SPRINGER BRIEFS IN COMPUTER SCIENCE

Anthony L. Caterini · Dong Eui Chang

Deep Neural Networks in a Mathematical Framework

Springer

SpringerBriefs in Computer Science

More information about this series at http://www.springer.com/series/10028

Anthony L. Caterini • Dong Eui Chang

Deep Neural Networks in a Mathematical Framework

Springer

Anthony L. Caterini
Department of Statistics
University of Oxford
Oxford, Oxfordshire, UK

Dong Eui Chang
School of Electrical Engineering
Korea Advanced Institute of Science
and Technology
Daejeon, Korea (Republic of)

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-75303-4 ISBN 978-3-319-75304-1 (eBook)
https://doi.org/10.1007/978-3-319-75304-1

Library of Congress Control Number: 2018935239

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my fiancée Alexandra

To my parents

Preface

Over the past decade, Deep Neural Networks (DNNs) have become very popular models for problems involving massive amounts of data. The most successful DNNs tend to be characterized by several layers of parametrized linear and nonlinear transformations, such that the model contains an immense number of parameters. Empirically, we can see that networks structured according to these ideals perform well in practice. However, at this point we do not have a full rigorous understanding of *why* DNNs work so well, and *how* exactly to construct neural networks that perform well for a specific problem. This book is meant as a first step towards forming this rigorous understanding: we develop a generic mathematical framework for representing neural networks and demonstrate how this framework can be used to represent specific neural network architectures. We hope that this framework will serve as a common mathematical language for theoretical neural network researchers—something which currently does not exist—and spur further work into the analytical properties of DNNs.

We begin in Chap. 1 by providing a brief history of neural networks and exploring mathematical contributions to them. We note what we *can* rigorously explain about DNNs, but we will see that these results are not of a generic nature. Another topic that we investigate is current neural network representations: we see that most approaches to describing DNNs rely upon decomposing the parameters and inputs into scalars, as opposed to referencing their underlying vector spaces, which adds a level of awkwardness into their analysis. On the other hand, the framework that we will develop strictly operates over these vector spaces, affording a more natural mathematical description of DNNs once the objects that we use are well defined and understood.

These mathematical objects are then presented in Chap. 2. Derivatives arise in the training of DNNs—the parameters are often learned using some form of gradient descent with respect to a loss function—so we first review some elementary facts concerning the derivatives of vector-valued maps. We will also distinguish

between *state variables* and *parameters* throughout this work when describing neural network layers; thus, we present notation for maps which depend on both, along with notation for their associated derivatives. We conclude this chapter by looking at *elementwise functions*: maps which operate on individual coordinates of their inputs. These comprise the nonlinear portion of DNNs and are therefore important to include in our study.

We proceed in Chap. 3 by building our generic DNN framework. We represent one layer of a DNN as a parameter-dependent map, and then represent the entire DNN as a composition of these individual layers, composed according to the current state variable. Here, we assume that the parameters are independent across layers, but in later chapters we will see how to modify this assumption. We also describe an algorithm for calculating one step of gradient descent, performed directly over the inner product space defining the parameters, by representing the ubiquitous *error backpropagation* step in a concise and compact form. Besides the standard squared or cross-entropy loss functions, we also demonstrate how to extend our framework to a more complex loss function involving the first derivative of the network.

After developing the generic framework, we apply it to three specific network examples in Chap. 4. We start with the Multilayer Perceptron, the simplest type of DNN, and show how to generate a gradient descent step for it. We then represent the Convolutional Neural Network (CNN), which contains more complicated input spaces, parameter spaces, and transformations at each layer. The CNN, however, still fits squarely into the generic framework of Chap. 3. The last structure that we consider is the Deep Auto-Encoder, which has parameters that are not independent at each layer and thus falls slightly outside of the framework. However, we are still able to extend the framework to handle this case without much difficulty.

In Chap. 5, we extend the framework of Chap. 3 even further to represent Recurrent Neural Networks (RNNs), the sequence-parsing DNN architecture. The parameters of an RNN are shared across all layers of the network, and so we rely on some of the tools from Chap. 2 to modify our framework for representing RNNs. We describe a generic RNN first and then the specific case of the vanilla RNN. For the sake of completeness, we represent and compare both the common *backpropagation through time* and the less-common *real-time recurrent learning* approaches to gradient calculation in an RNN. We conclude the chapter by discussing extensions to basic RNNs, but explicit representations of these networks are outside of the scope of this book.

This book is intended for neural network researchers—theoretical or otherwise—and those from the fields of mathematics, sciences, and engineering who would like to learn more about DNNs. It is quite clear that DNNs are brimming with potential, and we believe that solidifying their mathematical foundations will lead to even more impressive results in application. Furthermore, we expect that this book will make neural networks more accessible to those outside of the community, allowing additional researchers to contribute to this exciting field.

Anthony L. Caterini would like to acknowledge support from the NSERC of Canada. Dong Eui Chang would like to acknowledge support from the NSERC of Canada and the MSIP/IITP of Korea.

Oxford, UK — Anthony L. Caterini
Daejeon, Korea — Dong Eui Chang
November 2017

Contents