Prugel-Bennett, Network **6**, 403 (1995).

[45] M. Konishi, Z Tierpsychol **22**, 770 (1965).

[46] M. Brainard and A. Doupe, Nat Rev Neurosci. **1**, 31 (2000).

[47] K. Doya and T. Sejnowski, in *Advances in Neural Information Processing Systems 7*, edited by G. Tesauro, D. Touretzky, and T. Leen (MIT Press, Cambridge, MA, 1995), pp. 101–108.

[48] T. Troyer and A. Doupe, J Neurophysiol **84**, 1204 (2000).

[49] R. Williams, Machine Learning **8**, 229 (1992).

[50] P. Bartlett and J. Baxter, Technical report (unpublished).

[51] A. Yu and D. Margoliash, Science **273**, 1871 (1986).

[52] Y. Le Cun, I. Kanter, and S. Solla, Physical Review Letters **66**, 2396 (1991).

[53] D. Margoliash, J Neurosci. **6**, 1643 (1986).

[54] D. Margoliash and E. Fortune, J Neurosci. **12**, 4309 (1992).

[55] S. Volman, J Neurosci. **13**, 4737 (1993).

[56] M. Lewicki and M. Konishi, Proc Natl Acad Sci U S A. **92**, 5582 (1995).

[57] P. Foldiak, in *The Handbook of Brain Theory and Neural Networks*, edited by M. Arbib (MIT Press, Cambridge, MA, 1995), pp. 895–98.

[58] S. Edwards and R. Jones, J Physics A **9**, 1595 (1976).

[59] J. Werfel, X. Xie, and H. Seung, in *NIPS* (PUBLISHER, ADDRESS, 2003).

[60] A. Barto and P. Anandan, IEEE Transactions on Systems, Man, and Cybernetics **15**, 360 (1985).

[61] A. Barto, R. Sutton, and C. Anderson, IEEE Transactions on Systems, Man, and Cybernetics **13**, 835 (1983).

[62] W. Thorpe, *Bird-Song* (Cambridge Univ. Press, Cambridge, 1961).

[63] P. Marler, J. Comp. Physiol. Psychol. **71**, 1 (1970).

[64] M. Fee, B. Shraiman, B. Pesaran, and P. Mitra, Nature **395**, 67 (1998).

[65] O. Tchernichovski, P. Mitra, T. Lints, and F. Nottebohm, Science **291**, 2564 (2001).

[66] S. Bottjer, E. Miesner, and A. Arnold, Science. **224**, 901 (1984).

[67] C. Scharff and F. Nottebohm, J Neurosci. **11**, 2896 (1991).

[68] N. Hessler and A. Doupe, Nat Neurosci. **2**, 209 (1999).

[69] R. Canady, G. Burd, T. Devoogd, and F. Nottebohm, J Neurosci. **8**, 3770 (1988).

[70] K. Herrmann and A. Arnold, J Neurosci. **11**, 2063 (1991).

[71] H. Sakaguchi and N. Saito, Brain Res Dev Brain Res. **95**, 245 (1996).

[72] K. Immelmann, in *Bird Vocalizations*, edited by R. Hinde (Cambridge Univ. Press, New York, 1969), pp. 61–74.

[73] P. Marler, in *Acoustic Behavior of Animals*, edited by R. Busnel (Elsevier, Amsterdam, 1964).

[74] A. Lombardino and F. Nottebohm, J Neurosci. **20**, 5054 (2000).

[75] G. Cauwenberghs, NIPS (1993).

[76] G. Beckers, R. Suthers, and C. T. C, Proc Natl Acad Sci U S A. **100**, 7372 (2003).

[77] S. Nowicki, Nature. **325**, 53 (1987-7).