2-2018

Gradient Estimation for Attractor Networks

Thomas Flynn
The Graduate Center, City University of New York

How does access to this work benefit you? Let us know!
Follow this and additional works at: https://academicworks.cuny.edu/gc_etds
Part of the Artificial Intelligence and Robotics Commons, Dynamical Systems Commons, and the Theory and Algorithms Commons

Recommended Citation
Flynn, Thomas, "Gradient Estimation for Attractor Networks" (2018). CUNY Academic Works.
https://academicworks.cuny.edu/gc_etds/2456

This Dissertation is brought to you by CUNY Academic Works. It has been accepted for inclusion in All Dissertations, Theses, and Capstone Projects by an authorized administrator of CUNY Academic Works. For more information, please contact deposit@gc.cuny.edu.
GRADIENT ESTIMATION FOR ATTRACTOR NETWORKS

by

Thomas Flynn

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2018
Gradient Estimation for Attractor Networks
by
Thomas Flynn

This manuscript has been read and accepted by the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Professor Felisa Vázquez-Abad

Date
Chair of Examining Committee

Professor Robert Haralick

Date
Executive Officer

Professor Felisa Vázquez-Abad
Professor Robert Haralick
Professor Chao Chen
Professor Vikram Krishnamurthy
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK
Abstract
Gradient Estimation for Attractor Networks
by
Thomas Flynn

Advisor: Professor Felisa Vázquez-Abad

It has been hypothesized that neural network models with cyclic connectivity may be more powerful than their feed-forward counterparts. This thesis investigates this hypothesis in several ways. We study the gradient estimation and optimization procedures for several variants of these networks. We show how the convergence of the gradient estimation procedures are related to the properties of the networks. Then we consider how to tune the relative rates of gradient estimation and parameter adaptation to ensure successful optimization in these models. We also derive new gradient estimators for stochastic models. First, we port the forward sensitivity analysis method to the stochastic setting. Secondly, we show how to apply measure valued differentiation in order to calculate derivatives of long-term costs in general models on a discrete state space. Throughout, we emphasize how the proper geometric framework can simplify and generalize the analysis of these problems.
Acknowledgments

Firstly, I would like to thank my advisor Felisa Vázquez-Abad, who inspired me not only with her deep knowledge of probability but also with her keen sense for applications. Her door was always open and I feel that I learned a lot about being an effective researcher from her. Ioannis Stamos was extremely influential in teaching me about research, and I gained valuable hands on experience in the Vision and Graphics Lab. Olympia Hadjiliadis also collaborated with us and lent many great ideas.

I had the opportunity to work with people at many CUNY campuses during my studies as part of the fellowship I received. In connection with this I would like to thank Katherine St. John and Sameh Fakhouri of Lehman college, and Susan Ko of SPS.

I’m grateful for the for the camaraderie of my lab mates, including Allan Zellener, Juan Liu, Syed Ali Ahmed and Alexey Nikolaev,

Also, I would like to give special thanks to an early mentor George Metzger of MS 104, who encouraged me to purse my interest in computers and programming back then.

Lastly, I would like to thank Nina Lichtman, and my mother father and brother for all their invaluable support and encouragement.
Contents

Contents vi

List of Tables x

List of Figures xi

1 Introduction 1

1.1 Motivation ........................................... 1
1.2 Outline ............................................. 5
1.3 Notations ........................................... 6

2 Background 8

2.1 Machine Learning Problem ................................. 8
2.2 Models .............................................. 11
   2.2.1 Attractor Networks on a Continuous State Space ........ 11
   2.2.2 Stochastic Attractor Networks .......................... 13
   2.2.3 Discrete Attractor Networks ............................ 16
2.3 Contributions ........................................ 17
   2.3.1 Attractor Networks on a Continuous State Space .......... 17
   2.3.2 Stochastic Attractor Networks .......................... 25
   2.3.3 Discrete Attractor Networks ............................ 27
CONTENTS

3 Methods

3.1 Contraction Analysis ........................................ 31
  3.1.1 Hierarchies of Contractions .......................... 33
3.2 Discrete Time Contractions ................................. 34
  3.2.1 Hierarchies of Contractions ......................... 35
3.3 Analysis of Gradient Descent .............................. 36
  3.3.1 Continuous Time .................................. 36
  3.3.2 Discrete Time .................................. 38
3.4 Finsler Structures ...................................... 39
3.5 Wasserstein Distances .................................. 40
3.6 Measure Valued Differentiation ......................... 42
  3.6.1 Measure Valued Differentiation for Markov chains .. 43
3.7 Simultaneous Perturbation ................................ 45

4 Deterministic Attractor Networks ......................... 47

4.1 Continuous Time ...................................... 47
  4.1.1 Optimization System .................................. 50
  4.1.2 Stability Criteria .................................. 53
  4.1.3 Optimization Criteria .................................. 56
  4.1.4 Time-scale Selection .................................. 57
  4.1.5 Application to the Optimization System ............... 59
  4.1.6 Neural Networks .................................. 64
  4.1.7 Discussion .................................. 71
4.2 Discrete Time ...................................... 72
  4.2.1 Assumptions .................................. 75
  4.2.2 Optimization Criteria .................................. 75
### CONTENTS

4.2.3 Time-scale Selection .......................................................... 76
4.2.4 Step-size and Initial Condition Constraints .......................... 81
4.3 Numerical Experiments ........................................................... 83
  4.3.1 Synthetic Data ............................................................... 84
  4.3.2 MNIST Experiment .......................................................... 88
4.4 Discussion ............................................................................. 91

5 Stochastic Attractor Networks ...................................................... 93
  5.1 Gradient Estimation for Stationary Markov Chains .................. 93
  5.2 Overview of Main Results ...................................................... 95
  5.3 Notations ............................................................................ 97
  5.4 Assumptions ....................................................................... 97
  5.5 Contraction Framework ........................................................ 100
    5.5.1 Interconnections of Contractions ................................... 105
  5.6 Stationary Differentiability ..................................................... 109
  5.7 State Space Conditions ........................................................ 112
  5.8 Gradient Estimation ............................................................ 121
  5.9 Examples .......................................................................... 128
  5.10 Discussion ....................................................................... 130

6 Discrete Attractor Networks .......................................................... 132
  6.1 Ergodicity Properties of the Little Model ................................. 133
  6.2 Gradient Estimation ............................................................. 135
    6.2.1 SPMVD .................................................................. 136
    6.2.2 Application to the Little model ....................................... 138
  6.3 Numerical Experiments .......................................................... 143
  6.4 Discussion ....................................................................... 144
## List of Tables

4.1 Stability conditions for the network $\frac{dx}{dt} = f(x(t), w) = -x(t) + \sigma(wx(t) + \rho)$ for different choices of the vector norm. In the first row $\sigma'(u)$ refers to the diagonal matrix whose $(j, j)$ entry is $\sigma'(u_j)$. Note that the third column uses a matrix norm. ............................................................... 64

4.2 Derivatives and bounds for the recurrent network (4.17) ........................................ 66

4.3 Derivatives for the model defined by equations (4.18a) and (4.19) ....................... 70

4.4 Experiment results ................................................................. 90

4.5 Experiment results (restricted feedback) ......................................... 91
List of Figures

1.1 A feed-forward network (left) and an attractor network (right) . . . . . . 3
1.2 In standard gradient based optimization schemes (a), the search direction $\Delta_n$ at time $n$ is calculated based solely on the parameter $w_{n-1}$. In dynamic gradient estimation schemes (b), the search directions $\Delta_n$ are computed based on the current parameter and the state $y_n$ of an auxiliary system. . . . . . 5

2.1 Plots of two common activation functions used in neural networks. See text for formal definitions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.2 Gradient estimation and optimization in different dynamical settings for neural networks. The first level splits into networks that are stochastic or deterministic. At the next level the split is whether the state space of the network is discrete or continuous. At the third level we differentiate based on connectivity constraints; whether the network is acyclic, has symmetric connections, or allows general connectivity. At the fifth level the split is based on the order of updating the nodes - whether they are they updated all at once (synchronous) or one at a time, in an asynchronous manner. This work considers gradient estimation for three types of networks. A question mark (?) means the author is unaware of any works considering gradient estimation in models with those properties. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
3.1 Visualization of the angle condition for gradient descent used in Proposition 3.3.3 below. 38

4.1 Update pattern for the adjoint-method studied in Section 4.2. \((x, y)\) is the state of the joint system, where \(x\) is the state of the network and \(y\) is the state of the adjoint system. The left graph shows how \((x, y)\) and \(w\) are calculated at time \(t\) from their values at time \(t-1\). On the right is the update pattern of the map \(T\), showing details of how \((x(t), y(t))\) are calculated from \((x(t-1), y(t-1))\). 73

4.2 A ring network was used for experiments. The network activity proceeds along the solid arrows while the adjoint system flows in the opposite direction. 83

4.3 Error trajectories for different step-sizes \(\epsilon\) using the one-step method. Each row corresponds to a different choice of step-size and the column indicates the iteration. The color indicates the approximate error at that step of optimization. 85

4.4 Error trajectories for different step-sizes \(\epsilon\), when very accurate gradient estimates are used. The procedure is more stable at large step-sizes but this requires more time spent approximating gradients. 86

4.5 A network with “full” feedback. There is lateral feedback among the input and output nodes and top-down feedback from output to input. 89

4.6 Left: A network with no feedback. Right: An attractor network with no feedback connections on the input nodes. 90

6.1 Error trajectories of MVD-based optimization for different values of \(M^1\). 143
Chapter 1

Introduction

1.1 Motivation

There are arguably two approaches to using computers to solve a particular problem. In the first case, a skilled software developer can exactly express to the computer how a problem is to be solved. Starting from a precise description of a problem written in a natural language, they express the solution in one of any number of programming paradigms - be it functional, imperative, logic based or otherwise. This approach can be summarized by saying that the programmer “tells the computer what to do”. An amazing number of problems have been approached in this way - from airline reservations, networking and communications applications, financial systems, business information systems and databases, operating systems, and many more. However, it seems that there are problems that cannot be solved by this approach. Certain problems appear to be too complex for us to solve them by “telling the computer what to do”. These include image classification, natural language processing, speech understanding, and other problems that are related to extracting higher order information from “natural” low-level signals. In these problems, the most promising approach is one in which the engineer “shows the computer what to do” by presenting many examples
of how the software should operate. This is the machine learning approach.

In the machine learning approach, one first identifies a family of programs defining potential solutions to the problem. Data in the form of examples of how the program should respond to different inputs are used by an algorithm to automatically select a candidate solution from the family. This process of finding the best candidate solution given the data is an example of optimization. The goal of the optimization algorithm in this case is to find the model which minimizes a measure of the discrepancy between the current behavior and the desired behavior.

A popular choice for the family of models are the neural networks. A neural network is a collection of simple units with parameters encoded on the connections between the units. Conceptually they are at the same level of logic gates; a given neuron can not compute anything interesting, but hopefully by hooking them up in the right way a complex calculation can be carried out. The relation to logic gates is more than just an analogy - small neural networks can compute all the usual logical operations (and, or, xor, not), and this suggests it is a suitable class of models for general purpose computing. In this work, a neural network then is something like a logical circuit whose behavior depends continuously on its parameters.

Among circuits one can distinguish between those that are feed-forward and those with feedback connections. Feed-forward circuits can express computations that terminate in a fixed number steps. Roughly speaking, the depth of the circuit determines how many steps of computation the circuit can represent. Although the dynamics of such a circuit are very simple, as the network will reach a stable point in a finite number of steps. A general model of computation must have some sort of non-trivial control structure such as loops or recursion. Without these, even simple computations can be burdensome to express. The relevant network may require many nodes and/or weights and be of large depth. This has drawbacks for optimization - many parameters means the network requires more storage,
and is more susceptible to over-fitting. Furthermore it requires the user to guess at how many steps of computation would be required to solve a given problem. A more general class of circuits are those that allow feed-back connections, as shown in Figure 1.1. The network on the left represents a network that performs four steps of computation. The network on the right is an attractor network. Each of the four components continually exchanges information with its neighbors until a fixed-point of the computation is reached. This class is more powerful as it includes the feed-forward networks but can also express some computations more compactly. As a general principle, any computation that can be expressed using iteration can be naturally represented using a feedback network, but has to be “unrolled” to be represented as a feed-forward network. However, feedback adds several complications. Feedback means the network can have non-trivial dynamics. This makes the mathematical treatment and optimization are more complicated (59). Also, the dependency on the parameters can be much more complicated.
In either case, the family of models consists of all neural networks with a given connectivity pattern, and the individuals in the family differ in the exact weights on the connections between their nodes. The weights in a neural network are real numbers, hence the optimization problem is that of minimizing a function of (many) real variables and the machine learning problem reduces to a continuous optimization problem.

Neural network optimization problems are approached using gradient based optimization. At each step of the algorithm, the optimization program calculates the derivative of an objective function with respect to parameters of the model, in order to determine what direction to move in. In the ideal scenario, the derivatives can be calculated exactly at each iteration. For instance with a feed-forward neural network the back propagation algorithm can be applied. In networks with feed-back these derivatives can only be approximated. This is usually for the same reason that the long-term behavior of these networks can only be approximated. That is, one finds that the natural the gradient estimation process inherits the dynamic properties of the underlying network. An interesting approach that can be taken with models that involve feedback is to run the gradient estimation and optimization processes simultaneously. The idea is to use an iterative algorithm for computing the gradient, and carry over the state of the estimation procedure after each parameter update, to avoid starting from scratch each time. The structure of this algorithm is shown in Figure 1.2.

In this thesis we make several contributions in the area of gradient estimation and optimization of neural networks with general connectivity. We begin with networks on a continuous state space. These are more tractable from the optimization standpoint since they are directly subject to useful optimization operations such as taking derivatives. We prove a type of convergence of the overall optimization procedure when the underlying network satisfies a contraction property. Then turning to more general stochastic networks, we consider the forward sensitivity method for gradient estimation, and prove its correctness subject to
CHAPTER 1. INTRODUCTION

Figure 1.2: In standard gradient based optimization schemes (a), the search direction $\Delta_n$ at time $n$ is calculated based solely on the parameter $w_{n-1}$. In dynamic gradient estimation schemes (b), the search directions $\Delta_n$ are computed based on the current parameter and the state $y_n$ of an auxiliary system.

certain contraction and differentiability conditions. We then turn our attention to networks on a discrete state space. We show how the concept of measure valued differentiation can be applied to estimate gradients in a general setting that extends several well-known neural network models. We detail these contributions more in the next chapter.

1.2 Outline

The remainder of this thesis is organized as follows. In Chapter 2 we offer a more detailed definition of the problem at hand and survey the relevant literature. In Chapter 3 we introduce the mathematical methods that will be used to achieve our results. Chapters 4 through 6 have our main results. In Chapter 4 we consider the deterministic, continuous space attractor networks. In Chapter 5 we consider the stochastic attractor networks on a
continuous state space. In Chapter 6 we present our results for the stochastic networks on a discrete state space. We finish with concluding remarks in 7.

1.3 Notations

In this section we introduce for reference some notations that used throughout the thesis.

- \( n \) - dimensionality of the state space of a model. In a network based model, this will be the number of nodes in the network.

- \( m \) - dimensionality of the parameter space of a model.

- \( x \) - variable for the state of a neural network or other dynamic system.

- \( w_{i,j} \) - weight from node \( j \) into node \( i \)

- \( b_i \) - bias at node \( i \)

- \( \theta \) - parameter vector. In a neural network model, \( \theta \) is a pair \((w, b)\), consisting of a weight matrix and bias vector.

- \( \frac{\partial g}{\partial y}(y) \) - for a function \( g : \mathbb{R}^n \rightarrow \mathbb{R}^m \), this is the \( m \times n \) matrix with entries \( \left[ \frac{\partial g}{\partial y}(y) \right]_{i,j} = \frac{\partial g_i}{\partial y_j}(y) \).

- \( t \) - iteration number of algorithm.

- \( \theta(1), \theta(2), \ldots, \theta(t), \ldots \) - sequence of parameters generated by optimization algorithm.

- \( \pi_\theta \) - probability measure depending on a parameter \( \theta \).

- \( \delta_x \) - point mass centered at \( x \); the measure \( \delta_x(A) = 1_{x \in A} = \{1 \text{ if } x \in A, 0 \text{ otherwise.} \} \)

- \( P \) - A Markov kernel. \( P(x, A) \) is the probability \( P \) assigns to set \( A \) from the state \( x \).
• When the state space is discrete, we will often abuse notation and write \( P(x, y) \) instead of \( P(x, \{y\}) \).

• \((\nu P)\) - the measure which is the product of \( \nu \) and Markov kernel \( P \): \((\nu P)(A) = \int_X P(x, A)d\nu(x)\) is the probability that the next state is in \( A \) given that the initial state is distributed according to \( \nu \).

• \(\delta_x P\) - the measure \((\delta_x P)(A) = P(x, A)\).

• \(\nu(e)\) - expected value of random variable \( e \) under measure \( \nu \); \( \nu(e) = \int_X e(x)d\nu(x) \).

• \(\mathcal{P}(X)\) - collection of probability measures on the space \( X \).

• \(\Gamma(\nu_1, \nu_2)\) - collections of couplings of the probability measures \( \nu_1 \) and \( \nu_2 \). A coupling of \( \nu_1 \) and \( \nu_2 \) is a measure \( \gamma \) on \( X \times X \) such that \( \gamma(A, X) = \nu_1(A) \) and \( \gamma(X, A) = \nu_2(A) \) for all measurable \( A \subseteq X \).

• \(1_{p(x)}\) - the function of \( x \) which is 1 when the predicate \( p(x) \) is true and 0 otherwise.

• \(e\) - a loss function. For instance \( e(x, y) = 1_{x \neq y} \) or \( e(x, y) = \|x - y\|_2^2 \)

• \(J\) - a function to minimize.

• \(C^1\) - class of functions with continuous derivatives. More generally, \( C^n \) refers to functions with continuous \( n \)th derivative.

• \(\mathbb{R}_{\geq 0}\) - set of non-negative numbers. More generally, \(\mathbb{R}_{\geq k} = \{x \in \mathbb{R} \mid x \geq k\}\).

• \(\gamma^+, \gamma^-\) - for a real number \( \gamma \), these are \( \gamma^+ = \max\{0, \gamma\} \) and \( \gamma^- = -\min\{0, \gamma\} \).

• \(\gamma^\dagger\) - the function \( \gamma^\dagger = 2\gamma - 1 \).
Chapter 2

Background

In this chapter we introduce the main goals of this thesis and summarize our results. In Section 2.1 we set the stage by defining the machine learning problem. Then we formally define the neural network models in Section 2.2. We describe the contributions to the optimization and gradient estimation of these models in Section 2.3.

2.1 Machine Learning Problem

We begin with the formal definition of a machine learning problem. This starts with a data distribution, which is a probability distribution over pairs of inputs and outputs. We let $U_I$ denote a space of inputs and $U_O$ denote the space of outputs. The joint space $U$ is the product $U = U_I \times U_O$. For instance $U_I$ could be the space of images and $U_O$ could be possible classes or, more generally, interpretations of the image. We denote elements of $U_I$ by $v$ and elements of $U_O$ by $l$. A distribution on $\nu$ over $U$ is called a data distribution. A loss function is a function $e : U_O \times U_O \rightarrow \mathbb{R}_{\geq 0}$ that takes two elements of the output space and gives a real number. A deterministic classifier maps elements $U_I$ to elements of $U_O$. For optimization purposes the classifier typically also depends on a parameter $\theta \in \Theta$; we call $\Theta$ the parameter
space. Given these definitions, the test error is defined as

\[
\int_U e(c(v, \theta), l) \, d\nu(v, l).
\] (2.1)

In most cases one cannot compute the derivative of the test error with respect to the parameter \(\theta\) exactly, and instead we must consider approximations. The first approximation involves replacing the data distribution with a finite sample and the second replaces a possibly non-smooth objective with a smooth one. In detail, we replace \(\nu\) with a finite-sample approximation, denoted by \(\nu_N\). This is the measure defined by \(n\) i.i.d samples \((v_1, l_1), \ldots, (v_N, l_N)\) from \(\nu\). Using it we define the empirical error

\[
\int_U e(c(v, \theta), l) \, d\nu_N(v, l) = \frac{1}{N} \sum_{i=1}^{N} e(c(v_i, \theta), l_i).
\]

There is still potentially an issue in that the function \(\theta \mapsto e(c(v, \theta), l)\) may not be differentiable, and hence the empirical error function may fail to be differentiable. This can be approached by smoothing - replacing \(e\) or \(c\) (or both) by smooth surrogates. For example, when a neural network is used for classification, one computes the scores along \(m\) possible classes and assigns the class based on which score is highest. The result \(c\) is a vector with a 1 in the coordinate corresponding to the inferred class and 0's elsewhere. Typically \(e\) is the function \(e(c, l) = 1 - c^T l\), which means the test error is one minus the classification accuracy. However, this results in an empirical error function which is not smooth. A solution is to smooth the output of the network, by using the soft-max function instead of a hard maximum. For ease of notation, we are going to assume any smoothing has already been incorporated into the definitions, and use the name “empirical error” to refer to the resulting problem that has been appropriately smoothed.

Since the problem is now that of minimizing a finite sum, we will restrict our discussion
to the case that \( N = 1 \) for ease of notation. To extend the algorithms to the setting \( N > 1 \), one can calculate the derivatives for all the examples in parallel and sum them together. Hence the problem is now to minimize \( e(c(v_1, \theta), l_1) \) as a function of \( \theta \), and for notational purposes we write instead

\[
J(\theta) = e(c(\theta)).
\]

The classifier may also be probabilistic. In this case we model the output of the classifier not as a point in \( U_O \), but as a measure on \( U_O \) which depends on the input \( u \) and the parameter \( \theta \). This is denoted by \( \pi_{u,\theta} \). Then the test error is

\[
\int_U \int_{U_O} e(l', l) \, d\pi_{v,\theta}(l') \, d\nu(v, l),
\]

and the empirical version is

\[
\frac{1}{N} \sum_{i=1}^{N} \int_{U_O} e(l', l_i) \, d\pi_{v_i,\theta}(l').
\]

For a single example, the aim is to minimize \( \int_{U_O} e(l', l_1) \, d\pi_{v_1,\theta}(l') \) Since the \((v, l)\) is fixed, we can just write

\[
J(\theta) = \int_{U_O} e(l', l) \, d\pi_{\theta}(l').
\]

To summarize, in this thesis we will be interested in optimization of functions of the form (2.2) (or more generally (2.3)) and this function differs from the function of interest due to (1) being defined through a finite sample and (2) the effects of smoothing. The function \( c \) (or more generally \( \pi \)) is defined through an attractor neural network. We are concerned exclusively with gradient based approaches to solving the optimization problem. However, there are approaches to neural network training that don’t involve gradient based optimization. One avenue is to try to program the desired behavior into the network manually (47, 2). Genetic algorithms may also be used (89).


2.2 Models

We now review the three types of attractor networks that are the main focus of this work.

2.2.1 Attractor Networks on a Continuous State Space

The work of (79) generated much interest in gradient-based training of feed-forward networks, and shortly after there appeared works studying all variety of neural networks. The type of networks that we describe in this section, attractor networks, seem to first have been described in (4, 70).

First we consider attractor networks on a continuous state space. This means to classify an input \( u \), we run a neural network with input \( u \) until it reaches a stable point and then use this stable point to determine the class of \( u \). When the architecture is feed-forward the stable point is reached after finitely many steps, while feedback may mean this point is only reached asymptotically.

The state space is \( \mathcal{X} = \mathbb{R}^n \) and there is a set of edges that determines the connectivity between nodes. We let \( x = (x_1, \ldots, x_n) \) denote a state of the neural network. The weights \( w \in \mathbb{R}^{n \times n} \) and biases \( b \in \mathbb{R}^n \) are the parameters of the network. The number \( w_{i,j} \) is the weight into node \( i \) from node \( j \). A value of \( w_{i,j} = 0 \) means there is no connection to node \( i \) from node \( j \). We let \( \Theta = \mathbb{R}^n \times \mathbb{R}^{n \times n} \) represent the joint parameter space.

The following function \( f : \mathcal{X} \times \Theta \times \mathcal{X} \to \mathcal{X} \) gives the next state of the network given the current state, parameters, and input:

\[
    f_i(x, (w, b), v) = \sigma \left( \sum_{j=1}^{n} w_{i,j} x_j + b_i + v_i \right), i = 1, 2, \ldots, n. \tag{2.4}
\]

That is, the state of node \( i \) at time \( t+1 \) is determined by the external input \( v_i \) at that node, and the states of its neighbors at time \( t \). The function \( \sigma \) is referred to as the activation
function. Typical choices are the logistic function \( \sigma(x) = \frac{1}{1+\exp(-x)} \) or the hyperbolic tangent 
\( \sigma(x) = \frac{2}{1+\exp(-2x)} - 1 \). See Figure 2.1. The results we prove for these models hold for any activation functions with bounded first and second derivatives.

By iterating (2.4) starting from an initial point \( x(0) \), one obtains a sequence of network states \( x(1), x(2), \ldots \) where \( x(t+1) = f(x(t), \theta, v) \). Alternatively, we may write \( x(t+1) = f^{t+1}(x(0), \theta, v) \). Even very simple networks of units of the form (2.4) can have complex behaviors such as chaos (63). We rule this out by focusing on a well behaved subclass. A network has a globally attractive fixed-point when there is a state \( x^* \) that is a fixed-point for \( f \), meaning

\[
\lim_{t \to \infty} f^t(x(0), \theta, v) = x^*
\]

and this fixed-point is globally attractive, meaning

\[
\lim_{t \to \infty} f^t(x^*, \theta, v) = x^*
\]
for any initial point $x_0$. We will also use the name 'fixed-point networks' to refer to attractor networks. In general, this fixed-point will depend on the parameters $\theta = (w, b)$ and input $v$, and to make this explicit we will write $x^*(\theta, v)$. The question of whether a network admits a globally attractive fixed-point for a given value of the parameters $(w, b)$ is difficult; for one type of attractor network known as the Hopfield network, various hardness results have been obtained for this question (64). The problem remains computationally hard even for simple connectivity patterns such as cycle graphs (73). There are several known conditions, however. If the weights are symmetric then the network will converge to a point for any initial conditions (4, 46). However, this point may not be independent of the initial condition. Alternatively, a unique attractive point exists when the weights of the network are “small” in some sense. Formally, for any norm $\| \cdot \|$ on $\mathbb{R}^n$, there will be contraction when

$$\sup_{x \in X} \left\| \frac{\partial f}{\partial x} (x) \right\| < 1. \quad (2.5)$$

If we specialize to the form of $f$ in Equation 2.4, we can get a more specific condition. Recall that a norm $\| \cdot \|$ is called absolute when $\| (u_1, \ldots, u_n) \| = \| (|u_1|, \ldots, |u_n|) \|$ (48, Chapter 5).

Also, define $\| \sigma' \|_\infty = \sup_{x \in \mathbb{R}} |\sigma'(x)|$. Then for absolute norms, contraction will occur when

$$\|w\| \leq \frac{1}{\|\sigma'\|_\infty}.$$

Norms that are absolute include the norms $\| \cdot \|_p$ for $p \in \{1, 2, \ldots, \infty\}$. When $\sigma$ is the logistic function, $\|\sigma'\|_\infty = 4$.

### 2.2.2 Stochastic Attractor Networks

The neural networks we described in the previous section were deterministic. To each input they associate one output. There are a number of situations when one may be interested
in stochastic networks. For instance, if one is dealing with a very large network that has to be spread across multiple computers, this can cause random delays or synchronization issues. We can model this with a neural network that has noisy connections, meaning the set of neighbors of each node is random at each time step. Additionally, a noisy network can represent a one-to-many mapping that could be useful, for example, when there are multiple interpretations to a given image. When the network is well behaved, the process will be ergodic and the problem is to optimize the long-term average behavior. To guarantee that the network has a regular long-term behavior independent of its starting point, we will use a condition which is a stochastic analogue of the contraction condition we saw in the previous section.

The general form of our stochastic neural networks are as follows. Let $\xi(1), \xi(2), \ldots$ be an infinite sequence of independent identically distributed (i.i.d) $\Xi$-valued random variables distributed according to a measure $\eta$. Each $\xi(t)$ could be for instance a vector of uniform random variables in $[0,1]$. Then the state at time $t+1$, denoted $x(t+1)$, is obtained from the state $x(t)$, the parameters $\theta$ and the random noise $\xi(t+1)$ as

$$x(t+1) = f(x(t), \theta, \xi(t+1)). \quad (2.6)$$

Note we have suppressed the notation for the input $v$, as it is fixed. For instance, in the random connection model we described above the $\xi(t)$ could be a matrix of Bernoulli random variables, that determine which connections are activated. We interpret $\xi_{i,j} = 1$ to mean the connection from $j$ to $i$ is activated, and $\xi_{i,j} = 0$ means the connection is disabled. In this case, $f$ takes the form

$$f_i(x, (w, b), v, \xi) = \sigma \left( \sum_{i=1}^{n} \xi_{i,j} w_{i,j} x_j + b_i + v_i \right).$$
Since the input \( v \) is fixed, in what follows we omit it from the definition of \( f \) and will speak of \( f(x, \theta, \xi) \). We are interested in the problem of optimizing the long-term average behavior of recursions such as (2.6). This reduces to the deterministic attractor problem in case there is no noise (\( f \) does not depend on \( \xi \)). To define the objective function, we need to put some restrictions on our process so that it does in fact have a regular long-term behavior. The condition will be that the Markov chain defined by (2.6) possesses a unique variant measure \( \pi_{\theta} \), and that given any distribution on the initial state \( x(0) \), the distributions of the states \( x(1), x(2), \ldots \) converge to \( \pi_{\theta} \). We can formally state this as follows. Let \( P_{\theta} \) be the Markov kernel associated to the stochastic system (2.6). Given that one is in state \( x \), the quantity \( P_{\theta}(x, A) \) is the probability that the random next state \( f(x, \theta, \xi) \) is in the set \( A \). Formally, letting \( \eta \) be the noise measure, \( P_{\theta} \) is

\[
P_{\theta}(x, A) = \eta(\{\xi \mid f(x, \theta, \xi) \in A\}).
\]

For a function \( e : X \to \mathbb{R} \) we denote by \( P_{\theta}e \) the function

\[
(P_{\theta}e)(x) = \int_{X} e(y) \, d(\delta_{x}P_{\theta})(y) = \int_{\Xi} e(f(x, \theta, \xi)) \, d\eta(\xi)
\]

This number \( (P_{\theta}e)(x) \) is the expectation of the random variable \( e \) at the next state of the network, given that we start in state \( x \). For a measure \( \nu \) on \( X \) we denote by \( \nu P \) the measure

\[
(\nu P_{\theta})(A) = \int_{X} P_{\theta}(x, A) \, d\nu(x).
\]

Then the process (2.6) possesses a stationary measure \( \pi_{\theta} \) when

\[
\pi_{\theta} = \pi_{\theta}P_{\theta}.
\]  

Instead of using the terminology 'globally attractive', we will say that the network is ergodic.
This means that (2.7) holds, and for any initial measure \( \nu \),

\[
\nu P^t_\theta \to \pi_\theta \text{ as } t \to \infty.
\] (2.8)

The type of convergence in (2.8) is weak convergence. A sequence of measures \( \nu_t \) converges weakly to a measure \( \nu \) if \( \nu_t(\varphi) \to \nu(\varphi) \) as \( t \to \infty \) for all bounded Lipschitz functions \( \varphi : X \to \mathbb{R} \).

### 2.2.3 Discrete Attractor Networks

The third class of models we consider are probabilistic and operate on a finite state space. Each unit takes on either the values 0 or 1. They were first defined in (56), and are sometimes referred to as the Little model. They can be interpreted as threshold networks, where the thresholds are randomly chosen at each time step.

We define the attractor network on a discrete space as follows. Let the network have \( n \) nodes, and let \( \xi(1), \xi(2) \ldots \) be a sequence of noise vectors in \( \mathbb{R}^n \), with the entire collection \( \{\xi_i(t); i = 1, \ldots, n, t = 1, 2, \ldots\} \) independent and distributed according to the logistic distribution. That is, the cumulative distribution function (CDF) of \( \xi_i(t) \) is

\[
\eta(\xi_i(t) < x) = \frac{1}{1 + \exp(-x)}.
\]

Define \( f : \{0, 1\}^n \times \Theta \times \Xi \to \{0, 1\}^n \) as

\[
f_i(x, (w, b), \xi) = \begin{cases} 
1 & \text{if } \sum_{j=1}^{n} w_{i,j} x_j + b_i > \xi_i, \\
0 & \text{otherwise}.
\end{cases}
\] (2.9)

This function \( f \) and the noise \( \xi(1), \xi(2) \ldots \) determines the operation of the random threshold
CHAPTER 2. BACKGROUND

network; from the initial point $x(0)$ one follows the recursion

$$x(t + 1) = f(x(t), \theta, \xi(t + 1))$$

(2.10)

to generate the next state. We let $P_\theta$ be the Markov kernel corresponding to this recursion. Formally,

$$P_\theta(x, A) = \eta(\{\xi \mid f(x, \theta, \xi) \in A\})$$

(2.11)

for the function $f$ defined in (2.9). We show that $P$ is ergodic and find the convergence rate of the Markov chain in Section 6.1 below. We denote by $\pi_\theta$ be the stationary measure of the recursion (2.10).

2.3 Contributions

2.3.1 Attractor Networks on a Continuous State Space

First we consider the attractor networks that are deterministic and operate on a continuous state space. Some early works that considered these networks were (7, 70, 71, 72, 4). The dynamical nature means that one cannot compute the derivatives of interest easily, as one can in feed-forward networks. One can construct an optimization algorithm for attractor networks using a dynamic variant of back-propagation based algorithms, resulting in an algorithm with the structure of Figure 1.2. We propose to address the problems of how to tune the algorithm to guarantee a function decrease, and to obtain a long-term guarantee on the optimization algorithm. Some other works (See Figure 2.2) also considered these models, but the results they obtained for gradient estimation and optimization were mostly heuristic, or were asymptotic. In this setting we seek results that concern finite-step sizes and we aim to express the results in terms of available model information.
Adjoint and forward sensitivity analysis

We derive the optimization algorithm that we are interested in. The exposition is somewhat similar to that in the works (8, 65). An alternative derivation uses Lagrange multipliers (28, 54).

Let $X = \mathbb{R}^n$ and let $e : X \to \mathbb{R}$ be a loss function. The problem is to minimize the loss at the fixed-point:

$$\min_{\theta} J(\theta) \quad (2.12)$$
where
\[ J(\theta) = e(x^*(\theta)). \]

The differentiability of \( J \) follows by the implicit function theorem and the chain rule. Starting from the equation
\[ x^*(\theta) = f(x^*(\theta), \theta) \tag{2.13} \]
and using the contractivity and differentiability properties of \( f \), one can conclude that \( x^*(\theta) \) is differentiable. Then as long as \( e \) is differentiable we can apply the chain rule to get that \( J \) is differentiable. Using this and the formulas provided by the implicit function theorem, one obtains that
\[ \frac{\partial J}{\partial \theta}(\theta) = A(\theta)B(\theta)C(\theta). \tag{2.14} \]

where
\[ A(\theta) = \frac{\partial e}{\partial x}(x^*(\theta)), \quad B(\theta) = (I - \frac{\partial f}{\partial x}(x^*(\theta), \theta))^{-1}, \]
\[ C(\theta) = \frac{\partial f}{\partial \theta}(x^*(\theta), \theta). \]

(The contraction condition Inequality 2.5 guarantees the inverse used to define \( B \) exists.) From this formula, we can see two challenges to computing, or even approximating, the derivative. The first is that these terms involve \( x^*(\theta) \), which can only be approximated by iteration. Secondly, they involve the solution of linear systems (one can choose between either \( A(\theta)B(\theta) \) or \( B(\theta)C(\theta) \)). Below we describe two iterative algorithms that can address these problems.

We can calculate the term \( A(\theta)B(\theta)C(\theta) \) by computing \( A(\theta)B(\theta) \) and then post multiplying by \( C(\theta) \), or we can compute \( B(\theta)C(\theta) \) and premultiply by \( A(\theta) \). In each case, one can use iterative solver to jointly solve the fixed-point equation (2.13) and deal with the matrix inverse. These approaches are referred to as adjoint sensitivity analysis and forward sensitivity analysis respectively. The derivations are somewhat symmetric; we focus on the adjoint
method in what follows. One early work to mention this method of sensitivity analysis, for
the non-asymptotic, finite time, case, was (13).

Let the space $\mathcal{Z}$ be $\mathbb{R}^n \times \mathbb{R}^n$, and define the map $T^{Adj} : \mathcal{Z} \times \Theta \rightarrow \mathcal{Z}$ as

$$T^{Adj}((x, y), \theta) = \left(f(x, \theta), y \frac{\partial f}{\partial x}(x, \theta) + \frac{\partial e}{\partial x}(x)\right). \tag{2.15}$$

Assuming that $T^{Adj}$ possesses a fixed-point $z^*(\theta) = (x^*(\theta), y^*(\theta))$, it is easy to verify that $x^*(\theta)$ is a fixed-point for $f$ and

$$y^*(\theta) = A(\theta)B(\theta).$$

Therefore, if we could obtain the fixed-point $z^*(\theta)$ we could easily compute the gradient (2.14), since

$$\frac{\partial J}{\partial \theta}(\theta) = G(z^*(\theta), \theta)$$

where

$$G((x, y), \theta) = y \frac{\partial f}{\partial \theta}(x, \theta). \tag{2.16}$$

This map $T^{Adj}$ is essentially doing the same type of gradient estimation as in the back-
propagation procedure for neural networks. In the case that the network has no cycles, then
the gradient estimation converges in a finite number of steps. For example, if $f$ describes a
feed-forward network with $k$ layers then a fixed-point of $T^{Adj}$ will be reached by iterating for
$k$ steps. If there are cycles, then under certain contraction assumptions (such as Inequality
2.5), the operator $T^{Adj}$ also satisfies this contraction property. This can be verified using the
condition on the derivative in Inequality 2.5, for an appropriate choice of norm on the space $\mathcal{Z}$. This is discussed in (71) and other works concerning attractor networks. See Proposition
4.2.4 for a formal statement.

If $T^{Adj}$ defines a globally attractive process on $\mathcal{Z}$, this gives an iterative method to
estimate the gradient: Iterate $T^{Adj}$ enough times starting from an arbitrary point $(x_0, y_0)$.
to obtain point \((x_M, y_M)\) close to \((x^*(\theta), y^*(\theta))\), and then form the estimate \(G((x_M, y_M), \theta)\).

By continuity properties of \(f\), it should be that

\[
G((x_M, y_M), \theta) \approx G(z^*(\theta), \theta)
\]

where the quantity on the right which is the true gradient. The pseudocode for this procedure (termed adjoint sensitivity analysis) procedure in Algorithm 1.

**Algorithm 1: Deterministic Adjoint sensitivity analysis**

Define \(T^{Adj}: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n\) as

\[
T^{Adj}((x, y), \theta) = \left( f(x, \theta), y \frac{\partial f}{\partial x}(x, \theta) + \frac{\partial e}{\partial x}(x) \right)
\]

for \(t = 0, 1, \ldots, M - 1\) do

\( (x(t + 1), y(t + 1)) = T^{Adj}((x(t), y(t)), \theta) \)

end

- Set \(\Delta^{Adj} = y(M) \frac{\partial f}{\partial \theta}(x(M), \theta)\)

**return** \(\Delta^{Adj}\)

The output of Algorithm 1 is the gradient estimate \(\Delta^{Adj}\). It has the property that

\[
\Delta^{Adj} \to \left. \frac{\partial f}{\partial \theta} \right|_{\theta} \text{ as } M \to \infty,
\]

as we have explained above. This gradient estimation procedure is turned into an optimization procedure by interleaving the estimation and optimization processes as shown in
Algorithm 2.

Algorithm 2: Optimization using Adjoint sensitivity analysis

\[
\begin{align*}
\text{for } t = 0, 1, \ldots, \text{ do} \\
(x(t + 1), y(t + 1)) &= T^{Adj}((x(t), y(t), \theta(t))) \\
\theta(t + 1) &= \theta(t) - \epsilon y(t + 1) \frac{\partial f}{\partial \theta}(x(t + 1), \theta(t)) \\
\end{align*}
\]

end

The main result about this algorithm, Theorem 4.2.7, says that given boundedness of the derivatives of \( f \) and \( e \), and a uniform contraction property on \( f \), then one can choose \( \epsilon \) so that the algorithm generates parameter updates that are accurate enough so that the function decreases at each step. It can be roughly stated as follows:

**Theorem 2.3.1.** Assume \( f \) is uniformly contracting, with bounded derivatives up to order 2, and that \( e \) is bounded from below, with bounded derivatives up to order 2. Then there are \( \epsilon \) and \( c \) such that if the initial point \((x(0), y(0), \theta(0))\) satisfies

\[
\| (x(0), y(0)) - (x^*(0), y^*(0)) \| \leq c \left\| \frac{\partial f}{\partial w}(x(0), \theta(0))^T y(0) \right\|
\]

then \((e \circ x^*)(\theta(t))\) converges and \( \lim_{t \to \infty} \frac{\partial}{\partial \theta} (e \circ x^*)(\theta(t)) = 0 \).

See Section 4.2 for full details.

**Related work**

Some early works considered conditions for stability in attractor networks, but did not consider their gradient based optimization (33). Investigation of optimization in attractor networks started in (4, 70, 7). The back-propagation procedure, originally formulated as a protocol for optimizing the parameters of a feed forward network, also “makes sense” when cycles are present, but the analysis of the procedure is more complicated for the reasons described above. In particular, the work (71) formulated the “recurrent back-propagation”
algorithm in essentially the form of equations (4.21a, 4.21b), as three simultaneous processes: forward propagation (the $x_n$), backward propagation (the $y_n$), and the parameter adaptation process (the $w_n$). The need to balance the relative rates of these activities was not investigated rigorously except in a few limited instances. Some local convergence results were obtained in (74, 94). Similar considerations apply in a variety of dynamical settings for neural networks, including stochastic models (101, 93, 49).

These algorithms, which couple gradient estimation with optimization are well-known also in design optimization, where it is used in aerospace applications (31, 28, 38). Often a PDE-constrained problem is transformed into a discrete-time fixed-point optimization problem by spatial and temporal discretization; the map $f$ above then corresponds to an iteration of the numerical integrator, and the parameter of interest $w$ determines the fixed-point of the integrator. So called “one-shot” methods alternate between iterating the integrator and performing a parameter update or “design step”. The introduction of this method is often attributed to (92). It has been applied in a variety of situations, including aerodynamics problems (39, 34). Several recent works have analyzed the convergence of one shot methods (37, 38). One approach is to consider Algorithm 2 and to identify preconditioners, as opposed to mere step-sizes as we do, that guarantee contractivity of the overall procedure (32). In this context it is called the one-shot approach (32, 38). However, the results in these works also assume that the algorithm starts near a global minimum.

Perhaps the main difference between this work and previous analyses of this type of optimization procedure is that we do not assume any convexity, local or otherwise, of the overall optimization problem. The main “stability” type of assumption is that of the underlying dynamic system $f$. The goal is to guarantee a convergence property about the procedure that is independent of the initial parameter $w_0$ and without assuming the the problem is convex. Of course, the conclusion is correspondingly weaker; our result guarantees descent of the true objective $e \circ x^*$ at each step, but the long-term guarantee is typical of gradient
descent without convexity; \((e \circ x^*)(w_n)\) converges, \(\frac{\partial}{\partial w}(e \circ x^*)(w_n) \to 0\), but not convergence of \(w_n\) itself. With this in mind, in a recent work (22) the author studied a continuous-time version of the optimization procedure defined by Equations (4.21a, 4.21b), where the focus is on how to choose a time-scale or rate parameter that guarantees convergence. This work can be seen as a discrete time counterpart of those results.

There are a number of relevant works that attempt to analyze adjoint-based optimization procedures. The work of (75) analyzed a continuous time-version of the algorithm and obtained local convergence results using methods for singularly perturbed systems (81). The results were local in the sense that they assume optimization begins near an attracting local minimum. They were also not constructive, meaning they didn’t quantify the requirements on the algorithm, such as time-scale parameters. In (74) the same authors considered the algorithm in the discrete time setting, but did not pursue a convergence analysis.

A related set of works considers Hebbian learning in neural networks. This a type of learning process for a neural network that is not explicitly gradient based, instead being motivated by the neuroscientific theory of Hebbian learning. The similarity is that both involve simultaneous process of adaptation and underlying network dynamics, and the need to consider the relative rate of the two activities. Convergence of a continuous time Hebbian learning process was discussed in (18). The work of (60) also considered convergence of this type of algorithm, again using the singular perturbation methods of (81).

There are a number of applications that could be of interest. One idea is inspired by reservoir networks (58). We can consider a hierarchical network, the first component being a random attractor network, and the second being a regular feed-forward layer. We would keep the random weights fixed, but only optimize the output weights. The above results require a uniform contraction property (that is, the rate of convergence of the system should stay constant as optimization proceeds). This is accommodated by keeping the feed-back weights fixed.
The attractor networks can be generalized by making them stochastic. As shown in Figure 2.2, some other authors considered restricted cases of such models that have acyclic connectivity. We approach this in a way that generalizes the deterministic approach described in the previous section.

The optimization problem we are interested in is, given a cost function \( e : X \to \mathbb{R} \),

\[
\min_{\theta \in \Theta} J(\theta)
\]

where

\[
J(\theta) = \int_X e(x) \, d\pi_\theta(x).
\]

Our concern is with gradient estimation, meaning how to calculate the derivative of \( J \). The method of forward sensitivity analysis leads to a gradient estimation procedure as shown in Algorithm 3.

**Algorithm 3: Stochastic Forward sensitivity analysis**

- Define \( T^{Fwd} : \mathbb{R}^n \times \mathbb{R}^{n \times m} \times \Theta \times \Xi \to \mathbb{R}^n \times \mathbb{R}^{n \times m} \) as

\[
T^{Fwd}((x,u), \theta, \xi) = \left( f(x, \theta, \xi), \frac{\partial f}{\partial x}(x, \theta, \xi) u + \frac{\partial f}{\partial \theta}(x, \theta, \xi) \right)
\]

for \( t = 0, 1, \ldots, M - 1 \) do

\[
(x(n+1), u(n+1)) = T^{Fwd}((x(t), u(t)), \theta, \xi(t+1))
\]

end

- Set \( \Delta^{Fwd} = \frac{\partial e}{\partial x}(x_M) u_M \)

return \( \Delta^{Fwd} \)

We would like to know if the procedure does in fact help to calculate derivatives. We aim to show two things about Algorithm (3):

i. The process \( (x(t+1), u(t+1)) = T^{Fwd}(x(t), u(t), \theta, \xi(t+1)) \) is ergodic,
ii. $E[\Delta_{Fwd}] \rightarrow \frac{\partial J}{\partial \theta}(\theta)$.

An optimization algorithm which uses Algorithm 3 to compute gradients could be analyzed using results from stochastic approximation theory. This is because the gradient estimation procedure has a Markovian noise structure, also known as endogenous noise. Theorems based on weak convergence could be used in this case (52). Our concern however is only on the gradient estimation aspect of this algorithm, including its correctness and convergence rate.

A simple version of our main result for stochastic networks can be stated as follows (See Theorem 5.4.5 for a full statement.)

**Theorem 2.3.2.** If the function $f$ and the probability space $(\Xi, \Sigma, \eta)$ are such that

i. $\int_\Xi \|f(x, \xi, \theta)\|^2 d\eta(\xi) < \infty$ for all $(x, \theta) \in X \times \Theta$,

ii. $(x, \theta) \mapsto f(x, \xi, \theta)$ is a $C^2$ function for each $\xi \in \Xi$,

iii. For $0 < i + j \leq 2$, the functions $L_{X^i, \Theta^j}(x, \theta) = \int_\Xi \|\frac{\partial^{i+j}f}{\partial x^i \partial \theta^j}(x, \xi, \theta)\|^2 d\eta(\xi)$ are continuous and bounded on $X \times \Theta$, and in particular, $\sup_{(x,\theta)} L_X(x,\theta) < 1$,

Then the forward sensitivity process (5.2a, 5.2b) converges weakly to a stationary measure $\gamma_\theta$, and equation (5.1) holds for those $e : X \rightarrow \mathbb{R}$ that are $C^2$ with $\|\frac{\partial e}{\partial x}\|_\infty + \|\frac{\partial^2 e}{\partial x^2}\|_\infty < \infty$.

Note that the result only concerns gradient estimation, while the Theorem 2.3.1 for deterministic systems concerned both gradient estimation and optimization. On the other hand, Theorem 5.4.5 allows much more general contraction conditions.

Although we are concerned with the discrete time case for these models, there have been studies of gradient processes for continuous time stochastic neural networks (61).
2.3.3 Discrete Attractor Networks

We then turn our attention to networks that operate on a discrete state space. In this work, by “discrete” we mean finite. Several stochastic neural networks on discrete state spaces have been studied, and their gradient estimation procedures are based on having closed form solutions for the resulting probability distributions. The works (45, 66, 5) depend on constraints on network connectivity - for instance symmetry, or prohibiting cycles. See Figure 2.2. The earliest neural network models to be studied from the computational view were the deterministic threshold networks (59, 76). In this model, each unit senses the states of its neighbors, takes a weighted sum of the values, and applies a threshold to determine its next state (either on or off). For single layer versions of these networks, where the units are partitioned into input and output groups, with connections only from input to output nodes, the corresponding optimization problem can be solved by the perceptron algorithm. Any iterative algorithm for optimizing threshold networks has to address the credit assignment problem. This means that during optimization, the algorithm must identify which internal components of the network are not working correctly, and adjust those units to improve the output. The difficulty in solving the credit assignment problem for threshold networks with multiple layers prevents simple deterministic threshold models from being used in complex problems like image recognition. There have been a number of well-known approaches to the problem. For instance, one can abandon the threshold units, and work with units that have a smooth, graded, response such as the sigmoid neural networks described above. In this case methods of calculus are available to determine unit sensitivities. These new networks are still deterministic but now operate on a continuous state space.

Another approach is to keep the space discrete but make the network probabilistic, and use the smoothing effects of the noise to obtain a model one can apply methods of calculus to. One can interpret the Sigmoid Belief Networks in this way. These networks were introduced
in (62) and so named because they combine features of sigmoid neural networks and Bayesian networks. In these networks, when a unit receives a large positive input it is very likely to turn on, while a large negative input means the unit is likely to remain off. In fact, these networks can be interpreted as threshold networks with random thresholds. The use of the sigmoid function, which is the cumulative distribution function (CDF) of the logistic distribution, leads to an interpretation of a network with thresholds drawn from the logistic distribution. In (62), the author derived formulas for the gradient in these networks, and showed how Markov chain Monte Carlo (MCMC) techniques can be used to implement gradient estimators. The networks studied in (62) had a feed-forward architecture, but one could also define variants that allow cycles among the connections. In this way one is lead to the random threshold networks. In this case, one would be interested in the long-term average behavior of the network. Such a generalization would resemble the random threshold networks that are our focus. It would be interesting to obtain a gradient estimator for these new networks.

Another motivation to study general random threshold networks comes from the Boltzmann machine (45). This is a network of stochastic units that are connected symmetrically. This means there is feed-back in the network, and the problem in these networks is to optimize the long-term behavior. The symmetry in the network, and the use of the sigmoid function to calculate the probabilities, leads to a nice closed form solution for the stationary measure in this model. Based on formulas for the stationary distribution, expressions for the gradient of long-term costs can be obtained, leading to MCMC based gradient estimators. If one changes the model, by for instance using non-symmetric connections, or changing the type of nonlinearity, these formulas are no longer available. Instead, one winds up with a model like the random threshold networks. This provides another motivation for studying gradient estimation in the Little model.

Let $\pi_\theta$ be the stationary measure corresponding to the Little model. This will be a
measure on $X = \{0, 1\}^n$. Given a cost function $e : X \to \mathbb{R}$, the optimization problem is

$$\min_{\theta \in \Theta} J(\theta) \quad (2.17)$$

where $J(\theta) = \int_X e(x) \, d\pi_\theta(x)$, and the measure $\pi_\theta$ is defined as the solution to $\pi_\theta P_\theta = \pi_\theta$, for the Markov kernel $P_\theta$ defined in (2.11).

In the case of networks where cycles are allowed, the work (6) considered gradient estimation in the finite horizon setting. Our interest is in the long-term average cost in networks that have general connectivity, where only knowledge of the transition probabilities is available. Methods such as forward sensitivity analysis cannot be used in this case, as they rely on the differential structure of the underlying state space. Instead, we propose an algorithm that computes descent directions based on simultaneous perturbation analysis and measure valued differentiation (MVD). We describe this more fully in Chapter 6.

One of the main motivations for this work is optimization in the Boltzmann machine (45, 3, 84). The Boltzmann machine follows an update rule similar to 2.9, except that the weights are constrained to be symmetric ($w_{i,j} = w_{j,i}$) and the nodes are updated one at a time, instead of all at once. This model was an important ingredient in many machine learning systems. In an influential work, Boltzmann machines were used as part of a pretraining strategy for multilayer neural networks (44, 43). Later, the deep Boltzmann machine was introduced, which is a Boltzmann machine with the structure of many layers (82). The Deep Boltzmann Machine (DBM) enabled very interesting applications such as multimodal processing (88). In that work, a deep Boltzmann machine model is trained to relate images and their text captions. After training, a caption can be generated for an image, and to a certain extent, vice versa. Despite these impressive feats, the optimization procedure for these models seems to be not well understood. In general, optimization algorithms for Boltzmann machines are of the standard form in Figure 1.2. Some researchers in the field made the connection
with stochastic approximation (93, 102). The most relevant theoretical works try to prove
convergence of Boltzmann machine optimization using methods of stochastic approximation
(101, 100). This thesis was partly motivated by a desire to understand the issues involved
in analyzing this type of algorithm. Our goal was to have results with explicit formulas for
algorithm settings such as step-sizes, while avoiding statements that were only asymptotic in
nature, or that relied on convexity. To do this we had to make other simplifying assumptions,
such as bounded derivatives.
Chapter 3

Methods

Our focus is on neural networks with contraction properties. This includes feed-forward networks, but also allows certain networks with feed-back. In the case of a deterministic system, contraction means the network converges to a fixed-point which is independent of the initial state of the network. For a stochastic neural networks, it means the system is ergodic and has a unique invariant measure. We will review several contraction criteria. First we discuss a continuous time notion in Section 3.1, and then introduce some discrete time criteria in Section 3.2. To obtain generalized conditions, one can vary the metric used to define the contraction property. To do this we consider a class of metrics based on Finsler structures which we introduce in Section 3.4. For discussing contraction in the stochastic setting we will use the Wasserstein distances, introduced in Section 3.5.

3.1 Contraction Analysis

To define contraction in the continuous time setting we first recall the notion of a matrix measure. Fix a vector norm $\| \cdot \|$ on $\mathbb{R}^n$. The matrix measure induced by this norm is the
real-valued function $\mu$ defined on $n \times n$ matrices such that

$$
\mu(A) = \lim_{h \to 0^+} \frac{\|I + hA\| - 1}{h}
$$

(3.1)

where the norm in the numerator is the induced matrix norm, and this limit is taken as $h \to 0$ from the right. The matrix measure, also known as the logarithmic norm, is well defined for all matrices and any vector norm. A proof of this fact and other properties of matrix measures may be found in (97) and (16).

We now consider a continuous time dynamical system described by an ordinary differential equation (ODE):

$$
\frac{dz}{dt}(t) = u(z(t), w(t))
$$

(3.2)

Here, $z \in \mathbb{R}^n$, $w \in \mathbb{R}^m$, and $u : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. For $\beta > 0$, the system (3.2) is said to be contracting with rate $\beta$ when

$$
\mu \left( \frac{\partial u}{\partial z}(z, w(t)) \right) \leq -\beta
$$

(3.3)

for all $z$ and $t$. This condition guarantees that trajectories started from different locations converge towards each other (see (57), (86)). Precisely, letting $z_1(t), z_2(t)$ be any two solutions of the system $\dot{z} = u(z, w(t))$ corresponding to different initial conditions, one has

$$
\|z_1(t) - z_2(t)\| \leq \|z_1(0) - z_2(0)\| e^{-\beta t}
$$

(3.4)

in the norm in which the system is contracting. That is, in the norm of definition (3.1). If $w$ does not depend on $t$ in (3.3), the contraction property guarantees existence of and convergence to a unique equilibrium point.

As an example, consider the $\infty$-norm, defined as $\|x\|_\infty = \max_i |x_i|$. In this case the
matrix norm is the maximum-absolute-row-sum: For an $n \times n$ matrix $A$,

$$\|A\|_\infty = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{n} |A_{i,j}| \right\} \tag{3.5}$$

and the matrix measure is given by

$$\mu(A)_\infty = \max_{1 \leq i \leq n} \left\{ A_{i,i} + \sum_{j \neq i} |A_{i,j}| \right\}.$$

Alternatively, consider the 1-norm, defined as $\|x\|_1 = \sum_i |x_i|$. The matrix norm and measure are given by

$$\|A\|_1 = \max_{1 \leq j \leq n} \left\{ \sum_{i=1}^{n} |A_{i,j}| \right\}$$

and

$$\mu(A)_1 = \max_{1 \leq j \leq n} \left\{ A_{j,j} + \sum_{i \neq j} |A_{i,j}| \right\}.$$

See (16) and references therein for further details.

### 3.1.1 Hierarchies of Contractions

An important feature of contraction is that the property is preserved under various types of system combinations. The following result, which is from (86, Theorem 3), regards hierarchies of contracting systems:

**Theorem 3.1.1.** (86, Theorem 3.) Consider a hierarchy of contracting systems of the form

$$\frac{dx}{dt}(t) = f(x(t))$$

$$\frac{dy}{dt}(t) = g(y(t), x(t))$$
where $x$ is contracting with rate $\beta_x$ in the norm $\| \cdot \|_X$, $y$ is contracting with rate $\beta_y$ in the norm $\| \cdot \|_Y$, and $\sup_{x,y} \| \frac{\partial g}{\partial x} (x, y) \|_{X,Y} \leq k$ for some $k$. Then for any positive numbers $p_1, p_2$ such that $\beta_x - \frac{p_2}{p_1} k > 0$, the joint system $(x, y)$ is contracting with rate $\beta$ in the norm $\|(x, y)\| = p_1 \|x\|_X + p_2 \|y\|_Y$ where $\beta = \min\{\beta_x - \frac{p_2}{p_1} k, \beta_y\}$.

The above result means that the joint system consisting of variables $x$ and $y$ will converge to a unique point $(x^*, y^*)$, as long as each system on its own satisfies a contraction property.

### 3.2 Discrete Time Contractions

For the discrete time case we use the notion of contraction mappings. Let $f : X \times W \to X$ be a function on the state space $X$ and depending on a parameter in $W$.

Assume $X$ has the structure of a complete metric space. The function $f$ is a contraction mapping (uniformly in $W$) if there is a constant $0 \leq \beta < 1$ such that

$$
\sup_{x_1 \neq x_2} \frac{d_X(f(x_1, w), f(x_2, w))}{d_X(x_1, x_2)} \leq \beta
$$

for all $w \in W$.

As a consequence of contraction mapping theorem (78, Theorem 9.23), if $f$ is a contraction then for any starting point $x_0$ the iterates $f^n(x_0, w)$ tend to a point $x^*(w)$ that depends only on $w$; this $x^*(w)$ is the unique solution to $x^*(w) = f(x^*(w), w)$.

If $X$ is Euclidean space and $f$ is differentiable in $x$, the contraction property is equivalent to

$$
\sup_{x \in X} \| \frac{\partial f}{\partial x} (x, w) \| \leq \beta.
$$

for all $w \in W$. 
CHAPTER 3. METHODS

3.2.1 Hierarchies of Contractions

This section presents a sufficient condition for the interconnection of two discrete time contractions to be a contraction of contractions. It is analogous to Theorem 3.1.1. We use $d_X, d_Y$ to refer to metrics on the sets $X, Y$ and $Z$ respectively. The proof of this proposition is modeled after the proof of the continuous time result Theorem 3.1.1.

Proposition 3.2.1. Let $X, Y$ be complete metric spaces. Let $f : X \rightarrow X$ be a $\beta_f$-contraction. Let $g : X \times Y \rightarrow Y$ have the property $d_Y(g(x_1, y_1), g(x_2, y_2)) \leq \beta_g d_Y(y_1, y_2) + (L_x g) d_X(x_1, x_2)$ for any two points $(x_1, y_1)$ and $(x_2, y_2)$ of $X \times Y$, where $0 \leq \beta_g < 1$, and $L_x g \geq 0$. Let $p_1, p_2$ be positive numbers such that $\beta_f + \frac{p_2}{p_1} L_x g < 1$. Then the function $h(x, y) = (f(x), g(x, y))$ is a $\beta_h$-contraction in the metric $d_Z((x_1, y_1), (x_2, y_2)) = p_1 d_X(x_1, x_2) + p_2 d_Y(y_1, y_2)$ on $X \times Y$, where $\beta_h = \max \{\beta_f + \frac{p_2}{p_1} L_x g, \beta_g\}$.

Proof. Let $(x_1, y_1)$ and $(x_2, y_2)$ be any two points of $X \times Y$. By definition of $h$ and $d$ we have

$$d_Z(h(x_1, y_1), h(x_2, y_2)) = p_1 d_X(f(x_1), f(x_2)) + p_2 d_Y(g(x_1, y_1), g(x_2, y_2))$$

$$\leq \left(\beta_f + \frac{p_2}{p_1} L_x g\right) p_1 d_X(x_1, x_2) + \beta_g p_2 d_Y(y_1, y_2)$$

$$\leq \max \left\{\beta_f + \frac{p_2}{p_1} L_x g, \beta_g\right\} d_Z((x_1, y_1), (x_2, y_2))$$

The first step simple uses the definition of $d_Z$ and the second use the assumptions on $f$ and $g$. \qed

The relevance of this result will be evident when we apply it to analyze the gradient estimation procedure of Algorithm 1.
3.3 Analysis of Gradient Descent

When we analyze optimization procedures below we will typically invoke a result about approximate gradient descent methods to reach our conclusions. We present those results here.

3.3.1 Continuous Time

The following result concerns convergence of a perturbed continuous time gradient system.

**Theorem 3.3.1.** Let $J(z) : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function such that $\frac{\partial J}{\partial z}$ is Lipschitz continuous and $J$ is bounded from below. Consider the perturbed gradient system

$$\frac{dz}{dt}(t) = -\frac{\partial J}{\partial z}(z(t)) + u(t)$$

where for all $t > 0$,

$$\|u(t)\|_2 \leq \alpha \|\frac{\partial J}{\partial z}(z(t))\|_2$$

for $\alpha < 1$. Then $J(z(t))$ converges and $\frac{\partial J}{\partial z}(z(t)) \to 0$.

**Proof.** The norm used here is the 2-norm. Let $L$ be the Lipschitz constant of $\frac{\partial J}{\partial z}(z)$, so that $\|\frac{\partial J}{\partial z}(z_1) - \frac{\partial J}{\partial z}(z_2)\| \leq L\|z_1 - z_2\|$ for all $z_1, z_2$. It can be easily seen that if $\alpha < 1$ then $J(z(t))$ is decreasing. Under the additional assumption that $J$ is bounded from below we conclude that $J(z(t))$ converges to some value $J^*$. We show that $\frac{\partial J}{\partial z}(z(t)) \to 0$. Let $\phi(z, t)$ be the map which takes initial conditions $z(0) = z$ to the state of the gradient system at time $t$. Using the estimate

$$\|\phi(z, t) - \phi(z, t + h)\| \leq \frac{\|\frac{\partial J}{\partial z}(z(t))\|}{L} (e^{(1+\alpha)Lh} - 1)$$
and fixing an initial condition, one may obtain from Taylor’s formula that

\[ J(z(t + h)) \leq J(z(t)) + \| \frac{\partial J}{\partial z}(z(t)) \|^2 \left( -2h + \frac{1}{L}(e^{(1+\alpha)Lh} - 1) + \frac{h^2}{2L}(e^{(1+\alpha)Lh} - 1)^2 \right). \]

Note that this can be written as

\[ J(z(t + h)) \leq J(z(t)) + \| \frac{\partial J}{\partial z}(z(t)) \|^2 r(h) \]

where \( r(h) = -2h + \frac{1}{L}(e^{(1+\alpha)Lh} - 1) + h^2 k(L, \alpha, h) \) for some function \( k \). Since \( r'(0) = -1 + \alpha \) it follows that if \( \alpha < 1 \) then there exists some \( h, \beta \) both positive and depending only on \( L, \alpha, h \) so that for all \( t \),

\[ J(z(t + h)) \leq J(z(t)) - \beta \| \frac{\partial J}{\partial z}(z(t)) \|^2 \]

We use this estimate to derive a contradiction in the case that \( \frac{\partial J}{\partial z}(z(t)) \not\to 0 \). If this is so then there is an \( \epsilon \) so that for every \( t \) there is a \( t' > t \) where \( \| \frac{\partial J}{\partial z}(z(t)) \| \geq \epsilon \). In particular, there is an increasing sequence \( \{ t_i \} \) of times where \( t_{n+1} - t_n > h \) and \( \| \frac{\partial J}{\partial z}(z(t_i)) \| \geq \epsilon \) for all \( i \). Setting \( y_n = J(z(t_n)) - J(z(t_{n-1})) \) it follows that \( J(z(t_0)) + \sum_{n=1}^{\infty} y_n \to J^* \) where the series converges. However, since \( t_{n+1} - t_n \geq h \), and \( J(t) \) is strictly decreasing, we must have

\[
J(z(t_{n+1})) - J(z(t_n)) \leq J(z(t_n + h)) - J(z(t_n)) \\
\leq -\beta \| \frac{\partial J}{\partial z}(z(t_n)) \|^2 \\
\leq -\beta \epsilon^2
\]

which implies that \( \sum_{n=1}^{\infty} y_n \to -\infty \).

We will make use of the following corollary of this result:
Corollary 3.3.2. Let \( v \) be as in Theorem 3.3.1 and consider the system

\[
\frac{dz}{dt}(t) = -u(t)
\]

where for all \( t > 0 \),

\[
\| \frac{\partial v}{\partial z}(z(t)) - u(t) \|_2 \leq \alpha \| u(t) \|_2
\]

for \( \alpha < \frac{1}{2} \). Then \( v(z(t)) \) converges and \( \frac{\partial v}{\partial z}(z(t)) \to 0 \).

Proof. We can write \(-u(t) = -\frac{\partial v}{\partial z}(z(t)) + e(t)\) where \( e(t) = \frac{\partial v}{\partial z}(z(t)) - u(t) \), and it is easily seen that \( \|e(t)\|_2 \leq \frac{\alpha}{1-\alpha} \| \frac{\partial v}{\partial z}(z(t)) \|_2 \). Since \( \alpha < \frac{1}{2} \), \( e(t) \) satisfies the conditions of Theorem 3.3.1.

3.3.2 Discrete Time

In this section we present a result that is analogous to Theorem 3.3.1 and used to show convergence of an optimization algorithm in the discrete time setting. The condition on the gradient updates is visualized in Figure 3.1.
Proposition 3.3.3. Let \( J : W \to \mathbb{R} \) be differentiable, with an \( L \)-Lipschitz gradient, and bounded from below. Consider a sequence

\[
w(t + 1) = w(t) - \epsilon h(t)
\]

where \( \| h(t) - \frac{\partial J}{\partial w}(w(t)) \|_W \leq \alpha \| h(t) \|_W \) for some \( \alpha \in [0, \frac{1}{2}) \) and \( \epsilon \in (0, \frac{2(1-\alpha)}{L(1+\alpha)^2}) \). Then \( J(w(t)) \) converges and \( \frac{\partial J}{\partial w}(w(t)) \to 0 \).

Note that when there is no error in the derivatives (\( \alpha = 0 \)) this gives the usual \( \epsilon < \frac{2}{L} \) requirement of gradient descent.

### 3.4 Finsler Structures

To derive more general conditions for contraction, we can go beyond the metrics based on norms. These metrics are defined by minimizing a length functional, and form a subclass of the Finsler metrics. First we define this class of metrics then we will derive contraction conditions. Our main interest will be when we get to the stochastic systems. We will present ergodicity conditions which rely on pointwise contraction estimates involving such metrics.

Let \( X \) be a closed convex subset of the Euclidean space \( \mathbb{R}^n \) and let \([x \leadsto y]\) be the set of piecewise \( C^1 \) curves from \( x \) to \( y \). Given a norm \( \| \cdot \| \) on \( \mathbb{R}^n \) and a function \( x \mapsto A(x) \) taking values in the invertible \( n \times n \) matrices, one can define a metric on \( X \) as follows.

**Proposition 3.4.1.** Let \( \| \cdot \| \) be a norm on \( \mathbb{R}^n \) and let \( x \mapsto A(x) \) be a continuous function that assigns to each \( x \in X \) an invertible linear map \( A(x) \) on \( \mathbb{R}^n \), in such a way that \( \sup_{x \in X} \| A(x)^{-1} \| < \infty \). For a piecewise \( C^1 \) curve \( \gamma : [\gamma_s, \gamma_e] \to X \), define

\[
L(\gamma) = \int_{\gamma_s}^{\gamma_e} \| A(\gamma(t)) \gamma'(t) \| \, dt.
\]
CHAPTER 3. METHODS

Then the function

\[ d_A(x, y) = \inf_{\gamma \in [x \rightarrow y]} L(\gamma) \]

defines a metric on \( X \) compatible with the Euclidean topology, and \( (X, d_A) \) is complete.

**Proof.** The metric axioms can be shown as in (14, Chapter 2). We show the completeness. The condition on \( A(x)^{-1} \) means that for some \( k \) the inequality

\[ \|x - y\| \leq kd_A(x, y) \quad (3.6) \]

holds for all \( x, y \in X \). The continuity of \( A \) means that \( \|A\| \) is bounded on compact subsets of \( X \). Combining this with (3.6) it follows that \( d_A \) and the metric determined on \( \|\cdot\| \) are strongly equivalent on compact subsets of \( X \). Using (3.6) one can show that any \( d_A \)-Cauchy sequence is contained in a compact subset of \( X \). By the strong equivalence \( d_A \) is complete on this subset. \( \square \)

For instance taking \( A = I_n \) one recovers the norm \( d_A(x, y) = \|x - y\| \). Using \( A(x) = V(x)I_n \) for real-valued function \( V \) means a cost \( V(x) \) is assigned for going through each point \( x \). Using a general matrix allows the cost for traveling through each point \( x \) to also depend on the direction of the path at the point.

### 3.5 Wasserstein Distances

To generalize our discussion of contraction beyond deterministic systems, we will work with the space of probability distributions equipped with the Wasserstein distance. In this section we recall the definition of Wasserstein distance, and below we will derive ergodicity conditions using this notion.

The collection of Borel probability measures on \( X \) is denoted \( \mathcal{P}(X) \). We denote by \( \nu(\epsilon) \)
CHAPTER 3. METHODS

the expectation of $e$ under $\nu$. That is, $\nu(e) = \int_X e(x) \, d\nu(x)$.

Given a function $V : X \to \mathbb{R}_{\geq 0}$ the space $\mathcal{P}_{p,V}(X)$ is defined to be all Borel measures $\nu$ on $X$ which can integrate $V^p$:

$$\mathcal{P}_{p,V}(X) = \left\{ \nu \in \mathcal{P}(X) \left| \int_X V(x)^p \, d\nu(x) < \infty \right. \right\}.$$  

If $d_A$ is a metric, then we define $\mathcal{P}_{p,A}$ to be the set of measures which can integrate $x \mapsto d_A(x_0, x)^p$ ($x_0$ is an arbitrary base point.) This set is equipped with the Wasserstein metric $d_{p,A}$:

$$d_{p,A}(\nu_1, \nu_2) = \inf_{\gamma \in \Gamma(\nu_1, \nu_2)} \left( \int_{X \times X} d_A(x, y)^p \, d\gamma(x, y) \right)^{1/p}.$$  

The space $\mathcal{P}_{p,A}$ is complete if $(X, d_A)$ is. Furthermore, the Kantorovich duality formula holds for $p = 1$:

$$\sup_{\|e\|_{\text{Lip}} \leq 1} |\nu_1(e) - \nu_2(e)| = d_{1,A}(\nu_1, \nu_2).$$  \hspace{1cm} (3.7)

See (98) for more background.

The notion of convergence in the Wasserstein distance is closely related to the notion of weak convergence. A precise statement is that a sequence of measures $\nu_n$ converges to $\nu$ in the Wasserstein distance if and only if $\nu_n$ converges weakly to $\nu$ and $\lim_{n \to \infty} \int_X d_A(x_0, x) \, d\nu_n = \int_X d_A(x_0, x) \, d\nu$ (98).

The completeness means the contraction mapping theorem can be used to test if a Markov kernel is ergodic; if the Markov kernel $P$ defines a contraction on this space, then $\nu P^n$ converges to a unique invariant measure $\pi$ for any starting measure $\nu$. This is formalized below.

**Proposition 3.5.1.** Let $P$ be a Markov kernel on a Polish space $X$. Let the following
CHAPTER 3. METHODS

contraction condition hold:

\[ \sup_{x_1 \neq x_2} \frac{d(\delta_{x_1} P, \delta_{x_2} P)}{d(x_1, x_2)} := \rho < 1. \tag{3.8} \]

Then \( P \) has a unique stationary measure \( \pi \), and

\[ d(\nu P^t, \pi) \leq \rho^t d(\nu, \pi), \]

In Chapter 5 we will investigate sufficient conditions for 3.8.

3.6 Measure Valued Differentiation

When dealing with systems on a discrete space, we will have take a different approach to gradient estimation. The idea of measure valued differentiation is to express the derivative of an expectation as the difference of two expectations. Each of these expectations involves the same cost function, but the underlying measures are different. If these measures are easy to sample from, this leads to a simple, unbiased derivative estimator. For simplicity we will consider the setting of a finite state space \( X \) in these definitions. Formally, we consider a measure \( \nu_\theta \) that depends on a real parameter \( \theta \).

**Definition 3.6.1.** The measure \( \nu_\theta \) is said to be differentiable at \( \theta \in \mathbb{R} \) if there is a triple \((c_\theta, \nu^+_\theta, \nu^-_\theta)\), consisting of a real number \( c_\theta \), and two probability measures \( \nu^+_\theta, \nu^-_\theta \) on \( X \) such that

\[ \frac{\partial}{\partial \theta} \nu_\theta(e) = c_\theta[\nu^+_\theta(e) - \nu^-_\theta(e)] \]

for any function \( e : X \to \mathbb{R} \).

An MVD gradient estimator would consist of two parts: First, sample a random variable \( Y^+ \) distributed according to \( \nu^+_\theta \), then sample random variable \( Y^- \) according to \( \nu^-_\theta \), and
finally form the estimate $\Delta^{MVD} = c_\theta [e(Y^+) - e(Y^-)]$. Compared to finite differences, the advantage is that there is no bias and there is no division by a small number. For some background see (42).

**Example 3.6.2.** Let $\nu_1, \nu_2$ be two probability measures on a measurable space $X$, and define the measure $\rho_\theta$ that depends on a parameter $\theta \in \mathbb{R}$:

$$\rho_\theta(A) = e^{-\theta^2} \nu_1(A) + (1 - e^{-\theta^2}) \nu_2(A)$$

(3.9)

for any set $A$. The parameter determines which of the measures $\nu_i$ is more likely in this mixture. By simple calculus, it holds that for any bounded measurable function $e : X \to \mathbb{R}$,

$$\frac{\partial}{\partial \theta} \rho_\theta(e) = c_\theta [\nu_2(e) - \nu_1(e)].$$

where

$$c_\theta = 2\theta e^{-\theta^2}$$

Then according to Definition 3.6.1, the triple $(c_\theta, \nu_2, \nu_1)$ an MVD of the measure $\rho_\theta$.

See (42) for a number of other examples.

### 3.6.1 Measure Valued Differentiation for Markov chains

The concept of MVD can be extended from measures to Markov kernels, and then applied to derivatives of stationary costs (68, 67).

**Definition 3.6.3.** A Markov kernel $P_\theta$ that is defined on a finite space $X$ and which depends on a real parameter $\theta$ is said to be differentiable at $\theta$ if for each $x \in X$ there is a triple $(c_\theta(x), P^+_\theta(x, \cdot), P^-_\theta(x, \cdot))$ which is the measure valued derivative of the measure $P(x, \cdot)$ at the parameter $\theta$ in the sense of Definition 3.6.1.
Formally, this means \( \frac{\partial}{\partial \theta} (\delta_x P_\theta(e)) = c_\theta(x)[\delta_x P^+(e) - \delta_x P^-(e)] \) for all \( x \in X \) and all cost functions \( e : X \to \mathbb{R} \).

If the Markov kernel \( P_\theta \) is ergodic with stationary measure \( \pi_\theta \), then in certain cases it is possible to use the \( (c_\theta, P_\theta^+, P_\theta^-) \) to compute the stationary derivatives \( \frac{\partial}{\partial \theta} \pi_\theta(e) \) (67). We now describe this.

The procedure is presented in Algorithm 4. A theorem about its correctness is given in Theorem 3.6.4. For measure valued differentiation, like finite differences, it seems that \( O(m) \) simulations are required for a system with \( m \) parameters, although the variance characteristics are much more favorable compared to finite differences (see (69), Section 4.3). In finite differences, one must trade off bias for variance, but for MVD the variance can be shown to be bounded independently of the parameters \( M^1, M^0 \) which determine the bias.

**Algorithm 4: MVD gradient estimation for Markov chains**

```plaintext
for t = 0, 1, \ldots, M^0 - 1 do
  Sample x(t+1) from P_\theta(x(t), \cdot)
end

Sample x^+(0) from P_\theta^+(x(M^0), \cdot)
Sample x^-(0) from P_\theta^-(x(M^0), \cdot)

for t = 0, 1, \ldots, M^1 - 1 do
  Sample x^+(t+1) from P_\theta(x^+(t), \cdot)
  Sample x^-(t+1) from P_\theta(x^-(t), \cdot)
end

- Set \( \Delta^{MVD} = c_\theta(x(M^0)) \sum_{t=1}^{M^1} [e(x^+(t)) - e(x^-(t))] \)

return \( \Delta^{MVD} \).
```

We recall a theorem on measure valued differentiation for Markov chains. It gives a condition on a Markov chain \( P_\theta \) that guarantees the corresponding stationary costs \( \pi_\theta(e) \) are differentiable. This result is from (67).
Theorem 3.6.4. Let \((\delta_x P_\theta)(e)\) be differentiable for each bounded, Lipschitz continuous \(e\). That is, for each \(x\) there is a triple \((c_\theta(x), P^+_\theta(x, \cdot), P^-_\theta(x, \cdot))\) such that

\[
\frac{\partial}{\partial \theta} (\delta_x P_\theta)(e) = c_\theta(x)[P^+_\theta(e) - P^-_\theta(e)]
\]

for each bounded, Lipschitz \(e\). Furthermore, suppose that \(P_\theta\) is a contraction on the space \(\mathcal{P}(X)\) in the sense of Inequality (3.8). Then the stationary cost \(\pi_\theta(e)\) is differentiable, and Algorithm 4 can be used to estimate the derivatives. Specifically, if we let \(\Delta^{MVD}\) be the output of the algorithm, then

\[
\lim_{M^0 \to \infty, M^1 \to \infty} \Delta^{MVD} = \frac{\partial}{\partial \theta} \pi_\theta(e)
\]

More general results on MVD for stationary measures can be found in (40).

3.7 Simultaneous Perturbation

Another idea which inspired this work is simultaneous perturbation. We motivate the simultaneous perturbation estimator by showing how it addresses some of the shortcomings of finite difference.

In the case of a single, real-valued parameter, the finite difference method uses two copies of the system to estimate the derivative at a particular point \(\theta(0)\). One copy of the system runs with setting \(\theta(0) + \lambda\), and one uses \(\theta(0) - \lambda\). After running for a long-time, the error in both of these systems is sampled, and a difference quotient is formed as the gradient estimate. The extension to \(m\) parameters involves replicating the procedure \(m\) times, one for each coordinate direction. Typically, when dealing with an \(n\) node neural network there are \(\approx n^2\) parameters. Therefore finite differences would require running \(\approx 2n^2\) copies of the network, which is unfeasible. Furthermore, it has unfavorable variance properties.
One interesting solution to the cost of estimating derivatives with finite differences is known as simultaneous perturbation (87). In this scheme, one picks a random direction $v$, and then approximates the directional derivative using stochastic finite differences. In this case only two simulations are needed for a system with $m$ parameters. Using random directions in stochastic approximation was also studied in (53, Section 2.3.5) and (20).

The variance issues remain with this approach; in order to decrease the bias of the estimator, one has to deal with a larger variance. For generating the directions, one possibility is to let $v$ be a random point on the hypercube $\{-1/2, 1/2\}^m$, as suggested in (87). For the theoretical analysis, it is important that the directions have zero mean, and that the random variable $\frac{1}{\|v\|}$ is integrable. The procedure is shown in Algorithm 5.

**Algorithm 5: Simultaneous perturbation derivative estimation for Markov chains**

- Initialize $\lambda$ to a small positive number.
- Sample direction $v$ from the measure

\[
P(v) = \prod_{i=1}^{n} \left[ \frac{1}{2}\delta_{-1/2}(v_i) + \frac{1}{2}\delta_{1/2}(v_i) \right]
\] (3.10)

for $t = 0, 1, \ldots, M - 1$ do

Sample $x^+(t + 1)$ from $P_{\theta + \lambda v}(x^+(t), \cdot)$

Sample $x^-(t + 1)$ from $P_{\theta - \lambda v}(x^-(t), \cdot)$

end

- Set $\Delta^{SP} = \frac{e(x^+(M)) - e(x^-(M))}{2\lambda} v$

return $\Delta^{SP}$

This algorithm will form the basis for the gradient estimator we derive for discrete attractor networks in Chapter 6.
Chapter 4

Deterministic Attractor Networks

Our proposed work is to analyze an optimization algorithm based on the adjoint method for gradient estimation. To gain intuition for this problem we first consider the setting of continuous time systems - in this case certain aspects of the problem are simplified since there is no step-size to choose. Then we use this intuition to address the problem of interest - which is the setting of discrete time systems. The results of Section 4.1 were reported in (22) and the results of Section 4.2 appear in (23).

4.1 Continuous Time

Given a fixed input, the neural networks studied here “compute by convergence” to a steady state. The output of the network is taken to be this fixed point or some simple readout thereof. It is possible to implement backpropagation-like algorithms which take into account the recurrent nature of the network. Early works which explored this include (7, 71, 4). A desirable property of such algorithms is that they operate locally in time and space. Local in time means the algorithm only uses information along the trajectory of the network, as opposed to requiring asymptotic data (such as the fixed point of the system) at each
step. Spatial locality requires that during training information be transferred using the existing network topology. In a neural network this means the process taking place on a parameter along a particular edge only requires information from the two nodes incident at that edge. Such a process, which we review below, can be derived naturally from the equations describing the gradient in a recurrent network and is comprised of two activities: accumulating sensitivities along the trajectory of the network and a flow on the parameters. In the absence of dynamics on the weights, this process simply computes the sensitivities and one would expect that accurate estimates can still be obtained if the weights change slowly enough (alternatively, that the sensitivity accumulation happens fast enough). If the weights change too quickly the sensitivity estimates may not be able to keep up, and the optimization will fail. Thus, controlling the relative rates of these two activities is crucial for ensuring these optimization schemes work.

We show under reasonable assumptions on the network architecture that finite and fixed rates can be chosen so that this localized learning scheme has the same long time behavior as the gradient system associated to the network, which from the distributed perspective requires infinite power and non-local communication. The assumptions are that the network obeys a stability condition, known as contraction, uniformly for all parameter values and that the various first and second derivatives of the activity at each node are bounded. The main technical tools we use are stability results applicable to contracting systems and hierarchical combinations of such systems. The optimization problem is approached as a gradient system with error, and the time-scale is chosen so as to control this error.

The assumption of contraction plays a key role in our results. This stability criteria, which says roughly that any two trajectories of a system converge towards each other, has been fruitfully applied to problems in systems biology and the analysis of networks of dynamical elements generally. This has been used to study, for instance, phenomena such as synchronization (99) and entrainment to periodic signals (80). Here we show how it can be
applied in the context of distributed gradient-based optimization of such networks.

The main theoretical tool used by previous works to analyze this problem was singular perturbation theory. The basic idea is to analyze the process on the weights and the forward motion of the network separately, according to their natural time-scales. If certain stability criteria are met in each of these cases then stability of the original system can be concluded when the weights are adapted slowly enough (or the forward motion happens fast enough), i.e., using an appropriate time-scale separation. Using general results for systems with multiple time-scales, (74) analyzed a gradient-based supervised scheme from this perspective. In this situation the stable points of the parameter space are local minima of an objective function. They showed local stability of the system around such points, again for a sufficient choice of the time-scale separation. Those authors also looked at the problem in (75), where they also noted that the problem of learning a periodic trajectory could be reduced to a set of fixed-point learning problems. A supervised training scheme with a different learning rule was also analyzed in (94) and a similar conclusion was reached. That work also investigated conditions which guarantee the neuron dynamics remain stable as the connection weights are modified during training, which is important as in general dynamics on the weights may destroy stability. The issue of time-scales in supervised learning was also noted in (7, 71, 8), where heuristics were proposed for choosing it in practice.

In the above cited theoretical analyses the essential soundness of the schemes is only established when the procedure begins near a local minimum of the objective function. Additionally, previous works did not attempt to quantify this requirement or the bound on the time-scale in terms of information that might be available about the model. Here, the specific set of assumptions and the tools we develop allow one to conclude that the optimization will work under much milder conditions on the initial state of the system. It is only necessary that the neuron activity be close enough to its stable point before the optimization begins. This initial requirement and a sufficient time-scale parameter are
quantified in terms of information that may be known, such as bounds on derivatives and the rate of convergence of neuron dynamics. Given enough information about the model, we show how to “bootstrap” the system so that the procedure can be carried out from any starting point.

In this section we focus on supervised learning, where one has a specific objective on the networks’ behavior. A related line of research investigates the stability of Hebbian learning in recurrent networks. In (18) those authors show stability by constructing a Lyapunov function on the joint system consisting of the forward activity of the network and a Hebbian process on the weights and use this to show that the total system converges to a local fixed point. Singular perturbation methods were applied to an unsupervised learning rule in (60).

Similar considerations apply to probabilistic neural networks. For instance, in order to calculate gradients in the Boltzmann machine it is necessary to compute expectations over the stationary distribution of a Markov chain. Hence the applicability of a multiple time-scale approach, consisting of a Markov chain from which statistics are gathered and a comparatively slower weight adaptation process. Some theoretical aspects of such algorithms were analyzed in (101). In (102) a connection was noted between this class of stochastic optimization procedures and the contrastive divergence algorithm for training restricted Boltzmann machines. This algorithm was shown to give favorable results on some machine learning tasks in (93). A variant of this procedure also plays an important role in training algorithms for deep Boltzmann machines (82). In (62) the problem for sigmoid belief networks, which are directed acyclic models, was also considered, and that author suggests a two time-scale approach utilizing persistent Markov chains for these models as well.

4.1.1 Optimization System

In this section we introduce the optimization system, which consists of the forward and adjoint system, and the dynamics on the weights. The derivation here is somewhat standard,
CHAPTER 4. DETERMINISTIC ATTRACTOR NETWORKS

see also (8). We start with the system whose asymptotic behavior we wish to optimize:

\[
\frac{dx}{dt}(t) = f(x(t), w, \rho).
\]  \hfill (4.1)

Here \( x \in \mathbb{R}^n \) is the state of the neural network, \( w \) are the parameters and \( \rho \) is some input vector which remains fixed while the network is running. For instance, in the case of neural networks typically used in machine learning the \( f \) might be \( f(x, w, \rho) = -x + \sigma(wx + \rho) \) where \( \sigma \) is a sigmoid function applied component wise. For now we will drop the notation indicating the input \( \rho \). We assume that \( f \) has a unique fixed point for each choice of the parameter vector \( w \), and that the system tends towards this point regardless of the starting configuration, in a specific sense which we identify in the next section. We want to design a flow on the parameters \( w \) of \( f \) which finds a local minimum of some objective function on the equilibrium point. Let \( x^*(w) \) give the equilibrium point of (4.1), i.e., the solution of the equation

\[
f(x^*(w), w) = 0
\]  \hfill (4.2)

and let \( g \) be a function which calculates fitness of points in the phase space. To minimize or at least find a stationary point of \( g(x^*(w)) \) a natural process would be the gradient system

\[
\frac{dw}{dt}(t) = -\frac{\partial}{\partial w}(g \circ x^*)(w(t)).
\]  \hfill (4.3)

According to the implicit function theorem, \( x^* \) will be differentiable at \( w \) so long as \( \frac{\partial f}{\partial x} \) is invertible at the fixed point. Differentiating (4.2) and using the chain rule one can then express (4.3) as

\[
\frac{dw}{dt}(t) = \frac{\partial g}{\partial x} (x^*(w(t)), w(t)) \left( \frac{\partial f}{\partial x}(x^*(w(t)), w(t)) \right)^{-1} \frac{\partial f}{\partial w} (x^*(w(t)), w(t)).
\]  \hfill (4.4)
As is, this system poses problems from both the distributed and computational point of view. This is due to the presence of terms involving the equilibrium point \( x^*(w) \) and the inverse of \( \frac{\partial f}{\partial x} \). That is, it is local in neither time nor space. We introduce a new variable \( y \) with dynamics

\[
\frac{dy}{dt}(t) = y(t)\frac{\partial f}{\partial x}(x^*(w), w) - \frac{\partial g}{\partial x}(x^*(w)).
\]

(4.5)

The process (4.5) can be thought of as a relaxation method to solve the equation \( y \frac{\partial f}{\partial x} = \frac{\partial g}{\partial x} \).

Taking \( y^* = y^*(w) \) to be the fixed point of this system we can rewrite (4.4) as

\[
\frac{dw}{dt}(t) = y^* \frac{\partial f}{\partial w}(x^*(w(t)), w(t)).
\]

(4.6)

So far this equation defines the same flow on \( w \) as (4.3). Now we consider the localized system consisting of \( x, y, \) and \( w \) where we use only information along the trajectories of \( x \) and \( y \), as opposed to their equilibrium points.

\[
\tau \frac{dx}{dt}(t) = f(x(t), w(t)) \quad \text{(4.7a)}
\]

\[
\tau \frac{dy}{dt}(t) = y(t)\frac{\partial f}{\partial x}(x(t), w(t)) - \frac{\partial g}{\partial x}(x(t)) \quad \text{(4.7b)}
\]

\[
\frac{dw}{dt} = y(t)\frac{\partial f}{\partial w}(x(t), w(t)) \quad \text{(4.7c)}
\]

Here we have introduced two separate time scales via the parameter \( \tau \). Note that by setting \( \tau = 0 \) equation (4.7c) becomes equation (4.6). These three equations define the distributed optimization system. As noted in (74) this can also be viewed as a three-time-scale algorithm by introducing separate rates \( \tau_x \) and \( \tau_y \) on the corresponding subsystems. For simplicity we calculate a single rate which suffices for both systems. We refer to the individual systems (4.7a) and (4.7b) as the forward and adjoint systems respectively. At times we refer to \( z = (x, y) \) as the joint system.

Our goal in the analysis of the system is to show that, under proper conditions, this flow
does decrease the error and that it does so at a sufficient pace that its long time behavior matches that of (4.3). This can be ensured by keeping $z$ close enough to $z^*(w)$, in a sense we make precise in Section 4.1.3. In the next section we specify the stability properties we require of the forward system and give conditions under which this extends to the joint system. We then formalize the condition that the weight process (4.7c) should satisfy in Section 4.1.3, taking the view that it is a gradient system with error. In Section 4.1.4 we present a general result which provides time-scales for this type of condition and in Section 4.1.5 we verify that this is applicable to the optimization system. We demonstrate several neural network architectures to which these results apply in Section 4.1.6.

### 4.1.2 Stability Criteria

We will show that the $y$ system (4.7b) is contracting whenever the $x$ system (4.7a) is, and that the above proposition may be applied to conclude that the joint system $(x, y)$ is contracting. This will allow us to carry $(x, y)$ as a single system $z$ in the analysis of the learning procedure. Specifically, the next result shows that the $y$ system is contracting in the dual norm $\|x\|_* = \sup_{\|y\| = 1} |x^T y|$.

**Proposition 4.1.1.** If the $x$ system (4.7a) is contracting with rate $\beta$ in a given norm $\| \cdot \|$, then the $y$ system (4.7b) is contracting with rate $\beta$ in the dual norm $\| \cdot \|_*$.

**Proof.** It is easily seen that the Jacobian of the adjoint system (4.7b) is $\frac{\partial f}{\partial x}^T$, i.e. the transpose of the Jacobian of the forward system (4.7a). It is well known that for any vector norm, the induced matrix norm satisfies $\|A^T\|_* = \|A\|$. Then for any number $h$, $\|I + hA^T\|_* = \|I + hA\|$ and therefore $\mu_*(A^T) = \mu(A)$, where $\mu_*$ is the matrix measure with respect to the dual norm. In particular $\mu_* \left( \frac{\partial f}{\partial x}^T \right) = \mu \left( \frac{\partial f}{\partial x} \right)$. 

We would like to note that when speaking of a dual norm as in the norm $\|v\|_{X^*}$ of a vector in $\mathbb{R}^n$ this is meant in the sense of a particular norm on the vector space $\mathbb{R}^n$. Likewise, for
a matrix $M \in \mathbb{R}^{n \times m}$ and an expression like $\|M\|_{X,Y}$, we are referring to the norm of $M$ as a linear map between the vector spaces $\mathbb{R}^m$ and $\mathbb{R}^n$ with the norms $\|\cdot\|_X$ and $\|\cdot\|_Y$ respectively.

The assumptions on the network dynamics $f$ and the loss function $g$ we use are that the first and second derivatives are uniformly bounded for all values of the parameter vector $w$. These properties may be verified in any norm which is convenient or by just showing that all (mixed) partials are bounded. However, the specific time-scale $\tau$ that we obtain is phrased in terms of bounds on these quantities with respect to specific norms that we identify throughout.

The $y$ system is a linear time-varying system and the following bound applies:

**Proposition 4.1.2.** Assume $x$ is contracting and $\sup_x \|\partial g / \partial x(x)\|_* \leq (L_x g)$ for some $L_x g$. Then $\|y\|_* \leq \frac{L_x g}{\beta}$ is forward invariant for $y$.

**Proof.** Say $\|y(t)\|_* > \frac{L_x g}{\beta}$. Then $\|y(t) + h \frac{d}{dt} y(t)\|_* \leq \|I + h (\frac{\partial f}{\partial x}(x(t), w(t)))^T\|_\ast \|y(t)\|_* + h \|\partial g / \partial x(x(t))\|_*$ and therefore $\frac{d}{dt} \|y(t)\|_* \leq \mu (\|\partial f / \partial x(x(t), w(t))\|_\ast + \|\partial g / \partial x(x(t))\|_*)$. Under the assumption on $\|y\|_*$ then $\frac{d}{dt} \|y(t)\|_* < 0$. \hfill \Box

We are now in a position to show that the joint system is contracting. Technically the system is contracting on the set $\mathbb{R}^n \times B_\ast \left(\frac{L_x g}{\beta}\right)$ where $B_\ast(r) = \{y \in \mathbb{R}^n \mid \|y\|_* \leq r\}$ is the ball of a given radius in the norm $\|\cdot\|_*$.

**Proposition 4.1.3.** Assume that for all $x$ and $w$ the bounds $\|\partial^2 g / \partial x^2(x)\|_{X,X} \leq L_x^2 g$, $\|\partial f / \partial x^2(x,w)\| \leq L_x^2 f$, and that the conditions of Proposition 4.1.2 hold. Let $\beta_x$ the contraction rate of the forward system and let $p_1, p_2$ be two positive numbers such that $\beta_x - \frac{p_2}{p_1} k > 0$ where $k = \frac{(L_x g)}{\beta_x} (L_x^2 f) + (L_x^2 g)$. Then the joint system (4.7a, 4.7b) is contracting with rate $\beta$ in the norm $\|\cdot\|_Z$ on the set $U$ where $\beta = \beta_x - \frac{p_2}{p_1} k$, $\|(x,y)\|_Z = p_1 \|x\| + p_2 \|y\|_*$, and $U = \mathbb{R}^n \times B_\ast \left(\frac{L_x g}{\beta_x}\right)$. 
Proof. We verify the conditions of Theorem 3.1.1. By Proposition 4.1.1 we conclude the adjoint system $y$ is contracting in the norm $\| \cdot \|_*$. Define $e(x, y, w) = y \frac{\partial f}{\partial x}(x, w) - \frac{\partial g}{\partial x}(x)$, so that $\frac{\partial y}{\partial t}(y) = e(x(t), y(t), w(t))$. We will show that $\| \frac{\partial y}{\partial x}(x, y, w) \|_{X, X^*} \leq k$ for all $w$ and $(x, y) \in U$. Let $x_1, x_2$ be two possible states of the $x$ system. Holding $y, w$ fixed we have

$$\|e(x_1) - e(x_2)\|_* \leq \| (\frac{\partial f}{\partial x}(x_1) - \frac{\partial f}{\partial x}(x_2))^T y \|_* + \| \frac{\partial g}{\partial x}(x_1) - \frac{\partial g}{\partial x}(x_2) \|_*$$

Using the fact that $\| M^T \|_* = \| M \|$ for any matrix $M$ one has

$$\| (\frac{\partial f}{\partial x}(x_1) - \frac{\partial f}{\partial x}(x_2))^T y \|_* \leq \| (\frac{\partial f}{\partial x}(x_1) - \frac{\partial f}{\partial x}(x_2))^T \|_* \| y \|_*$$

$$\leq \| \frac{\partial f}{\partial x}(x_1) - \frac{\partial f}{\partial x}(x_2) \| \| y \|_*$$

$$\leq (L_{x^2} f) \| x_1 - x_2 \| \| y \|_*$$

By assumption on $\frac{\partial g}{\partial x}$ we have $\| \frac{\partial g}{\partial x}(x_1) - \frac{\partial g}{\partial x}(x_2) \|_* \leq (L_{x^2} g) \| x_1 - x_2 \|$. By definition of $U$ we may assume $\| y \|_*$ is bounded. Therefore Theorem 3.1.1 can be applied. \qed

This next result summarizes two important facts about contracting systems we shall use later. The first is how the fixed point $z^*$ changes with the parameter $w$. The second is how $\| u \|$, which may be thought of as measure of energy for a contracting system, changes when there is some dynamics on the parameter $w$.

Proposition 4.1.4. Assume $\frac{dz}{dt}(t) = u(z(t), w(t))$ is contracting with rate $\beta$ in $\| \cdot \|_Z$ and for all $(z, w)$ the function $u$ satisfies $\| \frac{\partial u}{\partial w}(z, w) \|_{W, Z} \leq (L_w u)$ for a norm $\| \cdot \|_W$. Then $\| \frac{\partial z^*}{\partial w}(w) \|_{W, Z} \leq \frac{L_w u}{\beta}$ and for all $t > 0$, $\frac{d}{dt} \| u(t) \|_Z \leq -\beta \| u(t) \|_Z + (L_w u) \frac{d}{dt} w(t) \| W$

Proof. In general for a matrix $A$, if $\mu(A) \leq -\alpha$ for $\alpha > 0$, then $\| A^{-1} \| \leq \alpha^{-1}$ (16). Turning to the system $z$, by the chain rule we have $\frac{\partial z^*}{\partial w} = -\frac{\partial z}{\partial z^*}(z^*(w), w)^{-1} \frac{\partial z}{\partial z}(z^*(w), w)$. Since $\mu(\frac{\partial u}{\partial w}) \leq -\beta$ where $\beta > 0$, we know that $\frac{\partial u}{\partial z}$ is invertible and $\| \frac{\partial u}{\partial z} \|_Z \leq \frac{1}{\beta}$. Then $\| \frac{\partial z^*}{\partial w} \|_{W, Z} \leq \frac{1}{\beta} \| \frac{\partial u}{\partial w} \|_{W, Z} \leq \frac{L_w u}{\beta}$.
For the second statement we proceed as in the proof of (97, Theorem 2.5.3). It is sufficient to show that  
\[ \lim_{h \to 0^+} \frac{1}{h} (\|u(t) + h \frac{d}{dt} u(t)\|_Z - \|u(t)\|_Z) \leq -\beta \|u(t)\|_Z + (L_w u) \frac{d}{dt} w(t) \|_W. \]
We have
\[ \|u(t) + h \frac{d}{dt} u(t)\|_Z = \|u(t) + h \left[ \frac{\partial u}{\partial z}(z(t), w(t)) u(t) + \frac{\partial u}{\partial w}(z(t), w(t)) \frac{d}{dt} w(t) \right] \|_Z \]
\[ \leq \|I + h \frac{\partial u}{\partial z}(z(t), w(t))\|_Z \|u(t)\|_Z + h \|\frac{\partial u}{\partial w}(z(t), w(t))\|_{W,Z} \|\frac{d}{dt} w(t)\|_W. \]

From here one can subtract \|u\|_Z from both sides, divide by \( h \), and take limits as \( h \to 0^+ \), while using the definition of matrix measure.

\[\boxed{\|h(t) - h^*(t)\|_2 \leq \alpha \|h(t)\|_2} \quad (4.8)\]

for some \( 0 \leq \alpha < \frac{1}{2} \). Such conditions or closely related criteria are commonly used in discrete gradient descent schemes, see for instance (11). By itself, condition (4.8) guarantees descent of the objective function and when the other conditions of Corollary 3.3.2 are satisfied convergence can be concluded as well. We now reformulate this condition so that we may utilize information about the online behavior of the joint system \( z \) which is available in the norm \( \| \cdot \|_Z \). Norms in Euclidean space being equivalent, a sufficient condition for (4.8) is
that
\[ \|h(t) - h^*(t)\|_W \leq \frac{\alpha}{k} \|h(t)\|_W \]
for some \( k \) such that
\[ \forall u, \quad \frac{1}{\sqrt{k}} \|u\|_W \leq \|u\|_2 \leq \sqrt{k} \|u\|_W. \]  \tag{4.9}
When \( \frac{\partial h}{\partial z} \) is bounded a sufficient condition for (4.8) is then
\[ \|z(t) - z^*(t)\|_Z \leq \frac{\alpha}{k(Lzh)} \|h(t)\|_W \]  \tag{4.10}
where \( Lzh \) is such that \( \|\frac{\partial h}{\partial z}(x,y,w)\|_Z,W \leq Lzh \). In any contracting system we have
\[ \beta \|z_1 - z_2\|_Z \leq \|u(z_1) - u(z_2)\|_Z \]  \tag{4.11}
for two states \( z_1, z_2 \), where \( \beta \) is the contraction rate (85). In particular \( \beta \|z - z^*\|_Z \leq \|u(z)\|_W \), so it suffices that
\[ \|u(t)\|_Z \leq \frac{\alpha \beta}{k(Lzh)} \|h(t)\|_W. \]  \tag{4.12}
In the next section we show that this condition can be satisfied for all \( \alpha \) below a certain threshold by choosing a corresponding rate \( \tau \) small enough.

\subsection*{4.1.4 Time-scale Selection}

One approach to the optimization problem is to view the system (4.7) as a singularly perturbed version of (4.3) and then apply standard results for such systems. This has been used in the context of Hebbian learning in (60) and in the supervised case in (74). In (74), those authors investigate the behavior of a “semirelaxed” form of the learning problem (4.4), in which only a temporal relaxation is introduced, and show that near a joint equilibrium of the total system \((x, w)\), one can conclude convergence of the relaxed learning algorithm to
CHAPTER 4. DETERMINISTIC ATTRACTOR NETWORKS

this local minimum, for an adequate choice of the time-scale parameter. Using the strong assumption of contraction, we are able to have weaker conditions on the optimization problem and the starting conditions. We do not assume that the original gradient system (4.3) obeys any particular stability criteria, only that it is Lipschitz continuous. Our condition on the starting point of the algorithm, stated in the theorem below, is only that the joint system \((x, y)\) starts sufficiently close to its equilibrium point. This is always possible to do by running the system for a “warm-up” period, since when there is no optimization happening the joint system is convergent. In (15) some topics of singular perturbation theory are revisited from the perspective of contraction theory. For instance, the results in that paper may be applied to conclude that \(\tau\) and the starting point may be chosen so that \(z\) remains within an arbitrarily small radius around \(z^*\) for all time.

As noted above, the specific property we require of the trajectory allows \(z\) to be far from \(z^*\) when the gradient is large, but requires that the distance goes to zero as the gradient does. The theorem we now prove is a general result for this type of condition and in the next section we apply it to the distributed optimization system.

**Theorem 4.1.5.** Consider a system of the form

\[
\tau \frac{dz}{dt}(t) = u(z(t), w(t))
\]

\[
\frac{dw}{dt}(t) = h(z(t), w(t))
\]

where \(z\) is contracting with rate \(\beta\) in the norm \(\| \cdot \|_Z\) and \(\| \cdot \|_W\) is a norm on \(w\) such that

\[
\| \frac{\partial u}{\partial w}(z, w) \|_{W, Z} \leq L_w u, \quad \| \frac{\partial h}{\partial z}(z, w) \|_{Z, W} \leq L_z h \quad \text{and} \quad \| \frac{\partial h}{\partial w}(z, w) \|_W \leq L_w h \quad \text{for all} \ (z, w).
\]

For any \(0 < \alpha < 1\), if

\[
c = \frac{\alpha \beta}{(L_z h)}
\]
and

\[ \tau \leq \frac{(1 - \alpha)\beta c}{c(L_w h) + (L_w u)} \]

then \( c\|h(z, w)\|_W \geq \|u(z, w)\|_Z \) is forward invariant for the system \((z, w)\).

Proof. Let \( H(t) = \|h(t)\|_W, U(t) = \|u(t)\|_Z \) and \( B(t) = cH(t) - U(t) \). We show that \( \frac{d}{dt} B(t) > 0 \) when \( B(t) < 0 \). Note that the conditions of Proposition 4.1.4 are satisfied so that

\[ \frac{d}{dt} U(t) \leq -\frac{\beta}{\tau} U(t) + (L_w u)H(t). \]

On the other hand, since in general \( \frac{d}{dt} \|h(t)\|_W \geq -\|D_t h(t)\|_W \), we have

\[ \frac{d}{dt} H \geq -\frac{(L_z h)}{\tau} U - (L_w h)H \]

Combining these, it follows that for \( B \)

\[ \frac{d}{dt} B(t) \geq c \left( -\frac{(L_z h)}{\tau} U(t) - (L_w h)H(t) \right) - (L_w u)H(t) + \frac{\beta}{\tau} U(t) \]

When \( B(t) < 0 \) we have \( cH(t) < E(t) \). Using this fact, and rearranging terms one obtains

\[ \frac{d}{dt} B(t) > U(t) \left( -(L_w h) - \frac{(L_w u)}{c} + \frac{1}{\tau}(\beta - c(L_z h)) \right) \geq 0 \]

where the last inequality follows by assumption on \( c \) and \( \tau \).

4.1.5 Application to the Optimization System

To show that the above result may be applied to the distributed optimization problem it suffices to exhibit the bounds \( L_w u, L_z h, L_w h \), and also show that it is possible to choose \( c \) small enough so that condition (4.12) may be satisfied. The latter problem is trivial once we show that \( D_z h \) is bounded, since the theorem provides a timescale for arbitrarily small
values of the parameter \( c \). Demonstrating the bounds on these derivatives is a straightforward application of the assumptions.

**Proposition 4.1.6.** Assume the conditions of Proposition 4.1.3. Let \( \| (x, y) \|_z = p_1 \| x \| + p_2 \| y \|_* \) be the norm provided by that result, let \( \beta_x \) be the contraction rate of the forward system, and further assume there are numbers \( L_w f, L_{x,w} f, \) and \( L_{w^2} f \) such that \( \| \frac{\partial f}{\partial w} (x, w) \|_W \leq L_w f, \| \frac{\partial^2 f}{\partial x \partial w} (x, w) \|_{X,W} \leq L_{x,w} f, \) and \( \| \frac{\partial^2 f}{\partial w^2} (x, w) \|_{W,W} \leq L_{w^2} f \). Then \( \| \frac{\partial h}{\partial z} (z, w) \|_{Z,W} \leq \max \left\{ \left( \frac{L_x g}{\beta_x} \right) \left( L_{x,w} f \right), \left( L_w f \right) \right\} \) and \( \| \frac{\partial h}{\partial w} (z, w) \|_{W} \leq \frac{(L_x g)}{\beta_x} (L_{w^2} f) \).

**Proof.** By Proposition (4.1.2) we may assume \( \| y \|_* \leq \frac{L_x g}{\beta_x} \). Fix \( w \) and let \( z_1 = (x_1, y_1), z_2 = (x_2, y_2) \) be two states of the joint system. Note that for any \( x \), \( \| (\frac{\partial f}{\partial w} (x, w))^T \|_{X,W} = \| \frac{\partial f}{\partial w} (x, w) \|_{W^*,X} \leq L_w f \) and

\[
\| (\frac{\partial f}{\partial w} (x_1, w) - \frac{\partial f}{\partial w} (x_2, w))^T \|_{X,W} = \| \frac{\partial f}{\partial w} (x_1, w) - \frac{\partial f}{\partial w} (x_2, w) \|_{W^*,X}
\]

Then we have

\[
\| h(z_1, w) - h(z_2, w) \|_W = \| \frac{\partial f}{\partial w} (x_1, w)^T y_1 - \frac{\partial f}{\partial w} (x_2, w)^T y_2 \|_W \\
\leq \| (\frac{\partial f}{\partial w} (x_1, w) - \frac{\partial f}{\partial w} (x_2, w))^T y_1 \|_W + \| \frac{\partial f}{\partial w} (x_2, w)^T (y_1 - y_2) \|_W \\
\leq (L_{x,w} f) \| x_1 - x_2 \|_X \| y_1 \|_* + (L_w f) \| y_1 - y_2 \|_*
\]

Setting \( k = \max \{ \frac{L_x g}{\beta_x} (L_{x,w} f), \frac{L_w f}{p_2} \} \) it follows that \( \| h(z_1, w) - h(z_2, w) \|_W \leq k \| z_1 - z_2 \|_Z \). For the bound on \( \frac{\partial h}{\partial w} \), fix some \( z \) and let \( w_1, w_2 \) be two parameter vectors. Then

\[
\| h(z, w_1) - h(z, w_2) \|_W = \| (\frac{\partial f}{\partial w} (z, w_1) - \frac{\partial f}{\partial w} (z, w_2))^T y \|_W \\
\leq \| \frac{\partial f}{\partial w} (z, w_1) - \frac{\partial f}{\partial w} (z, w_2) \|_{W^*,X} \| y \|_* \\
\leq (L_{w^2} f) \| y \|_* \| w_1 - w_2 \|_W
\]
Using this, the constraint (4.12) with \( L_z h \) equal to the bound on \( \| \frac{\partial h}{\partial z} \|_{Z,W} \) given by the above proposition is sufficient to ensure descent during the learning process.

The next result says the other conditions of the proposition are also true, i.e., \( \frac{\partial u}{\partial w} \) and \( \frac{\partial u}{\partial z} \) are uniformly bounded.

**Proposition 4.1.7.** Assume the conditions of Proposition 4.1.3. Let \( \| (x,y) \|_Z = p_1 \| x \| + p_2 \| y \|_* \) be the norm provided by that result, let \( \beta_x \) be the contraction rate of the forward system, and further assume that \( \| \frac{\partial f}{\partial w}(x,w) \|_{W,X} \leq L_w f \), \( \| \frac{\partial^2 f}{\partial x \partial w}(x,w) \|_{X,W,X} \leq L_{x,w} f \), \( \| \frac{\partial f}{\partial x}(x,w) \|_{X,X} \leq L_x f \), and \( \| \frac{\partial^2 f}{\partial x^2}(x,w) \|_{X,X,X} \leq L_x^2 f \) for all \((x,w)\). for some numbers \( L_w f, L_{x,w} f, L_x f, \) and \( L_x^2 f \).

Then \( \| \frac{\partial u}{\partial z}(z,w) \|_Z \leq (L_x f) + \frac{(L_{x,w} f)}{\beta_x} \| y \|_* \| w \|_W \)

or \( \| \frac{\partial u}{\partial z}(z,w) \|_W \leq (L_w f) + \frac{(L_{w,x} f)}{\beta_x} \| y \|_* \| w \|_W \).

**Proof.** Fixing \( x \) we have \( \| f(x,w_1) - f(x,w_2) \|_X \leq (L_w f) \| w_1 - w_2 \|_W \). For the \( y \) system we have

\[
\| (\frac{\partial f}{\partial x}(x,w_1) - \frac{\partial f}{\partial x}(x,w_2))^T y \|_* \\
\leq \| (\frac{\partial f}{\partial x}(x,w_1) - \frac{\partial f}{\partial x}(x,w_2))^T X^* \| y \|_* \\
\leq \| \frac{\partial f}{\partial x}(x,w_1) - \frac{\partial f}{\partial x}(x,w_2) \|_X \| y \|_* \\
\leq (L_{w,x} f) \| w_1 - w_2 \|_W \| y \|_*
\]

Then

\[
\| u(w_1) - u(w_2) \|_Z \leq p_1 (L_w f) \| w_1 - w_2 \|_W + p_2 (L_{w,x} f) \| y \|_* \| w_1 - w_2 \|_W \\
= (p_1 (L_w f) + p_2 (L_{w,x} f) \| y \|_*) \| w_1 - w_2 \|_W
\]

We now bound \( \frac{\partial u}{\partial z} \). Fixing \( w \), let \( z_1 = (x_1,y_1) \), \( z_2 = (x_2,y_2) \) be two states of the enlarged
system. Then \( \| f(x_1, w) - f(x_2, w) \|_X \leq (L_x f) \| x_1 - x_2 \|_X \) while for the \( y \) system we have
\[
\| \left( \frac{\partial f}{\partial x} (x_1, w) - \frac{\partial f}{\partial x} (x_2, w) \right)^T y_1 \|_* \leq \| \left( \frac{\partial f}{\partial x} (x_1, w) - \frac{\partial f}{\partial x} (x_2, w) \right)^T \|_* \|_{x, x^*} \| y \|_*
\]
\[
\leq \left( L_{x^2} f \right) \| x_1 - x_2 \|_X \| y \|_*
\]
and \( \| \left( \frac{\partial f}{\partial x} (x_2, w)^T (y_1 - y_2) \|_* \leq \| \left( \frac{\partial f}{\partial x} (x_2, w)^T \|_* \| y_1 - y_2 \|_* \leq (L_x f) \| y_1 - y_2 \|_* \). It follows that
\[
\| \left( \frac{\partial f}{\partial x} (x_1, w) - \frac{\partial f}{\partial x} (x_2, w) \right)^T y_1 \|_* \leq \left( L_{x^2} f \right) \| x_1 - x_2 \|_X \| y \|_* + (L_x f) \| y_1 - y_2 \|_*
\]
Then
\[
\| u(z_1, w) - u(z_2, w) \|_Z \leq \left( p_1 \left( L_x f \right) + p_2 \left( L_{x^2} f \right) \| y \|_* \right) \| x_1 - x_2 \|_X + \left( L_{x^2} f \right) \| y_1 - y_2 \|_*
\]
Setting \( k = \left( L_x f \right) + \frac{p_2}{p_1} \left( L_{x^2} f \right) \| y \|_* \), it follows that \( \| u(z_1, w) - u(z_2, w) \|_Z \leq k \| z_1 - z_2 \|_Z \).

The following theorem summarizes the result for the optimization system (4.7). We write \( A_u v \) for the assumption that the derivative \( \frac{\partial v}{\partial u} \) is uniformly bounded.

**Theorem 4.1.8.** Assume that the forward system (4.7a) is contracting, and properties \( A_x f, A_{x^2} f, A_w f, A_{w^2} f, A_{x, w} f, A_x g \) and \( A_{x^2} g \). Then there are \( 0 < \alpha < \frac{1}{2} \) and \( \tau > 0 \) such that the optimization system (4.7) verifies the descent condition (4.8) along the whole trajectory, given suitable initial conditions on the joint system \( (x, y) \). If, in addition, \( g \) is bounded from below then \( (g \circ x^*)(w(t)) \) converges and \( \frac{\partial}{\partial w} (g \circ x^*)(w(t)) \to 0 \).

**Proof.** The conditions of Proposition 4.1.3 are satisfied so one may construct a norm \( \| \cdot \|_Z \) in which the joint system \( z = (x, y) \) is contracting, with contraction rate \( \beta \). The conditions of Propositions 4.1.6 and 4.1.7 are also satisfied, so that the bounds on \( \frac{\partial u}{\partial w}, \frac{\partial h}{\partial z} \) and \( \frac{\partial h}{\partial w} \) exist. Then by Theorem 4.1.5, for all \( c \) sufficiently small there is a rate \( \tau \) so that the set...
\( \|u\|_Z \leq c\|h\|_W \) is forward invariant for the optimization system (4.7). In particular we can choose \( c \) small enough so that

\[
c < \frac{\beta}{2k(L_zh)}
\]

where \( k \) is defined as in equation (4.9) and the descent condition (4.8) will be satisfied. Note that this is equivalent to selecting \( \alpha < \frac{1}{2k} \) in Theorem 4.1.5.

Next we show that \( \frac{\partial}{\partial w}(g \circ x^*)(w) \) is Lipschitz continuous. Let \( L_w h, L_z h \) be the bounds on \( \|\frac{\partial h}{\partial w}\|_W \) and \( \|\frac{\partial h}{\partial z}\|_{Z,W} \) respectively provided by Proposition 4.1.6 above. Note that

\[
-\frac{\partial}{\partial w}(g \circ x^*)(w) = h(x^*(w), y^*(w), w) = h(z^*(w), w)
\]

For any two parameters \( w_1, w_2 \) we have

\[
\|h(z^*(w_1), w_1) - h(z^*(w_2), w_2)\|_W \leq (L_w h)\|w_1 - w_2\|_W \quad (4.13)
\]

and

\[
\|h(z^*(w_1), w_2) - h(z^*(w_2), w_2)\|_W \leq (L_z h)\|z^*(w_1) - z^*(w_2)\|_Z \leq (L_z h)K\|w_1 - w_2\|_W \quad (4.14)
\]

where \( K \) is the bound on \( \frac{\partial z^*}{\partial w} \) given by Proposition 4.1.4. Combining (4.13) and (4.14),

\[
\|h(z^*(w_1), w_1) - h(z^*(w_2), w_2)\| \leq ((L_w h) + (L_z h)K)\|w_1 - w_2\|_W
\]

Therefore Corollary 3.3.2 can be applied.

Lastly we would like to address how to bootstrap the optimization system. This is the problem of finding some initial \((x, y)\) which verify (4.12) for a given \( \alpha \). This can be done by...
Table 4.1: Stability conditions for the network $\frac{dx}{dt}(t) = f(x(t), w) = -x(t) + \sigma(wx(t) + \rho)$ for different choices of the vector norm. In the first row $\sigma'(u)$ refers to the diagonal matrix whose $(j, j)$ entry is $\sigma'(u_j)$. Note that the third column uses a matrix norm.

<table>
<thead>
<tr>
<th>Vector norm</th>
<th>$\mu(\frac{\partial f}{\partial x})$</th>
<th>Stability criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|\cdot|_2$</td>
<td>$\max_i {\lambda_i(-I + \frac{1}{2}(\sigma'(u)w + (\sigma'(u)w)^T))}$</td>
<td>$|w|<em>2 &lt; \frac{1}{L</em>{\sigma'}}$</td>
</tr>
<tr>
<td>$|\cdot|_1$</td>
<td>$-1 + \max_j {\sum_i \sigma'(u_i)</td>
<td>w_{i,j}</td>
</tr>
<tr>
<td>$|\cdot|_\infty$</td>
<td>$-1 + \max_i {\sigma'(u_i)\sum_j</td>
<td>w_{i,j}</td>
</tr>
</tbody>
</table>

running the joint system with $w$ fixed until the condition (4.12) is satisfied. The left and right side terms can be measured at run-time if the norms $\|\cdot\|_W$ and $\|\cdot\|_Z$ can be computed and the other constants are available. So long as $h^* \neq 0$ this inequality must eventually be verified, since $h \to h^*$ and $u \to 0$ exponentially fast. After this the optimization can be started.

### 4.1.6 Neural Networks

We now verify whether the above assumptions hold for various neural network architectures. These models use a sigmoid function which is bounded and has bounded derivatives. These can be for instance the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ or the hyperbolic tangent $\sigma_H(x) = \tanh(x) = \frac{e^x - e^{-x}}{1+e^{-2x}}$. We assume the existence of three numbers $L_{\sigma}, L_{\sigma'}$ and $L_{\sigma''}$ such that $|\sigma(x)| \leq L_{\sigma}, |\sigma'(x)| \leq L_{\sigma'},$ and $|\sigma''(x)| \leq L_{\sigma''}$. The error function $g$ should have $\|\frac{\partial g}{\partial x}\|$ and $\|\frac{\partial^2 g}{\partial x^2}\|$ bounded.

Various conditions have been identified which ensure convergence to a unique equilibrium point in Neural Networks. In (7) that author shows global stability when the sum-of-squares of the weights is small. Matrix measure conditions like the ones referred to here were also explored in (21). Many of the conditions can be enforced by selecting a suitable norm for the weight matrix and requiring that the weight matrix is not too large when measured this
way. For instance, consider the model where unit $i$ evolves according to

$$\frac{dx_i}{dt}(t) = -x_i(t) + \sigma \left( \sum_j x_j(t)w_{i,j} + \rho_i \right)$$  \hspace{1cm} (4.16)$$

where $\rho$ is an input vector and we assume $w_{i,i} = 0$. Writing $u_i = \sum_j x_j w_{i,j} + \rho_i$, the Jacobian of this system is

$$\frac{\partial f_i}{\partial x_j} = -\delta_{i,j} + \sigma'(u_i)w_{i,j}$$

From here one may obtain different conditions on the weight matrix which ensure convergence to a unique equilibrium point by choosing different norms on the state $x$. Several examples of this are summarized in Table 4.1. Note that any feed-forward networks satisfying equation (4.16) may be seen to be contracting by recursive application of Theorem 3.1.1.

To determine whether the results of this section are applicable to a given model it is necessary to verify the uniform stability in a particular norm, while the uniform boundedness of the maps $\frac{\partial f_i}{\partial x_i}$, $\frac{\partial f_i}{\partial w_i}$ for $i = 1, 2$ and $\frac{\partial^2 f_i}{\partial w \partial x}$ may be verified in any norm. As mentioned above, to actually set the rates and verify the initial conditions it is necessary to have bounds in particular norms.

**Example 4.1.9.** The simplest case to analyze is a “fixed feedback” network. The parameters have been partitioned into a matrix $v$ holding the weights between internal nodes $x$ and a matrix $w$ holding the weights between the internal nodes and the input $\rho$. We assume $v_{i,i} = 0$. The forward dynamics are

$$\frac{dx}{dt}(t) = f(x(t), w) = -x(t) + \sigma(vx(t) + w\rho)$$  \hspace{1cm} (4.17)$$

The system will be contracting for $\|v\|$ small enough in some norm, since $\sigma'$ is bounded and $\rho$ is fixed. Optimization takes place on the parameters $w$. It can be verified that all first and second derivatives with respect to $x$ and $w$, along with the mixed partials $\frac{\partial^2 f}{\partial x \partial w}$, are bounded
Table 4.2: Derivatives and bounds for the recurrent network (4.17)

<table>
<thead>
<tr>
<th>Derivative values</th>
<th>Norm required</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\partial f_i}{\partial w_{i,k}}$</td>
<td>$\sigma'(u_i)\rho_k$</td>
<td>$|\frac{\partial f}{\partial w}|_{F,2}$</td>
</tr>
<tr>
<td>$\frac{\partial^2 f_i}{\partial x_j \partial w_{i,r}}$</td>
<td>$\sigma''(u_i)v_{i,j}\rho_r$</td>
<td>$|\frac{\partial^2 f}{\partial x\partial w}|_{2,2,2}$</td>
</tr>
<tr>
<td>$\frac{\partial^2 f_i}{\partial x_k \partial w_{i,r}}$</td>
<td>$\sigma''(u_i)v_{i,j}\rho_r\rho_k$</td>
<td>$|\frac{\partial^2 f}{\partial x\partial w}|_{F,F,2}$</td>
</tr>
<tr>
<td>$\frac{\partial^2 f_i}{\partial x_j \partial x_k}$</td>
<td>$\sigma''(u_i)v_{i,j}v_{i,k}$</td>
<td>$|\frac{\partial^2 f}{\partial x^2}|_{2,2,2}$</td>
</tr>
</tbody>
</table>

uniformly. These are shown in Table 4.2. This shows the distributed optimization procedure is feasible for this model.

For this example we shall describe in detail how one may choose a suitable pair of the time-scale $\tau$, and the parameter $\alpha$ which controls how small the error in the gradient is. We start with the following assumptions:

- The 2-norm is used on the network state $x$, and the Frobenius norm is used for weight matrix $w$, i.e. $\|w\|_F = \left(\sum_i \sum_j (w_{i,j})^2\right)^{\frac{1}{2}}$.

- The feedback matrix $v$ has $\|v\|_F < \frac{1}{L_{\sigma'}}$, so that the network is contracting in the 2-norm with rate $\beta_x = 1 - L_{\sigma'}\|v\|_F$.

- The error function is $g(x) = \frac{1}{2}\|x - t\|_2^2$ for some target vector $t \in [0, 1]^n$ where $n$ is the number of nodes in the network. Then $\|\frac{\partial g}{\partial x}\| = \|x - t\| \leq \sqrt{n}$ and $\|\frac{\partial^2 g}{\partial x^2}\| = \|I\| = 1$.

This choice of norm on the weight matrix has the advantage that it is self-dual, i.e. $\|w\|_{F^*} = \|w\|_F$ so there are fewer bounds to compute. For instance, in general one needs bounds on both $\|\frac{\partial f}{\partial w}\|_{W,X}$ and $\|\frac{\partial f}{\partial w}\|_{W^*,X}$ (to compute the results of Propositions 4.1.7 and 4.1.6, respectively), but these coincide when using the Frobenius norm. It is also consistent with the 2-norm in the sense that $\|wx\|_2 \leq \|w\|_F \|x\|_2$, for a vector $x$, which simplifies computations of the bounds. Table 4.2 lists bounds on the derivatives of $f$ that are needed for the computation of the time-scale parameter $\tau$. To show how these bounds are computed,
Proposition 4.1.10. Consider the neural network defined by equation (4.17). The derivatives of $f$ in this case satisfy $\| \frac{\partial f}{\partial w} \|_{F,2} \leq L_{\sigma'} \| \rho \|_2$ and $\| \frac{\partial^2 f}{\partial w \partial f} \|_{F,F,2} \leq L_{\sigma''} \| \rho \|^2_2$.

Proof. The entries of the matrix $\frac{\partial f}{\partial w}$ are $\frac{\partial f}{\partial w_{j,k}} = \delta_{i,j} \sigma' (u_j) \rho_k$. We show that for a general matrix $n \times (n \times n)$ matrix $M$ with $M_{i,(j,k)} = \delta_{i,j} a_j b_k$, the norm $\| M \|_{F,2}$ satisfies $\| M \|_{F,2} \leq \| a \|_\infty \| b \|_2$. The first result follow from this by setting $a_j = \sigma' (u_j)$ and $b = \rho$. By definition $\| M \|_{F,2} = \sup_{\| N \|_F = 1} \| MN \|_2$. Straight forward computation shows that for any matrix $N$ the entries of the vector $MN$ are $(MN)_i = a_i (N \rho)_i$. Therefore $MN = D(a)(N \rho)$ where $D(a)$ is the diagonal matrix with the vector $a$ on the diagonal. Then $\| MN \|_2 \leq \| D(a) \|_2 \| N \rho \|_2$. In the 2-norm, a diagonal matrix $D(a)$ has $\| D(a) \|_2 = \| a \|_\infty$. Using this, and the fact that the Frobenius norm is consistent, we get $\| MN \|_2 \leq \| a \|_\infty \| \rho \|_2 \| N \|_F$. Since the $N$ is arbitrary, this shows $\| M \|_{F,2} \leq \| a \|_\infty \| \rho \|_2$.

To show the second part, fix $x$ and let $w_1, w_2$ be two different weight matrices. Then $\frac{\partial f}{\partial w}(x, w_1) - \frac{\partial f}{\partial w}(x, w_2) = M$ where $M_{i,(j,k)} = \delta_{i,j} a_j b_k$, where $a_j = \sigma' ((v_x + w_1 \rho)_j) - \sigma' ((v_x + w_2 \rho)_j)$ and $b = \rho$. By the above argument, $\| M \|_{F,2} \leq \| a \|_\infty \| b \|_2$. In general $\| a \|_\infty \leq \| a \|_2$, and in this case $\| a \|_2 \leq L_{\sigma''} \| (w_1 - w_2) \rho \|_2 \leq L_{\sigma''} \| \rho \|_2 \| w_1 - w_2 \|_F$. This shows $\| \frac{\partial f}{\partial w} (x, w_1) - \frac{\partial f}{\partial w} (x, w_2) \|_{F,2} \leq L_{\sigma''} \| \rho \|_2 \| w_1 - w_2 \|_F$, from which the second result follows. 

Given this, the rest of the constants may be computed as follows:

- Define a norm and compute contraction rate on the joint system (4.7a), (4.7b):

  - Define $K = \frac{L_{x} g}{\beta_x} L_{x^2} + L_{x^2} g$
  - Choose $p_1, p_2 > 0$ such that $\beta_x - \frac{p_2}{p_1} K > 0$
  - Define $\beta = \beta_x - \frac{p_2}{p_1} K$
• Compute bounds provided by Propositions 4.1.6 and 4.1.7:

- Define \( L_z h = \max \left\{ \frac{L_z g}{\beta_x} L_{x,w} \frac{L_x f}{p_1}, \frac{L_{w,f}}{p_2} \right\} \)

- Define \( L_{w,h} = \frac{L_z g}{\beta_x} L_{w^2} f \)

- Define \( L_{w,u} = p_1(L_w f) + p_2(L_{w,x} f) \frac{L_z g}{\beta_x} \)

• Compute a time-scale parameter:

- Choose \( 0 \leq \alpha < \frac{1}{2} \)

- Define \( c = \frac{\alpha \beta}{L_z h} \)

- Define \( \tau = \frac{(1-\alpha)\beta c}{c L_{w,h} + L_{w,u}} \)

There are a number of choices to be made when calculating the time-scale parameter. Beyond the choice of norms \( \| \cdot \|_X \) and \( \| \cdot \|_W \), one must also decide on the coefficients \( p_1, p_2 \) for the norm \( \| \cdot \|_Z \) and the parameter \( \alpha \) which controls in the error in the gradient. Intuitively, a small \( \alpha \) means less error in the gradient, and therefore a better guarantee in the rate of descent of the objective \( g(x^*) \), and one can see in the above that there is a trade off between \( \alpha \) and \( \tau \): A small \( \alpha \) requires a small \( \tau \), which means more energy must expended running the adjoint system. The nature of the trade-off is modulated by the choice of norms, and presumably one would want to select norms which exhibit an efficient trade-off.

A problem which can arise in recurrent network optimization is that the dynamics on the weights may destroy the global stability of the network (19). That is, bifurcations may occur which result in the creation of spurious fixed points or other undesired phenomena. The results in this section require that the network tends to a unique fixed point for each parameter value at a uniform rate. If this rate is not uniform, and the convergence rate is allowed to approach zero during training, a finite time-scale separation may not be sufficient to ensure the descent condition. The network in Example 4.1.9 didn’t present any of these
issues since we were able to bound the contraction rate independently of the parameters being optimized. In (50) the authors explored several approaches to stability in the discrete time case. In one suggested approach, a penalty term is added to the objective to keep the parameters within a stable regime. In (7) the training procedure scales the weights down if they grow so large that global stability cannot be guaranteed. (94) applied the concept of connective stability to derive conditions under which the network will remain stable during weight adaptation.

To get around the possibility of bifurcations, these next examples have uniform stability “built-in” by some mechanism which allows one to a priori bound the weight on each connection. Instead of having the dynamics act directly on the connection strengths, in several of the models we have the weight on the connection from \( j \) to \( i \) pass through another sigmoid function. For instance applying this to the model 4.16, the forward dynamics would then consist of

\[
\frac{dx(t)}{dt} = -x(t) + \sigma(\omega x(t) + \rho)
\]

\[
\omega = \sigma(w)
\]

The point of introducing this model is that although it still may be possible for the weights \( w_{i,j} \) to grow to infinity, the relevant quantities for optimization (the first and second derivatives of \( f \) with respect to \( x \) and \( w \)) remain bounded. It also enables one to separately control the magnitude and sign of each connection. Note that once the network is trained it is only necessary to retain \( \omega_{i,j} \) as the weight from \( i \) to \( j \) and \( w_{i,j} \) can be discarded.

**Example 4.1.11.** We extend the previous example by adding a readout of the feedback
Table 4.3: Derivatives for the model defined by equations (4.18a) and (4.19)

<table>
<thead>
<tr>
<th>$\frac{\partial f_i}{\partial x_j}$</th>
<th>$\sigma'(u_i)\omega_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\partial f_i}{\partial w_{i,k}}$</td>
<td>$\sigma'(u_i)x_k\alpha_k\sigma'(w_{i,k})$</td>
</tr>
<tr>
<td>$\frac{\partial^2 f_i}{\partial x_j \partial x_k}$</td>
<td>$\sigma''(u_i)\omega_{i,j}\omega_{i,k}$</td>
</tr>
<tr>
<td>$\frac{\partial^2 f_i}{\partial w_{i,k} \partial w_{i,r}}$</td>
<td>$\sigma''(u_i)x_kx_r\alpha_k\alpha_r\sigma'(w_{i,k})\sigma'(w_{i,r})$</td>
</tr>
<tr>
<td>$\frac{\partial^2 f_i}{\partial x_j \partial w_{i,r}}$</td>
<td>$\sigma''(u_i)\omega_{i,j}x_r\alpha_r\sigma'(w_{i,r})$</td>
</tr>
</tbody>
</table>

module which is also subject to optimization:

\[
\frac{dx_1}{dt}(t) = -x_1(t) + \sigma(\omega x_2(t))
\]
\[\omega = \sigma_H(w)\]
\[
\frac{dx_2}{dt}(t) = -x_2(t) + \sigma(v x_2(t) + u \rho)
\]

which has three sets of parameters $v, w, u$, of which $w$ and $u$ are subject to optimization. The presence of a fixed feedback module and dynamics on the read-out weights is a hallmark of reservoir computing (58). This system is a hierarchy of the form $x_2 \rightarrow x_1$ and Theorem 3.1.1 may be applied to conclude the overall system $(x_1, x_2)$ is contracting. The boundedness of the various derivatives holds as well.

\[\square\]

**Example 4.1.12.** We now turn to a network where optimization takes place on all connections. Let $in(i)$ and $out(i)$ denote the in-degree and out-degree respectively of node $i$, and let $E$ be the set of edges present in the network.

\[
\frac{dx_i}{dt}(t) = -x_i(t) + \sigma \left( \sum_{j:(j,i) \in E} \omega_{i,j}x_j(t) + \rho_i \right) \quad (4.18a)
\]
\[
\omega_{i,j} = \alpha_i\sigma_H(w_{i,j}) \quad (4.18b)
\]
This system is contracting in the $\infty$-norm, for instance, when

$$0 \leq \alpha_i < \frac{1}{L_{\sigma_L} L_{\sigma_L} \text{in}(i)}$$

This can be easily seen from the definition of $\mu_\infty$. Due to the $\omega$ terms being bounded, the other conditions required by the optimization procedure may also be verified. If we replace (4.18b) in this example with the definition

$$\omega_{i,j} = \alpha_j \sigma(w_{i,j}) \quad (4.19)$$

then the system (4.18a, 4.19) is contracting in the 1-norm when

$$|\alpha_j| < \frac{1}{L_\sigma L_{\sigma_L} \text{out}(j)}$$

Here the sign of $\alpha_j$ determines whether connections emanating from that node are facilitory or inhibitory.

4.1.7 Discussion

We believe that in addition to differentiability of the system components, the existence of appropriate time-scales as demonstrated here is key to concluding that optimization is feasible in recurrent networks. This allows one to implement a flow on the parameters which is naturally adapted to the distributed nature of the network, in the sense that it can be expressed in terms of local information. This is not only relevant to distributed or parallel implementations. When the optimization is phrased in terms of the system (4.7), one avoids having to solve a set of nonlinear equations at each step and can rely on simple computations involving the the derivatives of the activation functions. We also believe that the conditions required here are not too restrictive since, as shown in Section 4.1.6, a wide
variety of criteria may obtained by different choices of norm. Several of these conditions allow one to conclude stability based on local information at each node. The corresponding contraction rate also depends only on local data. This is important if the procedure is to be regulated in a decentralized manner. As is, the rate $\tau$ provided by these results does not have this property, since it involves the equivalence between the 2-norm and the norm in which the system is contracting, which may depend on global properties such as the number of nodes in the network.

Another interesting extension of this work would be to consider more detailed models based on chemical kinetics (17, 12). These models are motivated by the physiology of biological neurons. As a kind of halfway point, the G-networks of (27) seem promising. They are more complex than the networks considered here, but are still very simple compared to complete kinetic models.

### 4.2 Discrete Time

We consider the optimization problem for discrete time attractor networks as defined in Section 2.3.1. We begin with a dynamic system $x(t + 1) = f(x(t), w)$ that depends on a parameter $w$. We assume a strong convergence property on the underlying dynamics $f$. This is the notion of contraction as described in Section 3.2. Then $f$ has a unique fixed-point $x^*(w)$. Given an objective function $e : X \rightarrow \mathbb{R}$, one obtains the optimization problem:

Starting from $w_0 \in W$, find a local minimum of $(e \circ x^*)(w)$

The algorithm involves adjoint sensitivity analysis for computing gradients, as outlined in Algorithm 2. The auxiliary system for computing gradients involves the map $T$:

$$T((x, y), w) = \begin{pmatrix} f(x, w) \\ \left( \frac{\partial f}{\partial x}(x, w) \right)^T y - \frac{\partial e}{\partial x}(x) \end{pmatrix}$$

(4.20)
Figure 4.1: Update pattern for the adjoint-method studied in Section 4.2. \((x, y)\) is the state of the joint system, where \(x\) is the state of the network and \(y\) is the state of the adjoint system. The left graph shows how \((x, y)\) and \(w\) are calculated at time \(t\) from their values at time \(t - 1\). On the right is the update pattern of the map \(T\), showing details of how \((x(t), y(t))\) are calculated from \((x(t - 1), y(t - 1))\).

It is easy to verify that for each parameter \(w \in W\) the map \(T\) has as a fixed point the pair \((x^*(w), (I - \frac{\partial f}{\partial x}(x^*(w), w))^{-T} \frac{\partial e}{\partial x}(x^*(w)))\). With the assumptions on \(f\) and \(e\) we will employ, one can also show that this is the only fixed-point of \(T\) and that \(T\) converges to this point; in short that \(T\) inherits the contraction property from \(f\). More details on this are given below. The optimization procedure investigated alternates between iterating \(T\), and updating the parameter \(w\) using an approximate gradient derived from the variables \((x, y)\):

\[
(x(t + 1), y(t + 1)) = T((x(t), y(t)), w(t)) \\
w(t + 1) = w(t) + \epsilon \left( \frac{\partial f}{\partial w}(x(t + 1), w(t)) \right)^T y(t + 1)
\]

Figure 4.1 shows the (somewhat arbitrary) update pattern of the variables \(x, y, w\) at each step of the algorithm. Other patterns may be more appropriate depending on the setting. The parameter to tune is the step-size \(\epsilon\). After step (4.21a), if \((x(t + 1), y(t + 1))\) is close to the equilibrium point \((x^*(t), y^*(t))\), then the calculated search direction will be accurate. After performing the update in step (4.21b), there is a new parameter \(w(t + 1)\) and therefore
a new equilibrium point \((x^*(t+1), y^*(t+1))\). The following iteration of \(T\), which brings \((x(t+1), y(t+1))\) to \((x(t+2), y(t+2))\) takes a step towards this new equilibrium point. If the equilibrium has not moved very much, then perhaps only a single iteration of \(T\) suffices to guarantee feasibility of the next computed search direction. Intuitively, the smaller the step-size \(\epsilon\), then the smaller the motion on the equilibrium point \((x^*(t+1), y^*(t+1))\), and therefore it should require less work (fewer applications of \(T\)) to ensure that the next search directions are good. This illustrates the additional role of the step-size in such dynamic gradient approximation schemes. Formalizing this and showing that indeed \(\epsilon\) can be chosen small enough that some form of convergence of the optimization procedure can be guaranteed is the main contribution of this work.

A couple of remarks about this result. As long as the assumptions hold, the initial condition can be verified using only the data \((x(0), y(0), w(0))\) with the inequality \(|z(0) - z^*(0)| \leq \frac{1}{1 - \beta_T}||T(z(0), w(0)) - z(0)||\) from the contraction mapping principle (\(\beta_T\) is the Lipschitz constant of \(T\) in \(z\)). Secondly, a \(z(0)\) such that \(||T(z(0), w(0)) - z(0)|| \leq (1 - \beta_T)c||h(z(0), w(0)||\) can always be found if \(w(0)\) is not a stationary point, simply by iterating \(T\) long enough starting from an arbitrary point.

Our analysis uses basic properties of contraction mappings and gradient descent. As shown in Proposition 4.2.4, the map \(T\) is a contraction mapping under our assumptions on \(f\) and \(e\). We find in Section 4.2.2 that in order to have accurate gradient estimates, it suffices that the iterates \((x(t), y(t))\) stay near the equilibrium point of \(T(\cdot, w(t))\). Then we use a result about autonomous perturbations of contractions, Proposition 4.2.5, that enables us to satisfy this condition by choosing \(\epsilon\) properly. In Section 4.3 a numerical example on a ring of nonlinear units is presented. Each unit has a real valued parameter, and the goal of the optimization procedure is to make the network have a given target vector as its stable point.
4.2.1 Assumptions

We assume that $X$ is a convex region in $\mathbb{R}^n$, and that $W$ is a finite dimensional vector space. We use the 2-norm in both $X$ and $W$ but note that none of below the results intrinsically depend on the 2 norm or other inner-product norms.

The cost function $e : X \to \mathbb{R}$ and the dynamics $f : X \times W \to X$ should satisfy the following conditions.

**Assumption 4.2.1 (Contraction property).** There is a $\beta \in [0, 1)$ such that

$$
\sup_{(x,w) \in X \times W} \| \frac{\partial f}{\partial x}(x, w) \| \leq \beta.
$$

**Assumption 4.2.2 (Other derivatives are bounded).** There is an $L$ such that

$$
\sup_{(x,w) \in X \times W} \| \frac{\partial^{i+j} f}{\partial x^i \partial w^j}(x, w) \| \leq L \text{ for } 1 \leq i + j \leq 2.
$$

We make a number of assumptions on the error function $e : X \to \mathbb{R}$ as well.

**Assumption 4.2.3.** The function $e$ is bounded from below and has $\sup_{x \in X} \| \frac{\partial e}{\partial x}(x) \| \leq L$ for $i = 1, 2$.

At times we will refer to bounds on specific derivatives. For this we use the following notation: $L_{w} f$ refers to an upper bound on $\| \frac{\partial f}{\partial w} \|$, likewise for $L_{x} e, L_{x^2} e$, etc.

4.2.2 Optimization Criteria

Here we formulate a condition which the optimization system (4.21) should satisfy, so that descent of the objective at each iteration and overall convergence of the procedure can be guaranteed. This is a gradient based procedure and we aim to apply the result Proposition 3.3.3 on gradient descent with errors. According to Proposition 3.3.3, it suffices to show that the sequence of $h(t)$ defined by $h(t) = h(z(t+1), w(t)) = (\frac{\partial f}{\partial w}(x(t+1), w(t)))^T y(t+1)$ used in the parameter updates can be made to satisfy the condition

$$
\| h(t) - \frac{\partial (g \circ x^*)}{\partial w}(w(t)) \| \leq \alpha \| h(t) \|_W
$$

(4.22)
by a suitable choice of \( \alpha, \epsilon \) and initial conditions. The true gradient occurs at \( h(z^*(t), w(t)) \) where \( z^*(t) \) is the fixed point of \( T(\cdot, w(t)) \). If one defines a vector norm on \( Z = X \times \mathbb{R}^n \) and shows that \( h \) is an \( L_z h \)-Lipschitz function of \( z = (x, y) \), for some \( L_z h \), then a sufficient condition for inequality (4.22) is

\[
\|z(t + 1) - z^*(t)\|_Z \leq \frac{\alpha}{L_z h} \|h(z(t + 1), w(t))\|_W. \tag{4.23}
\]

We will show that this property can be maintained over each iteration by a suitable choice of \( \epsilon, \alpha \), and corresponding conditions on the initial point \( (x(t), y(t)) \).

### 4.2.3 Time-scale Selection

First we formalize that the adjoint system \( y \) inherits the stability of the forward system \( x \). This is a well known fact in sensitivity analysis. We present here perhaps a novel way of quantifying it. It is shown that the joint dynamics, consisting of the forward and adjoint systems, is contracting on a subspace of \( X \times \mathbb{R}^n \), in a norm \( \| \cdot \|_Z \) defined by bounds on the various derivatives of \( f \) and \( e \). In the next proposition the above is applied to the function \( T \) defined in (4.20) describing the joint dynamics. The closed ball of radius \( r \) centered at the origin in \( \mathbb{R}^n \) is denoted \( B_n(r) \).

**Proposition 4.2.4.** Let \( f, e \) satisfy (4.2.1, 4.2.2), and (4.2.3). Let \( p_1, p_2 \) be any positive numbers such that \( \beta + \frac{p_2}{p_1} k < 1 \) where \( k = \frac{(L_{z} f)(L_{z} e)}{1-\beta} + L_{z} e \). Then the map \( T \) defined by Eqn. 4.20 is a contraction mapping with rate \( \beta_T \) in the norm \( \| \cdot \|_Z \) on the set \( Z \) where \( \beta_T = \beta + \frac{p_2}{p_1} k \); \( \|(x, y)\|_Z = p_1 \|x\|_X + p_2 \|y\|_X \), and \( Z = X \times B_n(\frac{L_{z} e}{1-\beta}) \).

**Proof.** Set \( Y = B_n(\frac{L_{z} e}{1-\beta}) \) and let \( g : X \times Y \to Y \) be the function \( g(x, y) = \frac{\partial f}{\partial x}(x, w) y + \frac{\partial e}{\partial x}(x) \). By the assumption on \( f \) and \( e \) it is clear that \( g(Z) \subseteq Y \), so that \( g \) is well-defined. Additionally, it is evident that \( \| \frac{\partial g}{\partial y}(x, y) \| = \| \frac{\partial f}{\partial x}(x, y) \| \leq \beta \). Next, whenever \( (x, y) \in Z \) we
have \( \frac{\partial g}{\partial x}(x, y) \) \( \leq (L_{x^2}f) \frac{L_{x^2}f}{1-\beta} + L_{x^2}e \). These two inequalities imply \( \| g(x_1, y_1) - g(x_2, y_2) \| \leq \beta \| y_1 - y_2 \| + (L_xg)\| x_1 - x_2 \| \), for \( L_xg = \frac{(L_{x^2}f)(L_{x^2}f)}{1-\beta} + L_{x^2}e \). Then apply Proposition 3.2.1 for the pair \( (f, g) \) to obtain the result.

This next result regards autonomous perturbations of contractions. It assumes a discrete time system consisting of two components: a contracting system \( x \) and a “parameter” system \( w \). The system alternates between iterating the contraction and taking a step of the parameter process. For simplicity we first assume that multiple contraction steps are taken each time. The result gives conditions so that the distance to equilibrium of the contracting system \( \| z(t + 1) - z(t) \| \) remains bounded by the disturbance to the parameter controlling the fixed-point, \( \| w(t + 1) - w(t) \| \). It uses a bound on the Lipschitz constant of the fixed-point of a contraction. One such bound is \( K = \frac{L_wT}{1-\beta_T} \) where \( \beta_T \) is the Lipschitz constant of \( T \) in \( z \) and \( L_wT \) is the Lipschitz constant of \( T \) in \( w \).

**Proposition 4.2.5.** Let \( z_0, w_0 \) be given and consider the sequence

\[
\begin{align*}
z(t + 1) &= T^m(z(t), w(t)) \\
w(t + 1) &= w(t) + \epsilon h(z(t + 1), w(t))
\end{align*}
\]

where \( T \) has Lipschitz constants \( \beta_T < 1 \) in \( x \) and \( h \) has Lipschitz constants \( L_z h \) and \( L_w h \) in \( z \) and \( w \) respectively. If \( \| z(0) - z(0)^* \|_Z \leq c \| h(z(0), w(0)) \|_W \) then for all \( t \geq 0 \),

\[
\| z(t + 1) - z(t) \|_Z \leq c \| h(z(t + 1), w(t)) \|_W
\]

whenever \( \epsilon, c \) and \( m \) satisfy the following:

There are some \( \alpha_1, \alpha_2 < 1 \) such that

\[
c = \frac{\alpha_1}{2L_z h}, \quad \epsilon = \frac{\alpha_2(1 - \alpha_1)}{2(L_z h)K + L_w h}
\]
and $m$ is large enough so that

$$\beta^m_T \leq \frac{(1 - \alpha_1)(1 - \alpha_2)c}{c + K\epsilon}$$

where $K$ is a Lipschitz constant for $z^*$.

**Proof.** For $t = 0$, we show that if $\|z(0) - z(0)^*\|_Z \leq c\|h(z(0), w(0))\|_W$ then $\|z(1) - z(0)^*\|_Z \leq c\|h(z(1), w(0))\|_W$. Using the contraction property and assumption on $z(0)$ we have

$$\|z(1) - z(0)^*\|_Z \leq \beta^m_T c\|h(z(0), w(0))\|_W$$

(4.24)

Using the Lipschitz property we get

$$\|h(z(0), w(0))\|_W \leq \|h(z(1), w(0))\|_W + (L_z h)\|z(0) - z(1)\|_Z$$

(4.25)

The distance between the iterates $z_0$ and $z_1$ satisfies

$$\|z(0) - z(1)\|_Z \leq \|z(0) - z(0)^*\|_Z + \|z(0)^* - z(1)\|_Z$$

$$\leq (1 + \beta^m_T)c\|h(z(0), w(0))\|_W$$

$$\leq 2c\|h(z(0), w(0))\|_W$$

Combining this with (4.25) we have $\|h(z(0), w(0))\|_W \leq \frac{\|h(z(1), w(0))\|_W}{1 - 2(L_z h)c}$. Then applying (4.24),

$$\|z(1) - z(0)^*\|_Z \leq \frac{\beta^m_T c}{1 - 2(L_z h)c}\|h(z(1), w(0))\|_W \leq c\|h(z(1), w(0))\|_W$$

The last inequality follows from the fact that $1 - 2(L_z h)c = 1 - \alpha_1$, and that

$$\frac{\beta^m_T}{1 - \alpha_1} \leq (1 - \alpha_2)\frac{c}{c + K\epsilon} < 1$$
This concludes the case of $t = 0$.

For $t \geq 1$, assume $\|z(t) - z^*(t - 1)\|_Z \leq c\|h(z(t), w(t - 1))\|_W$. Using the contraction property we obtain

$$\|z(t + 1) - z^*(t)\|_Z \leq \beta_T^n \|z(t) - z^*(t)\|_Z. \quad (4.26)$$

By the assumption on $z(t)$ and the Lipschitz property of $z^*$,

$$\|z(t) - z^*(t)\|_Z \leq \|z(t) - z^*(t - 1)\|_Z + \|z^*(t - 1) - z^*(t)\|_Z \quad (4.27)$$

$$\leq (c + K\epsilon)\|h(z(t), w(t - 1))\|_W. \quad (4.28)$$

Applying both Lipschitz properties of $h$ one obtains

$$\|h(z(t), w(t - 1))\|_W \leq \|h(z(t + 1), w(t))\|_W$$

$$+ (Lzh)\|z(t) - z(t + 1)\|_Z \quad (4.29)$$

$$+ (Lwh)\|h(z(t), w(t - 1))\|_W. \quad (4.30)$$

For $\|z(t) - z(t + 1)\|_Z$ we have, as in the base case,

$$\|z(t) - z(t + 1)\|_Z \leq 2\|z^*(t) - z(t)\|_Z. \quad (4.32)$$

Combining (4.32) with (4.28) and (4.31) we have

$$\|h(z(t), w(t - 1))\|_W \leq \frac{\|h(z(t + 1), w(t))\|_W}{1 - 2(Lzh)(c + K\epsilon) - (Lwh)\epsilon}. \quad (4.33)$$
Applying (4.26), (4.28) and (4.33),
\[
\|z(t + 1) - z^*(t)\|_Z \leq \frac{\beta_T^m(c + K\epsilon)\|h(z(t + 1), w(t))\|_W}{1 - 2(L_z h)(c + K\epsilon) - (L_w h)\epsilon} \leq \frac{\beta_T^m(c + K\epsilon)\|h(z(t + 1), w(t))\|_W}{(1 - \alpha_1)(1 - \alpha_2)} \leq c\|h(z(t + 1), w(t))\|_W.
\]

The last two steps follow by the assumption on \(c, \epsilon\) and \(m\). \(\square\)

An inspection of the requirements on \(\epsilon\) and \(c\) shows that if they are chosen to be small enough, then only one step \((m = 1)\) of the contraction is required to maintain the invariant.

**Corollary 4.2.6.** If \(\epsilon\) and \(c\) are defined as in Proposition 4.2.5 then a single step \((m = 1)\) suffices to maintain the invariant if \(\alpha_1, \alpha_2\) satisfy the additional constraints

\[
\alpha_1 < 1 - \beta_T, \quad \alpha_2 \leq \frac{\alpha_1(1 - \alpha_1 - \beta_T)}{\beta_T + \alpha_1(1 - \alpha_1 - \beta_T)}.
\]

**Proof.** Based on the proof of Proposition 4.2.5, we will show \(\frac{\beta_T(c + K\epsilon)}{(1 - \alpha_1)(1 - \alpha_2)} \leq c\). Note that

\[
c + K\epsilon \leq \frac{\alpha_1 + \alpha_2(1 - \alpha_1)}{2L_z h}.
\]

Therefore it suffices that

\[
\beta_T \frac{\alpha_1 + \alpha_2(1 - \alpha_1)}{(1 - \alpha_1)(1 - \alpha_2)} \leq \alpha_1.
\]

After some manipulations, this is seen to be equivalent to

\[
\alpha_2 \leq \frac{\alpha_1(1 - \alpha_1 - \beta_T)}{\beta_T + \alpha_1(1 - \alpha_1 - \beta_T)}.
\]

Choose \(\alpha_1 < 1 - \beta_T\) to avoid the trivial constraint \(\alpha_2 = 0\). \(\square\)
We collect these results into the main statement about the algorithm defined by equations (4.21). The above result is combined with the result about approximate gradient descent to get initial conditions and a step-size that guarantee gradient convergence.

**Theorem 4.2.7.** Under the assumptions (4.2.1, 4.2.2), and (4.2.3) there is some choice of constants \((c,\epsilon)\) and a norm \(\| \cdot \|_Z\) so that if \(z(0) = (x(0),y(0))\) satisfies \(\|z(0) - z^*(0)\|_Z \leq c\|h(z(0),w(0))\|_W\) then the sequence of \(w(t)\) generated by (4.21), starting from the point \((x(0),y(0),w(0))\) has the property that \((e \circ x^*)(w(t))\) converges and \(\frac{\partial}{\partial w}(e \circ x^*)(w(t)) \to 0\)

**Proof.** Let \(\| \cdot \|_Z\) be a norm on \(X \times \mathbb{R}^n\) satisfying the conditions of Proposition 4.2.4.

By assumption on the derivatives of \(f\) and \(e\), the function \(h: Z \times W \to W\) where \(h((x,y),w) = \frac{\partial f}{\partial w}(x,w)^T y\) is a Lipschitz function of \(z\), with some Lipschitz constant \(L_z h\). Additionally, by those assumptions the function \(z^*\), which gives the fixed-point of \(T(\cdot,w)\), is also a Lipschitz function. Since \(\frac{\partial}{\partial w}(e \circ x^*)(w) = h(z^*(w))\), we can conclude that \(e \circ x^*\) has a Lipschitz gradient.

According to the gradient descent result Proposition 3.3.3, and the ensuing remarks, for all \(\alpha < \frac{1}{2}\) there is a step-size \(\epsilon(\alpha)\) so that, if the \(\{z(t)\}\) satisfy \(\|z(t+1) - z(t)^*\| \leq \frac{\alpha}{L_z h}\|h(t)\|\), then convergence will be guaranteed when \(\epsilon \leq \epsilon(\alpha)\).

According to Corollary 4.2.6, there is a pair \(c,\epsilon'\), with \(c < \frac{1-\beta}{2L_z h}\) so that \(\|z(t+1) - z^*(t)\| \leq c\|h(z(t+1),w(t))\|\) for all \(t\), whenever \(\|z(0) - z^*(0)\| \leq c\|h(z(0),w(0))\|\), and a step-size \(\epsilon \leq \epsilon'\) is used.

Therefore it suffices to use the pair \((c, \min\{\epsilon', cL_z h\})\)

\(\square\)

### 4.2.4 Step-size and Initial Condition Constraints

In this section we sketch out how to compute a sufficient choice of \(\epsilon\), the norm \(\|(x,y)\|_Z = p_1\|x\| + p_2\|y\|\), and \(c\), based on the given bounds on derivatives of \(f\) and \(e\). Assume the
following:

\[
\begin{align*}
&\| \frac{\partial f}{\partial x} \| \leq \beta, \| \frac{\partial f}{\partial w} \| \leq L_w f \\
&\| \frac{\partial f^2}{\partial x^2} \| \leq L_{x^2} f, \| \frac{\partial f^2}{\partial w^2} \| \leq L_{w^2} f, \| \frac{\partial f^2}{\partial x \partial w} \| \leq L_{x,w} f \\
&\| \frac{\partial e}{\partial x} \| \leq L_{x,e}, \| \frac{\partial e}{\partial x^2} \| \leq L_{x^2} e
\end{align*}
\]

First, \( p_1, p_2 \) and a \( \beta_T \) may be computed according to Proposition 4.2.4:

\[
\begin{align*}
&\text{Define } M_y = \frac{L_{x,e} - \beta}{1 - \beta}.
\\
&\text{Define } L_x T_y = (L_{x^2} f) M_y + (L_{x^2} e)
\\
&\text{Choose } p_1, p_2 \text{ be such that } \beta + \frac{p_2}{p_1} (L_x T_y) < 1
\\
&\text{Define } \|(x, y)\|_Z = p_1 \|x\| + p_2 \|y\|.
\\
&\text{Define } \beta_T = \beta + \frac{p_2}{p_1} (L_x T_y).
\end{align*}
\]

The norm \( \| \cdot \|_Z \) just defined should be used to verify the initial condition.

To compute \( \epsilon, c \) we need Lipschitz constants \( L_w T, L_w h, L_z h \) and \( L_w T^* \). The following choices suffice, and can be verified using the definitions of \( \| \cdot \|_Z \) and the assumptions on \( f \) and \( e \):

\[
\begin{align*}
&\text{Define } L_w h = (L_{w^2} f) M_y
\\
&\text{Define } L_z h = \max\left\{ \frac{(L_{x,w} f) M_y}{p_1}, \frac{L_{w} f}{p_2} \right\}
\\
&\text{Define } L_w T = p_1 (L_w f) + p_2 (L_{x,w} f) M_y.
\\
&\text{Define } L_w T^* = \frac{L_w T}{1 - \beta_T}
\end{align*}
\]

Now we can compute \( c, \epsilon \); according to the proof of Theorem 4.2.7, these should satisfy both Proposition 4.2.5 and the requirements of gradient descent.
Figure 4.2: A ring network was used for experiments. The network activity proceeds along the solid arrows while the adjoint system flows in the opposite direction.

- Choose $\alpha_1 < 1 - \beta_T$.

- Define $\alpha_2 = \frac{\alpha_1(1-\alpha_1-\beta_T)}{\beta_T+\alpha_1(1-\alpha_1-\beta_T)}$.

- Define $c = \frac{\alpha_1}{2(L_z h)}$.

- Define $\epsilon_{PC} = \frac{\alpha_2(1-\alpha_1)}{2(L_z h)(L_w T^*) + L_w h}$.

- Define $L_w^3 E = (L_z h)(L_w T^*)$.

- Define the function $\epsilon_{GD}(\alpha) = \frac{2(1-\alpha)}{(L_w^3 E)(1+\alpha)^2}$.

- Use the above $c$, and take $\epsilon = \min\{\epsilon_{GD}(\alpha_1/2), \epsilon_{PC}\}$.

4.3 Numerical Experiments

We applied the above procedure to optimize the fixed-point of a network of nonlinear units.
4.3.1 Synthetic Data

In this example with the synthetic data, the units form a ring, each unit having one real valued parameter $b_i$. Initially the network has a stable point $x^*(b)$ and the goal is to adapt the parameter vector $b$ so that a target vector $t$ becomes the stable point.

**Network dynamics**

The network was made up of $n = 30$ units with the dynamics of the $i$’th given by

$$f_i(x, b) = v_i S(x_{i-1}) + b_i$$  \hspace{1cm} (4.34)

Here $S : \mathbb{R} \to \mathbb{R}$ was taken to be $S = \sin$, and with the convention that $x_{i-1} = x_n$ for $i = 1$. The weights $v \in \mathbb{R}^{30}$ are fixed during optimization, and were chosen as follows. Each $v_i$, which is the weight on the connection from $i$ to it’s predecessor, was chosen uniformly at random from the set $\{-\beta, \beta\}$ where $\beta = \frac{3}{4}$.

**Contraction property**

There is a simple condition for the network dynamics (4.34) to be contracting. If $S$ has bounded derivative, $|S'| \leq M$, then $f$ is a contraction mapping in the 2-norm if $\|v\|_{\infty} < \frac{1}{M}$. In the case of $S = \sin$, it suffices that $\|v\|_{\infty} < 1$. Hence with $\beta = \frac{3}{4}$ as above, $f$ is a contraction. Importantly, this sufficient condition is independent of $b$, the parameter that is being optimized.
Figure 4.3: Error trajectories for different step-sizes $\epsilon$ using the one-step method. Each row corresponds to a different choice of step-size and the column indicates the iteration. The color indicates the approximate error at that step of optimization.

**Optimization problem**

A target vector $l \in \mathbb{R}^{30}$ was generated randomly. Each component $l_i$ was sampled from a normal distribution with mean 0 and unit variance. The error function $e$ was

$$e(x) = \sum_{i=1}^{30} a(x_i - l_i)$$

where $a$ is the smooth absolute-value-like function $a(u) = \log\left(\frac{1}{2}(\exp(u) + \exp(-u))\right)$. Therefore the optimization problem is

$$\min_b e(x^*(b))$$

where $x^*(b)$ is the unique solution for $x^*(b) = f(x^*(b), b)$. 
Figure 4.4: Error trajectories for different step-sizes $\epsilon$, when very accurate gradient estimates are used. The procedure is more stable at large step-sizes but this requires more time spent approximating gradients.

**Step-size**

For this problem the computation of the constants is somewhat simpler than the general case; one can use for the Lipschitz constants

$$L_x f = \beta, \quad L_b f = 1,$$

$$L_{b^2} f = 0, \quad L_{x,b} f = 0, \quad L_{x^2} f = \beta,$$

$$L_x e = 30, \quad L_{x^2} e = 1.$$

Given this, we can use the results of Section 4.2.4 to compute a $\| \cdot \|_{Z,c}$, and $\epsilon$ that enable the application of Theorem 4.2.7. An interesting future work would be to find optimal choices of
\( p_1, p_2 \) and \( \alpha_1 \), which are variables in the recipe of Section 4.2.4. In this work, our theorems only provide sufficient values.

Using the choice \( p_1 = 1, p_2 = 0.2 \ast (1.0 - \beta)/(LT_y) \approx 0.0005 \) yields a \( \beta_T \) of 0.8; Then \( \alpha_1 \) was chosen to be \( 0.5 \ast (1.0 - \beta_T) \). The final value of the step-size \( \epsilon \) is on the order of \( 10^{-6} \) and \( c \) is near \( 10^{-5} \). Not surprisingly, the numerical experiments showed that this choice of step-size was very conservative.

**Initialization**

The initial parameter \( b \) was selected randomly as well; \( b_i \) was sampled from a normal distribution with mean 0 and variance 1. The initial point \((x(0), y(0))\) was obtained as \((x(0), y(0)) = T^{100}(0, 0)\). That is, \( T \) was iterated 100 times starting from \((0, 0)\) before optimization was started.

**Algorithm**

After obtaining the initial point \((x(0), y(0))\), the optimization process (the dynamics on the \( b(t) \)) begins. For completeness, the algorithm takes the following form with this choice of \( f \) and \( e \):

\[
x^i(t+1) = v_iS(x^{i-1}(t)) + b^i(t) \tag{4.35a}
\]

\[
y^i(t+1) = v_{i+1}S'(x^i(t))y^{i+1}(t) + \tanh(x^i(t) - l^i) \tag{4.35b}
\]

\[
b^i(t+1) = b^i(t) - \epsilon y^i(t + 1) \tag{4.35c}
\]

**Results**

We ran the algorithm several times with varying values of \( \epsilon \). The results exhibit the usual behavior of gradient descent. When the step-size is very small, the objective function is decreasing but very slowly. Then comes an interval where the decrease happens rapidly
and the function values seem to converge. When the step size gets too large, the process is characterized by a rapid initial decrease followed by oscillations.

With the initial data \( \{(x(0), y(0), w(0))\} \) we performed optimization, as specified by (4.35), with values of \( \epsilon = 0.005, 0.01, 0.015, \ldots, 0.2 \). The (approximate) trajectory of the error \( e(x^*(t)) \) as optimization progressed was recorded and is plotted Figure 4.3. For instance, the first row in that figure shows how the error decreases as optimization progresses, when a step-size of \( \epsilon = 0.005 \) is used. In subsequent rows the step-size is increased by a constant amount \( \Delta \epsilon = 0.005 \) in each experiment, while the underlying problem and initial state remains the same. Near a step-size \( \epsilon = 0.10 \) the procedure begins to lose stability. When using such large step-sizes, optimization starts out working well but eventually exhibits oscillations. It seems to be reflective of the need to use more accurate gradients near a local minimum, a requirement expressed in Proposition 3.3.3 on gradient descent.

For comparison, we ran the same algorithm but where accurate gradients are used instead of the one-step update procedure. This was achieved by iterating \( T \) 10 times between each parameter update, instead of once. The results of this are plotted in Figure 4.4. When using accurate gradients, we did not notice oscillations in the function values during optimization until the step-sizes got significantly larger than \( \epsilon = 0.2 \).

### 4.3.2 MNIST Experiment

In this section we apply the attractor networks on the benchmark MNIST handwritten digit dataset. This dataset contains 60,000 training samples and 10,000 samples which are reserved for testing. Each sample is a 28x28 grey scale image and a label for that image, which is a number from 0 to 9. The classification task is to infer what digit is present in the image.

The attractor network was structured as follows. There are 28 \( \times \) 28 input nodes and 10
output nodes. The equation for the network update was as follows: For \( i = 1, 2, \ldots, 784 + 10, \)

\[
x_i(t + 1) = \sigma \left( \sum_{j \in N(i)} w_{i,j} x_j + b_i + v_i \right)
\]

with the convention that \( v_i = 0 \) if \( i \geq 784 \). Hence the first 784 units are directly connected to the input. The set of neighbors of node \( i \) is denoted \( N(i) \). This determines the network architecture. For instance, when \( N(i) = \{1, 2, \ldots, 794\} \) then the network is fully connected.

In experiments we compared several different architectures.

The optimization problem was to minimize the sum of squared errors. Setting \( e(x, y) = \sum_{i=1}^{10} (x_{784+i} - y_i)^2 \), then \( J \) is

\[
J(\theta) = \sum_{i=1}^{N} e(x^*(\theta, v_i), \ell_i)
\]

where \( v_i \) is the image for sample \( i \) and \( \ell_i \) is the label vector for sample \( i \). In the experiments we only used a small portion of the total data set (\( N = 5,000 \)).

**Arbitrary connectivity**

In the first experiments there were no constraints on the connectivity. We performed tests using networks of varying levels of sparsity. A sequence of 10 networks was trained, where in network \( n \) each edge was present with probability \( 0.1*n \) for \( n = 1, 2, \ldots 10 \). For two networks \( n, m \) with \( n < m \), the network \( m \) contained at least the set of edges present in network \( n \). The weights are initialized uniformly at random within \((-0.01, 0.01)\), while the biases were
initialized in $(-0.1, 0.1)$. In each case we eventually detected a loss of global stability during training, at which point training was stopped. If, for a given training sample, the joint system $(x, y)$ did not meet the convergence criteria within 1000 steps we declared that a bifurcation had occurred. This event tended to occur later in sparser networks. After detection the weight matrix from the end of the previous epoch is returned as the output of the training procedure. The idea of using a fixed, but randomly initialized, connectivity is inspired by Reservoir Networks (58). See Table 4.4 for a summary of the results. Networks with medium connectivity gave the highest accuracy.

<table>
<thead>
<tr>
<th>Connectivity</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.896</td>
<td>0.931</td>
<td>0.933</td>
<td>0.949</td>
<td>0.948</td>
<td>0.945</td>
<td>0.925</td>
<td>0.94</td>
<td>0.937</td>
<td>0.938</td>
</tr>
<tr>
<td>Epochs</td>
<td>14</td>
<td>29</td>
<td>26</td>
<td>25</td>
<td>21</td>
<td>17</td>
<td>15</td>
<td>14</td>
<td>15</td>
<td>12</td>
</tr>
</tbody>
</table>

The different network architectures we experimented with are shown in Figure 4.6.

**Restricted Feedback**

In further experiments, it was noted that certain types of restricted feedback did not lead to the stability issues. In these experiments, there are connections from the input nodes to the output nodes, connections among the output nodes, and connections from the output
CHAPTER 4. DETERMINISTIC ATTRACTOR NETWORKS

nodes to the input nodes. There are no connections among the inputs. Since there are 784 input nodes, this means 784 × 784 fewer parameters. Whereas the full-feedback networks lost stability after at most a few dozen epochs, the restricted feedback networks seemed fine up to 120 epochs, which was the limit in our experiments.

Within these restricted feedback networks, we experimented with varying levels of connectivity. For each of the levels of connectivity (0.1, 0.3, 0.5, 0.7, 0.9) we compared the attractor networks with a feed forward network. We find that as a rule the attractor network performed better, but there was only a marginal difference. The results are shown in Table 4.4

<table>
<thead>
<tr>
<th>Connectivity</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error (Attractor)</td>
<td>0.468</td>
<td>0.317</td>
<td>0.275</td>
<td>0.249</td>
<td>0.237</td>
</tr>
<tr>
<td>Error (Feed-forward)</td>
<td>0.481</td>
<td>0.328</td>
<td>0.293</td>
<td>0.268</td>
<td>0.26</td>
</tr>
<tr>
<td>Accuracy (Attractor)</td>
<td>72.8</td>
<td>86.2</td>
<td>88.8</td>
<td>90.3</td>
<td>90.5</td>
</tr>
<tr>
<td>Accuracy (Feed-forward)</td>
<td>72.0</td>
<td>85.7</td>
<td>88.3</td>
<td>89.6</td>
<td>90.2</td>
</tr>
</tbody>
</table>

4.4 Discussion

In this work we investigated conditions under which a distributed optimization procedure based on adjoint sensitivity analysis satisfies a convergence property. The requirements we gave, which concern the step size and initial condition, are based on bounds on the various derivatives of the system $f$ and error function $e$. Furthermore, the conditions can be satisfied whenever these derivatives are bounded and when the underlying system is uniformly contracting.

Currently, the update pattern (see Figure 4.1) leaves a strong mark in the main part of the analysis (Proposition 4.2.5). However there are several other reasonable update patterns one could use. An argument that is somehow independent of this choice may be more insightful.
Secondly, there is also the question of how to optimally select the parameters of the procedure, namely the step-size $\epsilon$. As shown above, using bounds on the derivatives of $f$ and $e$ one can calculate a sufficient $\epsilon$ in terms of these quantities, in a tedious but straightforward way. We did not investigate what is the optimal way to do so, but it may be crucial in complex problems in order to reach practicality.

Future work may consider the extension of these algorithms to other domains. As long as dynamic system has a contraction property and is sufficiently smooth, these results should be applicable. In particular, certain chemical reaction networks may be susceptible to optimization (29, 95).
Chapter 5

Stochastic Attractor Networks

In this chapter we extend the methodology of Chapter 4 to stochastic systems. We show that the forward method of gradient estimation is also valid in the stochastic setting. We also introduce some novel ergodicity criteria that generalize the contraction conditions used for deterministic systems. The results of this chapter were reported in (24).

5.1 Gradient Estimation for Stationary Markov Chains

Stationary gradient estimation starts with a Markov kernel $P$ that depends on a parameter $\theta$. Given a cost function $e$ defined on the states of the Markov chain, and assuming ergodicity of the process, the problem is to estimate the derivative of the average cost, at stationarity, with respect to the parameter $\theta$. That is, setting $\pi_\theta$ to the stationary measure of $P_\theta$, the problem is to estimate

$$\frac{\partial}{\partial \theta} \int_X e(x) \, d\pi_\theta(x)$$
In this work we investigate an approach to this problem based on forward sensitivity analysis, 
an algorithm used for estimating sensitivities in deterministic systems. Let $P_\theta$ take the form
\[
(P_\theta e)(x) = \int_{\Xi} e(f(x, \xi, \theta)) \, d\eta(\xi)
\]
for a probability space $(\Xi, \Sigma, \eta)$ and a function $f : X \times \Xi \times \Theta \to X$. We find that if certain 
contraction and differentiability conditions are satisfied, then
\[
\frac{\partial}{\partial \theta} \int_X e(x) \, d\pi_\theta(x) = \int_{X \times M} \frac{\partial e}{\partial x}(x) m \, d\gamma_\theta(x, m) \quad (5.1)
\]
where $\gamma_\theta$ is the stationary measure on $X \times M$ of the recursion
\[
x(t + 1) = f(x(t), \xi(t + 1), \theta) \quad (5.2a)
\]
\[
m(t + 1) = \frac{\partial f}{\partial x}(x(t), \xi(t + 1), \theta) m(t) + \frac{\partial f}{\partial \theta}(x(t), \xi(t + 1), \theta) \quad (5.2b)
\]
where the $\xi(t)$ form an i.i.d. sequence of $\eta$-distributed random variables. There are several 
challenges associated with this. The first is to extend the contraction framework to include 
probabilistically interesting systems. The contraction framework should enable us to show 
convergence of the forward sensitivity process (5.2a, 5.2b) as well as the underlying process. 
The second challenge is to show correctness of the procedure.

A simple version of our main result (Theorem 5.4.5) can be stated as follows.

**Theorem 5.1.1.** If the function $f$ and the probability space $(\Xi, \Sigma, \eta)$ are such that

i. $\int_{\Xi} \|f(x, \xi, \theta)\|^2 \, d\eta(\xi) < \infty$ for all $(x, \theta) \in X \times \Theta$,

ii. $(x, \theta) \mapsto f(x, \xi, \theta)$ is a $C^2$ function for each $\xi \in \Xi$,

iii. For $0 < i + j \leq 2$, the functions $L_{X, \Theta}(x, \theta) = \int_{\Xi} \|\frac{\partial^{i+j} f}{\partial x^i \partial \theta^j}(x, \xi, \theta)\|^2 \, d\eta(\xi)$ are continuous 
and bounded on $X \times \Theta$, and in particular, $\sup_{(x, \theta)} L_X(x, \theta) < 1$,
Then the forward sensitivity process (5.2a, 5.2b) converges weakly to a stationary measure \( \gamma_{\theta} \), and equation (5.1) holds for those \( e: X \to \mathbb{R} \) that are \( C^2 \) with \( \| \frac{\partial e}{\partial x} \|_{\infty} + \| \frac{\partial^2 e}{\partial x^2} \|_{\infty} < \infty \).

### 5.2 Overview of Main Results

First the contraction framework is introduced. Second, criteria for differentiability of the stationary costs are presented. The third component is a set of conditions on the function \( f \) that let us apply the abstract result on stationary differentiability, establish convergence of the sensitivity process \( (x(t), m(t)) \), and allow us to show that Equation (5.1) holds. Finally, we consider an application to neural networks.

**Contraction framework.**

Given a matrix valued function \( A(x) \) and a norm \( \| \cdot \| \) on \( \mathbb{R}^{nx} \), we consider the following ergodicity condition

\[
\sup_{x \in X} \left( \int_{\Xi} \| A(f(x, \xi)) \frac{\partial f}{\partial x}(x, \xi) A(x)^{-1} \|_p d\eta(\xi) \right)^{1/p} < 1 \tag{5.3}
\]

The object inside the norm is the composition of the three linear maps \( A(f(x, \xi)), \frac{\partial f}{\partial x}(x, \xi) \) and \( A(x)^{-1} \), and the norm in this inequality is that induced by \( \| \cdot \| \) on the space of linear maps \( L(\mathbb{R}^{nx}, \mathbb{R}^{nx}) \). Formally, the map \( (x, u) \mapsto \| A(x)u \| \) defines a Finsler structure on the space \( X \), which induces a metric \( d_A \) on \( X \). This is extended to a metric on probability measures using the Wasserstein distance \( d_{p,A} \). The condition (5.3) implies the Markov kernel \( P \) is a contraction mapping for this distance. This is developed in Section 5.5. In Section 5.5.1 we consider interconnections of contracting systems, obtaining sufficient conditions for both feedback and hierarchical combinations of contracting systems to again be contracting. This is useful to analyze the forward sensitivity process, as it exhibits a hierarchical structure.
Stationary differentiability.

In Section 5.6 we give abstract conditions for stationary differentiability, using a variant of the proof technique in (40). The equation

\[ l = lP_\theta + \pi_\theta \frac{\partial}{\partial \theta} P_\theta \]  

(5.4)

is shown to have a unique solution in the variable \( l \), and this \( l \) is shown to evaluate the stationary derivatives, meaning \( l(e) = \frac{\partial}{\partial \theta} \int_X e(x) \, d\pi_\theta(x) \). While similar formulas have been recovered by other authors (see (96, 40, 67)) we rederive this using assumptions that are relevant for the smooth systems we are interested in.

Gradient estimation.

To study the forward sensitivity process we define an appropriate metric on the space \( X \times M \) and prove a pointwise contraction inequality for the joint system (5.2a, 5.2b) in this distance. This is used together with a Lyapunov function for the joint system to establish ergodicity of the sensitivity process. This is done in Section 5.8. It is then established that the functional \( e \mapsto \int_{X \times M} \frac{\partial e}{\partial x}(x) m \, d\gamma_\theta(x, m) \) verifies Equation (5.4). We conclude that Equation (5.1) holds for the class of cost functions.

Before formally stating the assumptions and main results, we introduce some notation and conventions. For a function \( f : X \to \mathbb{R}^n \) where \( X \subseteq \mathbb{R}^m \), we denote by \( \frac{\partial f}{\partial x}(x_0) \) the derivative of \( f \) with respect to \( x \) at the point \( x_0 \), and for a vector \( u \in \mathbb{R}^m \), we denote by \( \frac{\partial f}{\partial x}(x_0)u \) the \( \mathbb{R}^n \)-valued result of applying this linear map to the vector \( u \). The second derivative of \( f \) with respect to \( x \) is \( \frac{\partial^2 f}{\partial x^2} \), and \( \frac{\partial^2 f}{\partial x^2}(x_0)[u, v] \) refers to the \( \mathbb{R}^m \)-valued result of applying this bilinear map to the arguments \( u, v \). A function \( f \) is \( C^1 \) if it is continuously differentiable. The function is \( C^2 \) if it is twice continuously differentiable. Given norms \( \| \cdot \|_X \) and \( \| \cdot \|_Y \) on the space
\( \mathbb{R}^m \) and \( \mathbb{R}^n \), recall that the norm of a linear map \( E : \mathbb{R}^n \to \mathbb{R}^m \) is \( \|E\| = \sup_{\|u\|_X=1} \|Eu\|_Y \).

For a bilinear map \( F \) defined on \( \mathbb{R}^n \times \mathbb{R}^m \) and taking values in a third space with norm \( \|\cdot\|_Z \), the norm is \( \|F\| = \sup_{\|u\|_X=\|v\|_Y=1} \|F[u,v]\|_Z \). Given two linear maps \( E \) and \( F \), their direct sum is the linear map \((E \oplus F)(u,v) = (Eu,Fv)\). For reference here is a summary of notations and definitions of spaces used throughout this chapter.

### 5.3 Notations

- \( \Theta \) - space of parameters,
- \( n_X \) - dimensionality of state space for underlying system,
- \( n_\Theta \) - dimensionality of parameter space,
- \( L(\mathbb{R}^n,\mathbb{R}^m) \) - space of linear maps from \( \mathbb{R}^n \) to \( \mathbb{R}^m \),
- \( M \) - the space \( L(\mathbb{R}^{n_\Theta},\mathbb{R}^{n_X}) \),
- \( \|V\|_{L^p(\nu)} \) - short hand for \( \left( \int_X \|V(x)\|^p \, d\nu \right)^{1/p} \),
- \( \|h\|_V \) - short hand for \( \sup_x |h(x)| \) with \( d_A(x,x_0) \) the metric induced by a Finsler structure,
- \( P(\mathbb{R}^n) \) - Borel probability measures on \( \mathbb{R}^n \),
- \( P_{p,V}(\mathbb{R}^n) \) - measures in \( P(\mathbb{R}^n) \) that such that \( \|V\|_{L^p(\nu)} < \infty \),
- \( d_{A,A} \) - Wasserstein distance on the space \( P_{p,A}(\mathbb{R}^n) \),
- \( \|\cdot\|_{\text{Lip}} \) - Lipschitz constant for a function between metric spaces,
- \((E \oplus F)(u,v) = (Eu,Fv)\).

### 5.4 Assumptions

Assumption 5.4.1. The set \( X \) is a closed, convex subset of \( \mathbb{R}^{n_X} \), and \( \mathbb{R}^{n_X} \) carries a norm \( \|\cdot\|_X \). The function \( A : X \to L(\mathbb{R}^{n_X},\mathbb{R}^{n_X}) \) is continuous, such that each \( A(x) \) is invertible, and \( \sup_{x \in X} \|A(x)^{-1}\|_X < \infty \).

We will require differentiability and integrability of \( f \):

Assumption 5.4.2. For an open set \( \Theta \subseteq \mathbb{R}^{n_\Theta} \), the function \( f : X \times \Xi \times \Theta \to X \) satisfies

i. \( \xi \mapsto d_A(x,f(x,\xi,\theta))^2 \) is \( \eta \)-integrable for all \((x,\theta) \in X \times \Theta\),
ii. \((x, \theta) \mapsto f(x, \xi, \theta)\) is twice continuously differentiable \((C^2)\) for each \(\xi \in \Xi\).

We also require some bounds on \(P\) as a function of \(\theta\), formulated with the help of a function \(B(x)\) taking values in the invertible \(n_\theta \times n_\theta\) matrices.

**Assumption 5.4.3.** \(\mathbb{R}^{n_\theta}\) has a norm \(\| \cdot \|_\Theta\). The function \(B : X \to L(\mathbb{R}^{n_\theta}, \mathbb{R}^{n_\theta})\) takes values in the invertible linear maps, and \(x \mapsto \|B(x)\|_\Theta\) is a \(d_A\)-Lipschitz function.

For an example when Assumption 5.4.3 is satisfied, consider the following. Let \(g : X \to \mathbb{R}_{\geq 0}\) be a function that is Lipschitz continuous with respect to the underlying norm \(\| \cdot \|_X\) on \(X\). Then use \(A(x) = \exp(g(x))I_{n_X}\) and \(B(x) = \exp(g(x))I_{n_\theta}\), where \(I_n\) is the \(n \times n\) identity matrix. Of course, the assumption always holds when \(B(x) = I_{n_\theta}\).

The next assumptions relate to the contraction property of \(P\) and the differentiability properties of \(P_\theta e\). Before continuing we define several norms derived from \(A\) and \(B\). At each \(x \in X\) the matrix \(A(x)\) defines a norm \(\| \cdot \|_{A(x)}\) on \(\mathbb{R}^{n_X}\) by \(\|u\|_{A(x)} = \|A(x)u\|\). and \(B(x)\) defines a norm on \(\mathbb{R}^{n_\theta}\) by \(\|v\|_{B(x)} = \|B(x)v\|\). These extend to norms on the various linear spaces. For example, if \(l \in L(\mathbb{R}^{n_X}, \mathbb{R})\) then \(\|l\|_{A(x)} = \|lA(x)^{-1}\|\). For a bilinear map \(Q \in L(\mathbb{R}^{n_X}, \mathbb{R}^{n_X}; \mathbb{R})\) we can write \(\|Q\|_{A(x), A(x)} = \|Q(A(x)^{-1} \oplus A(x)^{-1})\|.\) Further extend this to functions from \(X\) into the linear spaces by taking supremums, e.g. if \(h : X \to L(\mathbb{R}^{n_\theta}, \mathbb{R})\) then \(\|h\|_B = \sup_x \|h(x)\|_{B(x)}\). For the case of a real-valued \(h : X \to \mathbb{R}\), let \(\|h\|_A = \sup_x \frac{|h(x)|}{1 + d_A(x, x_0)}\), where \(x_0\) is an arbitrary basepoint in \(X\).

We introduce the space of cost functions \(E^2\):

\[
E^2 = \{h : X \to \mathbb{R} \mid h \text{ is } C^2 \text{ and } \|h\|_A + \|\frac{\partial h}{\partial x}\|_A + \|\frac{\partial^2 h}{\partial x^2}\|_{A, A} < \infty\}
\]

On \(E^2\) we put the norm

\[
\|h\|_{E^2} = \|h\|_A + \|\frac{\partial h}{\partial x}\|_A + \|\frac{\partial^2 h}{\partial x^2}\|_{A, A}
\] (5.5)
We consider bounds on the derivatives of $f$ formulated using the following functions:

\[
L_X(x, \theta) = \left( \int_{\Xi} \|A(f(x, \xi, \theta)) \frac{\partial f}{\partial x}(x, \xi, \theta) A(x)^{-1}\|^2 \, d\eta(\xi) \right)^{1/2}
\]

\[
L_\Theta(x, \theta) = \left( \int_{\Xi} \|A(f(x, \xi, \theta)) \frac{\partial f}{\partial \theta}(x, \xi, \theta) B(x)^{-1}\|^2 \, d\eta(\xi) \right)^{1/2}
\]

\[
L_{X^2}(x, \theta) = \int_{\Xi} \|A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x^2}(x, \xi, \theta) (A(x)^{-1} \oplus A(x)^{-1}) \| \, d\eta(\xi)
\]

\[
L_{\Theta^2}(x, \theta) = \int_{\Xi} \|A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial \theta^2}(x, \xi, \theta) (B(x)^{-1} \oplus B(x)^{-1}) \| \, d\eta(\xi)
\]

\[
L_{X, \Theta}(x, \theta) = \int_{\Xi} \|A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x \partial \theta}(x, \xi, \theta) (A(x)^{-1} \oplus B(x)^{-1}) \| \, d\eta(\xi)
\]

**Assumption 5.4.4.** The functions $L_{X^i, \Theta^j}$ satisfy

i. The various functions $L_{X^i, \Theta^j}$ are continuous on $X \times \Theta$,

ii. There is a $K_X \in [0, 1)$ such that $\sup_{(x, \theta) \in X \times \Theta} L_X(x, \theta) \leq K_X$,

iii. For $0 < i + j \leq 2$, there are $K_{X^i, \Theta^j}$ such that $\sup_{(x, \theta) \in X \times \Theta} L_{X^i, \Theta^j}(x, \theta) \leq K_{X^i, \Theta^j}$.

**Theorem 5.4.5.** Let Assumptions 5.4.1, 5.4.2, 5.4.3, and 5.4.4 be satisfied. Let $\theta$ be an arbitrary point of $\Theta$. Then the forward sensitivity process (5.2a, 5.2b) possesses a unique stationary measure $\gamma_\theta$ and for any $e \in \mathcal{E}$ equation (5.1) is valid. Furthermore, if the variables $(x_1, m_1)$ satisfy the integrability condition $E[|d_A(x_0, x_1) + \|A(x_1)m_1||] < \infty$ for an arbitrary basepoint $x_0$, then $E[\frac{\partial e}{\partial x}(x_n)m_n] \to \frac{\partial}{\partial \theta} \int_X e(x) \, d\pi_\theta(x)$ as $n \to \infty$.

**Neural network application.**

In Section 5.9 two examples are considered. The first involves neural networks. In neural networks, a central problem is to compute derivatives of cost functionals with respect to network parameters (weights on the connections between nodes). We are concerned with
long-term average cost problems, a type of problem that is relevant when a network has cycles. The back-propagation algorithm for calculating derivatives (79), originally formulated for a continuous state-space model with a finite horizon objective, is also valid for calculating gradients in long-term average cost problems under contraction assumptions (71). Our contribution addresses the long-term average cost problem for continuous stochastic networks.

The example system consists of a network with weights on connections between units. At each step every node updates its value based on the values of its neighbors, but only a random subset of possible connections are activated, leading to a stochastic process. We find contraction conditions based on a sparsity coefficient, and verify that stochastic forward sensitivity analysis can be used to calculate the derivative of stationary costs. We present a second example to illustrate using a non-trivial metric on the underlying system. We finish with a discussion in Section 5.10.

5.5 Contraction Framework

Let $d_A$ is the metric defined by the Finsler structure $\| \cdot \|_x = \| A(x) \cdot \|$. For a function $e : X \to \mathbb{R}$ we let $\| e \|_{Lip(A)}$ be the Lipschitz constant of a function $e : X \to \mathbb{R}$ with respect to the metric $d_A$. When the metric $d_A$ is clear we will just write $\| e \|_{Lip}$.

For a probability measure $\nu$ and $p \geq 1$ we write $\| V \|_{L^p(\nu)} = (\int_X \| V(x) \|^p \, d\nu(x))^{1/p}$.

Given a Markov kernel $P$, we denote the image of measure $\nu$ under $P$ by $\nu P$. That is, $(\nu P)(A) = \int_X P(x, A) \, d\nu(x)$. For $V : X \to \mathbb{R}_{\geq 1}$, let $\| e \|_V = \sup_{x \in X} \frac{|e(x)|}{V(x)}$. We say that $V : X \to \mathbb{R}_{\geq 1}$ is a $p$-Lyapunov function for $P$ if $V$ has compact sublevel sets and there exists numbers $\beta \in [0, 1), K \geq 0$ so that $(PV^p(x))^{1/p} \leq \beta V(x) + K$ for all $x$. A measure $\nu \in \mathcal{P}(X \times X)$ is a coupling of $\nu_1$ and $\nu_2$ if $\nu(A \times X) = \nu_1(A)$ and $\nu(X \times A) = \nu_2(A)$ for each measurable set $A$. We define $\Gamma(\nu_1, \nu_2)$ to be the set of all couplings of $\nu_1$ and $\nu_2$. 
Let the Markov kernel $P$ have an explicit representation as

$$(Pe)(x) = \int_{\Xi} e(f(x, \xi)) \, d\eta(\xi)$$

(5.6)

for a measurable function $f : X \times \Xi \to X$ and a probability space $(\Xi, \Sigma, \eta)$. In this section we present two separate conditions for the ergodicity of a Markov kernel given in the form (5.6). The first, Proposition 5.5.2, is weaker and is used to show convergence of the forward sensitivity system (consisting of the variables $x_n, m_n$). Proposition 5.5.3 relies on a stronger set of assumptions and is used to show differentiability of the stationary costs. Both results utilize the following pointwise estimate of Proposition 5.5.1.

In this proposition, and throughout the chapter, we consider a differentiable function defined on a closed subset $X$ of Euclidean space. In case $X$ is a strict subset of the space, we assume $f$ is the restriction of a function $\overline{f}$ that is defined and differentiable on an open set $U$ containing $X$. In this way there is no ambiguity in defining the derivative of $f$ at each point of $X$.

**Proposition 5.5.1.** Let $P$ be of the form (5.6) where

i. $x \mapsto f(x, \xi)$ is $C^1$ for each $\xi \in \Xi$,

ii. $\sup_{x \in X} \sup_{u \in \mathbb{R}^n, \|u\|=1} \left( \int_{\Xi} \|A(f(x, \xi))\frac{\partial f}{\partial x}(x, \xi)A^{-1}(x)u\|^p \, d\eta(\xi) \right)^{1/p} \leq \alpha,$

for some $\alpha \geq 0$. Then for any $x_1, x_2 \in X$ we have

$$\left( \int_{\Xi} d_A(f(x_1, \xi), f(x_2, \xi))^p \, d\eta(\xi) \right)^{1/p} \leq \alpha d_A(x_1, x_2).$$

(5.7)

**Proof.** Let $x_1 \neq x_2$ be points of $X$, let $\epsilon > 0$ and let $\gamma : [0, T] \to X$ be a piecewise $C^1$ path from $x_1$ to $x_2$ such that $L(\gamma) \leq d_A(x_1, x_2) + \epsilon$. We further assume that $\gamma$ is parameterized by arc length. For our definition of length this means $\|A(\gamma(t))\gamma'(t)\| = 1$ for all $t$ and that
\[ T = L(\gamma). \] Since \( t \mapsto f(\gamma(t), \xi) \) defines a curve from \( f(x_1, \xi) \) to \( f(x_2, \xi) \) we have

\[
\left( \int_\Xi d_A(f(x_1, \xi), f(x_2, \xi))^p \, d\eta(\xi) \right)^{1/p} \\
\leq \left( \int_\Xi \left( \int_0^T \| A(f(\gamma(t), \xi)) \frac{\partial f}{\partial x}(x, \xi) \gamma'(t) \| \, dt \right)^p \, d\eta(\xi) \right)^{1/p} \\
\leq L(\gamma)^{(p-1)/p} \left( \int_\Xi \int_0^T \| A(f(\gamma(t), \xi)) \frac{\partial f}{\partial x}(x, \xi) \gamma'(t) \|^p \, dt \, d\eta(\xi) \right)^{1/p}
\]

In the first step the definition of length was applied. Then Jensen’s inequality was used together with the fact that \( L(\gamma) = T \). Next, note the integrand in the final expectation is of the form \( (t, \xi) \mapsto g(t, \xi) \) where \( g \) is non-negative, continuous in \( t \) for each \( \xi \), and measurable in \( \xi \) for each \( t \). Then we may interchange the integrals, yielding

\[
= L(\gamma)^{(p-1)/p} \left( \int_0^T \int_\Xi \| A(f(\gamma(t), \xi)) \frac{\partial f}{\partial x}(x, \xi) \gamma'(t) \|^p \, d\eta(\xi) \, dt \right)^{1/p}
\]

Using the identity \( A(\gamma(t))^{-1} A(\gamma(t)) \gamma'(t) = \gamma'(t) \) and the assumption on \( \frac{\partial f}{\partial x} \) we get

\[
\leq L(\gamma)^{(p-1)/p} \left( \int_0^T \alpha^p \| A(\gamma(t)) \gamma'(t) \|^p \, dt \right)^{1/p}
\]

Then since \( \gamma \) is parameterized by arc length,

\[
= L(\gamma)^{(p-1)/p} \alpha L(\gamma)^{1/p} \leq \alpha d_A(x_1, x_2) + \alpha \epsilon
\]

As \( \epsilon > 0 \) was arbitrary, the result follows.

If a tuple \( \{(\Xi, \Sigma, \eta), f, (\| \cdot \|, A)\} \) satisfies the conditions of Proposition 5.5.1 for some \( \alpha < 1 \), we say that a pointwise \( p \)-contraction inequality holds for the process.

Combining this with the assumption that the system carries a Lyapunov function yields the following ergodicity result.
Proposition 5.5.2. Let the assumptions of Proposition 5.5.1 hold for $p \geq 1$ and $\alpha < 1$, and assume there is a $p$-Lyapunov function $V$ for $P$. Then $P$ has a unique invariant measure $\pi \in \mathcal{P}_{p,V}(X)$ and for any $\nu \in \mathcal{P}_{p,V}$, $\|e\|_{Lip} + \|e\|_V \leq 1$.

Proof. The existence of a unique invariant measure $\pi$ is an immediate result of Corollary 4.23 and Theorem 4.25 of (35). To show that $\pi \in \mathcal{P}_{p,V}$, reason as follows. If $V$ is a $p$-Lyapunov function, then $V^p$ is a 1-Lyapunov function (for possibly different values of the constants $\beta$ and $K$). Then apply Proposition 4.24 of (35).

We turn to convergence of the expectations $\nu P^t(e)$ as $t \to \infty$. Let $e$ have $\|e\|_{Lip} + \|e\|_V < \infty$. Using (5.7) we see $\|Pe\|_{Lip} \leq \alpha \|e\|_{Lip}$ and by iterating the inequality we see

$$|P^t e(x) - P^t e(y)| \leq \alpha^t \|e\|_{Lip} d_A(x, y)$$

(5.8)

By iterating the Lyapunov inequality, we see

$$|P^t e(x) - P^t e(y)| \leq \|e\|_V \beta^t [V(x) + V(y)] + \|e\|_V K'$$

(5.9)

where $K' = 2K/(1 - \beta)$. Combining (5.8) and (5.9), for any coupling $\gamma$ of $\nu$ and $\pi$,

$$|\nu P^t(e) - \pi(e)| \leq (\|e\|_{Lip} + \|e\|_V) \int_{X \times X} \min\{\alpha^t d_A(x, y), \beta^t [V(x) + V(y)] + K'\} d\gamma(x, y)$$

It remains to show that right hand side of this inequality tends to 0 as $t \to \infty$. Letting $f_t(x, y) = \min\{\alpha^t d_A(x, y), \beta^t [V(x) + V(y)] + K'\}$, it is clear the pointwise convergence of $f_t$ to 0 holds. Since also $|f_t| \leq V(x) + V(y) + K'$, the latter function being $\gamma$-integrable, the result follows by the dominated convergence theorem.

Let $x_0$ be an arbitrary basepoint in $X$. The next result strengthens the conclusion in
case $V(x) = 1 + d_A(x_0, x)$, and concerns contraction in the Wasserstein space $\mathcal{P}_{p,A}$.

**Proposition 5.5.3.** Let the assumptions of Proposition 5.5.1 hold for some $p \geq 1$ and $\alpha < 1$. Let $V(x) = 1 + d_A(x, x_0)$ be a $p$-Lyapunov function for the kernel $P$. Then $P$ determines a contraction mapping on the Wasserstein space $\mathcal{P}_{p,A}(\mathcal{X})$ and possesses a unique invariant measure $\pi \in \mathcal{P}_{p,A}$. Furthermore, if $\nu \in \mathcal{P}_{p,V}$,

$$
\sup_{\|\epsilon\|_{\text{Lip}} \leq 1} |\nu P^t(\epsilon) - \pi(\epsilon)| \leq \alpha^t \sup_{\|\epsilon\|_{\text{Lip}} \leq 1} |\nu(\epsilon) - \pi(\epsilon)|. \quad (5.10)
$$

\textbf{Proof.} Let $\gamma$ be any coupling in $\Gamma(\nu_1, \nu_2)$. For any points $x, y$ of $\mathcal{X}$ we can form a coupling of $\delta_x P$ and $\delta_y P$ using common random numbers. Formally, this is the measure $C(x, y)$ which arises as the pushforward of $\eta$ under the map $\xi \mapsto (f(x, \xi), f(y, \xi))$. Then $C$ is a well-defined Markov kernel on $\mathcal{X} \times \mathcal{X}$, and according to Proposition 5.5.1,

$$
\left( \int_{\mathcal{X} \times \mathcal{X}} d_A(x', y')^p d(\delta_{(x,y)} C)(x', y') \right)^{1/p} \leq \alpha d_A(x, y)
$$

Then

$$
d_{p,A}(\nu_1 P, \nu_2 P) \leq \left( \int_{\mathcal{X} \times \mathcal{X}} d_A(x, y)^p d(\gamma C)(x, y) \right)^{1/p} \leq \alpha \left( \int_{\mathcal{X} \times \mathcal{X}} d_A(x, y)^p d\gamma(x, y) \right)^{1/p}
$$

Since $\gamma$ was arbitrary, it follows that $P$ is a contraction. Since $\mathcal{P}_{p,A}$ is complete, $P$ has a unique stationary measure $\pi$ in $\mathcal{P}_{p,A}$. Inequality (5.10) results by combining the contraction property with the duality formula (3.7).

Conditions similar to those used in Proposition 5.5.1 have been mentioned in other works. The work of (90) considered the case of a scalar potential $A(x) = V(x)I$. The metric viewpoint for the scalar potential can be found in (36, 91). The results of (1) may be helpful.
to find scalar weight functions. The contraction conditions were also motivated by work on
contraction analysis for deterministic systems (57, 25).

Aside from generality, there is a reason related to gradient estimation for considering
matrix-valued functions $A$. Even if the underlying system has the unweighted average
contraction property, meaning inequality (ii) of Proposition 5.5.1 holds with the function
$A(x) = I$, this does not extend to the joint system (Eqns. 5.2a, 5.2b). This is due to the factor
$m$ in the auxiliary system (5.2b), which makes the Jacobian $\frac{dT}{dz}$ large at points $(x, m)$ where
$\|m\|$ is large. One approach is to look beyond the scalar potentials to metrics that weigh the
$x$ and $m$ directions differently. We will see in Section 5.8 that, for the case of unweighted
contraction, a suitable metric involves a matrix $H(x, m)(u_x, u_m) = (1 + h(x, m))u_x, u_m$
for a scalar function $h(x, m)$.

5.5.1 Interconnections of Contractions

This section gives conditions for the interconnection of two contracting systems to again be
contracting. It is relevant to gradient estimation since the system (5.2a, 5.2b) has a hierar-
chical form, the underlying system $x$ feeding into the system $m$. Interconnection theorems
for contracting systems hold in other dynamical settings as well; results for deterministic
continuous time systems can be found in (80, 83).

Let $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$ be closed, convex sets, and let $Z = X \times Y$. For instance, when these
results are applied later to the forward sensitivity process, the space $Y$ will be $L(\mathbb{R}^{nx}, \mathbb{R}^{n\phi})$.
Let $(\Xi, \Sigma, \eta)$ be a probability space and let $R$ be the Markov kernel that corresponds to
following stochastic recursion on $Z$:

\[
x(t + 1) = f(x(t), y(t), \xi(t + 1)) \\
y(t + 1) = g(x(t), y(t), \xi(t + 1))
\]
where the $\xi(t)$ are independent $\eta$-distributed random variables. For measurable $\phi : Z \to \mathbb{R}$, one has $(R\phi)(x, y) = \int_{\Xi} \phi(T(x, y, \xi)) \, d\eta(\xi)$ where $T(x, y, \xi) = (f(x, y, \xi), g(x, y, \xi))$. We find conditions on $f$ and $g$ that guarantee the joint system is contracting.

**Assumption 5.5.4.** Regarding the functions $f, g$ and the probability space $(\Xi, \Sigma, \eta)$,

i. The maps $(x, y) \mapsto f(x, y, \xi)$ and $(x, y) \mapsto g(x, y, \xi)$ are $C^1$ for each $\xi \in \Xi$;

ii. There are pairs $(\| \cdot \|_X, F), (\| \cdot \|_Y, G)$, such that $\| \cdot \|_X, \| \cdot \|_Y$ are norms on $\mathbb{R}^n, \mathbb{R}^m$ respectively, $F : X \times Y \to \mathbb{R}^{n \times n}$ and $G : X \times Y \to \mathbb{R}^{m \times m}$ are continuous with values in the invertible matrices, and $\sup_{(x, y) \in X \times Y} \|F(x, y)^{-1}\|_X + \|G(x, y)^{-1}\|_Y < \infty$;

iii. There are $\alpha_1$ and $\alpha_2$, both in $[0, 1)$, such that

$$
\sup_{z \in \Xi} \sup_{u \in \mathbb{R}^n : \|u\|_X = 1} \left( \int_{\Xi} \|F(T(z, \xi)) \frac{\partial F}{\partial x}(z, \xi) F^{-1}(z) u\|_X^p \, d\eta(\xi) \right)^{1/p} \leq \alpha_1,
$$

$$
\sup_{z \in \Xi} \sup_{u \in \mathbb{R}^m : \|u\|_Y = 1} \left( \int_{\Xi} \|G(T(z, \xi)) \frac{\partial G}{\partial y}(z, \xi) G^{-1}(z) u\|_Y^p \, d\eta(\xi) \right)^{1/p} \leq \alpha_2.
$$

We are concerned with pointwise contraction as in Proposition 5.5.1. With further integrability assumptions, convergence to a unique stationary measure can be obtained with results of the previous section.

**Proposition 5.5.5.** Let Assumption 5.5.4 hold. Let $K_1, K_2$, and $p \geq 1$ be such that

i. $\sup_{z \in \Xi} \sup_{\|u\|_Y = 1} \left( \int_{\Xi} \|F(T(z, \xi)) \frac{\partial F}{\partial y}(z, \xi) G^{-1}(z) u\|_Y^p \, d\eta(\xi) \right)^{1/p} \leq K_1$,

ii. $\sup_{z \in \Xi} \sup_{\|u\|_X = 1} \left( \int_{\Xi} \|G(T(z, \xi)) \frac{\partial G}{\partial x}(z, \xi) F^{-1}(z) u\|_X^p \, d\eta(\xi) \right)^{1/p} \leq K_2$,

iii. $K_1 K_2 < (1 - \alpha_1)(1 - \alpha_2)$. 

Choose \( \rho_1, \rho_2 \) so that \( \rho_2 K_2 < \rho_1 (1 - \alpha_1) \) and \( \rho_1 K_1 < \rho_2 (1 - \alpha_2) \). Then a pointwise \( p \)-contraction inequality holds for the system \( \{ (\Xi, \Sigma, \eta), T, (\| \cdot \|_Z, H) \} \) on \( Z \) where

\[
H(z)(u_x, u_y) = (F(z)u_x, G(z)u_y) \tag{5.11a}
\]

\[
\| (u_x, u_y) \|_Z = \rho_1 \| u_x \|_X + \rho_2 \| u_y \|_Y \tag{5.11b}
\]

**Proof.** We will apply Proposition 5.5.1. We must find an \( \alpha < 1 \) so that

\[
\sup_{z \in Z} \sup_{\| u \|_Z = 1} \left( \int_{\Xi} \| H(T(z, \xi)) \frac{\partial T}{\partial z}(z, \xi) H(z)^{-1} u \|_Z^p \, d\eta(\xi) \right)^{1/p} \leq \alpha.
\]

Let \( z \in Z \) and let \( u = (u_x, u_y) \) be any vector with \( \rho_1 \| u_x \|_X + \rho_2 \| u_y \|_Y = 1 \). Then

\[
\left( \int_{\Xi} \| H(T(z, \xi)) \frac{\partial T}{\partial z}(z, \xi) H(z)^{-1} u \|_Z^p \, d\eta(\xi) \right)^{1/p} = \left( \int_{\Xi} \left[ \rho_1 \| F(T(z, \xi)) \frac{\partial f}{\partial x}(z, \xi) F(z)^{-1} u_x + F(T(z, \xi)) \frac{\partial f}{\partial y}(z, \xi) G(z)^{-1} u_y \|_X + \rho_2 \| G(T(z, \xi)) \frac{\partial g}{\partial x}(x, \xi) F(z)^{-1} u_x + G(T(z, \xi)) \frac{\partial g}{\partial y}(x, \xi) G(z)^{-1} u_y \|_Y \right]^p \, d\eta(\xi) \right)^{1/p} \leq \rho_1 \alpha_1 \| u_x \|_X + \rho_1 K_1 \| u_y \|_Y + \rho_2 K_2 \| u_x \|_X + \rho_2 \alpha_2 \| u_y \|_Y \leq \max \left\{ \alpha_1 + \frac{\rho_2}{\rho_1} K_2, \alpha_2 + \frac{\rho_1}{\rho_2} K_1 \right\}
\]

Finally, note that satisfiability of the condition \( \max \{ \alpha_1 + \frac{\rho_2}{\rho_1} K_2, \alpha_2 + \frac{\rho_1}{\rho_2} K_2 \} < 1 \) is equivalent to the condition \( K_1 K_2 < (1 - \alpha_1)(1 - \alpha_2) \).

The above can be specialized to hierarchical interconnections:

**Corollary 5.5.6.** Let Assumption 5.5.4 hold. Say that \( f \) does not depend on \( Y \) \( (\frac{\partial f}{\partial y} = 0) \).

Let \( K \) be such that

\[
\sup_{z \in Z} \sup_{\| u_x \|_X = 1} \left( \int_{\Xi} \| G(T(z, \xi)) \frac{\partial g}{\partial x}(z, \xi) F(z)^{-1} u_x \|_Y^p \, d\eta(\xi) \right)^{1/p} \leq K. \tag{5.12}
\]
Choose $\rho_1, \rho_2$ so that $\rho_2 K < \rho_1(1 - \alpha_1)$. Then a pointwise $p$-contraction property holds for the system $\{(\Xi, \Sigma, \eta), T, (\| \cdot \|_Z, H)\}$ on $Z$ using the $H$ and $\| \cdot \|_Z$ of (5.11a, 5.11b).

The condition (5.12) in Corollary 5.5.6 can be relaxed using a kind of Lyapunov function for the interconnection of the two systems, while requiring a stronger form of contraction on the input system.

**Proposition 5.5.7.** Let Assumption 5.5.4 hold, with $p \geq 2q$ for some $q \geq 1$. Let $K$ and the continuous function $h : Z \to \mathbb{R}_{\geq 0}$ be such that, for all $z \in Z$,

i. $\sup_{\|u_x\|_X = 1} \left( \int_{\Xi} \|G(T(z, \xi)) \frac{\partial}{\partial x} (z, \xi) F^{-1}(z) u_x \|^q_X d\eta(\xi) \right)^{1/q} \leq h(z),$

ii. $\left( \int_{\Xi} h(T(z, \xi))^p d\eta(\xi) \right)^{1/p} \leq h(z) + K.$

Then there are some $\rho_1, \rho_2$ so that a pointwise $q$-contraction inequality holds for the system $\{(\Xi, \Sigma, \eta), T, (\| \cdot \|_Z, H)\}$ on $Z$ where

$$H(z)(u_x, u_y) = ((1 + \rho_1 h(z)) F(z) u_x, G(z) u_y))$$

$$\|(u_x, u_y)\|_Z = \|u_x\|_X + \rho_2 \|u_y\|_Y$$

**Proof.** Let $\alpha_1, \alpha_2$ be contraction coefficients for $f, g$ respectively. Let $F_1(z) = [1 + \rho_3 h(z)] F(z)$, using an $\rho_3 \geq 0$ such that $\alpha_1 (1 + \rho_3 K) < 1$. We aim to apply Corollary 5.5.6 to the pair of systems $f$ and $g$, using a metric defined by the pairs $(\| \cdot \|_X, F_1)$ and $(\| \cdot \|_Y, G)$, in order to find $q$-contraction of the joint system. Letting $\|u_x\|_X = 1$, then,

$$\left( \int_{\Xi} \|F_1(T(z, \xi)) \frac{\partial}{\partial x} (x, \xi) F_1(z)^{-1} u_x \|^q_X d\eta(\xi) \right)^{1/q} = \left( \int_{\Xi} \left\| \frac{1 + \rho_3 h(T(z, \xi))}{1 + \rho_3 h(z)} F(T(z, \xi)) \frac{\partial}{\partial x} (x, \xi) F(z)^{-1} u_x \right\|^q_X d\eta(\xi) \right)^{1/q}$$
Applying Hölder’s inequality and the assumption on $\frac{\partial f}{\partial x}$ yields

\[ \leq \frac{1}{1 + \rho_3 h(z)} \left( 1 + \rho_3 \left( \int_{\Xi} h(T(z, \xi))^2 \eta(\xi) \right)^{1/(2q)} \right) \alpha_1 \]
\[ \leq \frac{1 + \rho_3 (h(z) + K)}{1 + \rho_3 h(z)} \alpha_1 \leq \alpha_1 (1 + \rho_3) K \]

It remains to show that Inequality 5.12 holds. Let $\|u_x\|_X = 1$. Then

\[ \left( \int_{\Xi} \|G(T(z, \xi)) \frac{\partial g}{\partial x}(z, \xi) F_1(z)^{-1} u_x \|_Y^q \eta(\xi) \right)^{1/q} \]
\[ = \frac{1}{1 + \rho_3 h(z)} \left( \int_{\Xi} \|G(T(z, \xi)) \frac{\partial g}{\partial x}(z, \xi_2) F(x)^{-1} u_x \|_Y^q \eta(\xi) \right)^{1/q} \leq \frac{h(z)}{1 + \rho_3 h(z)} \leq \frac{1}{\rho_3} \]

Let $\rho_1, \rho_2$ be chosen so that $\rho_2 \frac{1}{\rho_3} < \rho_1 (1 - \alpha_1(1 + \rho_3) K)$. Then by Corollary 5.5.6 the tuple $\{(\Xi, \Sigma, \eta), T, (\|\cdot\|, H)\}$ determines a $q$-contracting system, where $\|(u, v)\|_Z = \rho_1 \|u\| + \rho_2 \|v\|$ and $H(z)(u_x, u_y) = ((1 + \rho_3 h(z)) F(z) u_x, G(z) u_y)$. One can take $\rho_1 = 1$ in these requirements, by choosing $\rho_2$ small enough that $\rho_2 \frac{1}{\rho_3} < (1 - \alpha_1(1 + \rho_3) K)$. \qed

### 5.6 Stationary Differentiability

Differentiability of stationary costs is established using properties of the Markov kernel $P$. Formally differentiating the equation $\pi_\theta = \pi_\theta P_\theta$ in $\theta$ suggests the stationary derivative $\pi'$ solves the equation $l = l P_\theta + \pi_\theta P'_\theta$ in the variable $l$. By defining $P'$ properly, as the linear map $e \mapsto \frac{\partial}{\partial \theta} P_\theta e$ on the space of cost functions, and considering this equation as being between functionals defined on the cost functions, one can show that it has a unique solution $l^*$, which is such that $l^*(e) = \frac{\partial}{\partial \theta} \int_{X} e(x) \, d\pi_\theta(x)$. The line of argument used in this section is a variant of Theorem 2 in (40), adapted to the specific ergodicity and state space conditions that we work with. In that work, a class of functions with a norm $\|e\| = \sup_x \frac{|e(x)|}{V(x)}$ is considered, while the norm we will use also involves the derivatives of $e$. This seems to make certain
steps in the proof more complicated. Primarily, these are Parts iv and v of Assumption 5.6.2. In the next section, the Assumptions are verified based on properties of the derivatives of the system.

We introduce the assumptions on $P$ and the cost functions $\mathcal{E}$:

**Assumption 5.6.1.** $X$ is a Polish space, $\mathcal{E}$ a vector space of real-valued functions on $X$ with norm $\| \cdot \|_\mathcal{E}$, and $\mathcal{P}$ a space of probability measures on $X$. For any $\nu \in \mathcal{P}$, it is required that $\sup_{\|e\|_\mathcal{E} \leq 1} |\nu(e)| < \infty$.

For a measure $\pi$ denote by $\Pi$ the Markov kernel $\Pi(x, A) = \pi(A)$. The parameter space is an open set $\Theta \subseteq \mathbb{R}^{\Theta}$ and we fix a $\theta_0 \in \Theta$. The space $\mathbb{R}^{\Theta}$ has a norm $\| \cdot \|_\Theta$. We show that the map sending a cost function $e$ to its stationary derivative at the fixed parameter $\theta_0$ is an element of the set $\mathcal{L}$ of linear maps from $\mathcal{E}$ to $L(\mathbb{R}^{\Theta}, \mathbb{R})$ that vanish on the constant functions and are bounded with respect to the norm $\|l\|_\mathcal{E} = \sup_{\|e\|_\mathcal{E} \leq 1} \|l(e)\|_\Theta$:

$$\mathcal{L} = \{l \in L(X, L(\mathbb{R}^{\Theta}, \mathbb{R})) \mid \|l\|_\mathcal{E} < \infty, \ l(1) = 0\}$$

where $1$ is the constant function $x \mapsto 1$. Note that $\mathcal{L}$ with the norm $\| \cdot \|_\mathcal{E}$ is a complete space.

To discuss stationary differentiability we introduce the operator $\frac{\partial}{\partial \theta} P_{\theta_0}$. If $e \in \mathcal{E}$ then $\frac{\partial}{\partial \theta} P_{\theta_0} e$ is the function from $X$ into $L(\mathbb{R}^{\Theta}, \mathbb{R})$ defined by $(\frac{\partial}{\partial \theta} P_{\theta_0} e)(x) = \frac{\partial}{\partial \theta} (P_{\theta_0} e(x))$.

**Assumption 5.6.2.** For any $\theta \in \Theta$ the following hold.

i. If $\nu \in \mathcal{P}$ then $\nu P_{\theta} \in \mathcal{P}$ and $P_{\theta}$ has a stationary measure $\pi_{\theta}$ in $\mathcal{P}$,

ii. If $e \in \mathcal{E}$ then $P_{\theta_0} e \in \mathcal{E}$, $\|P_{\theta_0} e\|_\mathcal{E} < \infty$, and $\sum_{i=0}^{\infty} \|P_{\theta_0}^i - \Pi_{\theta_0}\|_\mathcal{E} \leq K_{\theta_0}$ for some $K_{\theta_0} \geq 0$,

iii. For $e \in \mathcal{E}$ and $x \in X$ the function $\theta \mapsto P_{\theta_0} e(x)$ is differentiable at $\theta_0$ and $\|\pi_{\theta_0} \frac{\partial}{\partial \theta_0} P_{\theta_0}\|_\mathcal{E} < \infty$, 


Theorem 5.6.3. Under Assumptions 5.6.1 and 5.6.2 if $e \in \mathcal{E}^2$ then $\pi_\theta(e)$ is differentiable at $\theta_0$ and $\frac{\partial}{\partial \theta} \int_X e(x) \, d\pi_{\theta_0}(x) = l^*(e)$ where $l^* \in \mathcal{L}$ satisfies $l^* = l^* P_{\theta_0} + \pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0}$.

Proof of Theorem 5.6.3. First, define $T : \mathcal{L} \to \mathcal{L}$ as $T(l) := l P_{\theta_0} + \pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0}$. That $\pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0}$ is in $\mathcal{L}$ was one of our assumptions along with $\|P_{\theta_0}\|_\mathcal{E} < \infty$, which implies $T$ is well-defined.

Let $l^*$ be the functional $l^* = \sum_{i=0}^{\infty} (\pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0}) P^i_{\theta_0}$. This is in $\mathcal{L}$ since that space is Banach and by Part ii of Assumption 5.6.2,

$$\sum_{i=0}^{\infty} \| (\pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0}) P^i_{\theta_0} \|_\mathcal{L} = \sum_{i=0}^{\infty} \| (\pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0})(P^i_{\theta_0} - \Pi_{\theta_0}) \|_\mathcal{L} \leq \| \pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0} \|_\mathcal{L} K.$$

To see that $l^*$ is a fixed-point of $T$, note that $T(l^*) = \sum_{i=1}^{\infty} (\pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0}) P^i_{\theta_0} + \pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0} = l^*$.

To show $l^*$ is the unique fixed-point, let $l$ be any other fixed-point of $T$. Then

$$\|l - l^*\|_\mathcal{L} = \|T^n(l) - T^n(l^*)\|_\mathcal{L} = \|l - l^* (P^n_{\theta_0} - \Pi_{\theta_0})\|_\mathcal{L} \leq \|l - l^*\|_\mathcal{L} \| P^n_{\theta_0} - \Pi_{\theta_0} \|_\mathcal{L}.$$

Using Part ii of Assumption 5.6.2 again, the right hand side of this inequality goes to zero as $n \to \infty$, hence $T$ possesses a unique fixed-point $l^*$ in $\mathcal{L}$.

Define $c(\Delta \theta)$ as the functional $c(\Delta \theta)(e) = \pi_{\theta_0 + \Delta \theta}(e) - \pi_{\theta_0}(e) - l^*(e)(\Delta \theta)$. Assumption 5.6.1 and the definition of $\mathcal{L}$ guarantees $c(\Delta \theta) \in L(\mathcal{E}, \mathbb{R})$. It suffices that $\frac{1}{\|\Delta \theta\|_{\Theta}} \| c(\Delta \theta) \|_\mathcal{E} \to 0$.
as $\Delta \theta \to 0$. Using the fact that $T(l^*) = l^*$, we have

$$c(\Delta \theta) = \pi_{\theta_0} [P_{\theta_0 + \Delta \theta} - P_{\theta_0} - \frac{\partial}{\partial \theta} P_{\theta_0}(\Delta \theta)] + (\pi_{\theta_0 + \Delta \theta} - \pi_{\theta_0}) [P_{\theta_0 + \Delta \theta} - P_{\theta_0}] + c(\Delta \theta) P_{\theta_0}$$

Iterating this, and noting that each summand is a functional vanishing on the constant functions, we obtain that for any $k > 0$,

$$c(\Delta \theta) = \pi_{\theta_0} (P_{\theta_0 + \Delta \theta} - P_{\theta_0} - \frac{\partial}{\partial \theta} P_{\theta_0}(\Delta \theta)) \sum_{i=0}^{k-1} (P_{\theta_0}^i - \Pi_{\theta_0})$$

$$+ (\pi_{\theta_0 + \Delta \theta} - \pi_{\theta_0}) [P_{\theta_0 + \Delta \theta} - P_{\theta_0}] \sum_{i=0}^{k-1} (P_{\theta_0}^i - \Pi_{\theta_0})$$

$$+ c(\Delta \theta) (P_{\theta_0}^k - \Pi_{\theta_0})$$

Taking norms and letting $k \to \infty$, we see that

$$||c(\Delta \theta)||_{\mathcal{E}} \leq ||\pi_{\theta} (P_{\theta_0 + \Delta \theta} - P_{\theta_0} - \frac{\partial}{\partial \theta} P_{\theta_0}(\Delta \theta))||_{\mathcal{E}} K_{\theta_0} + ||(\pi_{\theta_0 + \Delta \theta} - \pi_{\theta_0}) [P_{\theta_0 + \Delta \theta} - P_{\theta_0}]||_{\mathcal{E}} K_{\theta_0}.$$ 

Finally, use Parts iv and v of Assumption 5.6.2 \qed

### 5.7 State Space Conditions

Let $P_\theta$ be the transition kernel of the Markov chain

$$x(t + 1) = f(x(t), \xi(t + 1), \theta)$$

(5.14)

with $\eta$-distributed random input $\xi(t)$. In this section we show how Assumptions 5.4.1, 5.4.2, 5.4.3, and 5.4.4 imply Assumptions 5.6.1 and 5.6.2, thereby establishing differentiability of the stationary costs for those cost functions $e \in \mathcal{E}^2$. 

Theorem 5.7.1. Let Assumptions 5.4.1 - 5.4.4 be satisfied. Then Assumptions 5.6.1 and 5.6.2 are verified for the space $P_{2,A}(X)$ of probability measures and the space of cost functions $E^2$, at any $\theta_0 \in \Theta$. Hence $\pi_{\theta_0}(e)$ is differentiable for any $\theta_0 \in \Theta$ and $e \in E^2$.

To show this, several preliminary results will be used. The first is concerned with how $P_\theta$ varies with $\theta$. Recall that $x_0$ denotes an arbitrary basepoint.

Proposition 5.7.2. Let $P_\theta$ be the transition kernel of the recursion (5.14), where

i. The map $\xi \mapsto d_A(x_0, f(x, \xi, \theta))^p$ is $\eta$-integrable for each $(x, \theta) \in X \times \Theta$,

ii. The function $(x, \theta) \mapsto f(x, \xi, \theta)$ is $C^1$ for each $\xi \in \Xi$,

iii. $\sup_{(x,\theta)\in X\times\Theta} \sup_{\|u\|=1} \left( \int_{\Xi} \|A(f(x,\xi,\theta))\frac{\partial f}{\partial \theta}(x,\xi,\theta)B(x)^{-1}u_\theta\|^p d\eta(\xi) \right)^{1/p} \leq K$.

Fix a $\theta_0 \in \Theta$. Then for all $\Delta \theta$ sufficiently small and all $\nu \in P_{2,A}(X)$ the inequality

$d_{p,A}(\nu P_{\theta_0}, \nu P_{\theta_0 + \Delta \theta}) \leq K \|B \Delta \theta\|_{L^p(\nu)}$ holds.

Proof. Let $\Delta \theta$ be so small that $\theta_0 + t \Delta \theta \in \Theta$ for $t \in [0, 1]$. If $(x, \xi)$ is distributed according to $\nu \times \eta$ then the law of $(f(x, \xi, \theta_0), f(x, \xi, \theta_0 + \Delta \theta))$ is a coupling of $\nu P_{\theta_0}$ and $\nu P_{\theta_0 + \Delta \theta}$. Let $\gamma : [0, 1] \to \mathbb{R}^{n \omega}$ be $\gamma(t) = \theta_0 + t \Delta \theta$. Then $t \mapsto f(x, \xi, \gamma(t))$, determines a curve from $f(x, \xi, \theta_0)$ to $f(x, \xi, \theta_0 + \Delta \theta)$, and reasoning as in Proposition 5.5.1,

$$\left( \int_X \int_{\Xi} d_A(f(x, \xi, \theta_0), f(x, \xi, \theta_0 + \Delta \theta))^p \, d\eta(\xi) \, d\nu(x) \right)^{1/p} \leq \left( \int_X \int_{\Xi} \left( \int_0^1 \|A(f(x, \xi, \gamma(t))\frac{\partial f}{\partial \theta}(x, \xi, \gamma(t))\Delta \theta\|^p \, dt \right)^p \, d\eta(\xi) \, d\nu(x) \right)^{1/p} \leq \left( \int_0^1 \int_X \int_{\Xi} \|A(f(x, \xi, \gamma(t)))\frac{\partial f}{\partial \theta}(x, \xi, \gamma(t))\Delta \theta\|^p \, d\eta(\xi) \, d\nu(x) \, dt \right)^{1/p} \leq \left( \int_0^1 \int_X K^p \|B(x)\Delta \theta\|^p \, dt \right)^{1/p} = K \|B \Delta \theta\|_{L^p(\nu)} \leq K \|B \Delta \theta\|_{L^p(\nu)}$$
The continuity assumptions on the $L_X, \Theta$ ensure that integration and differentiation can be exchanged. For discussing the differentiability it will be useful to introduce the following concept. A function $f : X \times \Xi \rightarrow \mathbb{R}^n$ is said to be $L^1(\eta)$-continuous when

1. $x \mapsto f(x, \xi)$ is continuous for each $\xi \in \Xi$,
2. $\xi \mapsto f(x, \xi)$ is measurable for each $x \in X$,
3. $x \mapsto \int_{\Xi} \|f(x, \xi)\| \, d\eta(\xi)$ is continuous.

The following two properties are not difficult to show. (i) If $f, g$ are $L^1(\eta)$-continuous functions then so are $\alpha f + \beta g$ for any numbers $\alpha, \beta$. (ii) A monotonicity property holds: If $f$ is a function satisfying the first two requirements of $L^1(\eta)$-continuity and if $\|f(x, \xi)\| \leq \|g(x, \xi)\|$ for an $L^1(\eta)$-continuous function $g$, then $f$ is $L^1(\eta)$-continuous.

Using this notion we state a condition for interchanging derivatives and integrals which is a generalized form of a result from (69), that considers a scalar parameter.

**Theorem 5.7.3** ((69), Theorem 3.13). Let $(\Xi, \Sigma, \eta)$ be a probability space and $W \subseteq \mathbb{R}^n$ be an open set. Let $h : W \times \Xi \rightarrow \mathbb{R}^m$ be a function such that

1. $\xi \mapsto h(w, \xi)$ is integrable for each $w \in W$,
2. $w \mapsto h(w, \xi)$ is continuously differentiable for each $\xi \in \Xi$,
3. $\frac{\partial h}{\partial w}$ is $L^1(\eta)$-continuous.

Then $\frac{\partial}{\partial w} \int_{\Xi} h(w, \xi) \, d\eta(\xi) = \int_{\Xi} \frac{\partial h}{\partial w}(w, \xi) \, d\eta(\xi)$ for all $w \in W$.

This criteria has the useful property that once it is established for $f$ it is easily extended to the function $e \circ f$. This is shown in the next proposition.

**Proposition 5.7.4.** Let Assumptions 5.4.1, 5.4.2, 5.4.3 and 5.4.4 hold. If $e \in \mathcal{E}^2$ and $i+j \leq 2$ then, for any $(x, \theta) \in X \times \Theta$, $\frac{\partial^{i+j}}{\partial x^i \partial \theta^j} \int_{\Xi} e(f(x, \xi, \theta)) \, d\eta(\xi) = \int_{\Xi} \frac{\partial^{i+j}}{\partial x^i \partial \theta^j} e(f(x, \xi, \theta)) \, d\eta(\xi)$. 
Proof. Consider the derivative $\frac{\partial}{\partial x}$. To apply Theorem 5.7.3, it is shown that the map $x \mapsto \int_\Xi \| \frac{\partial e}{\partial x}(f(x, \xi, \theta)) \frac{\partial f}{\partial x}(x, \xi, \theta) \| \, d\eta(\xi)$ is continuous. Noting that

$$\| \frac{\partial e}{\partial x}(f(x, \xi, \theta)) \frac{\partial f}{\partial x}(x, \xi, \theta) \| \leq \| \frac{\partial e}{\partial x} \|_A \| A(f(x, \xi, \theta)) \| \| A(x)^{-1} \| \| A(x) \|$$

the result follows by assumption on $\frac{\partial f}{\partial x}$ and the monotonicity property of $L^1(\eta)$-continuity.

Next, consider $\frac{\partial^2}{\partial \theta^2}$. We have that

$$\| \frac{\partial^2}{\partial \theta^2} e(f(x, \xi, \theta)) \| \leq \| \frac{\partial e}{\partial x} \|_A \| A(f(x, \xi, \theta)) \frac{\partial f}{\partial \theta}(x, \xi, \theta) B(x)^{-1} \| \| B(x) \|^2$$

$$+ \| \frac{\partial e}{\partial x} \|_A \| A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial \theta^2}(x, \xi, \theta) (B(x)^{-1} \oplus B(x)^{-1}) \| \| B(x) \|^2$$

The $L^1$-continuity of the left hand side follows by the $L^1$-continuity of the right side together with the monotonicity property. Similar reasoning yields the other cases.

Using this result, we can obtain the contraction property of $P$ with respect to the class $\mathcal{E}^2$, and find some bounds on the second order derivatives of $P_\theta e$:

**Proposition 5.7.5.** Let Assumptions 5.4.2 - 5.4.4 be in effect. For $e \in \mathcal{E}^2$ and $\theta \in \Theta$,

i. $\| \frac{\partial^2}{\partial x^2} P_\theta e \|_{A,A} \leq K_X^2 \| \frac{\partial e}{\partial x} \|_A + K_X \| \frac{\partial^2 e}{\partial x^2} \|_{A,A}$,

ii. $\| \frac{\partial^2}{\partial \theta^2} P_\theta e \|_{B,B} \leq K_\Theta^2 \| \frac{\partial e}{\partial x} \|_A + K_\Theta \| \frac{\partial^2 e}{\partial x^2} \|_{A,A}$,

iii. $\| \frac{\partial^2}{\partial x \partial \theta} P_\theta e \|_{A,B} \leq K_X, \Theta \| \frac{\partial e}{\partial x} \|_A + K_X K_\Theta \| \frac{\partial^2 e}{\partial x \partial \theta} \|_{A,A}$.

Furthermore, for each $\theta$ there is an $L_\theta \geq 0$ such that $\| P_\theta e \|_{\mathcal{E}^2} \leq L_\theta \| e \|_{\mathcal{E}^2}$ for all $e \in \mathcal{E}^2$.

**Proof.** We show Part ii; Parts i and iii are established similarly. We have

$$\frac{\partial^2}{\partial \theta^2} P_\theta e(x) \left( B^{-1}(x) \oplus B^{-1}(x) \right) = T_1 + T_2$$
where $T_1$ and $T_2$ are defined as

$$T_1 = \int_{\Xi} \frac{\partial e}{\partial x} (f(x, \xi, \theta)) \frac{\partial f}{\partial x}(x, \xi, \theta) \left( (B^{-1}(x))^{-1} \oplus B(x)^{-1}\right) \, d\eta(\xi),$$

$$T_2 = \int_{\Xi} \frac{\partial^2 e}{\partial x^2} (f(x, \xi, \theta)) \left( \frac{\partial f}{\partial \theta}(x, \xi, \theta) B^{-1}(x) \oplus \frac{\partial f}{\partial \theta}(x, \xi, \theta) B^{-1}(x) \right) \, d\eta(\xi).$$

Using the identity $A(f(x, \xi, \theta))^{-1} A(f(x, \xi, \theta)) \frac{\partial f}{\partial \theta}(x, \xi, \theta) = \frac{\partial f}{\partial \theta}(x, \xi, \theta)$, we get

$$\|T_1\| \leq \| \frac{\partial e}{\partial x} \|_{A} K_{\Theta^2}$$

while for $T_2$, use that $A(f(x, \xi, \theta))^{-1} A(f(x, \xi, \theta)) \frac{\partial f}{\partial \theta}(x, \xi, \theta) = \frac{\partial f}{\partial \theta}(x, \xi, \theta)$ to get

$$\|T_2\| \leq \| \frac{\partial^2 e}{\partial x^2} \|_{A,A} \left( \int_{\Xi} \|A(f(x, \xi)) \frac{\partial f}{\partial \theta}(x, \xi, \theta) B^{-1}(x) \|^2 \, d\eta(\xi) \right) \leq \| \frac{\partial^2 e}{\partial x^2} \|_{A,A} K_{\Theta^2}.$$

Combining this last inequality with Inequality (5.15), then,

$$\| \frac{\partial^2}{\partial \theta^2} P_\theta e(x) \|_{B(x), B(x)} \leq \| \frac{\partial e}{\partial x} \|_{A} K_{\Theta^2} + \| \frac{\partial^2 e}{\partial x^2} \|_{A,A} K_{\Theta^2}.$$

To show the boundedness with respect to $\| \cdot \|_{E^2}$, note that for any $e \in E^2$,

$$|(P_\theta e)(x)| \leq |e(x_0)| + \| \frac{\partial e}{\partial x} \|_{A} \int_{X} d_{A}(x_0, y) \, d(\delta_x P_\theta)(y)$$

$$\leq |e(x_0)| + \| \frac{\partial e}{\partial x} \|_{A} [C_\theta + K_X d_{A}(x, x_0)]$$

where $C_\theta$ is the number $C_\theta = \int_{X} d_{A}(x_0, y) \, d(\delta_x P_\theta)(y)$. This follows, since for the Lipschitz function $h(x) = d(x_0, x)$, $|(P h)(x)| \leq |Ph(x_0)| + |(Ph)(x_0) - (Ph)(x)| \leq C_\theta + K_X d_{A}(x_0, x)$. Also, for any $x \in X$, $\frac{|e(x_0)|}{1 + d_{A}(x_0, x)} \leq \frac{|e(x_0)|}{1 + d_{A}(x_0, x_0)} \leq \| e \|_{A}$. Therefore $P_\theta e \|_{A} \leq \| e \|_{A} + \max \{C_\theta, K_X\} \| \frac{\partial e}{\partial x} \|_{A}$. \qed
The following quadratic bound involving the metric \( d_A \) will be used as well.

**Proposition 5.7.6.** Let \( h : X \to \mathbb{R}^n \) be differentiable, such that \( \| \frac{\partial h}{\partial x} (x) A(x)^{-1} \| \leq B(x) \) where \( B : X \to \mathbb{R} \) is Lipschitz for the metric \( d_A \). Then the following inequalities hold:

i. \( \| h(x) - h(y) \| \leq B(x) d_A(x, y) + \frac{1}{2} B \| L_{lip} d_A(x, y)^2 \)

ii. For any \( \nu_1, \nu_2 \in \mathcal{P}_{2, A}(X) \),

\[
\left\| \int_X h(x) \, d\nu_1(x) - \int_X h(y) \, d\nu_2(y) \right\| \leq \| B \|_{L^2(\nu)} d_{2, A}(\nu_1, \nu_2) + \frac{1}{2} \| B \|_{L_{lip} d_{2, A}(\nu_1, \nu_2)}
\]

**Proof.** We will make use of the following inequality: Whenever \( \gamma : [0, T] \to X \) is a curve from \( x \) to \( y \) that is (i) parameterized by arc length, and (ii) such that \( L(\gamma) \leq d_A(x, y) + \epsilon \), then

\[
\int_0^T d_A(\gamma(t), x) \, dt \leq \frac{(d_A(x, y) + \epsilon)^2}{2} \tag{5.16}
\]

To see this, note that for any curve parameterized by arc length, \( d_A(\gamma(t), x) \leq t \). Integrating both sides of this inequality and using the first assumption yields the result.

We now proceed to the proof of part i. Let \( h : X \to \mathbb{R}^n \) be a function satisfying the assumptions of the Proposition. Given \( \epsilon > 0 \), let \( \gamma : [0, T] \to X \) be a piecewise \( C^1 \) curve from \( x \) to \( y \) with \( L(\gamma) \leq d_A(x, y) + \epsilon \). Assume that \( \gamma \) is parameterized by arc length. By the identity \( \gamma'(t) = A(\gamma(t))^{-1}A(\gamma(t))\gamma'(t) \), and the assumption on \( h \),

\[
\| h(x) - h(y) \| \leq \int_0^T \| \frac{\partial h}{\partial x} (\gamma(t)) \gamma'(t) \| \, dt
\]

\[
= \int_0^T \| \frac{\partial h}{\partial x} (\gamma(t)) A(\gamma(t))^{-1}A(\gamma(t))\gamma'(t) \| \, dt
\]

\[
\leq \int_0^T B(\gamma(t)) \, dt
\]
Seeing as $B$ is Lipschitz, and invoking Inequality (5.16),

$$
\leq \int_0^T \left( B(x) + \|B\|_{Lip} A(\gamma(t), x) \right) \, dt \\
\leq B(x) \int_0^T 1 \, dt + \|B\|_{Lip} \int_0^T A(\gamma(t), x) \, dt \\
\leq B(x) [d_A(x, y) + \epsilon] + \|B\|_{Lip} [d_A(x, y)^2 / 2 + d_A(x, y) \epsilon + \epsilon^2 / 2]
$$

Since $\epsilon$ was arbitrary, we have $\|h(x) - h(y)\| \leq B(x) d_A(x, y) + \frac{1}{2} \|B\|_{Lip} d_A(x, y)^2$.

For part ii, let $\gamma$ be any coupling of $\nu_1$ with $\nu_2$ such that

$$
\left( \int_{X \times X} d_A(x, y)^2 \, d\gamma(x, y) \right)^{1/2} \leq d_{2,A}(\nu_1, \nu_2) + \epsilon. \quad \text{Then}
$$

$$
\left| \nu(h(x)) - \nu(h(y)) \right| \leq \left| \int_X h(x) \, d\nu_1(x) - \int_X h(y) \, d\nu_2(y) \right| \\
\leq \int_{X \times X} \|h(x) - h(y)\| \, d\gamma(x, y) \\
\leq \int_{X \times X} B(x) d_A(x, y) \, d\gamma(x, y) + \frac{1}{2} \|B\|_{Lip} \int_{X \times X} d_A(x, y)^2 \, d\gamma(x, y) \\
\leq \|B\|_{L^2(\nu_1)} (d_{2,A}(\nu_1, \nu_2) + \epsilon) + \frac{1}{2} \|B\|_{Lip} (d_{2,A}(\nu_1, \nu_2) + \epsilon)^2
$$

Since $\epsilon > 0$ was arbitrary we are done.

With these tools in hand we can proceed to the proof of Theorem 5.7.1.

**Proof of Theorem 5.7.1.** In order to apply Theorem 5.6.3, we establish the requirements of Assumptions 5.6.1 and 5.6.2. Assumption 5.6.1 requires that for any $\nu$ in $P_{2,A}(X)$, the bound

$$
\sup_{\|e\|_{L^2(\nu)} \leq 1} |\nu(e)| < \infty \quad \text{holds. Note that } |e(x_0)| = \frac{|e(x_0)|}{1 + d_A(x_0, x_0)} \leq \|e\|_A. \quad \text{Then}
$$

$$
|\nu(e)| \leq \int_X \left[ |e(x_0)| + \frac{\partial e}{\partial x} d_A(x_0, x) \right] \, d\nu(x) \leq \max \left\{ 1, \int_X d_A(x, x_0) \, d\nu(x) \right\} \|e\|_{L^2(\nu)}.
$$
CHAPTER 5. STOCHASTIC ATTRACTOR NETWORKS

The integrability part of Assumption 5.4.2 and the contraction part of Assumption 5.4.4 allow us to apply Proposition 5.5.3. Hence $P_\theta$ is a contraction on the space $\mathcal{P}_{2,A}(X)$ with contraction coefficient $K_X$, and has a unique invariant measure $\pi_\theta$ for each $\theta \in \Theta$. Then part i of Assumption 5.6.2 holds. Proposition 5.7.5 affirms that $P_\theta e \in \mathcal{E}^2$ if $e \in \mathcal{E}^2$, and $P_\theta$ is bounded for the norm $\| \cdot \|_{\mathcal{E}^2}$. We now establish $\| P_n^\theta - \Pi_\theta \|_{\mathcal{E}^2} \leq \kappa_\theta K_X^n$ for some constant $\kappa_\theta$. We consider each of the terms in the norm $\| \cdot \|_{\mathcal{E}^2}$. First, for $e \in \mathcal{E}^2$,

$$\| P_n^\theta (e) - \Pi_\theta (e) \|_A \leq K_X^n \| \frac{\partial e}{\partial x} \|_A \max \{ C_\theta, 1 \}$$  \hspace{1cm} (5.17)

To see this, observe that

$$| (P_n^\theta (e) - \Pi_\theta (e))(x) | = | P_n^\theta (e)(x) - P_n^\theta (e)(x_0) + P_n^\theta (e)(x_0) - \pi_\theta (e) |$$

$$\leq K_X^n \| \frac{\partial e}{\partial x} \|_A d_A(x, x_0) + K_X^n \| \frac{\partial e}{\partial x} \|_A C_\theta$$

$$\leq K_X^n \| \frac{\partial e}{\partial x} \|_A \max \{ C_\theta, 1 \}(1 + d_A(x, x_0))$$

where $C_\theta = \int_X d_A(x_0, y) d\pi_\theta(y)$. Next,

$$\| \frac{\partial}{\partial \theta} (P_n^\theta (e) - \Pi_\theta (e)) \|_A \leq K_X^n \| \frac{\partial e}{\partial x} \|_A$$  \hspace{1cm} (5.18)

Finally, by recursive application of Part i of Proposition 5.7.5,

$$\| \frac{\partial^2}{\partial \theta^2} (P_n^\theta (e) - \Pi_\theta (e)) \|_{A,A} \leq K_X^2 K_X^{n-1} \frac{1}{1 - K_X} \| \frac{\partial e}{\partial x} \|_A + K_X^{2n} \| \frac{\partial^2 e}{\partial x^2} \|_{A,A}$$  \hspace{1cm} (5.19)

Adding Inequalities 5.17, 5.18, and 5.19, one obtains $\| P_n^\theta (e) - \Pi_\theta (e) \|_{\mathcal{E}^2} \leq K_X^n \kappa_\theta | e \|_{\mathcal{E}^2}$ where $\kappa_\theta = \max \{ C_\theta, 1 \} + 1 + K_X^2 \frac{1}{K_X(1 - K_X)}$. Thus item ii of Assumption 5.6.2 is satisfied.

Proposition 5.7.4 affirms that $\theta \mapsto P_\theta e(x)$ is differentiable for $e \in \mathcal{E}^2$ and $x \in X$. Proceeding as in the proof there, we see that $\| \frac{\partial}{\partial \theta} P_\theta e(x) \| \leq \| \frac{\partial e}{\partial x} \|_A K_\theta \| B(x) \|$. Therefore
\[ \| \pi_{\theta_0} \frac{\partial}{\partial \theta} P_{\theta_0} \|_L \leq K_\theta \| B \|_{L^1(\pi_{\theta_0})}, \] which confirms Part iii of Assumption 5.6.2.

Part ii of Proposition 5.7.5 means that for any \( e \in \mathcal{E}^2 \) and \( \theta \in \Theta \), \( \| \frac{\partial^2}{\partial \theta^2} P_{\theta_0}(x) \|_{B(x), B(x)} \leq k_1 \| e \|_{\mathcal{E}^2} \) where \( k_1 = \max\{K_{\Theta}^2, K_{\theta_0}^2\} \). Using the 2nd order version of Taylor’s theorem, this implies that for all \( \Delta \theta \) sufficiently small, for all \( e \in \mathcal{E}^2 \), and \( x \in X \), we have

\[
| P_{\theta_0 + \Delta \theta} e(x) - P_{\theta_0} e(x) - \frac{\partial}{\partial \theta} P_{\theta_0}(x)(\Delta \theta) | \leq \frac{1}{2} k_1 \| e \|_{\mathcal{E}^2} \| B(x) \Delta \theta \|^2 \quad (5.20)
\]

Integrating inequality (5.20) and dividing by \( \| \Delta \theta \| \) leads to

\[
\frac{1}{\| \Delta \theta \|} \| \pi_{\theta_0} [ P_{\theta_0 + \Delta \theta} - P_{\theta_0} - \frac{\partial}{\partial \theta} P_{\theta_0}(\Delta \theta) ] \|_{\mathcal{E}^2} \leq \frac{1}{2} k_1 \| B \|_{L^2(\pi_{\theta_0})} \| \Delta \theta \|
\]

and the right hand side goes to zero as \( \| \Delta \theta \| \to 0 \). Only Part v of Assumption 5.6.2 remains.

By the fundamental theorem of calculus,

\[
(P_{\theta_0 + \Delta \theta} - P_{\theta_0}) e(x) = \int_0^1 \int_\Xi \frac{\partial}{\partial \theta} (f(x, \xi, \theta + \lambda \Delta \theta) \frac{\partial f}{\partial \theta}(x, \xi, \theta + \lambda \Delta \theta)) \Delta \theta \, d\eta(\xi) \, dt
\]

Differentiating the above with respect to \( x \) and using Part iii of Assumption 5.4.4 yields

\[
\| \frac{\partial}{\partial x} ((P_{\theta_0 + \Delta \theta} - P_{\theta_0}) e(x)) A(x)^{-1} \| \leq \| e \|_{\mathcal{E}^2} k_2 \| \Delta \theta \| \| B(x) \|
\]

where \( k_2 = \max\{K_{X, \Theta}, K_X K_\theta\} \). Applying Proposition 5.7.6 we have

\[
\|(\pi_{\theta_0 + \Delta \theta} - \pi_{\theta})(P_{\theta_0 + \Delta \theta} - P_{\theta_0}) e\|
\leq k_2 \| \Delta \theta \| \| e \|_{\mathcal{E}^2} \left[ \| B \|_{L^2(\pi_{\theta_0})} d_{2, A}(\pi_{\theta_0 + \Delta \theta}, \pi_{\theta_0}) + \frac{1}{2} \| B \|_{Lip} d_{2, A}(\pi_{\theta_0 + \Delta \theta}, \pi_{\theta_0})^2 \right]
\]
For the terms $d_{2,A}$, first apply the contraction property of $P$ and Proposition 5.7.2:

$$d_{2,A}(\pi_{\theta+\Delta \theta}, \pi_{\theta}) \leq d_{2,A}(\pi_{\theta+\Delta \theta}P_{\theta+\Delta \theta}, \pi_{\theta}P_{\theta}) + d_{2,A}(\pi_{\theta}P_{\theta+\Delta \theta}, \pi_{\theta}P_{\theta})$$

$$\leq K_X d_{2,A}(\pi_{\theta+\Delta \theta}, \pi_{\theta}) + K_{\Theta}\|B\Delta \theta\|_{L^2(\pi_{\theta})}$$

Rearranging terms yields $d_{2,A}(\pi_{\theta+\Delta \theta}, \pi_{\theta}) \leq \frac{1}{1-K_X}K_{\Theta}\|B\Delta \theta\|_{L^2(\pi_{\theta})}$. Hence

$$\|(\pi_{\theta_0+\Delta \theta} - \pi_{\theta})(P_{\theta_0+\Delta \theta} - P_{\theta_0})\|_L$$

$$\leq k_2\|B\|_{L^2(\pi_{\theta_0})}\|\Delta \theta\|\left[\frac{1}{1-K_X}K_{\Theta}\|\Delta \theta\| + \frac{1}{2}\|B\|_{Lip}\left(\frac{1}{1-K_X}K_{\Theta}\|\Delta \theta\|\right)^2\right],$$

and Part v of Assumption 5.6.2 is verified. \qed

### 5.8 Gradient Estimation

The goal of this section is to prove Theorem 5.4.5. The standing assumptions are Assumptions 5.4.1 - 5.4.4. We let $Z = X \times M$ and denote elements of this space by $z = (x,m)$. Denote by $R_{\theta}$ the Markov kernel corresponding to the recursion (5.2a, 5.2b). In Proposition 5.8.1 and Corollary 5.8.2 we establish convergence of the forward sensitivity system in the sense of Proposition 5.5.2. It involves finding an appropriate Lyapunov function $V$ and metric $d_H$ on $X \times M$. In Proposition 5.8.3 we show that $(x,m) \mapsto \frac{\partial e}{\partial z}(x)m$ is an integrable function for $\gamma_{\theta}$, thereby establishing that the right hand side of (5.1) is finite. Finally, we want to show that the functional $l$ defined by

$$l(e) = \int_{X \times M} \frac{\partial e}{\partial z}(x)m \, d\gamma_{\theta}(x,m)$$

(5.21)

is bounded for the norm $\| \cdot \|_L$ and satisfies the derivative equation of Theorem 5.6.3.
Define $g$ and $T$ to be the functions

$$g((x,m),\xi,\theta) = \frac{\partial f}{\partial x}(x,\xi,\theta)m + \frac{\partial f}{\partial \theta}(x,\xi,\theta),$$

(5.22)

$$T((x,m),\xi,\theta) = (f(x,\xi,\theta), g(x,m,\xi,\theta)).$$

As $\theta$ is fixed in this section, we simplify notation and denote the values of $g$ by $g(z,\xi)$. We use $u_x, u_\theta, u_m$ to denote vectors in $\mathbb{R}^{nx}, \mathbb{R}^{n\theta}$, and $L(\mathbb{R}^{n\theta}, \mathbb{R}^{nx})$, respectively.

**Proposition 5.8.1.** Define $h: Z \rightarrow \mathbb{R}_{\geq 0}$ as $h(z) = \rho_1 \|A(x)m\| + \rho_2 \|B(x)\| + \rho_3 d_A(x_0, x)$. Then there are $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5$ so that $\{((\Xi, \Sigma, \eta), T, (\|\cdot\|_Z, H)\}$ satisfies a 1-contraction inequality where

$$H(z)(u_x, u_m) = \left((1 + \rho_4 h(z))A(x)u_x, A(x)u_m\right),$$

$$\|(u_x, u_m)\|_Z = \|u_x\| + \rho_5 \|u_m\|.$$

**Proof.** We will apply Proposition 5.5.7 to the map $T(z, \xi) = (f(x,\xi,\theta), g(x,m,\xi))$, to find contraction in the metric $d_H$. The norm $\|\cdot\|_M$ is the usual norm on $M$ induced by $\|\cdot\|_X$ and $\|\cdot\|_\Theta$. For Part iii of Assumption 5.5.4, we have

$$\sup_{\|u_m\|=1} \int_\Xi \|A(f(x,\xi,\theta))\frac{\partial g}{\partial m}(z,\xi)A(x)^{-1}u_m\|\,d\eta(\xi)$$

$$= \sup_{\|u_m\|=1} \int_\Xi \sup_{\|u_x\|=1} \|A(f(x,\xi,\theta))\frac{\partial f}{\partial x}(x,\xi,\theta)A(x)^{-1}u_mu_x\|\,d\eta(\xi) \leq K_X$$

and, directly by assumption,

$$\sup_{\|u_x\|=1} \left(\int_\Xi \|A(f(x,\xi,\theta))\frac{\partial f}{\partial x}(x,\xi,\theta)A(x)^{-1}u_x\|^2\,d\eta(\xi)\right)^{1/2} \leq K_X.$$

We now establish Part i of Proposition 5.5.7. The function $\frac{\partial g}{\partial x}(z,\xi)$ is a linear map from $\mathbb{R}^{nx}$ to $L(\mathbb{R}^{n\theta}, \mathbb{R}^{nx})$, and we identify this with a bilinear map from $\mathbb{R}^{nx} \times \mathbb{R}^{n\theta}$ to $\mathbb{R}^{nx}$. 
Specifically,

\[
\frac{\partial g}{\partial x}(z, \xi)[u_x, u_{\theta}] = \frac{\partial^2 f}{\partial x^2}(x, \xi, \theta)[u_x, m u_{\theta}] + \frac{\partial^2 f}{\partial x \partial \theta}(x, \xi, \theta)[u_x, u_{\theta}]
\]

and \( A(f(x, \xi, \theta)) \frac{\partial g}{\partial x}(z, \xi) A(x)^{-1} \) is the linear map from \( \mathbb{R}^n \times \) to \( L(\mathbb{R}^n, \mathbb{R}^n) \) where

\[
A(f(x, \xi, \theta)) \frac{\partial g}{\partial x}(z, \xi) A(x)^{-1}[u_x, u_{\theta}] =
A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x^2}(x, \xi, \theta)[A(x)^{-1} u_x, m u_{\theta}] + A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x \partial \theta}(x, \xi, \theta)[A(x)^{-1} u_x, u_{\theta}]
\]

For the first term we have, using the assumption on \( \frac{\partial^2 f}{\partial x^2} \) from Assumption 5.4.4 and the identity \( m u_{\theta} = A(x)^{-1} A(x)m u_{\theta} \),

\[
\sup_{\|u_x\|=1} \int_{\Xi} \sup_{\|u_{\theta}\|=1} \|A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x^2}(x, \xi, \theta)[A(x)^{-1} u_x, m u_{\theta}]\| \, d\eta(\xi) \leq K_{X^2} \|A(x)m\|
\]

For the second, use the identity \( u_{\theta} = B(x)^{-1} B(x)u_{\theta} \) and our assumption on \( \frac{\partial^2 f}{\partial x \partial \theta} \),

\[
\sup_{\|u_x\|=1} \int_{\Xi} \sup_{\|u_{\theta}\|=1} \|A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x \partial \theta}(x, \xi, \theta)[A(x)^{-1} u_x, u_{\theta}]\| \, d\eta(\xi) \leq K_X \|B(x)\|
\]

Combining these two inequalities, while assuming \( K_{X^2} \leq \rho_1 \) and \( K_X \leq \rho_2 \),

\[
\sup_{\|u_x\|=1} \int_{\Xi} \|A(f(x, \xi, \theta)) \frac{\partial g}{\partial x}(z, \xi) A(x)^{-1} u_x\| \, d\eta(\xi) \leq K_{X^2} \|A(x)m\| + K_X \|B(x)\|
\]

\[
\leq h(z)
\]

Next, we confirm Part ii of Proposition 5.5.7, by showing the Lyapunov property of the
function \( h \). We consider the three terms of the function, starting with \( \|A(x)m\| \):

\[
\left( \int_{\Xi} \|A(f(x,\xi,\theta))g(z,\xi)\|^2 \, d\eta(\xi) \right)^{1/2} \leq \left( \int_{\Xi} \|A(f(x,\xi,\theta))\frac{\partial f(x,\xi,\theta)}{\partial x}\|^2 \, d\eta(\xi) \right)^{1/2} \\
+ \left( \int_{\Xi} \|A(f(x,\xi,\theta))\frac{\partial f(x,\xi,\theta)}{\partial \theta}\|^2 \, d\eta(\xi) \right)^{1/2} \\
\leq K_X \|A(x)m\| + K_\theta \|B(x)\|
\]

Next is \( \|B(x)\| \). Fix a basepoint \( x_0 \) and set \( B_0 = \left( \int_{\Xi} \|B(f(x_0,\xi,\theta))\|^2 \, d\eta(\xi) \right)^{1/2} \). Then

\[
\left( \int_{\Xi} \|B(f(x,\xi,\theta))\|^2 \, d\eta(\xi) \right)^{1/2} \leq B_0 + \|B\|_{Lip} \left( \int_{\Xi} d_A(f(x_0,\xi,\theta), f(x,\xi,\theta))^2 \, d\eta(\xi) \right)^{1/2} \\
\leq B_0 + \|B\|_{Lip} K_X d_A(x_0, x)
\]

The first inequality uses Assumption 5.4.3 and the second uses the pointwise contraction property of \( f \) which comes from Proposition 5.5.1. For the term \( d_A(x_0, x) \) we have, setting \( D_0 = \left( \int_{\Xi} d_A(x_0, f(x_0,\xi,\theta))^2 \, d\eta(\xi) \right)^{1/2} \),

\[
\left( \int_{\Xi} d_A(x_0, f(x,\xi,\theta))^2 \, d\eta(\xi) \right)^{1/2} \leq D_0 + \left( \int_{\Xi} d_A(f(x_0,\xi,\theta), f(x,\xi,\theta))^2 \, d\eta(\xi) \right)^{1/2} \\
\leq D_0 + K_X d_A(x_0, x)
\]

Combining these we get

\[
\left( \int_{\Xi} h(T(z,\xi))^2 \, d\eta(\xi) \right)^{1/2} \leq \rho_1 K_X \|A(x)m\| + \rho_1 K_\theta \|B(x)\| + (\rho_2 \|B\|_{Lip} K_X + \rho_3 K_X) d_A(x_0, x) + K_4
\]

where \( K_4 = \rho_2 B_0 + \rho_3 D_0 \). Based on this inequality, it is evident that \( \rho_1, \rho_2, \rho_3 \) can be chosen so that the Lyapunov condition on \( h \) is satisfied. Specifically, take \( K_{X_2} \leq \rho_1 \),
max\{K_{X,\Theta}, \rho_1 K_{\Theta}\} < \rho_2\), and \(\rho_2 \|B\|_{Lip} K_X < \rho_3 (1 - K_X)\).

We can use \(h\) to get a Lyapunov function, yielding ergodicity of the sensitivity process:

**Corollary 5.8.2.** Let the \(\rho_1, \rho_2, \rho_3\) of Proposition 5.8.1 be chosen so that they are all positive. Let \(V\) be the function

\[
V(z) = \rho_1 \|A(x)m\| + \rho_2 \|B(x)\| + \rho_3 d_A(x_0, x) + 1.
\]

Then the kernel \(R_\theta\) has a unique invariant measure \(\gamma_\theta \in \mathcal{P}_1(Z, V)\), and for \(\nu \in \mathcal{P}_1(Z, V)\),

\[
\sup_{\|g\|_{Lip(H)} + \|g\|_{V} \leq 1} |\nu R_\theta^n(g) - \gamma_\theta(g)| \to 0 \quad \text{as} \quad n \to \infty.
\]

**Proof.** We apply Proposition 5.5.2, using the metric \(d_H\) defined in Proposition 5.8.1. Proposition 5.8.1 established the pointwise contraction inequality needed for Proposition 5.5.2. For some \(\beta \in [0, 1)\), the inequality

\[
\int_{\Xi} V(T(z, \xi, \theta)) d\eta(\xi) \leq \beta V(z) + (K_4 + 1)
\]

holds at \(z \in Z\), as we have already shown in the proof of Proposition 5.8.1. It remains to show that \(V\) has compact sublevel sets. Note that if \(V(x, m) \leq r\) then \(d_A(x_0, x) \leq \frac{r}{\rho_3}\) and \(\|m\| \leq \frac{rK}{\rho_1}\), where \(K\) is such that \(\sup_{x \in X} \|A(x)^{-1}\| \leq K\). Thus \(V^{-1}[0, r]\) is contained in the compact set \(\{(x, m) \in Z \mid d_A(x_0, x) \leq \frac{r}{\rho_3}\) and \(\|m\| \leq \frac{rK}{\rho_1}\}\).

To ensure that the function \((x, m) \mapsto \frac{\partial e}{\partial x}(x)m\) is integrable for the measure \(\gamma_\theta\) it suffices that it is Lipschitz for the metric \(d_H\), and bounded for Lyapunov function \(V\):

**Proposition 5.8.3.** For any \(e \in E^2\) the map \((x, m) \mapsto \frac{\partial e}{\partial x}(x)m\) is a Lipschitz function in the metric \(d_H\) of Proposition 5.8.1, and is also bounded for the norm \(\cdot \|V\|\).

**Proof.** Let the \(\rho_i\) be as in Proposition 5.8.1. Let \(g(x, m) = \frac{\partial e}{\partial x}(x)m\). We have

\[
\|g(x, m)\| \leq \|\frac{\partial e}{\partial x}\|_A \|A(x)m\| \leq \|e\|_{Lip} \|A(x)m\| \leq \frac{1}{\rho_1} \|e\|_{Lip} \|\varepsilon^2 V(x, m)\|
\]

hence \(\|g\|_V \leq \frac{1}{\rho_1} \|e\|_{Lip}\). Next, we show that \(\|g\|_{Lip} < \infty\) for the metric \(d_H\). This is equivalent to showing \(\|\frac{\partial g}{\partial x}\|_H < \infty\). Let \((u_x, u_m)\) be a vector in \(\mathbb{R}^{n_x} \times L(\mathbb{R}^{n_\Theta}, \mathbb{R}^{n_x})\). Then \(H(z)^{-1}(u_x, u_m)\)
is $H(z)^{-1}(u_x, u_m) = \left(\frac{1}{1 + \rho_4 h(z)} A^{-1}(x) u_x, A(x)^{-1} u_m\right)$ and $\frac{\partial g}{\partial z}(z)$ is the linear map from $\mathbb{R}^{n \times X} \times L(\mathbb{R}^{n \Theta}, \mathbb{R}^{n X})$ to $L(\mathbb{R}^{n \Theta}, \mathbb{R})$ where

$$\frac{\partial g}{\partial z}(z)[u_x, u_m][u_\theta] = \frac{\partial^2 e}{\partial x^2}(x)[u_x, mu_\theta] + \frac{\partial e}{\partial x}(x)[u_m u_\theta]$$

Fix $(u_x, u_m)$ with $\|u_x\| + \rho_5 \|u_m\| = 1$. Then

$$\left\| \frac{\partial g}{\partial z}(z) H(z)^{-1}(u_x, u_m) \right\|$$

$$= \sup_{\|u_\theta\| = 1} \left| \frac{\partial^2 e}{\partial x^2}(x)[A^{-1}(x) u_x, mu_\theta] + \frac{\partial e}{\partial x}(x) A^{-1}(x) u_m u_\theta}{1 + \rho_4 h(z)} \right|$$

$$\leq \sup_{\|u_\theta\| = 1} \frac{\|\partial^2 e\|_{A,A} \|u_x\| \|A(x) m\| \|u_\theta\|}{1 + \rho_4 h(z)} + \|\frac{\partial e}{\partial x}\|_{A} \|u_m\| \|u_\theta\|$$

To continue, note by definition of $h$ that $\|A(x)m\| \leq \frac{1}{\rho_1 \rho_4}$. Then,

$$\leq \frac{\|\partial^2 e\|_{A,A} \|u_x\|}{1 + \rho_4 h(z)} + \frac{\rho_5}{\rho_5} \|\frac{\partial e}{\partial x}\|_{A} \|u_m\|$$

$$\leq \max \left\{ \|\partial^2 e\|_{A,A}, \frac{1}{\rho_1 \rho_4}, \frac{1}{\rho_5} \|\frac{\partial e}{\partial x}\|_{A} \right\}$$

$$\leq \|e\|_{E^2} \max \left\{ \frac{1}{\rho_1 \rho_4}, \frac{1}{\rho_5} \right\}$$

Therefore a Lipschitz constant for the function $g$ is $\|e\|_{E^2} \max \left\{ \frac{1}{\rho_1 \rho_4}, \frac{1}{\rho_5} \right\}$. \(\square\)

We now continue to the proof of Theorem 5.4.5.

**Proof of Theorem 5.4.5.** By Corollary 5.8.2, the forward sensitivity process converges to a unique stationary measure $\gamma_\theta$ in $\mathcal{P}_{1,V}(Z)$. Let $g$ be the function $g(x, m) = \frac{\partial e}{\partial x}(x)m$. By Proposition 5.8.3 we see that $\|g\|_{Lip} + \|g\|_V < \infty$, which means in particular that the integral on the right side of Equation (5.21) is well-defined.
We show that the functional $l$ of (5.21) is bounded for the norm $\| \cdot \|_L$. We have $\| l(e) \| \leq \| e \|_E \int_Z \| A(x)m \| d\gamma_\theta(z)$, with the latter integral being finite since $\gamma_\theta \in P_{1,V}(Z)$. Then $\| l \|_L < \infty$. It remains to show $T(l) = l$. By the identity $\gamma_\theta = \gamma_\theta R_\theta$,

$$l(e) = \int_{X \times M} \frac{\partial e}{\partial x}(x,m) d\gamma_\theta(x,m)$$

$$= \int_{X \times M} \left( \int_{\Xi} \frac{\partial}{\partial x}(f(x,\xi,\theta)) \left( \frac{\partial f}{\partial x}(x,\xi,\theta)m + \frac{\partial f}{\partial \theta}(x,\xi,\theta) \right) d\eta(\xi) \right) d\gamma_\theta(x,m) \quad (5.23)$$

Recall the definition of $T$ is $T(l)e = lP_\theta e + \pi_\theta \frac{\partial}{\partial \theta} P_\theta e$. With our definition of $l$, and applying Proposition 5.7.4, these two terms are

$$lP_\theta(e) = \int_{X \times M} \frac{\partial}{\partial x}(P_\theta e)(x,m) d\gamma_\theta(x,m)$$

$$= \int_{X \times M} \left( \int_{\Xi} \frac{\partial}{\partial x}(f(x,\xi,\theta)) \frac{\partial f}{\partial x}(x,\xi,\theta) d\eta(\xi) \right) m d\gamma_\theta(x,m), \quad (5.24)$$

and

$$\pi_\theta \frac{\partial}{\partial \theta} P_\theta e = \int_X \left( \int_{\Xi} \frac{\partial}{\partial x}(f(x,\xi,\theta)) \frac{\partial f}{\partial \theta}(x,\xi,\theta) d\eta(\xi) \right) d\pi_\theta(x). \quad (5.25)$$

Add Equation (5.24) to Equation (5.25) and compare with Equation (5.23) to see $T(l) = l$. \hfill \Box

To finish this section, let us discuss how this estimator can be implemented. One option is to iterate the joint recursion (5.2a, 5.2b) for a large number of steps, to obtain a sample $(x(t), m(t))$, and then prepare the estimate by forming the product $\Delta(t) = \frac{\partial}{\partial x}(x(t))m(t)$. This requires the ability to compute the derivatives of $e$ and $f$. According to Theorem 5.4.5, the estimate $\Delta(t)$ has the property that $\mathbb{E}[\Delta(t)] \rightarrow \frac{\partial}{\partial \theta} \int_X e(x) d\pi_\theta(x)$ as $t \rightarrow \infty$.

To obtain an estimate with low variance, one can average the iterates. This would involve using the running averages $A(t) = \frac{1}{t} \sum_{i=1}^{t} \Delta(i)$. As we have shown that the underlying process
satisfies an ergodicity property, one could appeal to the recent results of (51) to find a convergence rate of the averaged estimates.

5.9 Examples

Example 5.9.1. We consider a stochastic neural network where at each time only a subset of the edges in the network are activated. There are $N$ nodes so that the state space $X$ is $[0, 1]^N$. The random input is a binary vector in $\Xi = \{0, 1\}^{N \times N}$. Let $\sigma$ be the sigmoid function $\sigma(x) = (1 + \exp(-x))^{-1}$. The function $f : X \times \Xi \times \Theta \to X$ is

$$f_i(x, \xi, \theta) = \sigma(u_i(x, \xi, \theta))$$

where $u_i(x, \xi, \theta) = \sum_{k=1}^{n} \xi_{i,k} \theta_{i,k} x_k$. The $b_i$ are biases and considered fixed. A vector $\xi \in \Xi$ indicates which edges are active at each time step; The edge $(i, j)$ from $j$ to $i$ is only used if $\xi_{i,j} = 1$. The probability measure on $\Xi$ is defined by $\eta(\xi) := \prod_{(i,j) \in E} r^{1-\xi_{i,j}} (1 - r)^{\xi_{i,j}}$. Under this law, in the extreme $r = 1$ we have $\xi_{i,j} = 0$ for all $i, j$ with probability 1. The parameter space $\Theta$ is the $N \times N$ matrices $\mathbb{R}^{N \times N}$, which are the weights $\theta_{i,j}$ between each unit. We set $A(x) = I$ and $\| \cdot \|_X = \| \cdot \|_\infty$, hence $d_A(x, y) = \|x - y\|_\infty$. We set $B(x) = I$. We must find conditions so that Assumptions 5.4.1, 5.4.2, 5.4.3 and 5.4.4 hold. After setting $\Theta$ to be an arbitrary open ball, the only non-trivial part is the contraction criteria, part ii of Assumption 5.4.4. Observe that $\frac{\partial f_i}{\partial x_j}(x, \xi, \theta) = \sigma'(u_i(x, \xi, \theta)) \xi_{i,j} \theta_{i,j}$. With the norm $\| \cdot \|_\infty$ on $X$ and as $|\sigma'(u)| \leq \frac{1}{4}$,

$$\|\frac{\partial f_i}{\partial x_j}(x, \xi, \theta)\|_\infty \leq \frac{1}{4} \|\theta\|_\infty \sup_{i,j} \xi_{i,j}.$$ 

Note that $\left(\int_\Xi (\sup_{i,j} \xi_{i,j})^2 \, d\eta(\xi)\right)^{1/2} = (1 - \eta(\xi = 0))^{1/2} = (1 - r^{|E|})^{1/2}$, so a sufficient condition for contraction in $d_2$ is $\|w\|_\infty (1 - r^{|E|})^{1/2} < 4$. The matrix norm induced by $\| \cdot \|_\infty$ is the maximum absolute row sum; then the condition is that the sum of magnitudes of
incoming weights at each node must be bounded in this way.

The requirements for applying forward sensitivity analysis are met. For completeness we derive the exact form of the sensitivity system. The space $M$ consists of the linear maps from $\mathbb{R}^{N \times N}$ to $\mathbb{R}^N$ and $\frac{\partial f_i}{\partial \theta_{(j,k)}}(x, \xi, \theta) = \delta_{i,j} \sigma'(u_i(x, \xi, \theta)) \xi_{i,k} x_k$. The notation $v_k$ means the $k^{th}$ component of vector $v$.

$$x_i(t+1) = \sigma(u_i(x(t), \xi(t+1), \theta)$$

$$m_{i,(j,k)}(t+1) = \sigma'(u_i(x(t), \xi(t+1), \theta) \left[ \delta_{i,j} \xi_{i,k}(t+1)x_k(t) + \sum_{q=1}^{n} \xi_{i,q}(t+1)\theta(i,q)m_{q,(j,k)}(t) \right]$$

At time $t+1$, node $i$ has to pull from each node $q$ that connects to it the data $m_{q,(j,k)}(t)$ and the state variable $x_q(t)$.

**Example 5.9.2.** Let $\Xi = \mathbb{R}^2$ and let $\eta$ be the law of two independent random variables $\xi_1, \xi_2$, such that $\mathbb{E}[\exp(6|\xi_1|) + |\xi_2|^2] < \infty$. Let $f : \mathbb{R}^2 \times \Xi \times \Theta \to \mathbb{R}^2$ be the function

$$f(x, \xi, \theta) = \left( f_1(x_1, \xi, \theta), f_2(x_1, x_2, \xi, \theta) \right)$$

where $f_1(x_1, \xi, \theta) = \frac{1}{2} x_1 + \theta + \epsilon \xi_1$ and $f_2(x_1, x_2, \xi, \theta) = \frac{1}{2} x_1 x_2 + \epsilon \xi_2$. Let $g_1, g_2$ be the real valued functions $g_1(x) = \exp(2|x_1|)(1 + |x_2|)$ and $g_2(x) = \exp(2|x_1|)$. The metric $d_A$ will be defined using the pair $(\| \cdot \|, A)$ where $\|(u,v)\| = p_1|u| + p_2|v|$ and $A(x) = g_1(x) \oplus g_2(x)$, with $p_1, p_2$ determined below. The parameter $\theta$ is a number and $B$ is $B(x) = g_1(x)$. We seek conditions on $\epsilon$ and $\theta$ that guarantee contraction and the applicability of stochastic forward sensitivity analysis. We find the following:

**Proposition 5.9.3.** Let the following hold

1. The parameter space is $\Theta = (-\frac{1}{4} \log 2, \frac{1}{4} \log 2)$,
ii. $\epsilon < 1$ and \( \left(1 + \epsilon \left( \int_{\Xi} |\xi_2|^2 \, d\eta(\xi) \right)^{1/2} \right) \left( \int_{\Xi} \exp(2\epsilon|\xi_1|)^2 \, d\eta(\xi) \right)^{1/2} < 2^{1/4} \),

iii. The coefficients $p_1, p_2$ are any positive numbers such that $1 + \frac{p_2}{p_1} < 2^{1/4}$.

For $\theta \in \Theta$ the stochastic forward sensitivity method is applicable for the system (5.26).

Proof. See Appendix A.1.1.

Based on the definition of $E^2$, the cost functions are those $e : \mathbb{R}^2 \to \mathbb{R}$ satisfying
\[
\sup_x |\frac{\partial e}{\partial x_i}(x)g_i(x)|^{-1} < \infty \quad \text{and} \quad \sup_x |\frac{\partial^2 e}{\partial x_i \partial x_j}(x)g_i^{-1}(x)g_j^{-1}(x)| < \infty \quad \text{for} \quad 1 \leq i, j \leq 2.
\]

Note that since $g_i \geq 1$ the functions in $E$ include those with $\sup_x \|\frac{\partial e}{\partial x}(x)\| < \infty$ and $\sup_x \|\frac{\partial^2 e}{\partial x^2}(x)\| < \infty$. The joint process takes the following form.
\[
\begin{align*}
x_1(t+1) &= \frac{1}{2} x_1(t) + \theta + \epsilon \xi_1(t+1) \\
x_2(t+1) &= \frac{1}{2} x_1(t)x_2(t) + \epsilon \xi_2(t+1) \\
m_1(t+1) &= \frac{1}{2} m_1(t) + 1 \\
m_2(t+1) &= \frac{1}{2} x_2(t)m_1(t) + \frac{1}{2} x_1(t)m_2(t)
\end{align*}
\]

5.10 Discussion

Our approach to establishing differentiability can be compared with works on measure-valued differentiation, such as (41, 40). The ergodicity framework in those works is based on normed ergodicity (1), while ours is also based on a norm but involves the derivatives of the cost functions as well. The approach to establishing differentiability is based on setting up a certain equation between linear functionals, showing that any solution to that equation must evaluate the stationary derivative, and showing that the equation indeed has a solution. In this sense it is similar to (96), which works with the class of bounded measurable cost functions, and in a different ergodicity framework. The work (67)
CHAPTER 5. STOCHASTIC ATTRACTOR NETWORKS

also used contraction in the Wasserstein distance in an ergodicity framework for stationary gradient estimation.

This work was motivated by derivative estimation and optimization in neural networks. The back-propagation procedure is based on *adjoint sensitivity analysis*, as opposed to the forward sensitivity analysis studied here. Adjoint sensitivity analysis is often preferred as the auxiliary system in this case evolves in a space which has dimension \( n_X \) as opposed to \( n_\theta \times n_X \). In (22, 23) the author analyzed joint gradient estimation/optimization schemes based on adjoint sensitivity analysis. It may be that the methods of this chapter can be extended to adjoint sensitivity analysis.

Another interesting extension may be to recursively apply the construction to obtain estimators for higher derivatives. Calculating \( \frac{\partial^2}{\partial \theta^2} E_{\pi_\theta} [e(x)] \) should be equivalent to computing \( \frac{\partial}{\partial \theta} E_{\gamma_\theta} [g(x)] \) for the “cost function” \( g(x) = \frac{\partial e}{\partial x}(x)m \).

Finally, besides neural networks there should be other interesting applications of these ideas. Any smooth, contracting stochastic system should be subject to this form of gradient system. This includes the iterated function systems which generate fractals (9, 10). Hence a future work might consider how these results may be applied to tune the parameters of an iterated function system in order to produce a fractal with desired properties.
Chapter 6

Discrete Attractor Networks

In this section we discuss a stochastic network on a discrete space known as the Little model (56). Gradient estimation has been studied for closely related models, such as the Boltzmann machine and sigmoid belief networks, as we discuss below. This model is somewhat more challenging since there is not a known closed-form solution for the stationary distribution. One has to focus on gradient estimation methods that only use the Markov kernel associated to the process. We propose an estimator which combines features of SP and MVD. The algorithm generates a random direction, as in SP, and then uses measure valued differentiation to approximate this directional derivative.

Let us make some definitions that are important to the upcoming analysis. The state space is $X = \{0, 1\}^n$, the binary strings of length $n$. We denote by $P_{\theta}(x^0, x^1)$ be the probability of going to state $x^1 \in X$ from state $x^0 \in X$. The function $u_i(x)$, that determines the input to each node at the state $x$ is defined as

$$u_i(x, (w, b)) = \sum_{j=1}^{n} w_{i,j} x_j + b_i.$$  (6.1)
Then
\[ P_\theta(x^0, x^1) = \prod_{i=1}^{n} \sigma(u_i(x^0, \theta))^{x_i^1}(1 - \sigma(u_i(x^0, \theta)))^{1-x_i^1}. \] (6.2)

Alternatively, we can use the following notation of (62): for \( x \in \{0, 1\} \), define
\[ x^\dagger = 2x - 1. \] (6.3)

Using this with the identity \( 1 - \sigma(x) = \sigma(-x) \), we get an equivalent expression to (6.2):
\[ P_\theta(x^0, x^1) = \prod_{i=1}^{n} \sigma((x_i^1)^\dagger u_i(x^0, \theta)). \] (6.4)

The Little model is related to the sigmoid belief network and the Boltzmann machine. If the connectivity graph is acyclic, then one obtains a model resembling the sigmoid belief network. We can enforce this by requiring \( w_{i,j} = 0 \) if \( i < j \). A model like the Boltzmann machine is obtained if the weights are symmetric, meaning \( w_{i,j} = w_{j,i} \). Technically, if one puts a symmetry requirement on our threshold networks, one does not exactly recover the Boltzmann machine, but a variant known as the synchronous or parallel Boltzmann machine (66). The synchronous Boltzmann machine also has a known, simple, stationary distribution (66).

6.1 Ergodicity Properties of the Little Model

Let us first establish the ergodicity properties of the Markov kernel \( P_\theta \) is ergodic. We will work with the total variation metric: The total variation distance between two probability measures \( \mu_1, \mu_2 \) on \( X \) is
\[ d_{TV}(\nu_1, \nu_2) = \sup_{B \subseteq X} |\nu_1(B) - \nu_2(B)| \]
where the supremum is over all subsets of \( X \).
We will show that $P_\theta$ has a contraction property in $d_{TV}$ of the form

$$d_{TV}(\nu_1 P^t, \nu_2 P^t) \leq \alpha^t d_{TV}(\nu_1, \nu_2).$$

The contraction coefficient $\alpha$ will depend on the weights of the network; larger weights will lead to a worse bound on the convergence.

We use the coupling representation of $d_{TV}$ (55, Section 4.2):

$$d_{TV}(\nu_1, \nu_2) = \inf_{\gamma \in \Gamma(\nu_1, \nu_2)} \gamma(\{(x_1, x_2) \mid x_1 \neq x_2\}).$$

This shows that the total variation distance is a special case of the Wasserstein distance, for the case of the discrete metric $d(x, y) = 1_{x \neq y}$. Then according to Proposition 3.5.1, we can show contraction of $P_\theta$ by finding an $\epsilon > 0$ so that

$$d_{TV}(\delta_x P_\theta, \delta_y P_\theta) \leq 1 - \epsilon$$

for all $x \neq y$. To do this, first consider the following general scenario involving two measures $\nu_1, \nu_2$ such that $\min\{\nu_1(x), \nu_2(x)\} \geq \epsilon$ for all $x$. Consider the coupling $\gamma(x, y) = \nu_1(x)\nu_2(y)$ which is simply the product measure. Then

$$d_{TV}(\nu_1, \nu_2) \leq \gamma(\{(x_1, x_2) \mid x_1 \neq x_2\})$$

$$= 1 - \sum_{x \in X} \nu_1(x)\nu_2(x)$$

$$\leq 1 - \sum_{x \in X} \nu_1(x)\epsilon = 1 - \epsilon.$$

Let us calculate this lower bound $\epsilon$ for the transition probability $P_\theta(x_0, x_1)$. According to the product form of $P_\theta$, (6.4), it suffices to find an $\epsilon_1$ such that $\sigma((x_i^0)^\dagger u_i(x^0, \theta)) \geq \epsilon_1$ and
then we can set $\epsilon = \epsilon_1^0$. Note that for any value of $x_i^1$, we have

$$\sigma((x_i^1)^t u_i(x^0, \theta)) \geq \sigma(-|u_i(x^0, \theta)|) \geq \sigma\left(-\sum_{j=1}^{n} |w_{i,j}| - |b_i|\right) \geq \sigma\left(-\|w\|_\infty - \|b\|_\infty\right).$$

where $\|w\|_\infty$ is the $\infty$-norm of the matrix $w$ (defined in Equation (3.5)). Hence we can set $\epsilon = \sigma(-\|w\|_\infty - \|b\|_\infty)^n$, resulting in the formula

$$d_{TV}(\nu_1 P, \nu_2 P) \leq (1 - \sigma(-\|w\|_\infty - \|b\|_\infty)^n)d_{TV}(\nu_1, \nu_2)$$

This has the (not surprising) feature that the predicted speed of convergence decreases as the weights increase. Note that a big difference compared to the deterministic continuous time case is that the number of nodes appears as an extra term.

Given the contraction property, we now consider the differentiability of the stationary distribution. Firstly, we are dealing with a finite state space and a Markov kernel with smooth transition probabilities (due so the smoothness of $\sigma$). Hence we are in a simple setting and the differentiability is relatively easy to establish. For instance Lemma 4 in (67) can be applied. The conclusion is that for any cost function $e$, the stationary expectation $\pi_\theta(e)$ is a differentiable function of $\theta$.

### 6.2 Gradient Estimation

We propose a gradient estimator that works by picking a random direction, as in SP, and computes the directional derivative using measure valued differentiation. In this way one deals with a small number of simulations, as in SP, while avoiding the variance issues with
finite differences. The method is termed simultaneous perturbation measure valued differentiation. The only requirement is that one can compute the measure valued derivative along arbitrary directions. We then consider the method in the context of the Little model.

6.2.1 SPMVD

Definition 6.2.1. Let $\nu_\theta$ be a measure depending on an $n$-dimensional vector parameter $\theta$. Let $v \in \mathbb{R}^n$ be a direction. A triple $(c_{\theta,v}, \nu^+_{\theta,v}, \nu^-_{\theta,v})$ is called a measure valued directional derivative at $\theta$ in the direction $v$ if for all $e : X \to \mathbb{R}$,

$$\frac{\partial}{\partial \theta} \nu_\theta(e) v = c_{\theta,v}[\nu^+_{\theta,v}(e) - \nu^-_{\theta,v}(e)].$$

In practice, one can try to calculate the MVD in direction $v$ as follows. By basic calculus,

$$\frac{\partial}{\partial \theta} \nu_\theta(e) v = \frac{\partial}{\partial \lambda} \bigg|_{\lambda=0} \nu_{\theta+\lambda v}(e).$$

Therefore, to find the MVD of $\nu_\theta$ in direction $v$ it suffices to find the normal, scalar, MVD for $\nu_{\theta+\lambda v}$ at $\lambda = 0$. This is the approach in the following example.

Example 6.2.2. Let $\nu_1, \nu_2, \ldots, \nu_m$ be $m$ probability measures, and for a vector parameter $\theta \in \mathbb{R}^n$ define $\nu_\theta$ as

$$\nu_\theta = \sum_{i=1}^{m} \frac{e^{-\theta_i}}{Z_\theta} \nu_i,$$

where $Z_\theta = \sum_{i=1}^{m} e^{-\theta_i}$. For any function $e : X \to \mathbb{R}$ and direction $v \in \mathbb{R}^m$, then, we have

$$\nu_{\theta+\lambda v}(e) = \sum_{i=1}^{m} \frac{e^{-(\theta_i+\lambda v_i)}}{Z_{\theta+\lambda v}} \nu_i(e).$$

(6.5)

Recalling the notation $\gamma^+, \gamma^-$ for the positive and negative part of a number respectively,
we make use of the following identities: $v = \gamma^+ - \gamma^-$ and $|v| = \gamma^+ + \gamma^-$. Define $K_{\theta,v} = \sum_{j=1}^{m} e^{-\theta_j |v_j|}$. Differentiating (6.5) at $\lambda = 0$, and doing some algebra, one can get the following representation for the directional derivative:

$$
\frac{\partial}{\partial \theta} \nu_{\theta}(e)\nu = \frac{\partial}{\partial \lambda} \bigg|_{\lambda=0} \nu_{\theta + \lambda v}(e) = c_{\theta,v} \left[ \sum_{i=1}^{m} \alpha_{i,v} \nu_i(e) - \sum_{i=1}^{m} \beta_{i,v} \nu_i(e) \right]
$$

where

$$
c_{\theta,v} = \frac{K_{\theta,v}}{Z_{\theta}},
$$

$$
\alpha_{i,v} = \frac{1}{Z_{\theta}K_{\theta,v}} \left[ v_i^- Z_{\theta} + \sum_{j=1}^{m} e^{-\theta_i v_j^+} \right],
$$

and

$$
\beta_{i,v} = \frac{1}{Z_{\theta}K_{\theta,v}} \left[ v_i^+ Z_{\theta} + \sum_{j=1}^{m} e^{-\theta_i v_j^-} \right].
$$

Therefore the triple $(c_{\theta,v}, \sum_{i=1}^{m} \alpha_{i,v} \nu_i, \sum_{i=1}^{m} \beta_{i,v} \nu_i)$ is the measure valued derivative of $\nu$ at the parameter $\theta$ in the direction $v$.

Defining the extension to Markov chains is straightforward.

**Definition 6.2.3.** A triple $(c_{\theta,v}, P_{\theta,v}^+, P_{\theta,v}^-)$ is a measure valued derivative for the Markov kernel $P$ at $\theta$ in the direction $v$ if for each $x$, $(c_{\theta,v}(x), P_{\theta,v}^+(x, \cdot), P_{\theta,v}^-(x, \cdot))$ is an MVD at $\theta$ in the direction $v$ for the measure $P_{\theta}(x, \cdot)$ in the sense of Definition 6.2.1.

The gradient estimator for stationary costs proceeds by choosing a random direction, and applying the stationary MVD procedure (Algorithm 4). The pseudocode is presented as Algorithm 6.

This estimator will be applied to our motivating example, the Little networks.
Algorithm 6: Simultaneous perturbation measure valued differentiation (SPMVD)

- Sample a random direction $v$ according to the distribution $p$:

$$p(v) = \prod_{i=1}^{n} \left[ \frac{1}{2} \delta_{-1/2}(v_i) + \frac{1}{2} \delta_{1/2}(v_i) \right]$$  \hspace{1cm} (6.6)

- Let $(c_{\theta,v}, P_{\theta,v}^+, P_{\theta,v}^-)$ be a measure valued derivative for $P_\theta$ in the direction $v$.
- Pass $P_\theta$ and $(c_{\theta,v}, P_{\theta,v}^+, P_{\theta,v}^-)$ to scalar MVD (Algorithm 4), to obtain the number $\Delta_{MVD,v}$.
- return $\Delta_{MVD,v}$.

6.2.2 Application to the Little model

Let us now give the measure valued directional derivatives for the Little model. Fix a parameter $\theta$. A direction in the parameter space is represented as a vector $v \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$, where $v_{i,j}$ is a direction along the weight from $j$ to $i$ and $v_i$ is a direction along the bias at node $i$. Using the definitions (6.1), (6.3), (6.4), and after some algebra, one can obtain the following expression for the directional MVD:

$$\frac{\partial}{\partial \theta} \left[ \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \right] v = c_{\theta,v}(x^0) \left[ \sum_{x^1} e(x^1) P_{\theta,v}^+(x^0, x^1) - \sum_{x^1} e(x^1) P_{\theta,v}^-(x^0, x^1) \right],$$  \hspace{1cm} (6.7)

where

$$c_{\theta,v}(x) = \sum_{i=1}^{n} |v_i| \sigma(u_i(x^0, \theta)) + \sum_{i=1}^{n} \sum_{j=1}^{n} |v_{i,j}| x^0_j \sigma(u_i(x^0, \theta)),$$

$$P_{\theta,v}^+(x^0, x^1) = \frac{P_\theta(x^0, x^1)}{c_\theta(x^0)} \sum_{i=1}^{n} \left[ x^1_i \left( v^+ + \sum_{j=1}^{n} v_{i,j} x^0_j \right) + \sigma(u_i(x^0, \theta)) \left( v^- + \sum_{j=1}^{n} v_{i,j} x^0_j \right) \right],$$  \hspace{1cm} (6.8)

and

$$P_{\theta,v}^-(x^0, x^1) = \frac{P_\theta(x^0, x^1)}{c_\theta(x^0)} \sum_{i=1}^{n} \left[ x^1_i \left( v^- + \sum_{j=1}^{n} v_{i,j} x^0_j \right) + \sigma(u_i(x^0, \theta)) \left( v^+ + \sum_{j=1}^{n} v_{i,j} x^0_j \right) \right].$$  \hspace{1cm} (6.9)
This yields two Markov kernels $P^{+}_{\theta,v}$ and $P^{-}_{\theta,v}$, that depend not just on the parameter $\theta$ (as would be the case in scalar MVD) but also the direction of interest $v$.

In order for SPMVD to be useful, there must be a practical procedure for running the Markov kernels $P^{+}_{\theta,v}$ and $P^{-}_{\theta,v}$. The Little networks operate on a large state space, of size $2^n$ when $n$ nodes are used, so this is not necessarily trivial. The original Markov kernel $P_\theta$ has a relatively simple structure, allowing each node to be updated independently (see Equation (6.4)), and one may hope for a similar situation with the MVD pair $P^{+}_{\theta,v}, P^{-}_{\theta,v}$.

To investigate this, let us consider the Markov kernel $P^{+}_{\theta,v}$. Fix an $x_0$ and a $\theta$. We want to see how to sample from $P^{+}_{\theta,v}(x_0, x^1)$.

Define the following variables:

\[
\begin{align*}
\text{Pi. } & d = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma(u_i(x_0, \theta)) v_{i,j} x_0^j + \sum_{i=1}^{n} \sigma(u_i(x_0, \theta)) v_i \\
\text{Pii. } & a_i = v_i^+ + \sum_{j=1}^{n} v_{i,j} x_0^j, \quad i = 1, \ldots, n, \\
\text{Piii. } & \beta_i = \sigma(u_i(x_0, \theta)), \quad i = 1, \ldots, n, \\
\text{Piv. } & c = c_{\theta,v}(x_0).
\end{align*}
\]

Each of these depends on $x_0, \theta$ and $v$. Then the probability measure $Q(x) = P^{+}_{\theta,v}(x_0, x)$ has the representation

\[
Q(x) = \frac{1}{c} \left( d + \sum_{i=1}^{n} x_i a_i \right) \prod_{i=1}^{n} \beta_i^x (1 - \beta_i)^{1-x_i}.
\]  
(6.10)

We will sample from this distribution by sequentially generating random variable $x_1, \ldots, x_n$. First we will calculate and sample from the marginal distribution $Q(x_1)$. Then we sample from the conditional distribution $Q(x_2 \mid x_1)$, followed by sampling from $Q(x_3 \mid x_1, x_2)$ and so on until finally sampling from $Q(x_n \mid x_1, \ldots, x_{n-1})$. This is a standard technique for
generating random vectors (see Section 4.6 of (77)). Each time we are sampling from a probability measure on the two outcomes 0 and 1, and as we shall see it is feasible since the conditional probabilities are easy to compute. We will calculate the complexity of the sampling, and show it to be $O(n^2)$ when dealing with $n$-nodes. That is, we are going to describe a procedure whose input is a state $x^0$ and whose output is distributed according to $P_{\theta,v}(x^0, x)$, and this procedure takes $O(n^2)$ steps to run - proportional to the number of parameters.

To start, note that by the definition of conditional probability,

$$Q(x_k | x_{k-1}, \ldots, x_1) = \frac{Q(x_k, x_{k-1}, \ldots, x_1)}{Q(x_{k-1}, \ldots, x_1)} \tag{6.11}$$

Based on this equation, if we can compute the marginal probabilities quickly then we can compute the conditional probabilities quickly. Starting from the formula (6.10), one can show that for $x_1 \in \{0, 1\}$,

$$Q(x_1) = \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} Q(x_1, x_2, \ldots, x_n)$$

$$= \frac{1}{c} \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \left( d + \alpha_1 x_1 + \sum_{k=2}^{n} \alpha_i \beta_i \right) \tag{6.12}$$

and for any $(x_k, \ldots, x_1) \in \{0, 1\}^k$,

$$Q(x_k, x_{k-1}, \ldots, x_1) = \frac{1}{c} \left( d + \sum_{i=1}^{k} \alpha_i x_i + \sum_{i=k+1}^{n} \alpha_i \beta_i \right) \prod_{i=1}^{k} \beta_i^{x_i} (1 - \beta_i)^{1-x_i}. \tag{6.13}$$

See Appendix A.2.2 for a derivation of (6.12) and (6.13). Based on these formulas, we construct the sampling algorithm (Algorithm 7). Using the identity (6.13), we can write
Algorithm 7: Sampling from a measure of the form (6.10)

Set $\delta_1 = Q(x_1 = 1)$ via Equation (6.12).
Set $x_1 = 1$ with probability $\delta_1$, otherwise $x_1 = 0$.

for $k = 2, \ldots, n$ do

Set $\delta_k = Q(x_k = 1 \mid x_{k-1}, \ldots, x_1)$ via Equation (6.14).
Set $x_k = 1$ with probability $\delta_k$, otherwise $x_k = 0$.

end

return $(x_1, \ldots, x_n)$.

(6.11) as

$$Q(x_k = 1 \mid x_{k-1}, \ldots, x_1) = \frac{\left( d + \sum_{i=1}^{k-1} \alpha_i x_i + \alpha_k + \sum_{i=k+1}^{n} \beta_i \alpha_i \right) \beta}{d + \sum_{i=1}^{k-1} \alpha_i x_i + \sum_{i=k}^{n} \beta_i \alpha_i}$$

(6.14)

The following proposition certifies the correctness of Algorithm 7.

Proposition 6.2.4. For any data $c, d, a_1, \ldots, a_n, \beta_1, \ldots, \beta_n$, the output of Algorithm 7 is distributed as follows:

$$p(x_1, \ldots, x_n) = \frac{1}{c} \left( d + \sum_{i=1}^{n} x_i a_i \right) \prod_{i=1}^{n} \beta_i^{x_i} (1 - \beta_i)^{1-x_i}.$$  

Proof. This follows from the identity (6.11) and the computational formulas (6.12), (6.13), whose correctness is established in the appendix.

Analogous to the variables (Pi) - (Piv), we have definitions for $P^-$:

Mi. $d = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma(u_i(x^0, \theta)) v_{i,j}^+ x_j^0 + \sum_{i=1}^{n} \sigma(u_i(x^0, \theta)) v_i^+$

Mii. $a_i = v_i^- + \sum_{j=1}^{n} v_{i,j}^- x_j^0, \ i = 1, \ldots, n$

Miii. $\beta_i = \sigma(u_i(x^0, \theta)), \ i = 1, \ldots, n$.

Miv. $c = c_{\theta,v}(x^0)$. 

**Algorithm 8**: MVD gradient estimation for the Little model

- Sample a random direction $v$ from the distribution (6.6).

for $t = 0, 1, \ldots, M^0 - 1$ do
  - Sample $x(t + 1)$ from $P_\theta(x(t), \cdot)$
end

- Calculate $c^+, d^+, a^+_1, \ldots, a^+_n, \beta^+_1, \ldots, \beta^+_n$ according to formulas (Pi) - (Piv).
- Run Algorithm 7 using data $c^+, d^+, \ldots,$ to obtain $x^+(0)$.

- Calculate $c^-, d^-, a^-_1, \ldots, a^-_n, \beta^-_1, \ldots, \beta^-_n$ according to formulas (Mi) - (Miv).
- Run Algorithm 7 using data $c^-, d^-, \ldots,$ to obtain $x^-(0)$.

for $t = 0, 1, \ldots, M^1 - 1$ do
  Sample $x^+(t + 1)$ from $P_\theta(x^+(t), \cdot)$
  Sample $x^-(t + 1)$ from $P_\theta(x^-(t), \cdot)$
end

- Set $\Delta^{MVD} = c_\theta(x(M^0)) \sum_{t=0}^{M^1} [e(x^+(t)) - e(x^-(t))]$

return $\Delta^{MVD}$.

**Corollary 6.2.5.** Let $c, d, a_1, \ldots, a_n, \beta_1, \ldots, \beta_n$ be defined as in (Pi) - (Piv). Then the distribution of the output of Algorithm 7 is $P_{\theta,v}^+(x_0, x_1)$.

Alternatively, if the $c, d, a_1, \ldots, a_n, \beta_1, \ldots, \beta_n$ are defined as in (Mi) - (Miv) then the distribution of the output of Algorithm 7 is $P_{\theta,v}^-(x_0, x_1)$.

One can use Algorithm 7 to simulate the Markov chains $P_{\theta,v}^+$ and $P_{\theta,v}^-$ that are needed in the directional MVD algorithm. Two copies would be used - one with the data for $P_{\theta,v}^+$ and one for $P_{\theta,v}^-$.

The gradient estimation procedure is summarized below. It is the MVD gradient estimation algorithm (Algorithm 4), customized to include random directions and the special sampling algorithm (Algorithm 7).

In the next section we empirically investigate the behavior of Algorithm 8.
Figure 6.1: Error trajectories of MVD-based optimization for different values of $M^1$.

6.3 Numerical Experiments

We have implemented Algorithm 8 and to estimate gradients in a neural network training procedure. The network was trained to classify digits from the MNIST dataset. The network was trained using the first 5000 digits of the dataset. We report how the empirical error evolved during training. The network has $784 + 10 = 794$ fully connected units. The first 784 units receive input from the image pixels and the last ten nodes are output units. Each weights $w_{i,j}$ is initialized with a sample from a uniform distribution on $[-0.01, 0.01]$. The biases $b_i$ are initialized with a sample from the same distribution. The parameters of Algorithm 8 were set up as follows. The value of $M^0$ was 10 and $M^1$ was 50. Common random numbers were used in the routines for updating $x^+$ and $x^-$ for variance reduction. The algorithm ran for 30000 parameter updates and the empirical error is reported every
500 updates.

We experimented with different choices for the step sizes, either $\epsilon = 0.001$ or $\epsilon = 0.0001$. Results are shown in Figure 6.1. The trend is similar but using $M^1 = 50$ iterations inside Algorithm 8 seems to have less variance.

### 6.4 Discussion

In this section we studied gradient estimation for the Little model. Since a closed form solution for the stationary distribution is not known, methods which work for similar models such as the Boltzmann machine or Sigmoid belief network cannot be used. To address this we introduced a method to calculate derivatives in the Little model based on measure-valued differentiation. To get a gradient estimate this way, one has to run the Little model for some time to get an initial state, then generate a random direction, and then run two Markov kernels $P_{\theta,v}^+$ and $P_{\theta,v}^-$ and use the observed errors in both chains to form the gradient estimate. There are several parameters of the method - the $M^0$ and $M^1$ of Algorithm 8. Future work will study the dependence of the algorithm performance, such as bias and variance, on these parameters. A more detailed numerical study will also aid in tuning these parameters. The author believes the general idea of pairing random directions with measure-valued-differentiation could enable optimization in other models as well.
Chapter 7

Conclusion

In this thesis we investigated gradient estimation and optimization for several neural network models that involve feedback. Each of these models required some form of contraction property. The contraction property guarantees their long-term behavior is well-defined. In deterministic systems, we required that the dynamics converge to a fixed-point. For stochastic systems, we required some form of ergodicity. For the systems on a continuous state space, the contraction conditions are expressed using bounds on derivatives of the underlying system. For discrete neural networks, we found contraction with respect to the total variation norm. For the stochastic systems on a continuous state space, we found contraction with respect to a Wasserstein distance.

To analyze the gradient estimation process in the continuous setting, we constructed a metric on the auxiliary state space and found contraction in this metric. In the deterministic setting, this metric was simple (a sum of norms). For stochastic systems, however, we went to a more general class of metric. To allow very general systems, we this simple approach is unsuitable. To allow probabilistically interesting systems, we found it necessary to move to a more general geometric setting. Inspired by results for deterministic contracting systems, we introduced some novel results on hierarchies of contracting stochastic systems. These
results were used to prove the contraction properties of the gradient estimation system. The two contributions of this chapter were (1) the novel contraction framework, including the interconnection results, and (2) the proof of correctness for forward sensitivity analysis. These should be generally useful for the analysis and optimization of contracting stochastic systems. We showed that the results are applicable in particular to a stochastic neural network model where the connections between nodes behave randomly.

For the case of deterministic attractor network we considered not just gradient estimation but also optimization. The optimization procedure alternates between gradient estimation and a parameter update in an approximate gradient direction. To ensure that a function decrease occurs, the gradients must be accurate. We showed how to tune the optimization procedure to guarantee the gradient estimates are accurate. This enabled us to apply a basic theorem on gradient descent. Previous work did not give a detailed treatment of this problem. Future work will try to port this convergence analysis to the stochastic setting. We are also interested in attempting the treatment of optimization in a more general geometric setting, in order to relax the strict bounds on the derivatives that the analysis currently requires.

For the case of discrete attractor networks, we investigated gradient estimation using measure valued differentiation. This results in yet another gradient algorithm that is dynamical process inheriting the contraction property of the system of interest. The model of interest, the Little model, is interesting as special cases have attracted a lot of interest in machine learning applications. In these special cases, formulas for the stationary distribution can be calculated and these formulas serve as the basis for optimization. Our contributions in this area were to show how a gradient estimator based on simultaneous perturbation and measure valued derivatives can be constructed for this more general model. We showed also that the gradient estimator can be implemented efficiently. The theoretical developments in this chapter only concerned gradient estimation. Future work in this area will theoretically
investigate a combined optimization/gradient estimation procedure for the Little model.
Appendix

A.1 Derivations Related to Stochastic Forward Sensitivity Analysis

A.1.1 Proof of Proposition 5.9.3

We verify Assumptions 5.4.1 - 5.4.4. For Assumption 5.4.1, the continuity is obvious. As $A$ has a diagonal structure, $\|A(x)^{-1}\| = \max\{g_1(x)^{-1}, g_2(x)^{-1}\}$, so it is clear that $\|A(x)^{-1}\| \leq 1$ for all $x$.

For Assumption 5.4.2, the differentiability is evident. For the integrability, using the base-point $(0, 0)$ it suffices that $\left(\int_{\Xi} d_A(0, f(x, \xi, \theta))^2 d\nu(\xi)\right)^{1/2} < \infty$ for any $(x, \theta) \in X \times \Theta$.

Consider the curve $t \mapsto t f(x, \xi, \theta)$, for $t \in [0, 1]$, from 0 to $f(x, \xi, \theta)$. Then $d_A(0, f(x, \xi, \theta)) \leq \int_0^1 \|A(t f(x, \xi, \theta))f(x, \xi, \theta)\| dt$. Next, by definition of $\| \cdot \|$, 

$$\|A(t f(x, \xi, \theta))f(x, \xi, \theta)\| = p_1 |g_1(t f(x, \xi, \theta))f_1(x, \xi, \theta)| + p_2 |g_2(t f(x, \xi, \theta))f_2(x, \xi, \theta)|$$
For the first term on the right hand side of this equation we have

\[
|g_1(t f(x, \xi, \theta)) f_1(x, \xi, \theta)| = \exp(2|t \frac{1}{2} x_1 + t \theta + t \epsilon \xi_1|) (1 + |t \frac{1}{2} x_1 x_2 + t \epsilon \xi_2|) \frac{1}{2} x_1 + \theta + \epsilon \xi_1 |
\leq \exp(|x_1|) \exp(2|\theta|) \exp(2|\epsilon \xi_1|) (1 + \frac{1}{2} |x_1||x_2| + \epsilon |\xi_2|) (\frac{1}{2} |x_1| + |\theta| + \epsilon |\xi_1|)
\leq \exp(2|x_1| + |x_1||x_2|) \exp(2|\theta|) \exp(3|\epsilon \xi_1|) (1 + \epsilon |\xi_2|)
\]

In the last inequality we used the fact that \( \theta < \frac{1}{2} \). Likewise, for the second term,

\[
|g_2(t f(x, \xi, \theta)) f_2(x, \xi, \theta)| = \exp(2|t \frac{1}{2} x_1 + t \theta + t \epsilon \xi_1|) \frac{1}{2} x_1 x_2 + \epsilon \xi_2 |
\leq \exp(|x_1| + |x_1 x_2|) \exp(2|\theta|) \exp(2|\epsilon \xi_1|) \epsilon |\xi_2|
\]

Combining these we obtain a bound for \( d_A(0, f(x, \xi, \theta)) \):

\[
d_A(0, f(x, \xi, \theta)) \leq p_1 \exp(2|x_1| + |x_1 x_2|) \exp(2|\theta|) \exp(3|\epsilon \xi_1|)(1 + \epsilon |\xi_2|)
+ p_2 \exp(|x_1| + |x_1 x_2|) \exp(2|\theta|) \exp(2|\epsilon \xi_1|) \epsilon |\xi_2|
\leq (p_1 + p_2) \exp(2|x_1| + |x_1 x_2|) \exp(2|\theta|) \exp(3|\epsilon \xi_1|)(1 + \epsilon |\xi_2|)
\quad \text{(A.1)}
\]

Let \( Q = (\int_{\Xi} |\xi_2|^2 d\nu(\xi))^{1/2} \) and set \( R = (\int_{\Xi} \exp(2|\epsilon \xi_1|)^2 d\nu(\xi))^{1/2} \). Squaring and integrating (A.1) yields

\[
\left( \int_{\Xi} d_A(0, f(x, \xi, \theta))^2 d\nu(\xi) \right)^{1/2}
\leq (p_1 + p_2) \exp(2|x_1| + |x_1 x_2|) \exp(2|\theta|) \left( \int_{\Xi} \exp(3|\epsilon \xi_1|)^2 d\nu(\xi) \right)^{1/2} (1 + \epsilon Q)
\]

which is finite by assumption that \( \exp(6|\xi_1|) \) is integrable and that \( \epsilon < 1 \).

For Assumption 5.4.3, the invertibility of \( B(x) \) follows since \( g_1 > 1 \). Next, we show
\[ \|B(x)\| \] is Lipschitz for \( d_A \). Since \( \|e\|_{L^p} = \| \frac{\partial e}{\partial x} \|_{A} \) when \( e \) is differentiable, the Lipschitz continuity of \( g_1 \) can be shown as follows. Let \( x = (x_1, x_2) \) be a point of differentiability for \((|x_1|, |x_2|)\), and let \( p_1|u| + p_2|v| = 1 \). Then

\[
\left| \frac{\partial g_1}{\partial x}(x)A(x)^{-1}(u, v) \right| = \left| \frac{\partial g_1}{\partial x}(x)(g_1(x)^{-1}u, g_2(x)^{-1}v) \right| \\
= \left| \frac{\partial g_1}{\partial x}(x)g_1(x)^{-1}u + \frac{\partial g_1}{\partial x_2}(x)g_2(x)^{-1}v \right| \\
\leq \max \left\{ \frac{1}{p_1} \left| \frac{\partial g_1}{\partial x_1}(x)g_1(x)^{-1} \right|, \frac{1}{p_2} \left| \frac{\partial g_1}{\partial x_2}(x)g_2(x)^{-1} \right| \right\}
\]

where \( \left| \frac{\partial g_1}{\partial x_1}(x)g_1(x)^{-1} \right| \leq 2 \) and \( \left| \frac{\partial g_1}{\partial x_2}(x)g_2(x)^{-1} \right| \leq 1 \). By an argument using a mollification of \( |\cdot| \), this is extended to all points of \( X \). Therefore \( \|g\|_{L^p} \leq \max \left\{ \frac{2}{p_1}, \frac{1}{p_2} \right\} \). We turn to the functions \( L_{X^t, \Theta^t} \), starting with \( L_X \). Observe the inequalities

\[
g_1(f(x, \xi, \theta)) \left| \frac{\partial f_1}{\partial x_1}(x, \xi, \theta) \right| g_1(x)^{-1} \\
\leq \frac{1}{2} \exp(2|\theta|) \exp(2\epsilon|x_1|) \exp(|x_1|)(1 + \frac{1}{2}|x_1| + \epsilon|x_2|) \exp(-2|x_1|), \tag{A.2}
\]

\[
g_2(f(x, \xi, \theta)) \left| \frac{\partial f_2}{\partial x_1}(x, \xi, \theta) \right| g_1(x)^{-1} \\
\leq \frac{1}{2} \exp(2|\theta|) \exp(2\epsilon|x_1|) \exp(|x_1|) \exp(-2|x_1|), \tag{A.3}
\]

and

\[
g_2(f(x, \xi, \theta)) \left| \frac{\partial f_2}{\partial x_2}(x, \xi, \theta) \right| g_2(x)^{-1} \\
\leq \frac{1}{2} \exp(2|\theta|) \exp(2|\epsilon x_1|) \exp(|x_1|) \exp(-2|x_1|). \tag{A.4}
\]
Next, note that

\[
\| A(f(x, \xi, \theta)) \frac{\partial f}{\partial x}(x, \xi, \theta) A(x)^{-1} \| \\
\leq \max \left\{ g_1(f(x, \xi, \theta)) \frac{\partial f_1}{\partial x_1}(x, \xi, \theta) g_1(x)^{-1} + \frac{p_2}{p_1} g_2(f(x, \xi, \theta)) \frac{\partial f_2}{\partial x_2}(x, \xi, \theta) g_2(x)^{-1}, \\
g_2(f(x, \xi, \theta)) \frac{\partial f_2}{\partial x_2}(x, \xi, \theta) g_2(x)^{-1} \right\}
\]

Combining this with the three inequalities (A.2), (A.3), (A.4), we get

\[
\| A(f(x, \xi, \theta)) \frac{\partial f}{\partial x}(x, \xi, \theta) A(x)^{-1} \| \\
\leq \frac{1}{2} \exp(2|\theta|) \exp(2\epsilon|\xi_1|) \exp(|x_1|) \max\{1 + \frac{1}{2}|x_1| + \epsilon|\xi_2| + \frac{p_2}{p_1}, |x_1|\} \exp(-2|x_1|) \\
\leq \frac{1}{2} \exp(2|\theta|) \exp(2\epsilon|\xi_1|) \exp(|x_1|)[1 + |x_1| + \epsilon|\xi_2| + \frac{p_2}{p_1}] \exp(-2|x_1|)
\]

Squaring and integrating the right-hand side of the last inequality, and using the independence of the \(\xi_1\) and \(\xi_2\) variables yields

\[
L_X(x, \theta) \leq \frac{1}{2} \exp(2|\theta|) R \exp(|x_1|)(1 + \epsilon Q + \frac{p_2}{p_1} + |x_1|) \exp(-2|x_1|)
\]

This is a continuous function of \((x, \theta)\), so the continuity of \(L_X\) holds. We now show the contraction property. Using the inequality \(a + x \leq a \exp\left(\frac{x}{a}\right)\) we get

\[
\leq (1 + \epsilon Q + \frac{p_2}{p_1}) \frac{1}{2} \exp(2|\theta|) R \exp\left(\left[1 + (1 + \epsilon Q + \frac{p_2}{p_1})^{-1}\right] |x_1|\right) \exp(-2|x_1|)
\]

Based on this, the contraction property holds if \(\epsilon, \theta, p_1, p_2\) are such that \((1 + \epsilon Q + \frac{p_2}{p_1}) \exp(2|\theta|) R < 2\) and one can verify that Assumptions (i), (ii) and (iii) mean that this indeed is the case. Now consider \(L_\theta\). Let \(\| \cdot \|_\Theta = | \cdot |\). Then \(\| A(f(x, \xi, \theta)) \frac{\partial f}{\partial \Theta}(x, \xi, \theta) B(x)^{-1} \| = \)
$g_1(f(x, \xi, \theta))g_1(x)^{-1}$. Using a similar analysis as above,

$$g_1(f(x, \xi, \theta))g_1(x)^{-1} \leq \exp(2|\theta|) \exp(2\epsilon|\xi_1|) \exp(|x|)(1 + \frac{1}{2}|x_1| + \epsilon|\xi_2|) \exp(-2|x_1|)$$

Squaring and integrating the right-hand side of this equation yields

$$L_\Theta(x, \theta) \leq \exp(2|\theta|) R \exp(|x_1|)(1 + \epsilon Q) \exp(-2|x_1|) \leq (1 + \epsilon Q) \exp(2|\theta|) R$$

From the first inequality we can see that $L_\Theta$ is continuous. From the last we can see that $L_\Theta$ is bounded on the set $X \times \Theta$. It remains to verify conditions on the higher derivatives.

The higher derivatives vanish except for $\frac{\partial^2 f}{\partial x^2}$. This is defined as follows

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x, \xi, \theta) = \begin{cases} \frac{1}{2} & \text{if } k = 2 \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

For $i = 1, 2$ we have $A(x)^{-1}e_i = g_i^{-1}(x)e_i$ and by basic properties of bilinear maps,

$$A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x_i \partial x_j}(z)(A(x)^{-1}e_i, A(x)^{-1}e_j) = A(f(x, \xi, \theta))g_i^{-1}(x)g_j^{-1}(x) \frac{\partial^2 f}{\partial x_i \partial x_j}(x, \xi, \theta)$$

Note that $\frac{\partial^2 f}{\partial x_i \partial x_j}(x, \xi, \theta) = 0$ if $i = j$. When $i \neq j$ we have $\frac{\partial^2 f}{\partial x_i \partial x_j}(x, \xi, \theta) = (0, \frac{1}{2})$ and $A(f(x, \xi))g_i^{-1}(x)g_2^{-1}(x)(0, \frac{1}{2}) = \left(0, g_2(f(x, \xi))g_1^{-1}g_2^{-1}(x)\right)$. Then for any $i, j$,

$$\|A(f(x, \xi, \theta)) \frac{\partial^2 f}{\partial x_i \partial x_j}(x, \xi, \theta)(A(x)^{-1}e_i, A(x)^{-1}e_j)\| \leq p_2 g_2(f(x, \xi, \theta))g_1(x)^{-1}g_2(x)^{-1}.$$
With this we get

\[
\|A(f(x, \xi, \theta))\frac{\partial^2 f}{\partial x^2}(x, \xi, \theta)(A(x)^{-1} \oplus A(x)^{-1})\| \\
\leq \left( \max\left\{ \frac{1}{p_1}, \frac{1}{p_2} \right\} \right)^2 p_2 g_2(f(x, \xi, \theta))g_2(x)^{-1} \\
= \max\left\{ \frac{p_2}{p_1}, \frac{1}{p_2} \right\} g_2(f(x, \xi, \theta))g_2(x)^{-1} = \max\left\{ \frac{p_2}{p_1}, \frac{1}{p_2} \right\} \exp(2|\theta| + \epsilon|\xi|)
\]

Integrating yields

\[
L_{X^2}(x, \theta) \leq \max\left\{ \frac{p_2}{p_1}, \frac{1}{p_2} \right\} \exp(2|\theta|) \int_{\Xi} \exp(2\epsilon|\xi|) \, d\nu(\xi),
\]

which is bounded and continuous on $X \times \Theta$.

### A.2 Derivations Related to the Little Model

#### A.2.1 Derivation of Equation 6.7

Fix an $x^0$ and a direction $v \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$. Then

\[
\frac{\partial}{\partial \lambda} P_{\theta+\lambda v}(x^0, x^1) = P_{\theta+\lambda v}(x^0, x^1) \frac{\partial}{\partial \lambda} \log P_{\theta+\lambda v}(x^0, x^1) \\
= P_{\theta+\lambda v}(x^0, x^1) \sum_{i=1}^n \frac{\partial}{\partial \lambda} \log \left( \sigma((x^1_i)^\dagger u_i(x^0, \theta + \lambda v)) \right) \\
= P_{\theta+\lambda v}(x^0, x^1) \sum_{i=1}^n (1 - \sigma((x^1_i)^\dagger u_i(x^0, \theta + \lambda v))) \, (x^1_i)^\dagger \frac{\partial}{\partial \lambda} u_i(x^0, \theta + \lambda v) \\
= P_{\theta+\lambda v}(x^0, x^1) \sum_{i=1}^n (1 - \sigma((x^1_i)^\dagger u_i(x^0, \theta + \lambda v))) \, (x^1_i)^\dagger \left( \sum_{j=1}^n v_{i,j} x^0_j + v_i \right)
\]

Evaluating this at $\delta = 0$ we find that

\[
\frac{\partial}{\partial \theta} P_{\theta}(x^0, x^1)v = P_{\theta}(x^0, x^1) \sum_{i=1}^n (1 - \sigma((x^1_i)^\dagger u_i(x^0, \theta))) (x^1_i)^\dagger \left( \sum_{j=1}^n v_{i,j} x^0_j + v_i \right) \quad (A.5)
\]
Note also that
\[(1 - \sigma(x^\top u))x^\top = \begin{cases} (1 - \sigma(u)) & \text{if } x = 1 \\ -\sigma(u) & \text{if } x = 0 \end{cases}\]
which means
\[(1 - \sigma(x^\top u))x^\top = x - \sigma(u) \quad (A.6)\]

Combining (A.5) and (A.6),
\[
\frac{\partial}{\partial \theta} \sum_{x^1} e(x^1) P_\theta(x^0, x^1)v \\
= \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \sum_{i=1}^n (1 - \sigma((x^1_i)^\top u_i(x^0, \theta))) (x^1_i)^\top \left( \sum_{j=1}^n v_{i,j} x^0_j + v_i \right) \\
= \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \sum_{i=1}^n (x^1_i - \sigma(u_i(x^0, \theta))) \left( \sum_{j=1}^n v_{i,j} x^0_j + v_i \right) \\
= \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \left[ \sum_{i=1}^n x_i \left( \sum_{j=1}^n v_{i,j} x^0_j + v_i \right) - \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v_{i,j} x^0_j + v_i \right) \right].
\]

(A.7)
Combining (A.8) with (A.9) and the definitions (6.8) and (6.9) we obtain (6.7).

Note that

\[
\sum_{x^1} e(x^1) P_\theta(x^0, x^1) \left[ \sum_{i=1}^n x^1_i \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) - \sum_{i=1}^n x^1_i \left( \sum_{j=1}^n v^-_{i,j} x^0_j + v^-_i \right) \right]
\]

\[
- \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \left[ \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) - \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^-_{i,j} x^0_j + v^-_i \right) \right]
\]

\[
= \left( \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \right) \left[ \sum_{i=1}^n x^1_i \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) + \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) \right]
\]

\[
- \left( \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \right) \left[ \sum_{i=1}^n x^1_i \left( \sum_{j=1}^n v^-_{i,j} x^0_j + v^-_i \right) + \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^-_{i,j} x^0_j + v^-_i \right) \right]
\]

\[
= \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \sum_{i=1}^n \left[ x^1_i \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) + \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) \right]
\]

\[
- \sum_{x^1} e(x^1) P_\theta(x^0, x^1) \sum_{i=1}^n \left[ x^1_i \left( \sum_{j=1}^n v^-_{i,j} x^0_j + v^-_i \right) + \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^-_{i,j} x^0_j + v^-_i \right) \right]
\]

\[
(A.8)
\]

Note that

\[
\sum_{x^1} P_\theta(x^0, x^1) \sum_{i=1}^n \left[ x^1_i \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) + \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) \right]
\]

\[
= \sum_{i=1}^n \left( \sum_{x^1} P_\theta(x^0, x^1) x^1_i \right) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) + \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right)
\]

\[
= \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right) + \sum_{i=1}^n \sigma(u_i(x^0, \theta)) \left( \sum_{j=1}^n v^+_{i,j} x^0_j + v^+_i \right)
\]

\[
(A.9)
\]

Combining (A.8) with (A.9) and the definitions (6.8) and (6.9) we obtain (6.7).
A.2.2 Derivation of Equations 6.12 and 6.13

We have

\[
Q(x_1) = \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} Q(x_1, x_2, \ldots, x_n)
\]

\[
= \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \frac{1}{c} \prod_{i=1}^{n} \beta_i^{x_i} (1 - \beta_i)^{1-x_i} \left( d + \sum_{i=1}^{n} \alpha_i x_i \right)
\]

\[
= \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \frac{1}{c} \prod_{i=2}^{n} \beta_i^{x_i} (1 - \beta_i)^{1-x_i} \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \left( d + \alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i x_i \right)
\]

\[
= \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \frac{1}{c} \prod_{i=2}^{n} \beta_i^{x_i} (1 - \beta_i)^{1-x_i} \left( d + \alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i x_i \right)
\]

\[
+ \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \frac{1}{c} \prod_{i=2}^{n} \beta_i^{x_i} (1 - \beta_i)^{1-x_i} \alpha_1 x_1
\]

\[
= \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \frac{1}{c} \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \prod_{i=2}^{n} \beta_i^{x_i} (1 - \beta_i)^{1-x_i} \left( d + \sum_{i=2}^{n} \alpha_i x_i \right)
\]

\[
(A.10)
\]
To simplify this equation, note that for $n > 1$,

$$
\sum_{x_1 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \prod_{i=1}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=1}^{n} a_i x_i \right)
= \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \left[ \beta_1 \prod_{i=2}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=2}^{n} a_i x_i + a_1 \right) + (1 - \beta_1) \prod_{i=2}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=2}^{n} a_i x_i \right) \right]
= \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \left[ \beta_1 a_1 \prod_{i=2}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} + \prod_{i=2}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=2}^{n} a_i x_i \right) \right]
= \beta_1 \alpha_1 + \sum_{x_2 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \prod_{i=2}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=2}^{n} a_i x_i \right),
\tag{A.11}
$$

and if $n = 1$ then

$$
\sum_{x_1 \in \{0, 1\}} \prod_{i=1}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=1}^{n} a_i x_i \right) = \beta_1 (d + a_1) + (1 - \beta_1) d
= \beta_1 d + \beta_1 a_1 + d - \beta_1 d = \beta_1 a_1 + d.
\tag{A.12}
$$

Combining Equation (A.11) and Equation (A.12), we see that for any $n \geq 1$,

$$
\sum_{x_1 \in \{0, 1\}} \ldots \sum_{x_n \in \{0, 1\}} \prod_{i=1}^{n} \beta_i^{x_i}(1 - \beta_i)^{1-x_i} \left( d + \sum_{i=1}^{n} a_i x_i \right) = d + \sum_{i=1}^{n} \beta_i \alpha_i.
\tag{A.13}
$$

Combining Equation (A.10) with Equation (A.13),

$$
Q(x_1) = \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \frac{\alpha_1 x_1}{c} + \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \frac{1}{c} \left( d + \sum_{i=2}^{n} \beta_i \alpha_i \right)
= \beta_1^{x_1} (1 - \beta_1)^{1-x_1} \frac{1}{c} \left[ d + \alpha_1 x_1 + \sum_{i=2}^{n} \beta_i \alpha_i \right].
$$
In general,

\[
Q(x_k, x_{k-1}, \ldots, x_1) = \sum_{x_{k+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} Q(x_1, \ldots, x_k, \ldots, x_n)
\]

\[
= \sum_{x_{k+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} \frac{1}{c} \prod_{i=1}^{k} \beta_i^{x_i} (1 - \beta_i)^{\sum_{i=1}^{k} a_i x_i} \left( d + \sum_{i=1}^{n} a_i x_i \right)
\]

\[
= \sum_{x_{k+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} \frac{1}{c} \prod_{i=1}^{k} \beta_i^{x_i} (1 - \beta_i)^{\sum_{i=1}^{k} a_i x_i} \left( d \sum_{i=1}^{n} a_i x_i \right)
\]

\[
= \frac{1}{c} \prod_{i=1}^{k} \beta_i^{x_i} (1 - \beta_i)^{\sum_{i=1}^{k} a_i x_i} \left( d + \sum_{i=1}^{k} a_i x_i \right).
\]
Bibliography


[38] Adel Hamdi and Andreas Griewank. Reduced quasi-newton method for simultaneous
design and optimization. *Computational Optimization and Applications*, 49(3):521–
548, 2011.

optimization using the navier–stokes equations. *International Journal for Numerical

[40] Bernd Heidergott and Arie Hordijk. Taylor series expansions for stationary markov

[41] Bernd Heidergott, Arie Hordijk, and Heinz Weisshaupt. Measure-valued differentiation
2006.

Gradient estimation for discrete-event systems by measure-valued differentiation. *ACM

[43] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with


*Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.*
IEEE, 1983.

[46] J. J. Hopfield. Neurons with graded response have collective computational properties
like those of two-state neurons. *Proceedings of the National Academy of Sciences*,

[47] John J Hopfield. Neural networks and physical systems with emergent collective com-
1982.


[50] Liang Jin and Madan M Gupta. Stable dynamic backpropagation learning in recurrent


