MATRIX FACTORIZATION AND DEEP LEARNING IN SCIENTIFIC DOMAINS: UNDERSTANDING WHEN AND WHY IT WORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Nils Johan Bertil Bjorck

December 2021

MATRIX FACTORIZATION AND DEEP LEARNING IN SCIENTIFIC DOMAINS: UNDERSTANDING WHEN AND WHY IT WORKS

Nils Johan Bertil Bjorck, Ph.D.

Cornell University 2021

Propelled by large datasets and parallel compute accelerators, deep neural networks have recently demonstrated human-like performance in many domains previously beyond the reach of machines. Computers can now recognize objects in images, transcribe speech and exhibit reading comprehension at the level of an average human. However, there are many domains that require expert knowledge beyond that of the average person, e.g. medical diagnosis and scientific analysis. This raises the question – beyond average humans, can modern statistical models perform as well as domain experts?

Attempting to answer this question, this thesis considers modern machine learning as applied to several expert domains. In particular, we consider problems related to sustainability, placing our work in the domain of computational sustainability – a nascent field at the intersection of computer science and sustainability. Firstly, we consider using neural networks to identify invasive species habitats from remote sensing images, showing that unsupervised learning can make use of sparse expert labels and cheap satellite images. Secondly, we consider passive acoustic monitoring of endangered animals and introduce a novel data-driven compression scheme for this setting. Thirdly, we consider the use of non-negative matrix factorization (NMF) to spectroscopic datasets in materials science, and show how combining discrete and continuous optimization can yield solutions that accelerate scientific discovery.

In expert domains such as these, empirical performance is not always enough. Instead, one needs to know when and why machine learning methods work to utilize model predictions. Paradoxically, most machine learning models are NP-hard to optimize, yet work remarkably well in practice. Inspired by our practical problems, we make empirical and theoretical contributions towards a principled understanding of when and why machine learning works. On the theoretical side, we introduce a randomized average case model for NMF and prove that certain convexity properties arise naturally in this model. On the empirical side, we consider a popular method in deep learning – batch normalization – which cannot improve model expressivity yet improves performance in practice. We demonstrate that the improved conditioning this normalization confers enables larger learning rates which has a regularizing effect.

BIOGRAPHICAL SKETCH

Johan Bjorck is advised by prof. Carla Gomes and prof. Bart Selman at the Department of Computer Science at Cornell University. Johan's work centers around the principles and practices of modern machine learning, especially in the context of scientific domains. Researched areas include automatic analysis of x-ray crystallography, unsupervised learning of remote sensing data, and machine learning for passive acoustic monitoring for endangered species. Additionally, Johan aims to develop a principled understanding of deep learning techniques whose practical utility is poorly understood, hoping to further machine learning as a scientifically grounded method. Johan completed his bachelor's in engineering physics at Chalmers, Sweden in 2015.

Dedicated to Henrik, Eva, and Ziwei.

ACKNOWLEDGEMENTS

It is said that it takes a village to raise a child. For a Ph.D. student, it takes at least a village, a department, professors, collaborators, friends, and family. I would here like to extend my gratitude to people and institutions that have directly shaped my Ph.D. experience.

I would like to thank friends and fellow doctoral students I've met at Cornell: Najva Akbari, David Specht, Lexi Hartley, Michael Butholp, Baris Bircan, Melanie, Artur Gorokh, Carla Vidal, Andrey Kostin, David Eriksson, Gideon Dresdner, and Vilja Järvi. For many years I lived at the Triphammer cooperative, and want to especially thank Tianyu Wang, Hannah George, Nico Hirscl, Connor Rosenblatt, and Ben Jablonski. I have also spent several years at the Telluride house, and would especially like to thank Anmol Kabra, Sidarth Ramanujan Raghunathan, and Ehab Ebeid for making it so welcoming. I've also relied on support from people back home in Sweden. Foremost my family: my father Henrik Björck, my mother Eva Johansson, and my siblings Hannes and Katrin Björck. I have many dear friends in Sweden, especially supporting are Fredrik Steen, Loe Lindström, and Axel Andersson. I would also like to thank my undergraduate friends: Johan Runeson for sharing his doctoral experiences at ETH and Alexander Kuzmin for telling me in 2015 that machine learning might be worth looking into. Six years later, I'm inclined to believe he was right. At last, I want to thank partner Ziwei Liu for sticking by my side and motivating me to finish this thesis. She finished her Ph.D. in Materials Science at Cornell in 2020, so I am also indebted to her advisor Chris Ober for keeping her around long enough for us to meet. During the years of the COVID-19 pandemic, Ziwei has been an invaluable partner. In addition to providing support, she has also helped me by proofreading multiple manuscripts.

I would also thank the many professors and senior researchers I've had the good fortune to interact with. Foremost, I'd like to thank prof. Carla Gomes and prof. Bart Selman for advising me throughout my Ph.D., sharing their grand vision with me, and showing how solid empirics can be impactful be it in SAT or deep learning. I am deeply grateful for advice and encouragement from them. I would also of course like to thank the members of my special committee. Kilian Weinberger for providing inspiration regarding research in machine learning. Bruce van Dover for fruitful collaboration in materials science. Rachit Agarwal for introducing me to the joys of systems. From applied physics, I'd like to thank David Muller and Lena Kourkotis for sage advice. From my undergraduate institution, I'd also like to thank professors Jana Madjarova, Bo Berndtsson, Martin Cederwall, Mats Halvarson, and Per Delsing for equipping me for my Ph.D. journey. Finally, I would like to thank managers during my summer internships for hosting me: Vladlen Koltun and Subhojit Sum.

On the research side, I have had the good fortune to publish research with many wonderful collaborators. On the materials science side, these include John Gregoire, Santosh Suram, Lan Zhou, and R Bruce Van Dover. On the ecology and conservation side, this includes Mark Whitmore, Peter Wrege, Carrie Brown-Lima, Jennifer Dean, and Angela Fuller. From the computer science department, collaborators include Chris de Sa, Anmol Kabra, and Xiangyu Chen. I've also been fortunate to work in a wonderful lab group, and I especially want to thank senior lab members Yexiang Xue and Guillaume Perez for sage advice. I'd also like to thank Rich Bernstein for tireless help with computers and writing. Furthermore, I'd like to thank my co-authors from the lab: Di Chen, Junwen Bai, Ronan Le Bras, Brendan Rappazzo, Yiwei Bai, Xiaojian Wu, and Qinru Shi – and also other members: Wenting Zhao, Niko Grupen, Dieqiao Feng for creating a

collaborative environment. I would also like to thank CAC, and especially staff members Jodie Sprouse and Resa Reynolds, for help with our cluster. Other staff members are Anna Matusiewicj and Brian Rieger, who provided invaluable help.

At last, I want to thank the Swedish Fulbright commission, the Telluride Association, and the Thanks to Scandinavia Foundation for supporting my studies and allowing me to independently pursue research. This material is supported by NSF awards CCF-1522054 and CNS-0832782 (Expeditions), CNS-1059284 (Infrastructure), and IIS-1344201 (INSPIRE); and ARO awards W911-NF-14-1-0498 and W911NF-17-1-0187. This material is also based upon work supported by the Air Force Office of Scientific Research under award numbers FA9550-18-1-0136 and FA9550-17-1-0292. Materials science experiments were supported through the Office of Science of the U.S. Department of Energy under Award No. DE-SC0004993. and by the Cornell Center for Materials Research with funding from the NSF MRSEC program (DMR-1719875). Use of the Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-76SF00515. Datasets and efforts related to conservation were funded through the New York State Department of Environmental Conservation (Award C008698 among others), USDA Forest Service, and the New York State Environmental Protection Fund as administered by the New York State Department of Environmental Conservation. We would like to thank the Elephant project, the Cornell Lab of Ornithology its volunteers and the Wildlife Conservation Society, the U.S. Fish and Wildlife Service, and the Robert G. and Jane V. Engel Foundation. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Mathematical models are ubiquitous when trying to understand our messy world, often filled with noise and outliers. Gauss famously used least-square fitting to determine comet orbits from noisy manual measurements [Gauss, 1809]. Student pioneered statistical modeling to infer chemical properties of barley from small samples at the Guinness Brewery [Student, 1908]. Traditionally, such mathematical models were often applied to small, manually collected datasets.

With the digital revolution, it has become increasingly easy to store and transfer data. In 1956, the first hard disk [1] was bulky, expensive, and allowed for less than 5 MB of storage. Today, consumers can buy 128GB storage for less than 30$. Similarly, capturing data electronically with cameras and microphones has become exceedingly easy. This digital revolution has created large modern datasets such as Imagenet, containing more than 1 million annotated images [Deng et al., 2009], and Vox Populi [Wang et al., 2021], containing 400,000 hours of recorded speech in 23 languages. Large-scale datasets have also become common in scientific domains, e.g. the human genome projects contain billions of recorded base-pairs [Lander et al., 2001]. Beyond their size, modern datasets are often high dimensional; a single image with resolution 200-by-200 contains more than 100,000 variables. For such high-dimensional datasets, traditional linear models often perform poorly. E.g., one cannot accurately determine the number of dogs in a picture by a linear function of pixel intensities.

One class of models able to capitalize on large and high-dimensional datasets is artificial neural networks, which are loosely inspired by the brain. These

[1] the IBM Model 350 Disk File which came with the IBM 305 RAMAC, priced at 3,200$.

models are highly non-linear and expressive, in fact, given enough parameters they can approximate any function [Hornik et al., 1989]. Propelled by modern datasets and parallel compute accelerators, researchers have demonstrated that neural networks can reach everyday capabilities previously only achieved by humans. Neural networks can now recognize objects in images at a human level [He et al., 2015a]. Similarly, they can transcribe speech better than humans [Xiong et al., 2017] and exhibit reading comprehension at a superhuman level [Nangia and Bowman, 2019, Raffel et al., 2019].

While neural networks often can outperform the average person in everyday tasks, many important domains require expert knowledge not possessed by the average person. Examples include medical diagnosis and scientific analysis. This raises the question – beyond average humans, can modern machine learning models perform as well as domain experts? To answer this question, this thesis considers the use of modern machine learning models in several high-dimensional scientific domains. In particular, we consider several problems related to sustainability, placing our work in the domain of computational sustainability [Gomes et al., 2019] – a nascent field at the intersection of computer science and sustainability.

The first scientific domain we consider is invasive species mapping, important for ecology and conservation work. Invasive species are species invading ecosystems they are not native to, such invasions annually cause tens of billions in damages. Together with ecology collaborators, we use modern machine learning to detect invasive habitats from high-resolution satellite images. Whereas obtaining satellite images is easy, obtaining accurate labels is expensive and requires expert ecology field workers. Leveraging sparse historical data, we

show that unsupervised methods can detect habitats even with few labels.

The second domain we consider is acoustic monitoring of African forest elephants. As a low-cost method to monitor these populations, one can place microphones in their habitats. These large datasets contain many outlier sounds such as guns and weather phenomena, and detecting elephant calls often requires expert knowledge. Together with our bioacoustics collaborators, we develop a data-driven compression method relevant in sub-Saharan Africa where bandwidth is limited.

The third scientific domain we consider is the analysis of spectroscopic data from X-ray crystallography. To analyze crystal structures, material scientists have analyzed spectroscopic data manually which is often time-consuming. To accelerate this process, we consider matrix factorization methods for demixing spectroscopic signals. Together with materials science collaborators, we propose efficient algorithms for demixing such data, and verify that our methods outputs solutions that adhere to physical constraints.

Our work empirically demonstrates that with the right inductive biases, modern machine learning models achieve excellent performance in scientific domains. Empirical performance is sufficient for many applications. However, in scientific domains, it is often important to know *when* and *why* models work. In medicine, for example, one needs to trust a model's prediction to predicate a risky operation upon it. Unfortunately, theoretical predictions are at odds with the empirical success of modern machine learning. Neural networks are NP-hard to solve, suggesting that any efficient algorithm must fail for some hard problem instances, i.e. in the *worst case*. Inspired by our practical application, we study the gap between machine learning *in practice* and in theory.

For modeling our spectroscopic datasets, we relied on non-negative matrix factorization (NMF). NMF is known to be NP-hard in the worst case, yet in practice, it often works well. To reconcile these facts, we consider a theoretical model of NMF with a random planted solution – for which worst-case analysis does not apply. We prove that certain convexity properties arise naturally in this model. Specifically, due to the random nature of the planted solution, with high probability, the NMF problem exhibits *star-convexity*. This means that the loss is convex on straight lines to the optima, and implies efficient algorithms. Our proof relies on a sum-of-squares lower bound and random matrix theory, and we empirically demonstrate that our predictions arise in real-world datasets. This is a first step to bridge the gap between empirical performance and worst-case analysis in NMF.

Neural networks are notoriously hard to debug, and a principled understanding of when and why they work is lacking. Inspired by our use of neural networks, we empirically study the effects of *batch normalization*. Batch normalization standardizes intermediate features in neural networks by subtracting the mean and dividing by the standard deviation. This linear transformation cannot improve expressivity, yet in practice, it is often crucial for obtaining good results. We demonstrate that this normalization has a conditioning effect on neural networks, allowing for larger learning rates. This increases stochasticity in the optimization procedure, which has a regularizing effect. These results bring us closer to a principled understanding of a common technique for neural networks, which is important when applying these models in sensitive expert domains.

This research would not have been possible without our wonderful collabo-

rators. Our research on invasive species has been a collaboration with the New York State Heritage Program, which coordinates efforts to combat the spread of invasive species in the state of New York. It has also been a collaboration with several researchers from the Department of Natural Resources at Cornell. The work on materials science has been conducted in collaboration with the Joint Center for Artificial Photosynthesis (JCAP) at Caltech, and several materials science researchers at the department of materials science at Cornell. At last, our work on passive acoustic monitoring has been joint work with researchers from the Cornell Lab of Ornithology, and their bio-acoustics division.

1.1 Machine Learning in Scientific Domains

1.1.1 Detecting Invasive Species Habitats from Satellite Images

Invasive species are species that invade an ecosystem, often to the detriment of local flora and fauna. Examples in North America include the Hemlock Wolly Adelgid which eats Hemlock trees, a keystone species providing important ecosystem services. Over 30 years, New York Natural Heritage Programs has collected an expert-annotated dataset consisting of 200,000 unique observations of invasive species in the state of New York. This dataset is relatively small – however, one can easily obtain high-quality satellite images over the state. We thus propose using remote sensing images to determine what habitats might contain invasive species.

To utilize this abundance of unannotated data, we propose an unsupervised method for representation learning which can combine extensive annotations with inexpensive satellite images. To make use of the geographic data, we take

two remote sensing images of nearby patches of land and train a neural network to recognize which patches fit together. This training extracts useful features which can be used for downstream prediction. This scheme is illustrated in Figure 1.1.

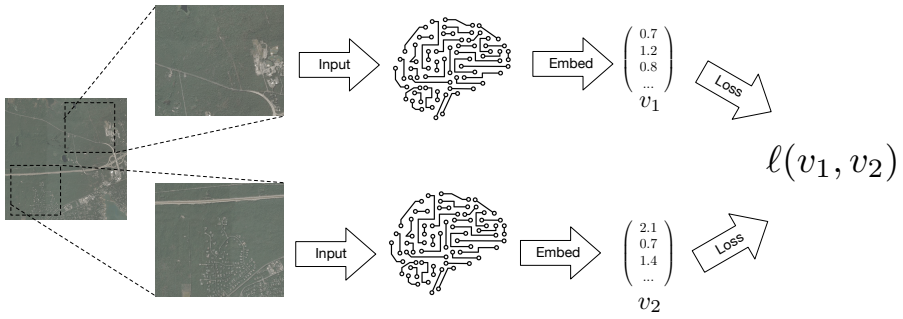

Figure 1.1: Given a remote sensing landscape, we consider two patches close to each other. We train the network to classify neighbours as such. This training procedure generates informative intermediate features.

After training a neural network in this unsupervised fashion, we evaluate the quality of the features extracted by using the labels. We consider several competing methods for extracting features and find that our unsupervised method outperforms them all. In addition to providing better features, we consider its use in active learning and show that it improves performance there too.

1.1.2 Elephants

Monitoring the African Forest elephant is paramount to ensure its survival. Despite its size, this mammal often resides under a dense canopy which makes aerial photography impossible. Consequently, researchers have proposed passive acoustic monitoring, where a large set of microphones are placed in their

habitats to record their characteristic calls. A sample call is illustrated in Figure 1.2. Compared to video, microphones are inexpensive and consume less energy and bandwidth. However, there are many sounds that might be confused with elephant calls – e.g. weather patterns, illegal logging, and other animals. Thus, detecting elephant calls requires expert knowledge, but a multitude of microphones that record around the clock produces large datasets infeasible to label manually.

It is often desirable to transmit audio streams to respond to poachers and other threats. The lack of wireless infrastructure makes high-fidelity transfer expensive, and typical audio compression schemes developed for the human ear are ill-suited for low-frequency elephant calls. To alleviate this issue, we propose a novel data-driven compression scheme. We use a differentiable proxy for the quantization error which allows us to jointly optimize for a high compression ratio and the transmission of information vital to detection efforts. We empirically demonstrate that this results in improved bit-allocation.

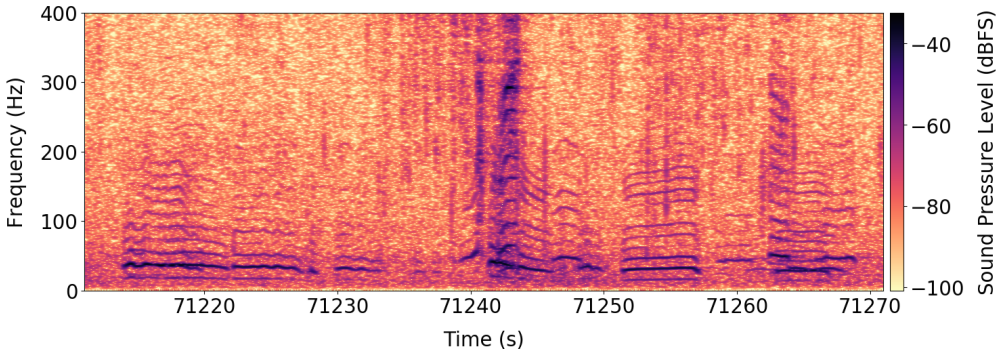

Figure 1.2: A spectogram of a sample elephant call. Detecting these, without confusing them for other activity such as logging and other animals, requires expertise.

1.1.3 Materials

Engineered materials crucially enable our modern world – doped semiconductors form the basis of the modern transistor, advanced polymers are used for packaging and manufacturing, and solar power depends upon complex materials with photovoltaic properties. To characterize novel materials, material scientists often employ x-ray crystallography to obtain high-dimensional spectroscopic measurements. Inferring material properties from spectroscopic datasets is labor-intensive and requires expertise. To accelerate materials discovery, we consider the use of machine learning methods to characterize these datasets.

In x-ray crystallography, each material gives rise to a unique spectroscopic fingerprint. When materials are mixed, a linear combination of their fingerprints arises. Thus, given enough measurements of mixed materials, finding the underlying individual fingerprints can be formulated as a non-negative matrix factorization (NMF) problem:

$$\min_{\mathbf{U},\mathbf{V}\geqslant 0} \|\mathbf{X}-\mathbf{U}\mathbf{V}\|_2^2 \qquad \mathbf{U}\in R^{n\times k}, \mathbf{V}\in R^{k\times m} \tag{1.1}$$

The non-negative constraint arises from the fact that these spectroscopic fingerprints and their mixture weights cannot be negative. To make matters worse, there are several rules derived from thermodynamics that the solutions to problem eq. (1.1) must obey. Otherwise, the solution breaks physical rules and cannot correspond to the ground truth. We propose to incorporate these physical rules into the optimization of eq. (1.1) by formulating them as a set of discrete and continuous problems called mixed-integer programs. For this class of programs, commercial solvers are available but run slowly. To speed up the training speed,

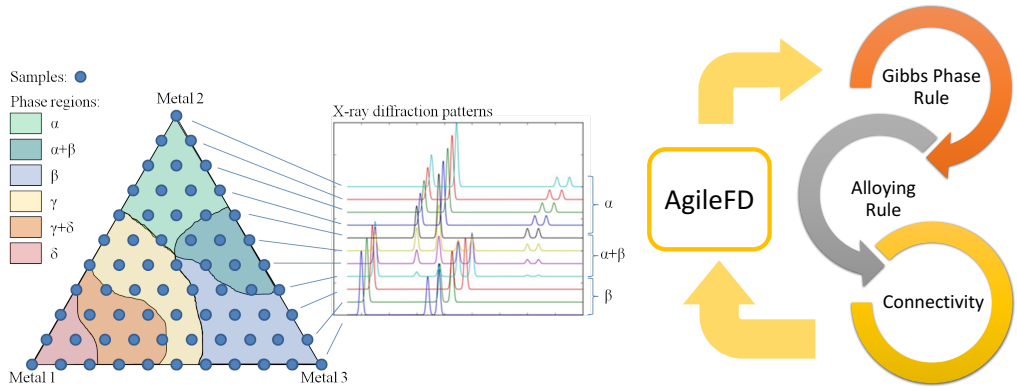

Figure 1.3: (Left) When mixing base elements, many materials can form, each with its unique spectroscopic fingerprint. Mixing materials leads to linear combinations of these patterns, and recovering individual materials becomes a non-negative matrix factorization (NMF) problem. (Right) Our proposed methods alternatives between continuous gradient-type updates and discrete optimization problems which enforces physical constraints for the factorization problem.

we instead iteratively apply these mixed-integer programs together with greedy continuous updates. Over a dataset of real-world spectroscopic data, we show that this results in solutions of good quality which obey physical rules and thus can be used by expert material scientists. The high-level idea is illustrated in Figure 1.3.

1.2 Machine Learning Beyond the Worst Case

1.2.1 Non-Negative Matrix Factorization

We have shown that NMF can be useful for demixing high-dimensional spectroscopic data. While NMF commonly succeeds in practice, it is in fact NP-hard. Thus, one cannot be guaranteed to find a good solution, which can be problem-

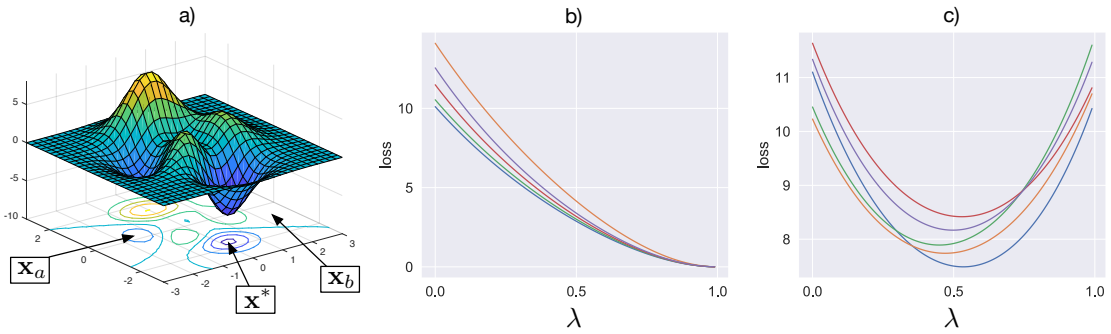

Figure 1.4: Despite being non-convex in general, NMF is often convex on straight lines. In an randomized average-case model, we prove that this behaviour is highly probable.

atic for expensive datasets. To study the gap between the theory and practice of NMF, we consider a model with a planted random solution. Since we restrict the class of NMF problems, hardness results based upon worst-case analysis no longer apply. We hope that a randomized model will roughly model noisy real-world datasets. The randomized model we consider is:

Assumption 1. *Let* $\mathbf{X} = \mathbf{U}^*\mathbf{V}^*$ *where the entries of the matrices* $\mathbf{U}^*, \mathbf{V}^*$ *are sampled iid from rectified Gaussian distribution.*

Empirically, one can observe that this model exhibits certain convexity properties, despite NMF being non-convex in general. Specifically, on straight lines towards the planted solution, the loss function is often convex. This is illustrated in Figure 1.4. Such properties have been studied by Lee and Valiant [2016], and they are defined as star-convexity. Formally, the definition is

Definition 1. *A function* $f : \mathcal{R}^n \to \mathcal{R}$ *is* ***star-convex*** *towards* $\boldsymbol{x}^*$ *if for all* $\lambda \in [0, 1]$ *and* $\boldsymbol{x} \in \mathcal{R}^n$*, we have* $f(\lambda \boldsymbol{x} + (1 - \lambda)\, \boldsymbol{x}^*) \leqslant \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{x}^*)$.

Lee and Valiant [2016] have shown that this property implies that the func-

tion can be efficiently optimized. Using a combination of sum-of-squares lower bounds and random matrix theory, we can show that for the model of Assumption 1, star-convexity arises with high probability. We formalize this in the following theorem

Theorem 1. ***(Informal)*** *Let matrices* $\mathbf{U}, \mathbf{V}, \mathbf{U}^*, \mathbf{V}^*$ *be sampled according to Assumption 1. NMF is then convex on the line* $(\mathbf{U}_1, \mathbf{V}_1) \rightarrow (\mathbf{U}^*, \mathbf{V}^*)$ *with probability* $\geqslant 1 - c_1 \exp(-c_2 n^{1/3})$ *for some constants* $c_1 > 0, c_2 > 0$.

For proving this, our strategy relies on deriving an inequality that implies star-convexity. While this inequality does not hold for all NMF problems, we show that it holds with high probability for the random matrices we consider here. Furthermore, the assumptions can be made slightly weaker and our results hold in a setting with partially observed matrices. Empirically we show that our predictions hold in practice. This provides an attempt to bridge the gap between the theoretical analysis and the empirical success of NMF.

1.2.2 Batch Normalization

Beyond NMF, a gap between empirical success and theoretical hardness exists for neural networks too – optimizing them is known to be NP-hard [Blum and Rivest, 1992]. Improvements in deep learning often come from methods with no or little theoretical basis. Over the years, this has led to deep learning to be described as a black art ruled by trial and error rather than universal principles. In scientific domains, one needs to want to understand when and why the models work well to use their predictions effectively.

Motivated by this, we study one method commonly used in neural networks

– *batch normalization*. This method simply subtracts the mean and divides by the standard deviation for intermediate features. Such a linear transformation cannot increase network expressivity, yet in practice, it often yields improvements. The original authors Ioffe and Szegedy [2015b] hypothesize that batch normalization alleviates *internal covariate shift*, but this hypothesis is easy to experimentally refute. To understand the benefits of batch normalization, we take an empirical approach.

Hyperparameters used for neural networks are often selected in tandem with the network architecture used. We empirically demonstrate that the learning rate – the step size used in gradient descent – interacts strongly with normalization. Specifically, when using normalization much larger learning rates can be used without divergence. Let us decouple SGD into a gradient and a noise term

$$\alpha \nabla_{SGD}(x) = \underbrace{\alpha \nabla \ell(x)}_{\text{gradient}} + \underbrace{\frac{\alpha}{|B|} \sum_{i \in B} \left(\nabla \ell_i(x) - \nabla \ell(x) \right)}_{\text{error term}}.$$

Under mild assumptions, the noise term can be estimated $\mathbb{E}\left[\|\alpha \nabla \ell(x) - \alpha \nabla_{SGD}(x)\|^2\right] \leqslant \frac{\alpha^2}{|B|} C$. This suggests that the larger learning rates enabled by normalization introduce noise into SGD, and such noise is widely believed to have a regularizing effect. In fact, empirically we show that almost the entire benefit of batch normalization can be explained by the regularizing effect in the following sense: if we don't use the larger learning rates normalization allows, we don't see any benefit from it. This is illustrated in Figure 1.5. This result is a first step towards understanding how and why an important component in modern neural works. We argue that this type of understanding is critical for using and trusting model predictions in sensitive domains.

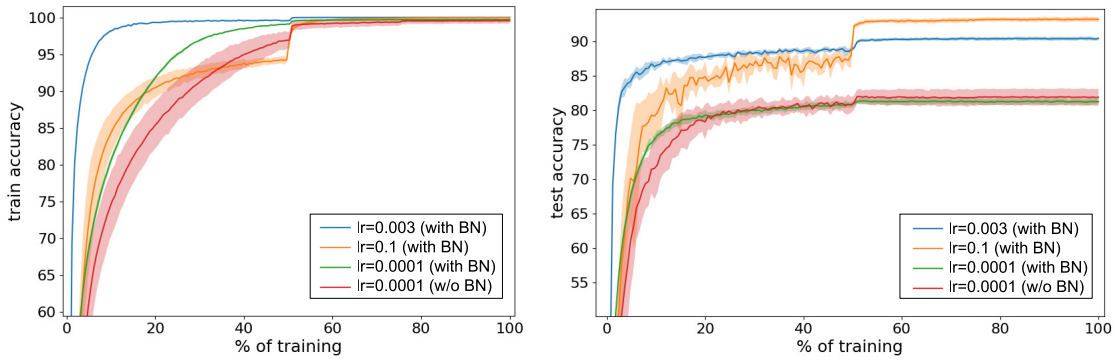

Figure 1.5: Learning curves for different learning rates, using networks with and without batch normalization. When using a small learning rate, batch normalization confers no advantages. Only by enabling larger learning rates does batch normalization improve generalization.

1.3 Summary

In this thesis, we have studied modern machine learning models in scientific domains. For detecting invasive species' habitats from remote sensing images, we have proposed an unsupervised feature learning method that leverages geographical features. For the problem of acoustically monitoring elephants, we have introduced a novel framework of data-driven compression. For the problem of demixing spectroscopic data in materials science, we have proposed a method of incorporating physical constraints into continuous optimization to yield physically meaningful solutions. In addition to demonstrating that machine learning can assist in scientific domains, we have made contributions explaining when and why these methods work. On the theoretical side, we have introduced an average-case model for NMF and proved that it gives rise to certain convexity properties which imply efficient optimization. This sheds light on the gap between theory and practice for NMF. On the empirical side,

we have studied the effects of normalization in neural networks, a method that empirically improves performance but theoretically offers no improvements. We have shown that this normalization procedure improves the conditioning of the neural networks which allows a larger step size to be used and that this, in turn, has a regularizing effect. Results in this thesis have been reported in the following peer-reviewed publications:

- **Bjorck, Johan** and Rappazzo, Brendan H and Shi, Qinru and Brown-Lima, Carrie and Dean, Jennifer and Fuller, Angela and Gomes, Carla, *Accelerating Ecological Sciences from Above: Spatial Contrastive Learning for Remote Sensing*, Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021)
- **Bjorck, Johan** and Rappazzo, Brendan H and Chen, Di and Bernstein, Richard and Wrege, Peter H and Gomes, Carla P, *Automatic detection and compression for passive acoustic monitoring of the African forest elephant*, Proceedings of the AAAI Conference on Artificial Intelligence (AAAI2019)
- Bai, Junwen and **Bjorck, Johan** and Xue, Yexiang and Suram, Santosh K and Gregoire, John and Gomes, Carla, *Relaxation methods for constrained matrix factorization problems: solving the phase mapping problem in materials discovery*, International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR2017)
- **Bjorck, Johan** and Kabra, Anmol and Weinberger, Kilian Q and Gomes, Carla, *Characterizing the Loss Landscape in Non-Negative Matrix Factorization*, Proceedings of the AAAI Conference on Artificial Intelligence (AAAI2021)
- **Bjorck, Johan** and Gomes, Carla and Selman, Bart and Weinberger, Kilian Q, *Understanding batch normalization, Proceedings of the Neural Information Processing Systems Conference (NeurIPS 2018)*

CHAPTER 2

DETECTING HABITATS FROM REMOTE SENSING IMAGES

2.1 Introduction

In recent years, neural networks have made impressive strides in their potency and can now accurately predict faces [Ye et al., 2020] and translate natural language [Devlin et al., 2018]. Besides progress in network architecture and optimization, this progress has been driven by large data sets such as Imagenet [Deng et al., 2009] and hugely parallel accelerators such as graphics processing units (GPUs) or tensor processing units (TPUs) [Jouppi et al., 2017]. Besides these advances in machine learning, another field that has improved with more efficient data processing is remote sensing. While remote sensing via satellites has been used since the cold war, the first commercial satellite (IKONOS) was only launched in 1999. Researchers have been increasingly interested in applications of remote sensing to environmental questions [Jensen, 2009, Lentile et al., 2006].

Machine learning for problems in computational sustainability [Gomes et al., 2019] is an active area of research [Gholami et al., 2019, Chen et al., 2016, Xue et al., 2017, Xie et al., 2015], and one problem that stands to benefit from both access

Species	Description	Observations	% of dataset
Water Chestnut	Floating aquatic plant that hinders boats and crowds out native plants	285	4.39
Honeysuckle	Terrestrial plants that form monotypic stands and reduces diversity	295	4.54
Oriental Bittersweet	Woody vine that smothers and uproot trees	307	4.72
Japanese Knotweed	Terrestrial plant that forms monotypic stands and reduces diversity	1287	19.81
Garlic Mustard	Terrestrial plant that forms monotypic stands and reduces diversity	653	10.05
Japanese Barberry	Terrestrial plant that forms monotypic stands and reduces diversity	459	7.06
Common Reed Grass	Terrestrial plant that forms monotypic stands and reduces diversity	1174	18.07
Purple Loosestrife	Shoreline plant that clogs waterways and reduces wetland habitat	918	14.13
EurasianWater-milfoil	Submerged aquatic plant that hinders boats and crowds out native plants	441	6.79
Multiflora Rose	Terrestrial plant that forms monotypic stands and reduces diversity	679	10.45

Table 2.1: Class names and distribution for the invasive species data set, as well as a description of their ecological relevance.

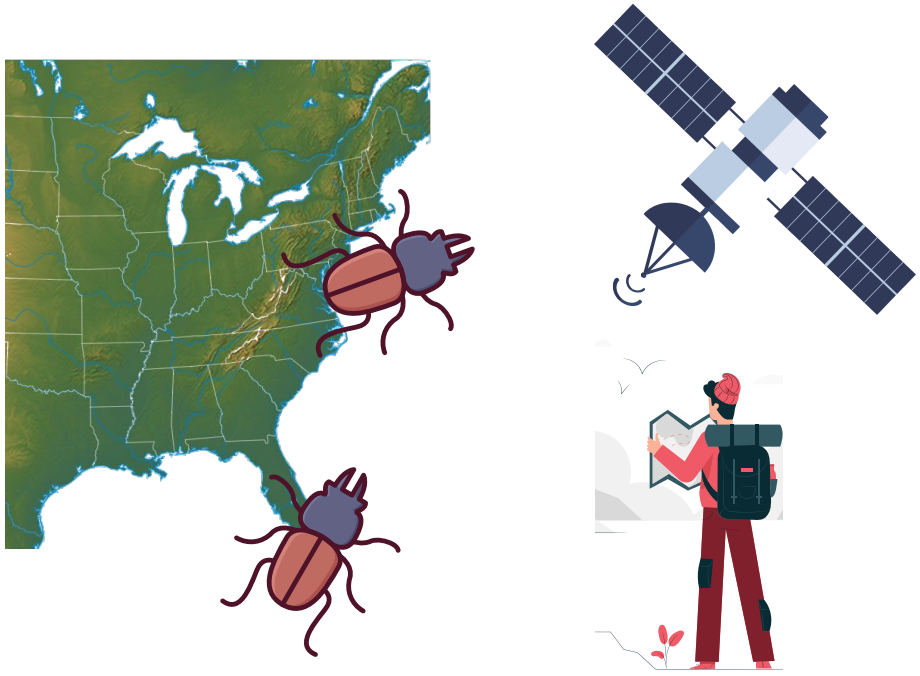

Figure 2.1: Invasive species management requires sending out observers to suitable locations across a large landscape. To effectively use limited resources it is imperative to send observers to the most important locations. In this work, we consider using unsupervised machine learning on remote sensing data to aid these efforts, using deep embedding techniques to improve sample complexity of species classification.

to high-quality remote sensing data and machine learning methods is invasive species management. As part of an ongoing collaboration with the New York Natural Heritage Program, which manages and coordinates invasive species data in the state of New York [Department of Environmental Conservation, 2018], this work considers remote sensing data for invasive species management. As part of the ImapInvasives project [NatureServe, 2020], the New York Natural Heritage Program collects data of observations across the state, and their current database includes over 200,000 observations of invasive species spread over 30 years. Collecting these data is laborious and requires sending out professionals

or volunteer citizen scientists to conduct field surveys. To effectively utilize available workers, it is paramount to send them to suitable locations that have a high likelihood of containing invasive species. While exhaustively searching the state is impossible, the task is made easier by the fact that suitable habitats of various invasive species are often known, e.g., the hemlock wooly adelgid lives in coniferous hemlock forests [Holmes et al., 2005]. In practice, the allocation of observers to actual locations is often made by ecological experts.

Towards automating the task of deciding suitable locations for observations, we consider the task of predicting invasive species' locations from satellite images, see Figure 2.1 for a schematic and further description. A central problem with this approach is sample complexity; neural networks often require large labeled data sets, whereas many species might have few observations. Specifically, in this setting, closely monitoring such species before large outbreaks (meaning few observations) is ecologically important. However, in this setting, satellite imagery is easy to obtain, which suggests the use of unsupervised learning. With this in mind, we consider augmenting contrastive learning [Oord et al., 2018] by utilizing the spatial structure of remote sensing data; training a neural network to classify nearby but non-identical satellite images as such. This naturally induces the network to generate low-dimensional embeddings, which can later be used for tasks like classification or active learning. As we demonstrate, this improves sample complexity over supervised methods and also is an improvement over previous methods of unsupervised learning of remote sensing images. In addition to evaluating our method on satellite images geo-referenced to an invasive species data set from New York Natural Heritage Program, we also consider using our method for another important problem in computational sustainability – landcover classification. We here consider the publicly available data sets

Eurosat [Helber et al., 2019] and the U.S. Department of Agriculture's National Agricultural Imagery Program (NAIP) [Jean et al., 2019], and again show that our method beats baselines. Lastly, we simulate field deployment of our method via active learning and propose to perform active learning in the latent space of images, showing that it can improve upon traditional active learning. We summarize our contributions as follows.

- We introduce a new data set of remote sensing images for invasive species management, where images correspond to observations on the ground.
- We consider spatially augmented contrastive learning for remote sensing data as a method to improve sample complexity, and consistently find that it outperforms baselines across three data sets.
- We simulate field deployment of the method, proposing active learning in the *latent* space of satellite images.

2.2 Remote Sensing and Invasive Species

In this work we consider invasive species that are non-native and that cause some type of harm to the environment, economy, or human health. Examples include zebra mussels invading United States (US) freshwater bodies [Nienhuis et al., 2014]. It has been estimated that invasive species cause damages in the billions of dollars annually, just in the US [Pimentel et al., 2005]. A famous example is the hemlock woolly adelgid, which initially came to the US from Japan [Oten et al., 2014]. The insect feeds on the sap at the base of hemlock needles, disrupting nutrient flow and eventually killing the tree. Due to the ecological importance of hemlocks in many forest ecosystems, researchers across

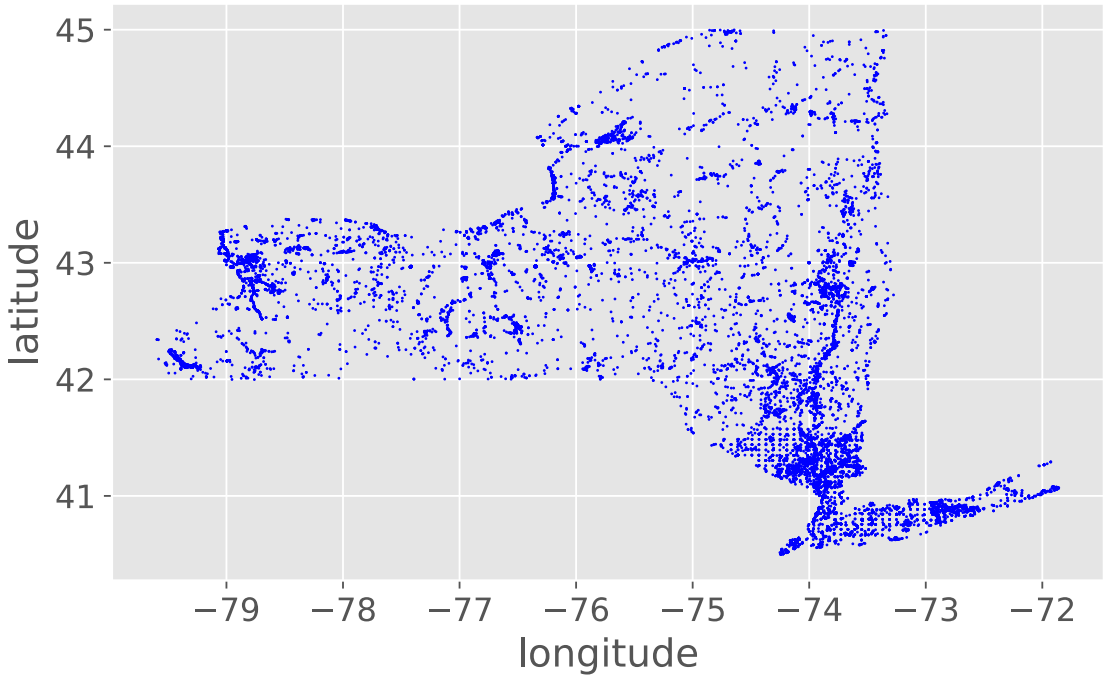

Figure 2.2: Our data set of invasive species observations covers the state of New York, spanning over 200 species and 30 years. Each dot corresponds to a unique observation.

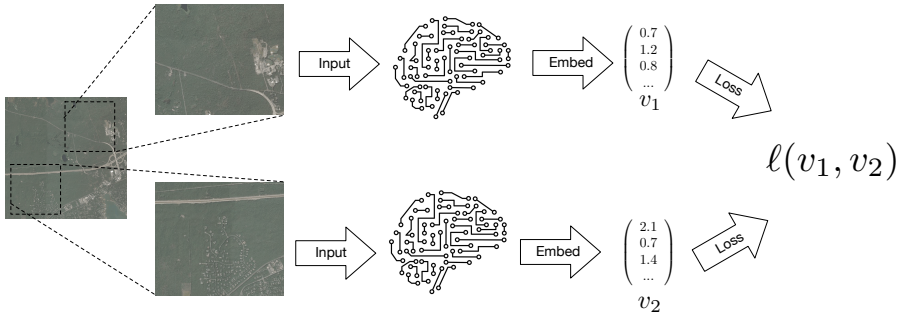

Figure 2.3: Given a remote sensing landscape, we consider two patches close to each other. These images are then fed into the same neural network which generate embeddings v_1, v_2 of the images. Given a collection of such embeddings, we want to be able to classify neighbours as such, and use the inner product $v_1^T v_2$ as the logit.

the US are working on finding efficient strategies to monitor, mitigate, and eradicate the hemlock woolly adelgid as well as all other invasive species. As part of an ongoing effort in the state of New York to monitor invasive species, the New York Natural Heritage Program uses iMapInvasives [NatureServe, 2020] to collect and synthesize invasive species data across the state going back more than 30 years [NatureServe, 2020]. The state is divided into eight invasive species regions, each with partnerships which monitor their region using a combination of paid employees and citizen scientists. Records of observed invasive species are reported to iMapInvasives as the central database. The database currently consists of over 200,000 individual observations, each containing a location and time, the species found, the observer's name, etc. Figure 2.2 illustrates the geographical spread of recorded observations. We will consider the ten most observed invasive species, listed in section 2.1, and will construct a remote sensing data set from these observations to be used for downstream tasks. On the fine spatial scale, observations are strongly correlated, as it is typical for one observer to observe multiple invasive species some meters away from another observer. Additionally, some locations have more observations than others, such as data near large cities. To make the data set approximately spatially balanced, we randomly sub-sample the observations across a grid. We divide the state of New York into a grid corresponding to 0.01 degrees latitude and longitude, and only select one observation per square in this grid, and further make sure that there are no neighboring (horizontally, vertically or diagonally) observations. This results in a data set of 6498 observations, and ten classes – corresponding to unique species – that are roughly balanced between species. For this work we do not consider any temporal information about the observations, such as what date or time an observation was made. See Section 2.1 for further details.

We then obtain 512x512 pixels red, green, blue (RGB) remote sensing images corresponding to these locations via Google Maps. While most invasive species cannot be seen from satellite, their tendency to prefer certain cover types, e.g., the Hemlock Woolly Adelgid prefers hemlock trees found in coniferous forest, will be useful as ecosystem traits can be observed via satellites. Given this data set, we first consider using unsupervised machine learning to generate low-dimensional embeddings that efficiently allows us to classify what invasive species inhabit what regions based upon historical data. The ultimate goal of this line of work is to use machine learning predictions to actually decide what pieces of land are susceptible to invasive species. Later in the paper, we simulate this by considering an active learning approach on this historical data set and defer field deployment to future work.

2.3 Embeddings

Given a remote sensing image x we wish to be able to generate low-dimensional embeddings $y = f(x)$ for some mapping f. The perhaps most well-known applications of embeddings are so-called word-embeddings, where individual words are represented by dense vectors suitable for neural network computation [Mikolov et al., 2013]. Given some corpora of text D, one initializes each word w in a language to be represented by some vector $v(w)$ and then obtains the final word embeddings as the solution to some optimization problem. Typically, one considers some loss function ℓ using the word w and its neighbor n, i.e., we have

$$\min \mathbb{E}_{w,n\sim D}\Big[\ell\big(v(w), v(n)\big)\Big]$$

It is often desirable to choose the loss function ℓ such that words that are used in similar contexts, i.e., have similar neighbors, have similar embeddings. The motivation for defining the loss function in this manner can be motivated by the J.R. Firth quote, "You shall know a word by the company it keeps". We consider an embedding for a satellite image x, but whereas there is a discrete fixed number of words in a language, we will let the embeddings be given by a neural network f. Inspired by this strategy of considering close words, it is natural to apply the same idea that the embeddings of satellite images of nearby locations should have similar embeddings, see Tile2Vec or Patch2Vec [Jean et al., 2019, Fried et al., 2017] that rely on triplet loss. Instead of directly optimizing embeddings via the triplet loss, we obtain them as a byproduct of a classification task, extending contrastive learning [Oord et al., 2018, Bachman et al., 2019] to spatial domains by utilizing the neighborhood relationship induced by the spatial distribution of

Figure 2.4: Examples images from all three data sets. The invasive species data set correspond to observations of invasive species (given in section 2.1) across the state of New York. The Eurosat data set corresponds to satellite images over ten types of landcovers across continental Europe [Helber et al., 2019]. The NAIP data set corresponds to images from California obtained via the national agriculture imaging program [Jean et al., 2019].

remote sensing images. Specifically, given two remote sensing images x, n, where n is a neighbor to x, we train the neural network f to generate embeddings that allow us to conclude that x and n indeed are neighbors. For an illustration, see Figure 2.3. A simple strategy is to cast this as a classification problem and use the softmax loss. If the embeddings are column vectors, we treat their inner product as the actual logit and then consider the soft-max cross-entropy loss.

$$\ell(x, n) = -\log \frac{\exp\left(f(x)^T f(n)\right)}{\sum_j \exp\left(f(x)^T f(j)\right)} \tag{2.1}$$

In practice, computing the denominator is expensive, and one can approximate it by only considering negative examples from the same batch. For a schematic illustration of the method, see Figure 2.3. One can further enlarge the data set by considering augmentations such as, rotations that the natural landscape is approximately invariant under. It has been observed that one can slightly improve contrastive learning by scaling the logits by some fixed parameter T and not using the embeddings of the final layer, but instead adding a small head multi-layer perceptron (MLP) on top of the convolutional neural network (CNN) for training but then using intermediate representations from the CNNs as representations [Chen et al., 2020].

2.4 Experiments

In this section, we primarily focus on our invasive species data set to evaluate the feature extraction from unlabeled remote sensing images. The dataset is constructed as per previous section. We also perform active learning experiments to simulate deploying our method in the field and additionally perform

	Invasive Data Set	
Embeddings	RFC	LR
Contrastive	25.63 ± 1.54	26.71 ± 1.71
Tile2Vec	23.07 ± 1.03	24.16 ± 1.57
AutoEncoder	22.50 ± 0.78	21.91 ± 1.14
PCA	22.16 ± 0.80	22.36 ± 1.37
ICA	22.24 ± 1.11	19.55 ± 1.23

Table 2.2: Accuracy for unsupervised setting. All experiments were run for 10 rounds, and the average value is given $\pm$ the standard deviation.

experiments for two external remote sensing data sets.

2.4.1 Unsupervised Experiments

We first evaluate whether the embeddings generated via our methods are useful for classification, comparing to Tile2vec [Jean et al., 2019] and some further baselines which we describe here. **Tile2Vec** uses a triplet loss [Hoffer and Ailon, 2015] to train a feature extracting network to push geographically nearby tiles

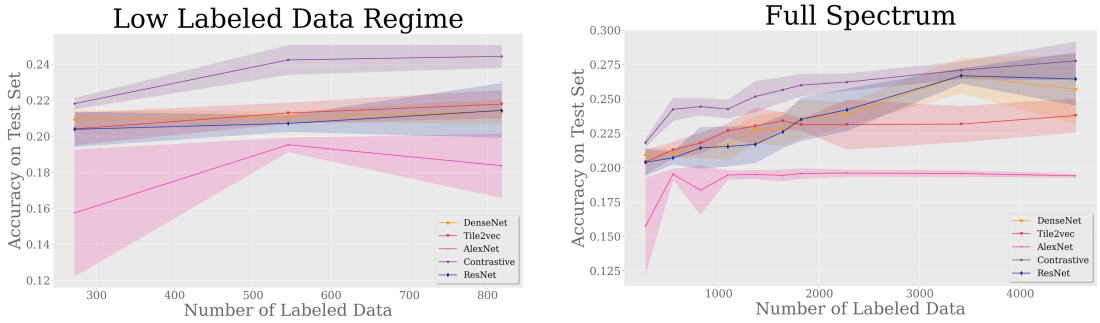

Figure 2.5: Available labeled data vs accuracy for supervised methods and unsupervised methods trained on all images (but not all labels). In this low-data regime (left) and zoomed out full spectrum of available labels (right), spatial contrastive learning outperforms classical supervised methods.

close together in the extracted feature space. The Tile2Vec and contrastive feature extractors are built with the ResNet-18 architecture [He et al., 2015b], with the last layer set to have 256 neurons/features. The contrastive method also uses a ResNet-18 architecture, plus a two-layer top module with 256-neurons for embedding which is discarded after training (i.e. features are obtained from the underlying ResNet) [Chen et al., 2020]. The models are trained for 150 epochs with a batch size of 256 and a learning rate of 0.1, both using the Adam optimizer [Kingma and Ba, 2014]. Tile2Vec uses the triplet loss with a margin of 0.1 following Jean et al. [2019], whereas the contrastive method is trained with the loss in eq. (2.1). The **autoencoder** [Kramer, 1991] baseline has an encoding module consisting of three convolutional layers with 8, 16 and 32 filters, respectively. This is followed by two fully connected layers of size 256 and 128, meaning the feature space has 128 dimensions. The decoding module had a single, fully connected layer of size 512, followed by three transpose convolutional layers. All convolutional layers were followed by a max-pooling layer, and all layers, except the output layer, were passed through a Leaky ReLu activation with a negative slope of 0.01. The autoencoder was optimized to minimize the mean squared error between the input image and reconstruction, training over 40 epochs with a batch size of 256 and using the Adam optimizer with a learning rate of 0.001. For principal component analysis (**PCA**) [Tipping and Bishop, 1999] and independent component analysis (**ICA**) [Hyvärinen and Oja, 2000] each image was flattened to be a vector of size 12,288. The top 10 principle or independent components are then computed, and the activations of these components for each image is treated as extracted features.

For all methods, the data were prepared in the same manner. The images are normalized to have zero mean and unit variance. We consider random 64x64

parts of the original 512x512 image, for Tile2Vec and contrastive two neighbors consists of two such parts from the same base image. In addition to using nearby parts of a remote sensing image, we also rotate and flip the images randomly and randomly zero out 16-by-16 sub-images to extend the data set further. The data set is first divided into a 70 percent set for unsupervised training of the feature extractors; the remaining 30 percent of the data is then randomly split into two 15 percent sets for training and evaluating the top-level classifier. We assess feature quality for a total of 10 rounds by evaluating the accuracy of a top layer classifier using the extracted features and ground truth labels – using either a logistic regression classifier (LR) or a random forest classifier (RFC) [Pedregosa et al., 2011]. We chose these methods as they are well-known and often perform well in practice. The RFC is trained using 100 decision tree learners and Gini impurity as the criterion for splitting. We report test accuracy for the top layer classifier using the extracted features, giving the mean and standard deviation accuracy for these methods in Table 2.2. As can be seen, our contrastive method outperforms all baselines for both classifiers. The predictions are likely to further improve with a larger data set, and we emphasize that the task is difficult as one cannot directly see invasive species from the images but must instead consider what habitat might be suitable for them.

2.4.2 Supervised Experiments

In ecological or biological domains, it is often easy to obtain unlabeled remote sensing data but generating labels requires sending domain experts to the field, which is a laborious process. A strong unsupervised feature extractor can potentially lead to a high accuracy classifier with much fewer labels, correspond-

ing to substantial savings in fieldwork. With this in mind, we investigate our method's accuracy compared to fully supervised methods with different amounts of labeled data available. We consider strong baseline deep learning classifiers DenseNet [Huang et al., 2017], ResNet [He et al., 2015b] and AlexNet [Krizhevsky et al., 2012], and additionally compare to the features extracted via Tile2vec. The data set is split into a training set of size 70 percent and a testing data set composed of the remaining 30 percent. From the training set, a variable percentage of the labels were then removed. This was done to simulate a real-world setting where there are ample unlabeled data for unsupervised methods to use, but few labels for supervised methods. Within the training set, we ran experiments with the following percent of labeled data available: 6, 12, 18, 24, 30, 36, 40, 50, and 75 percent. For the unsupervised methods (contrastive and Tile2Vec), we first train the unsupervised feature extraction on the entire training data set, using the same hyperparameters as the unsupervised experiments. We then train a top-level classifier (RFC) on the available labeled data using the extracted features as input. For the fully supervised methods, we train them on the available labeled data for 40 epochs using the Adam optimizer with a learning rate of 0.1 to optimize the cross-entropy loss, resulting in convergence for the loss. We then test each model's accuracy on the test set. We repeat each of these experiments for five rounds and report the mean and standard deviation accuracy. As can be seen in Figure 2.5, our method outperforms all others. This highlights how our method can be used to greatly reduce the number of needed labels, and therefore the cost, to obtain an accurate classifier. With a larger amount of labeled data, fully supervised methods, like DenseNet or ResNet, likely would match the performance of our method, but the experiments suggest that when labeled data are scarce, spatial contrastive learning provides efficient feature extraction.

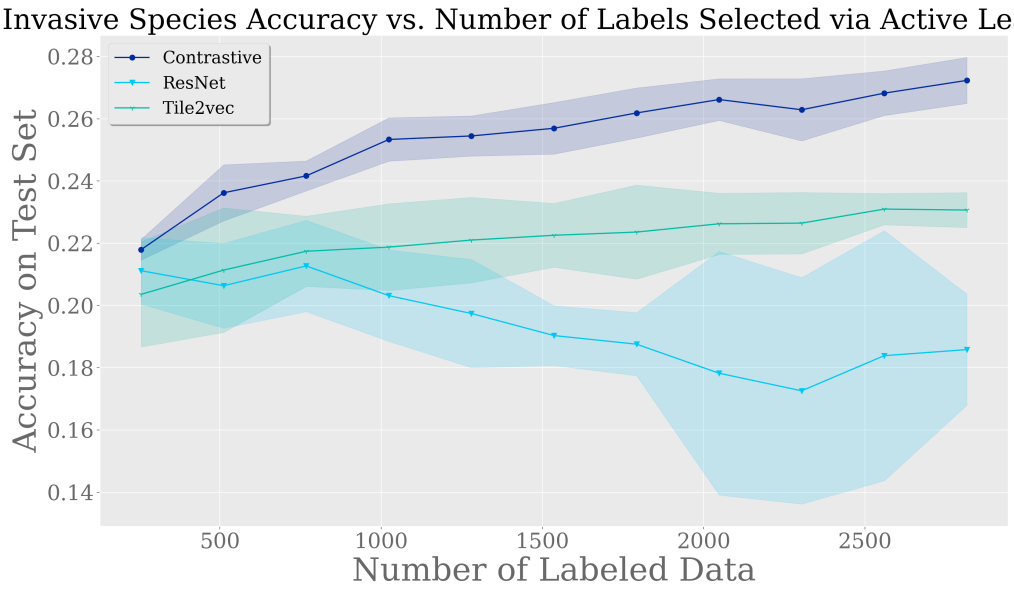

Figure 2.6: We simulate field deployment by performing active learning across the invasive species data set. Unlike classical active learning, where one queries for both images and labels, we propose to perform the active learning in the embedding space obtained from unsupervised models. This approach outperforms traditional active learning, and spatial contrastive learning outperforms Tile2vec.

2.4.3 Active Learning

The ultimate goal of our collaboration with the New York Natural Heritage Program is to use remote sensing images to direct ecologists to locations deemed likely to contain invasive species. To roughly simulate this setting, we consider performing active learning over our invasive species data set. We are given a fixed number of queries for labels and must use these to train as accurate a prediction model as possible (evaluated on a held-out test-set). Unlike traditional active learning where one chooses both images and labels to add to the train set, we propose to use all images for unsupervised pre-training, and then only conduct active learning on the labels. In our setting, remote sensing data are

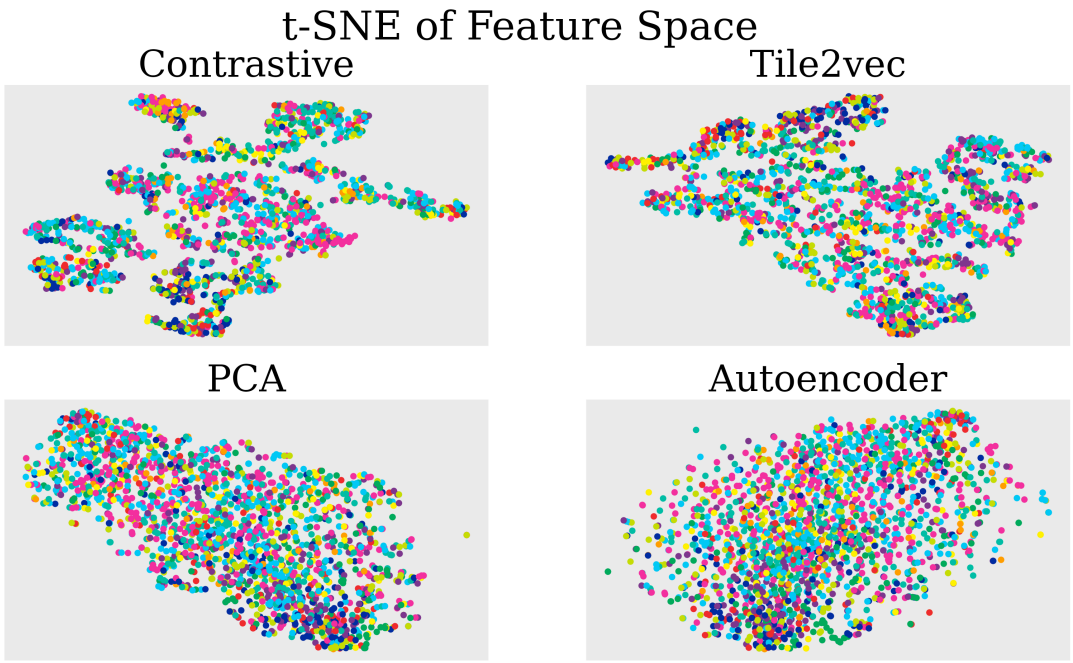

Figure 2.7: t-SNE visualization of the feature space for invasive species data set for our method, Tile2Vec, PCA and Autoencoder, where each color indicates a particular invasive species class. The illustration suggests that Tile2Vec encourage looser clusters that can lead to generalization error, which could be a reason our method performs better.

inexpensive but sending ecologists to perform observations on the ground is more expensive. The idea of conducting active learning only on the labels has the potential to greatly speed up ecological work but is only possible if useful features can be extracted in an unsupervised fashion.

In this experimental setting, we compare our contrastive model, Tile2Vec and standard active learning (i.e. selecting both images and labels) using ResNet-18. The data are prepared in the same manner as the supervised experimental setting. We first train the contrastive method and Tile2Vec unsupervised feature extraction using the same setting as described in the unsupervised experiments.

We then randomly initialize each method to have 256 labels and allow it to train a supervised model. For the contrastive method and Tile2Vec we trained a RFC from the features extracted to the labels, using 100 decision tree learners and the Gini purity as the splitting criterion. For ResNet, for each round the ResNet model was trained on all available labeled data for 40 epochs, with a batch size of 256, optimizing the cross-entropy loss with the Adam optimizer and a learning rate of 0.1. Then we run a series of 10 rounds of active learning, using the entropy sampling method for active learning [Settles, 2012]. Each round we generate predictions on all unlabeled images in the training set and take the set (of size 256) which produced the largest entropy in the classification predictions. This selected set is then added to the available labeled data for each method, and the model re-trains on the now larger train set and reports its accuracy on the test set. This experiment was run five times for the contrastive method, Tile2Vec, and ResNet. In Figure 2.6, we plot the mean and standard deviation accuracy against the amount of labeled data available. This demonstrates that our method can be used in an active learning setting to guide which labels should be taken; we hope to study this approach further in the future. This model is, of course, a simplification compared to actual field deployment, and many practical differences compared with real deployment remain.

2.4.4 Qualitative Analysis

To probe the learned features, we use t-SNE [Maaten and Hinton, 2008] to visualize the features extracted by our method, Tile2vec, the autoencoder and PCA, as seen in Figure 2.7. For this experiment, the embedding data were extracted from each method for the test set after training each method to convergence,

using the same parameters as per the unsupervised experimental setting. The illustration suggests that Tile2Vec and spatial contrastive learning results in a clearer structure than PCA and autoencoders. Further, we suspect that the L2 loss used in Tile2vec may not constrain the clusters as can be seen in this visualization, perhaps leading to a weakened ability for the model to generalize.

2.5 Additional Data Sets

Eurostat

While the invasive species application is the main focus of this work, we conduct experiments on additional data sets to show the generality of the proposed method. We consider landcover classification, where one attempts to classify a remote sensing image as belonging to some specific landcover type (forest, road, river, etc.). This task has practical implications in computational sustainability and can, e.g., be used for monitoring deforestation. We first consider the Eurostat data set, which consists of 27,000 Sentinel-2 satellite images of various landcover types from Europe [Helber et al., 2019]. The data and baselines were all prepared as for the invasive species data set, and the results of our experiments on this data set can be seen in Table 2.3. As can be seen, our method outperforms all other feature extractors on this data set.

NAIP

We additionally consider the NAIP data set [Jean et al., 2019], which contains a fourth spectrum band, which highlights our method's ability to handle multi-

	EuroStat	
Embeddings	RFC	LR
Contrastive	71.47 ± 0.40	71.23 ± 0.67
Tile2Vec	60.49 ± 0.63	49.77 ± 0.56
AutoEncoder	60.22 ± 0.92	57.27 ± 0.51
PCA	65.72 ± 0.80	43.13 ± 0.94
ICA	65.30 ± 0.86	21.25 ± 5.11

Table 2.3: Accuracy for Eurosat [Helber et al., 2019]. All experiments were for 10 rounds, and the average value is given $\pm$ the standard deviation.

	NAIP	
Embeddings	RFC	LR
Contrastive	66.47 ± 1.88	58.38 ± 1.96
Tile2Vec	62.70 ± 1.51	53.17 ± 1.52
AutoEncoder	61.50 ± 2.77	40.65 ± 2.03
PCA	60.92 ± 1.43	57.55 ± 2.51
ICA	62.58 ± 1.97	34.08 ± 1.58

Table 2.4: Accuracy for NAIP [Jean et al., 2019]. All experiments were for 10 rounds, and the average value is given $\pm$ the standard deviation.

spectral remote sensing. A difference in this experimental setting is that the NAIP data set has train and test set sources from different geographical locations and that one must obtain feature extraction that is robust under such distributional shift. See Jean et al. [2019] for details. Again, for this data set, we consider the same unsupervised experimental setting as per the invasive species data set. We use the entire training set to train our unsupervised methods, and then split the test set into two equal sets. For the PCA and ICA feature extractors, because of the fourth color channel, the input vectors were of size 16,384 as opposed to 12,288; otherwise, the data and baselines were all prepared in the same manner. The results of our experiments on this data set can be seen in Table 2.4. As can be seen, our method outperforms all other feature extractors on this data set.

2.6 Challenges and Opportunities

Whereas this paper has focused on computational aspects of invasive species management, deploying and using our models has many practical challenges and opportunities that we here expand upon. Firstly it is important to note that even an accuracy around 25 % can be useful for directing fieldwork and that it can complement classical approaches. We also note that the problem is hard as we only observe the habitat and not the invasive species themselves. Certain habitats can be favorable for invasive species, but that does not necessarily imply species presence. We also emphasize that there are variations within invasive species habitats. Ecologists know that the hemlock wooly adelgid lives off of hemlock trees, but an exact understanding of how the forest characteristics interact with the spreading rate is lacking [Oten et al., 2014]. Not all hemlock forests are identical, and there might be variations in e.g. tree density that influence spreading. Furthermore, land cover types are often coarse and might be on the level of "evergreen forest" rather than specifying e.g. tree species composition. Proximity to roads, trails, and water bodies can often impact invasive species spread, and their presence is easily detected from satellite images but not necessarily captured by land cover. The habitats can also pose a problem for our machine learning models. Many of the terrestrial plants we used can occur in the same or very similar forested habitats (and same for the aquatic plants in aquatic habitats). This could result in misclassifications in machine learning outputs for a particular species. Misclassification could have cost implications for managers by either sending managers to unsuitable sites or possibly missing a key population that should be managed.

Secondly, we highlight how the data are collected. The New York State (NYS)

invasive species program is managed by a collection of regional organizations which use paid professionals and citizen scientists. Strategies include recruiting citizen scientists for shorter fieldwork excursions, allowing citizens to report invasive species via an online reporting tool, or inviting the public to participate in plant removal events. How the data are collected likely leads to some bias, for example, locations that are easier to reach might be monitored more frequently. Bias is common in citizen science applications, and e.g. the eBird project suffers from road-side bias [Chen and Gomes, 2019]. However, we note that many invasive species spread via humans, so bias towards populated areas is not necessarily bad.

2.7 Related Work

Unsupervised deep learning has a long history [Kramer, 1991]. A popular line of work employs auto-encoders [Hinton and Zemel, 1994]. Contrastive predictive coding has also been researched since Oord et al. [2018], and typically relies on predicting parts of the input given other parts [Bachman et al., 2019, Srinivas et al., 2020]. Specifically, in Oord et al. [2018] the method relies on finely dividing natural images into subparts and then autoregressively making predictions. The idea of contrastive coding has inspired a lot of recent work [Chen et al., 2020, He et al., 2020]. We spatially augment contrastive learning methods, and instead of considering crops of the same image, we use non-overlapping parts of the same landscape – relying on spatial smoothness of landscape features. The strategy of considering neighbors is popular in natural language processing (NLP) [Devlin et al., 2018], and is also used for word embeddings [Mikolov et al., 2013, Pennington et al., 2014]. With the advent of deep learning, machine learning

for remote sensing has received much attention. The most closely related work is Tile2Vec [Jean et al., 2019], which uses a strategy reminiscent of word embeddings to generate remote sensing embeddings, specifically using the triplet loss. The work of Fried et al. [2017] also uses the triplet loss for geographic data but instead relies on supervision. Contemporaneous work also includes Kang et al. [2020], which similarly to this study considers unsupervised learning for remote sensing, we emphasize that our work also includes applications to active learning. An important application of remote sensing is poverty mapping, where given access to remote sensing data, one tries to predict economic conditions "on the ground" [Xie et al., 2015]. Another important use case is the prediction of crop yield from remote sensing data [Setiyono et al., 2014, Wang et al., 2018]. Invasive species management is an important ecological problem with economic implications, and computational aspects of the problem have received considerable interest. Researchers have used remote sensing via airplanes to identify invasive species from handcrafted features [Ustin et al., 2002, Asner et al., 2008, Piiroinen et al., 2018]. Researchers have used reinforcement learning [Taleghan et al., 2015], mixed integer programming [Büyüktahtakın et al., 2014] and stochastic dynamic programming [Shea and Possingham, 2000] to generate management strategies. Modelling work includes Hawkes processes [Gupta et al., 2018], extensions of the firefighter problem [Spencer, 2012] and predator-prey dynamics [Bjorck et al., 2018].

2.8 Discussion

In this work, we have considered the use of remote sensing data for invasive species management, motivated by an ongoing collaboration with the New

York Natural Heritage Program. By spatially augmenting contrastive coding methods, we show how to obtain low-dimensional embeddings of remote sensing data. Our experiments show that this method outperforms baselines, and we additionally show how one can perform active learning in this embedding space to improve sample complexity. For future work, we hope to further study how to integrate these methods into deployment.

CHAPTER 3

PASSIVE ACOUSTIC MONITORING WITH MACHINE LEARNING

3.1 Introduction

Poaching, illegal logging, and infrastructure expansions are some of many current threats to biodiversity, and large mammals are particularly susceptible. To effectively allocate conservation resources and develop conservation strategies, endangered animal populations need to be accurately and economically surveyed, but for species that roam large or inaccessible areas monitoring by humans becomes intractable. A promising approach for species using acoustic signals for communication is passive acoustic monitoring (PAM), which involves the use of autonomous recording devices scattered throughout habitats. Compared to video monitoring, acoustic monitoring is not limited by line of sight and is typically considerably cheaper and requires less bandwidth for transferring the data. However, extracting useful data from these soundscapes is non-trivial and automatic approaches are necessary.

In this work, we focus on passive acoustic monitoring in the context of African Forest Elephants which are a keystone species in the rainforests of the Congo Basin (the second largest expanse of rainforest on earth and among the most speciose). Conserving viable populations of forest elephants protects the biodiversity of their landscape, but the expansiveness of the rainforest and the difficulty of monitoring animals within it makes this problematic. Since elephants communicate over long distances via infrasonic signals referred to as rumbles [Hedwig et al., 2018] they are particularly suited to an acoustic approach. These characteristic vocalizations provide information on occupancy, landscape use,

population size, and the effects of anthropogenic disturbances [Wrege et al., 2017].

In real-time threat-detection and population monitoring the bandwidth of the

Figure 3.1: The African forest elephant (Loxodonta cyclotis) is the smallest of the three extant elephant species, a keystone species in the rainforests of the Congo Basin, and is entirely relied upon by many trees to disperse their seeds [Campos-Arceiz and Blake, 2011]. Due to their highly-valued ivory tusks, the elephant is a typical target for poachers in central Africa and the population has fallen by more than 60% in the last decade [Morelle, 2016]. Population monitoring is critical for the elephant's survival, and in this work, we consider combining passive acoustic monitoring and artificial intelligence towards this end.

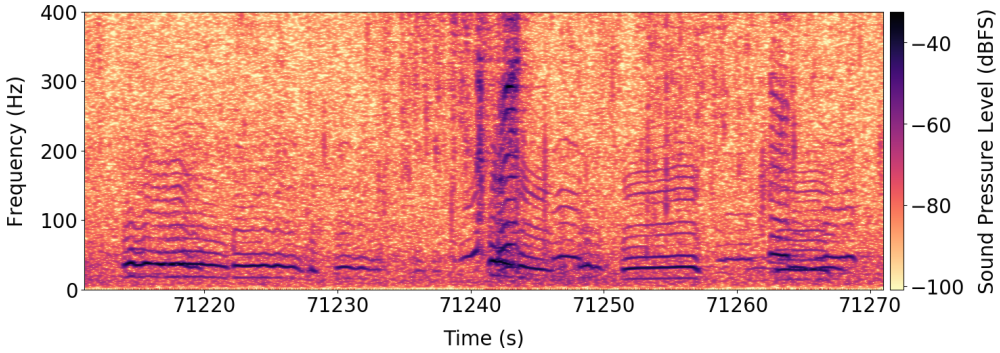

Figure 3.2: A spectrogram of several elephant rumble vocalizations within a 60-second segment of sound. The rich harmonic structure is typical of rumbles, however, since higher frequency elements attenuate rapidly with distance, recording these higher frequency elements depends on source amplitude and distance. Thus, it is difficult to infer distance from harmonic structure alone.

wireless networks becomes a bottleneck, and one additionally has to use efficient data representations to only communicate the necessary information. In many lossy compression schemes signal components inaudible to humans such as low frequencies are given low bit-rates. In the context of low-frequency elephant calls, this is a very poor strategy. Using a differentiable proxy for non-differentiable bit truncation, we are able to cast this problem as an end-to-end differentiable setup, which can be trained via stochastic gradient descent (SGD) to get improved compression.

3.2 The Dataset

3.2.1 Data Sourcing

Established in 2000, the Elephant Listening Project at the Cornell Lab of Ornithology uses acoustic methods to study the ecology and behavior of forest elephants in order to improve evidence-based decision making concerning their conservation. The Elephant Listening Project has recorded sounds from over 150 different locations, amassing more than 700,000 hours of recordings. These varying environments provide the source material for generating training data for algorithm development. The dataset we consider in this work was collected between 2007 and 2012 from three sites in Gabon and one in the Central African Republic, which will be referred to as Ceb1, Ceb4, Dzanga, and Jobo. At all locations, a single recording device was placed in a tree 7-10 meters above the ground near forest clearings (25 to 50ha) where elephants congregate for multiple purposes. The recording devices obtained audio recordings at sampling rates of

2000 (12-bit resolution) or 4000Hz (16-bit) and elephant calls up to approximately 0.8 km away were recorded. As is typical in bioacoustic applications the animals are detected infrequently and different locations have variable density, see Table 3.1. Additionally, multiple other sources of sound are recorded, both man-made and natural. For example, Ceb4 is close to a road and the recordings include signals associated with logging and gunshots.

3.2.2 Acoustic Characteristics of the Dataset

The primary mode of communication among elephants is a low-frequency vocalization known as a rumble, typically lasting between 2 and 8 seconds. These sounds have distinct frequency characteristics, with a low fundamental frequency (8 - 34Hz), often several higher harmonics, and slight frequency modulation. A typical recording is shown in Figure 3.2. At large gatherings, multiple elephants often make simultaneous or overlapping calls (see for example Figure 3.2 where two calls overlap). Other complications are the variability of the dataset, for example, some recording sites are closed to logging concessions which are often visited by motorized vehicles which can be recorded by the detector. Natural sources of noise include heavy wind, rainfall, insects chirping and thunderstorms, see Figure 3.3 for further examples.

3.2.3 Data Quality

The labeling of rumbles to be used in the training and testing of detection algorithms was done by both experts and trained volunteers at the Cornell Lab

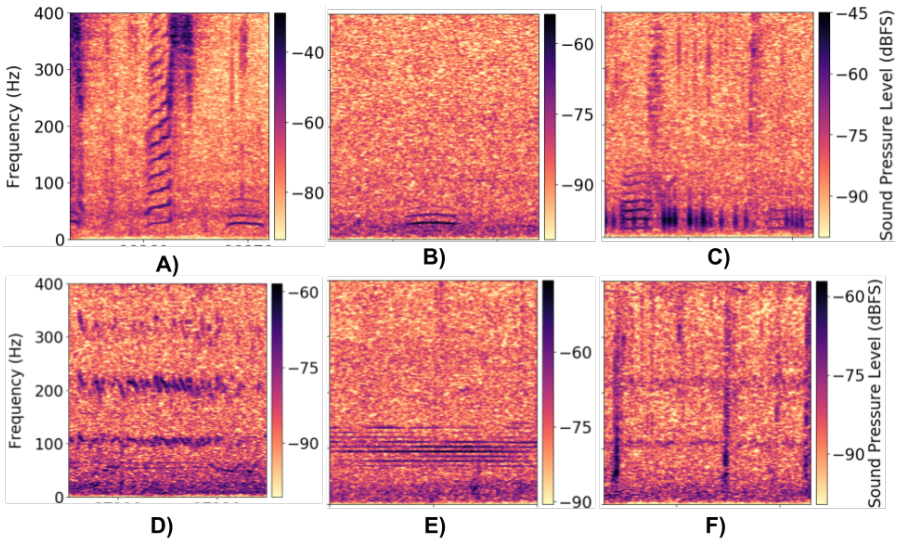

Figure 3.3: Examples of the diversity of acoustic signals encountered in sound streams from Central African forest environments. **A)** an elephant call combining both tonal and chaotic (broadband) sounds, often produced in agonistic situations. **B)** an elephant rumble with few harmonics (source far from microphone and/or low amplitude). **C)** signals emitted by a dwarf crocodile (Osteolaemus tetraspis), including some harmonics similar to those of elephants. **D)** the buzzing of insects **E)** a motorized vehicle **F)** sound of splashing of water as elephants move through a stream.

of Ornithology. The volunteers were recruited by a combination of work-study positions and information spread via word-of-mouth and were asked to identify individual elephant calls and their temporal extent in the recording. Positive labeling was based on a set of criteria developed by experts with more than ten years of experience with forest elephant vocalizations and experience with potentially confusing environmental sounds. Volunteers followed a detailed training program that concluded with them labeling rumbles in two 24 hour long test sound files. The labels generated by the volunteers for the test files were compared to those of an expert. If the results were within 5% of each other, the volunteer was considered trained; if not, he/she repeated the process on

Location	Dates Collected	Labelled hours	Num. calls	Apx. % Calls
Ceb1	09/04 - 11/06	1870	52810	0.784 %
Ceb4	08/06 - 11/03	1280	23038	0.500 %
Jobo	09/05 - 11/06	1437	28609	0.553 %
Dzan	11/04 - 12/02	312	63792	21.8 %

Table 3.1: The statistics of the datasets by location. The Apx. percentage of calls refer to what portion of the audio recordings contained elephant calls. The dates are given in YY/MM format.

other sound files until the 5% or less difference was achieved. The occasional further review of volunteer labeling efforts by the experts maintained reasonable consistency among all labelers (reliability $> 98\%$). Statistics about the dataset and the labeling can be seen in Table 3.1. To facilitate online crowdsourcing we have created an online labeling application for labeling. The application contains a tutorial where participants can first learn about the characteristics and variations of elephant calls, and then other sounds that might occur in recordings. Once trained, participants can then label elephant calls in audio segments by using the application's annotation tool. By giving the same spectrum to multiple participants one can gauge the accuracy of individual users and can encourage truthful responses. These issues will be addressed further in future work.

3.3 Compression

3.3.1 Background

The ultimate aim of passive acoustic monitoring is to provide accurate real-time detections of elephant vocalizations and threats. It is infeasible to perform neural

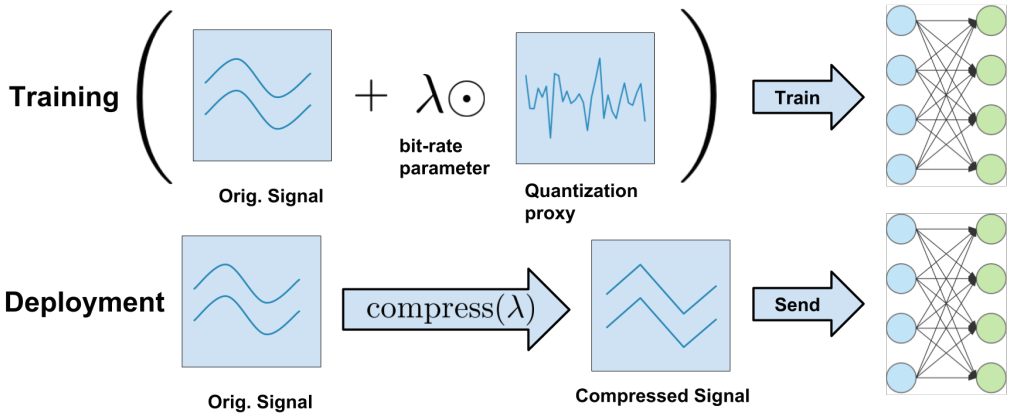

Figure 3.4: The main idea behind our end-to-end compression scheme is to introduce a bit-rate vector λ and i.i.d. noise that serves as a proxy for the quantization error. By optimizing lambda λ one can adjust the quantization level for different frequency bands since it is differentiable we can optimize λ jointly with a neural-network classifier to find compression strategies that result in signals that are useful for classification. At deployment, the bit-rates of individual frequency channels are used for compression at the recording devices so that the data sent is minimized.

network computations on the recording devices, and hence the devices need to send their data over the wireless networks of sub-Saharan Africa. Unfortunately, wireless infrastructure is still lagging behind in this area of the world [Aker and Mbiti, 2010] and available bandwidth is small and data-transfers are expensive. To make real-time passive acoustic monitoring cost efficient one has to only transfer the most relevant information across the wireless network.

A natural strategy for reducing the data-transfers across the wireless network is to compress the acoustic data. Most lossy compression codecs crucially rely on the specifics of the human auditory system to remove data that are irrelevant to the experience of a human listener. For example, it is well known that the sensitivity of the human auditory system varies with frequency [Painter and Spanias, 2000], and hence many lossy compression algorithms remove low-

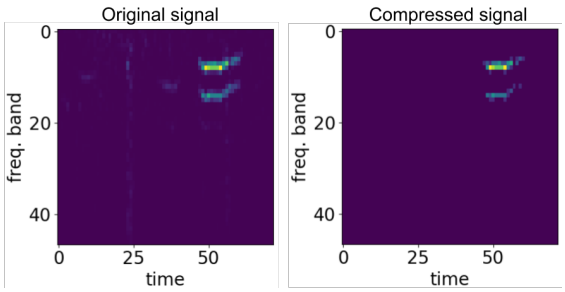

Figure 3.5: We here illustrate an example of quantization of a signal with elephant calls with extremely low bit-rate. Background signal almost disappears with quantization while the elephant call loses much of its nouances.

frequency components or simply use a low bit-rate for them. In the context of elephant monitoring, this is poor strategy since the elephants communicate by low-frequency rumbles. It is clear that we need to develop compression strategies uniquely suited for the elephant calls and for the neural networks that will analyze them. As neural networks are well known to be resistant to minor random perturbations [Micikevicius et al., 2017] lossy compression is a promising avenue. Additionally, as passive acoustic monitoring has applications to many species, from small birds [Bardeli et al., 2010] to marine mammals [Bittle and Duncan, 2013], data-driven approaches such as ours avoids the laborious process of manually crafting audio codecs and can easily be adapted to new species. As it does not require any hand-crafted features or any specific information regarding the structure of animal vocalization (save for an approximate frequency range, information that is easily obtainable for most species), one would "only" need data to train the networks on to adapt our framework to other species.

3.3.2 End-to-end differentiable compression codecs

The most salient difference between our setup and the setting of typical audio compression applications is the differences in the frequency spectrum of the source and the fact that the listener is not human. To isolate this phenomenon and achieve a simple setup we only consider compression in terms of the different frequency bands. Other aspects of lossy compression, for example, lossless compression on top of lossy strategies, can be added to all methods we consider. We assume that the one-dimensional X that describes the sound waves has been transformed via FFT as a pre-processing step into $\hat{X}$, and consider the problem of assigning bit-rates to the different frequency bands. Simple operations such as FFT and bit-truncation can easily be implemented in the rudimentary hardware of the recording devices. We propose a method that jointly optimizes for low bit-rates of the frequency channels and high classification accuracy.

Our algorithmic setup is illustrated in Figure 3.4. We want to assign different bit-rates to different frequency-channels, we simply truncate the bit representation of elements of the channels which has the effect of lowering the precision. Our key insight is to exchange a non-differentiable bit-truncation by a differentiable proxy – we simply model truncation as additive Gaussian noise, a common model of quantization error [Gray and Neuhoff, 1998]. We let the components of the vector λ denote the bit-rates of various frequency channels, and let β be a matrix with dimensions $t \times f$ with independent standard Gaussian entries, where there are t time-steps and f frequency bands. The truncation error is the proportional to by the matrix $\exp(-\lambda) \odot \beta$, where the entries (i, j) are equal to $\exp(-\lambda_j)\beta_{ij}$. This ensures that the additive errors in the original elephant spectrogram $\hat{X}$, which models bit-truncation, are independent but that each frequency

band has its own error scale. The input to the neural networks is thus $\hat{X} + \lambda \odot \beta$, and we simultaneously optimize the network parameters ω for large classification accuracy and the total bit-rate which is simply expressed as $\sum_i \lambda_i$, balancing these two objectives with the hyper-parameter μ

$$\frac{1}{|D|} \sum_{X,y \in D} \mathop{\mathbb{E}}_{\beta \sim N} \left[L\left(y, \mathrm{DNN}_\omega\left(\exp(-\lambda) \odot \beta + \hat{X} \right) \right) \right] + \mu \sum_i \lambda_i \tag{3.1}$$

Here the dataset D contains tuples $(\hat{X}, y)$ of data $\hat{X}$ and labels y, $L(y, \hat{y})$ denotes the loss function used to measure goodness of fit between ground-truth label y and estimated label $\hat{y}$. We again use the cross-entropy for the loss function. This function can be optimized via SGD, where we exchange the expectation $\mathbb{E}[\,\cdot\,]$ by sample averages.

3.3.3 Experiments

We compare different compression strategies by how well they transmit the important information as measured by how well a classifier can be trained to classify compressed elephant spectrograms given a fixed bit-rate. For all compression strategies, we will use a Densenet model. The original Fourier signal has elements put into one of the 2^{32} bins represented as 32 bit signed integers, lowering the bit-rate simply corresponds to removing the least significant bits with the sign bit is removed last. This has the effect of quantizing the signal and removing small variations in signal strength while keeping the large variations, see Figure 3.5. We enforce that no less than 5 bits are used for each frequency band as the dynamic range of the audio signal has the effect of completely erasing

Method / Bit-rate	Ceb1	Ceb4	Jobo	Dzan
Ours / 47	84.57	83.98	86.31	78.43
Human / 47	83.62	81.31	85.76	69.44
Ours / 141	92.81	92.21	93.19	77.96
Human / 141	86.61	91.90	90.32	73.51
Ours / 235	93.05	93.11	93.84	77.46
Human / 235	90.25	92.34	91.64	76.93

Table 3.2: The classification accuracy on the test-set for the given bit-rates.

the signal for smaller bit-rates. For assigning bit-rates via optimizing (3.1) we use a Densenet architecture for evaluating the compression quality. To ensure specific total bit-rates we assign bit-rates to various frequency bands proportional to the values of the components of λ. We compare our method against the method of assigning bit-rates proportional to the sensitivity of human hearing, using the well-known model of how human auditory sensitivity vary with frequency of [Painter and Spanias, 2000]. The proportional allocation excludes the 5 bits needed for the dynamic range of the signal. The results for various locations and bit-rates are given in Table 3.2, where we can clearly see that our proposed method achieves superior performance for the same bit-rates. For very small and very large bit-rates the difference becomes smaller. Implementing our method leads to data compression of a factor roughly 116 compared to näively storing the 1000Hz signal in 32-bit floating point numbers while achieving little performance degradation. These savings are significant for the often poor wireless networks of sub-Saharan Africa.

3.4 Related Work

3.4.1 Bioacoustics

The field of bioacoustics has for a long time been interested in automatic approaches towards detecting and classifying animal sounds with the ultimate goal to accurately survey population size and behavior [McDonald and Fox, 1999]. As sound waves attenuate less in water, passive acoustic monitoring can cover vast underwater areas. Much effort has been in terms of large marine animals with characteristic vocalizations – predominately various whale species (Humpback, right [Thode et al., 2017], Baleen [Baumgartner and Mussoline, 2011], Blue and Fin [Širović et al., 2007]) and dolphins [Erbs et al., 2017]. Acoustic signals are the primary mode of communication for many underwater species and for large gatherings vocalizations typically overlap which together with long reverberation times becomes challenging. Techniques used to overcome these issues include blind source separation [Zhang and White, 2017], pitch-tracking via dynamic programming [Baumgartner and Mussoline, 2011] and kernel methods [Thode et al., 2017].

On land, efforts towards bioacoustics have primarily focused on various bird species, owing to the characteristic songs many of them use for mating and communication. As bird species typically have unique songs PAM makes it possible to accurately survey populations of endangered species, whereas using direct visual observations becomes problematic for species that are small and/or occupy canopies [Bardeli et al., 2010]. Popular strategies include SVMs based upon MFCC [Dufour et al., 2014], segmentation via deep learning [Koops et al., 2015] and dictionary learning [Salamon et al., 2017]. Beyond birds, insects

[Ganchev and Potamitis, 2007], bats [Mac Aodha et al., 2018] and monkeys [Turesson et al., 2016] have all been considered. Elephants have been studied from a similar perspective as ours in [Pleiss et al., 2016]. For many of these species, especially many birds, the vocalizations occupy a relatively small frequency band making models less sensitive to noise and intra-population variability in vocalizations, hence making them unsuitable for elephant monitoring.

3.4.2 Machine-learning for Audio

Machine learning for audio-signals has primarily focused on human speech due to applications such as virtual assistants, automatic transcription, and translation. For a long time, mainstream research was primarily propelled by using the EM-algorithms for training Hidden-Markov-Models [Hinton et al., 2012]. Features for audio input could often be encoded via MFCC [Sahidullah and Saha, 2012], and rich distributions could be represented via Gaussian-Mixture-Models [Juang et al., 1986]. While using neural networks for acoustic applications was conceived more than 25 years ago [Bourlard and Morgan, 2012], it was in only 2009 that deep learning approaches were shown to be competitive with more traditional "hand-crafted" machine learning approaches [Mohamed et al., 2009]. Deep learning has now gained mainstream traction and it has become the dominant paradigm. State-of-the-art speech recognition often relies on recurrent neural networks [Graves and Jaitly, 2014] [Sak et al., 2014], where convolutional layers can automatically extract features [Sainath et al., 2015]. Beyond speech recognition, deep learning for acoustic sensing in smartphones has been investigated [Lane et al., 2015].

3.4.3 Compression for Audio

Compression for acoustic signals has been studied for a long time due to applications such as storing music on handheld devices and sending human conversations across networks, and many audio compression methods rely on essentially handcrafted features, for example, wavelets [Jagadeesh and Kumar, 2014]. Most methods for lossy compression of audio has the goal of ensuring signals are audible to humans, and hence most models are based upon the models of human hearing, so-called psychoacoustic models. A salient feature of human hearing is that its sensitivity varies with frequency [Painter and Spanias, 2000], a common strategy is to transform the audio-signal with the modified discrete cosine transform (MDCT) and address frequency bands individually. Another phenomenon of human hearing is called simultaneous masking where signal A can make signal B (which is of a different frequency and intensity) inaudible [Jagadeesh and Kumar, 2014].

While traditional compression schemes have typically relied on handcrafted features, the advent of deep learning has spurred interest in data-driven approaches to compression. Previous research has primarily focused on images and video, proposing various continuous and differentiable proxies for entropy and quantization, see for example Ballé et al. [2016] and Agustsson et al. [2017]. The only work on audio compression known to the authors is on human speech Kankanahalli [2017] which has is different in terms of frequency distribution, complexity, and dataset cleanliness; the proposed architecture relies on softmax quantization.

3.5 Discussion

Managers of the protected areas designed for the forest elephants are interested in better conservation tools but need to see definitive proof of their efficacy. This collaboration will be instrumental in developing a rapid work-flow for the current acoustic monitoring project in northern Congo, which covers 1500 square km of rainforest and generates seven terabytes of sound data quarterly. If useful information about elephant populations and human encroachments can reach managers within a reasonable timeframe, the potential to expand acoustic monitoring across the Congo Basin becomes a reality. The aim of our work is to contribute to making automatic PAM a reality. We have addressed how wireless network infrastructure is often lacking in sub-Saharan Africa data transfer quickly becomes a bottleneck for real-time systems. To circumvent this issue we have introduced a novel scheme for jointly optimizing bit-rates and prediction accuracy, which beats a baseline based upon models of human hearing.

CHAPTER 4

UNMIXING SPECTROSCOPIC DATA VIA MATRIX FACTORIZATION

4.1 Introduction

Matrix factorization has become a ubiquitous technique in data analysis, with applications in a variety of domains such as computer vision [Shashua and Hazan, 2005], topic modeling [Lee and Seung, 1999], audio signal processing [Smaragdis, 2004], and crystallography[Suram et al., 2016a]. Often the phenomena considered is naturally non-negative. In non-negative matrix-factorization, the goal is to explain a non-negative signal as the product of (typically) two non-negative low rank matrices. Nonnegative matrix factorization is known to be NP-Hard [Vavasis, 2009a], so a general algorithm for matrix factorization most likely scales exponentially in the worst case.

We consider a challenging and central problem in materials discovery, so-called phase-mapping, an inverse problem whose goal is to infer the materials' crystal structure based on X-ray sample data, see Figure 4.1(Left). Phase-mapping was shown to be NP-Hard [Le Bras et al., 2011]. Existing approaches to phase mapping, discussed in the next section, do not satisfy all the problem constraints. Furthermore, approaches that explicitly try to incorporate the main problem constraints have prohibitive run times on typical real-world data, hours or days, while still not producing solutions that are completely physically meaningful.

We propose a novel **Interleaved Agile Factor Decomposition (IAFD)** approach that "lazily" relaxes and postpones non-convex constraint sets (the lazy constraints), iteratively enforcing them when violations are detected, see Figure

4.1(Right). IAFD uncovers the main underlying problem structure revealed by the sample data by rapidly performing a large number of lightweight gradient-based moves. In order to incorporate more intricate combinatorial constraints, the algorithm interleaves the multiplicative gradient-based updates with efficient modular algorithms that detect and repair constraint violations, while still ensuring fast run times, scaling up to large scale real-world problems. Our experimental results show that IAFD is several orders of magnitude faster and its solutions are also in general considerably better than previous approaches. Our work provides an efficient approach to solving a central problem in materials discovery, while paving the way towards tackling constrained matrix factorization problems in general, with broader implications for data science.

4.2 The Phase Mapping Problem

In search of new materials a common experimental method is to deposit several elements onto a sample wafer at different angles. The sample locations on the wafer receive different concentrations of the elements. As a result, distinct and potentially undiscovered materials are formed at different locations. All materials can be characterized by a one-dimensional X-ray diffraction pattern $F(q)$, which can be measured at high energy accelerators. However, several phases might be present at one sample location and the X-ray diffraction pattern at that location then becomes a linear combination of a set of basis patterns, each corresponding to the pattern of one pure phase. Figure 4.1(Left) illustrates this phenomenon.

In the mathematical model of the problem, a matrix A representing a set of X-ray measurements on a sample wafer is obtained. Each column of A is a vector

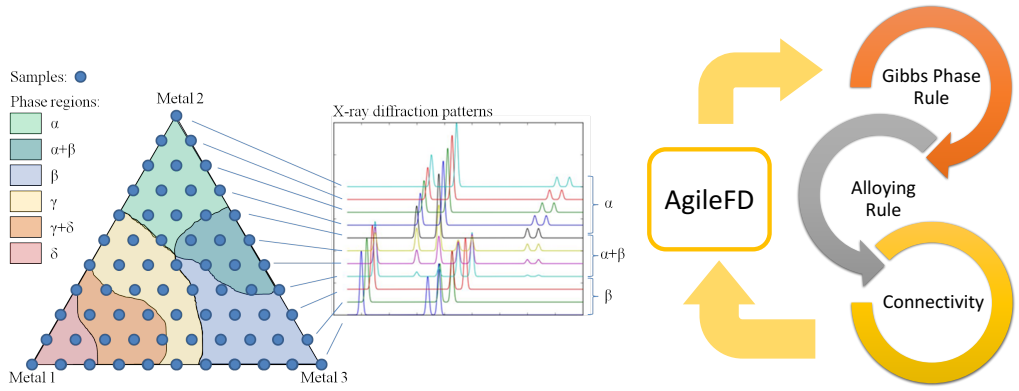

Figure 4.1: **(Left)** The goal of the phase mapping problem is to explain observed X-ray diffraction patterns at multiple sample locations in terms of the underlying phases or crystal structures of the materials. Here the X-ray diffraction patterns of sample locations on the right edge of the triangle are shown in the middle plot. The top four sample locations only have phase α, the bottom three only have phase β, while the middle four sample locations have both α and β. In addition, the X-ray diffraction patterns of both phase α and β are shifting to the right. **(Right)** At a high level, our Interleaved Agile Factor Decomposition (IAFD) algorithm starts with solving a relaxed problem using the multiplicative update rules of AgileFD [Xue et al.], without enforcing combinatorial constraints. Violations of the Gibbs' phase rule, the alloying rule, and the connectivity constraint in the relaxed solutions are then addressed by efficient modular algorithms, in an interleaving manner. This procedure is iterated, creating a closed loop involving AgileFD and the three modules.

representing the pattern $F(q)$ obtained at one sample location, sampled for Q fixed values of q. The phase mapping problem entails factorizing A into the product of W and H such that $A \approx WH$.

The matrix W encodes the characteristic patterns of pure phases while H represents how much of the different phases are present at individual sample location. A complicating factor of the phase-mapping problem is that the laws of thermodynamics induce a set of physical constraints on the possible underlying low rank representation. The solutions must satisfy these constraints, defined

below, and must additionally be nonnegative as the physical quantities described by the matrices cannot be negative. Efficient methods of solving this problem accelerates materials science and enables automatic experimentation in search of tomorrow's semiconductor and photvoltaic materials.

Shifting A phenomenon that complicates the matrix factorization is "shifting", where the X-ray patterns are changed in the sense $F(q) \rightarrow F(\lambda_k q)$, for some real number λ_k that is fixed for each phase k and column in A. For example, the X-ray patterns in figure 4.1 are shifting to the right. The problem can be circumvented by resampling the signal uniformly on a logarithmic scale, where multiplicative shifts becomes additive. For fixed m and k, the vector $(0, \ldots, 0, W_{1,k}, \ldots, W_{Q-m,k})^T$ formed by shifting the k-th column of W down by m entries (and filling 0 for remaining entries) describes basis pattern of phase k shifted by an amount controlled by m. We can then allow λ_k to attain M different discrete values by letting $m \in 0, 1...M-1$. By characterizing the H matrix with three indices, one per phase k, sample point n, and allowed discrete value of λ_k m, we can now express a linear combination of shifted basis patterns as $A_{qn} \approx \sum_{km} W_{q-m,k} H_{kmn}$. Since this specific formulation will be used, the constraints of the phase mapping problem will be given in terms of W_{qk} and H_{kmn}, however other formulations of the rules are possible[Ermon et al., 2012].

Gibbs' phase rule In a setting with three elements deposited, such as in figure 4.1, Gibbs' phase rule [Atkins and De Paula, 2006] states that the number of phases present at each sample location is at most three. Mathemati-

cally, it is equivalent to constraining the number of non-zero elements in vector $(\sum_m H_{1mn}, \sum_m H_{2mn}, \ldots)$ for any phase k to be no more than three. Thus, for fixed n we have $\| \sum_m H_{kmn} \|_0 \leqslant 3$.

Connectivity The connectivity rule requires that the sample points where a specific phase is present form a continuous domain on the sample wafer. For example, in Figure 4.1, each pattern occupies a continuous region. Mathematically, since we have a discrete set of measurements we describe the constraint via a graph G where sample points are nodes and nearby sample points are connected with an edge. This graph is obtained through Delauney triangulation [Lee and Schachter, 1980] of the sample points. A continuous domain then corresponds to a connected component on this graph, and we require that all sample points n with phase k present, i.e. $\sum_m H_{kmn} > 0$, form a connected component on G.

Alloying rule The shifting parameter λ_k for phase k may shift continuously across the sample points as a result of so called alloying. The alloying rule states that for points where λ_k is changing, Gibbs' phase rule becomes even stricter and requires $\| \sum_m H_{kmn} \|_0 \leqslant 2$. In this discrete setting we interpret λ_k of a point n as $\sum_m H_{nkm} m / \sum_m H_{nkm}$, which can be thought of as the expectation of m when we normalize H_{kmn} to a probability distribution. Two neighboring sample points n and n' with phase k present, which means $\sum_m H_{kmn} > 0$ and $\sum_m H_{kmn'} > 0$, are considered shifting if

$$\left\| \frac{\sum_m H_{kmn} m}{\sum_m H_{kmn}} - \frac{\sum_m H_{kmn'} m}{\sum_m H_{kmn'}} \right\| > \epsilon, \tag{4.1}$$

The alloying rule states that if Equation 4.1 is satisfied for any phase k and neighbouring sample points n' and n, then we must have $\| \sum_m H_{kmn} \|_0 \leqslant 2$.

4.2.1 Previous Approaches

Many algorithms have been proposed for solving the phase mapping problem, for example [Long et al., 2009], [Le Bras et al., 2011] and [Long et al., 2007]. Recently an efficient algorithm called AgileFD [Xue et al.], based on coordinate descent using multiplicative updates, has been proposed. If we let the matrix R represent the product of H and W, i.e. $R_{qn} = \sum_m W_{q-m,k} H_{kmn}$ these updates are

$$H_{kmn} \leftarrow H_{kmn} \frac{\sum_q W_{q-m,k}(A_{qn}/R_{qn})}{\sum_q W_{q-m,k} + \gamma}, \tag{4.2}$$

$$W_{qk} \leftarrow W_{qk} \frac{\sum_{mn} \frac{A_{q+m,n}}{R_{q+m,n}} H_{kmn} + W_{qk} \sum_{q'nm} H_{kmn} W_{q'k}}{\sum_{nk} H_{kmn} + W_{qk} \sum_{q'nm} \frac{A_{q'+m,n}}{R_{q'+m,n}} H_{kmn} W_{q'k}}. \tag{4.3}$$

The algorithm relies on manual refinement by domain experts to enforce combinatorial constraints, which makes it problematic to use in a scalable fashion.

Another approach called combiFD, able to express all constraints, has been proposed [Ermon et al., 2014]. It relies on a combinatorial factor decomposition formulation, where iteratively H or W are frozen while the other is updated by solving a MIP. This formulation allows all constraints to be expressed upfront, however solving the complete MIP programs is infeasible in practice.

4.3 Interleaved Agile Factor Decomposition

Given a non-negative Q-by-N measurement matrix A and the dimensions K and M of the factorization, the phase mapping problem entails explaining A as a generalized product of two low rank non-negative matrices W, H. The entire mathematical formulation becomes:

$$\begin{aligned} \min \quad & \sum_{qn} |A_{qn} - \sum_{mk} W_{q-m,k} H_{kmn}|, \;\; s.t. \;\; H \in \mathbb{R}_+^{K \times M \times N}, \;\; W \in \mathbb{R}_+^{Q \times K}, \\ & H, W \text{ satisfies Gibbs' phase rule, Connectivity, Alloying rule.} \end{aligned} \tag{4.4}$$

Representing the combinatorial rules as integer constraints has previously been tried [Ermon et al., 2014], however the resulting large MIP formulations are not feasible to solve in practice. Instead, we propose a novel iterative framework that interleaves efficient multiplicative updates with compact subroutines able to address specific constraints, called Interleaved Agile Factor Decomposition (IAFD). The algorithm is illustrated, at a high level, in figure 4.1 (Right). The central insight is that our constraints are too expensive to explicitly encode and maintain, however finding and rectifying individual violations can be done efficiently. This motivates a lazy approach that relaxes and postpones non-convex constraint sets (the lazy constraints), iteratively enforcing them only as violations are detected. For each constraint we provide an efficient method to detect violations and repair them through much smaller optimization problems.

The IAFD algorithm starts with solving the relaxed problem, with only the convex non-negativity constraint, using the multiplicative updating rules (4.2) and (4.3) of AgileFD [Xue et al.]. This relaxed solution is then slightly refined by

three subroutines which sample and rectify violations of Gibbs' phase rule, the alloying rule, and the connectivity constraint respectively, by solving small scale optimization problems. The refined solution is then relaxed again and improved through the multiplicative updates. This process is reapeated in an interleaving manner which creates a closed loop involving AgileFD and the three refining modules. A reason why this interleaving can be expected to not produce much duplicate effort is due to the following observation:

Proposition 1. *The number of non-zero entries in* H *:* $\|\{(n,k)|\sum_m H_{nkm} > 0\}\|$ *is nonincreasing under updates* (4.2) *of AgileFD.*

This comes from the fact that every component is updated through multiplication with itself in (4.2), which ensures that zero-components stay zero. Thus, if Gibbs' phase rule is satisfied before the multiplicative updates, it will still be satisfied after. We now describe the subroutines handling the constraints.

Gibbs' phase rule refinement After obtaining the matrix W and H, we find violations of Gibbs' phase rule by scanning sample points and noting which ones have more than three phases present. One key insight is that the problem of enforcing Gibbs' phase rule decouples between sample points once the matrix W is fixed. In order to represent the constraint that no more than three phases are present, we introduce a binary variable δ_{kn} denoting whether phase k is present at sample location n (i.e., $\sum_m H_{kmn}$ is nonzero). The constraint is now enforced by solving the following mixed integer program with W fixed for each violated

sample point, which results in a very light-weight refinement:

$$\min_{\delta, H_{kmn} \forall k,m} \sum_{q} |A_{qn} - \sum_{mk} W_{q-m,k} H_{kmn}|,$$
$$\text{s.t. } \forall k, m \; H_{kmn} \leqslant M\delta_{nk}, \; \sum_{k} \delta_{nk} \leqslant 3. \tag{4.5}$$

Here, $H_{kmn} \leqslant M\delta_{nk}$ is a big-M constraint, which enforces that phase k is zero if δ_{nk} is zero. We use $\sum_k \delta_{nk} \leqslant 3$ to enforce that only three phases are allowed. These compact programs typically contains two orders of magnitude fewer variables then the complete program, and can be quickly solved in parallel.

Alloying rule refinement Violations of the alloying rule can be found by comparing the shift parameter λ_k of some sample point n, here interpreted as $\sum_m H_{kmn} m / \sum_m H_{kmn}$, to that of its neighbors in graph G. This simply amounts to a linear scan through all sample points. It is again possible to decouple the constraint by taking W and n as fixed, which allows for a compact mixed integer program formulation. We fix the violating sample point n, denote the set of its neighbors as $N(n)$, and then calculate $\lambda_{kn'} = \sum_m m H_{kmn'} / \sum_m H_{kmn'}$ for all neighbors $n' \in N(n)$ where phase k is present. In the MIP the binary variable δ_{kn} is used to denote whether phase k is present at sample point n, another binary variable τ_n is then introduced to denote whether the sample point undergoes shift. By using a large M-constraint as in Gibbs' phase rule module we can encode that unless the sample point is shifting or doesn't contain the phase k, the λ_k has to be close

to that of it's neighbors as follows:

$$\begin{aligned}
\min_{\tau,\delta,H_{kmn}\forall k,m} \quad & \sum_{q} |A_{qn} - \sum_{mk} W_{q-m,k} H_{kmn}|, \\
\text{s.t.} \quad & |\sum_{m} H_{kmn} m - \lambda_{kn'} \sum_{m} H_{kmn}| \leqslant \epsilon \sum_{m} H_{kmn} + M\tau_n + M(1-\delta_{kn}), \\
& \forall k, m, n' \in N(n), \qquad H_{kmn} \leqslant M\delta_{nk}, \qquad \sum_{k} \delta_{nk} + \tau_n \leqslant 3. \qquad (4.6)
\end{aligned}$$

Connectivity refinement While explicitly encoding the constraint is computationally expensive, finding violations can be done in a lightweight manner. For each phase k we find all continuous regions containing phase k by simply finding the connected components of our graph G where phase k is present. To rectify the constraint, every connected component C is then weighted by the total amount of present phase, which amounts to calculating the quantity $\sum_{n\in C,m} H_{kmn}$. This weight corresponds to the amount of present signal. We then zero out components in H corresponding to phase k and sample points in the least weighted connected components. This procedure ensures that all the phases correspond to a single contiguous regions, without deteriorating much (if at all) the objective function in general.

4.4 Experimental Results

IAFD is evaluated on several real world instances of the phase mapping problem, available at Le Bras et al. [2014]. We randomly initialize the matrices, and as the interleaving with the connectivity-subroutine and the alloying-subroutine

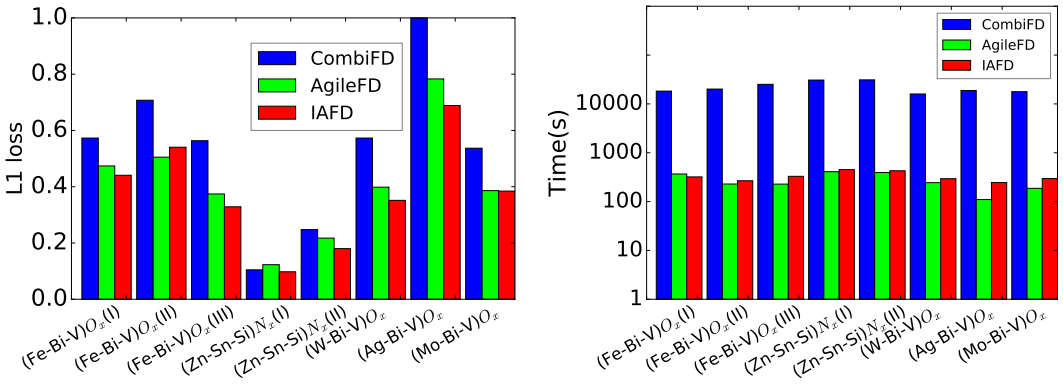

Figure 4.2: (Left) Normalized L1 Loss of the difference between ground-truth and reconstructed X-ray patterns for the algorithms on 8 real world systems. IAFD performs best, with combiFD lagging behind the two other methods. (Right) Runtime for CombiFD, AgileFD, IAFD to solve 8 real systems, note the logarithmic time scale. We can clearly see that the heavy duty MIP formulation of combiFD results in run times of hours, while the two lightweight methods run in a matter of minutes.

assumes structured data, the whole algorithm starts with several rounds of AgileFD interleaving with Gibbs' rule followed by the other two subroutines. The diffraction patterns are probed at around 200 locations of the respective wafers with approximately 1700 values of q sampled, we set $K = 6$ and $M = 8$ which gives us around two million variables per problem. More rounds of interleaving lead to better results but of course it takes more time. We chose to do three rounds of AgileFD interleaving with Gibbs' rule followed by enforcing the other two constraints to balance these tradeoffs. Our method is compared against CombiFD [Ermon et al., 2014], with a mipgap of 0.1 and 15 iterations. Due to its poor scaling properties only the Gibbs' phase rule is enforced for CombiFD. We also compare IAFD against AgileFD [Xue et al.], with termination constant set to 10^{-5}.

The most important metric when comparing different methods is the solution

	Alloying Constraint			Connectivity Constraint		
system	CombiFD	AgileFD	IAFD	CombiFD	AgileFD	IAFD
(Fe-Bi-V)O_x(I)	0.57	0.15	**0.00**	**1.00**	1.65	**1.00**
(Fe-Bi-V)O_x(II)	0.55	0.30	**0.00**	2.40	1.65	**1.00**
(Fe-Bi-V)O_x(III)	0.18	0.03	**0.00**	2.50	2.18	**1.00**
(Zn-Sn-Si)N_x(I)	0.06	0.01	**0.00**	**1.00**	2.38	**1.00**
(Zn-Sn-Si)N_x(II)	0.05	0.02	**0.00**	2.00	1.38	**1.00**
(W-Bi-V)O_x	0.54	0.08	**0.00**	1.67	2.31	**1.00**
(Ag-Bi-V)O_x	0.84	0.16	**0.00**	3.60	1.96	**1.00**
(Mo-Bi-V)O_x	0.46	0.08	**0.00**	1.60	1.72	**1.00**

Table 4.1: To the left we see the fraction of sample points violating the alloying rule for different algorithms, where IAFD consistently has no violations. The right side gives the average number of connected components per phase, and here only IAFD always contain a single continuous region as required by the connectivity constraint.

quality, measured by L1 loss. Results shown in Figure 4.2 (Left). It is evident that CombiFD in Ermon et al. [2014] has subpar performance, while IAFD wins by a slight margin over AgileFD. This suggests that enforcing the constraints actually improves the reconstruction error. The area where we expect IAFD to perform the best is in terms of enforcing the physical constraints, which is illustrated in Table 4.1. Here IAFD consistently performs the best with zero violations, which results in physically meaningful solutions to the phase mapping problem.

The smaller subroutines are evidently able to handle all constraints and additionally provide a low loss, which might lead one to suspect that IAFD has long run times. That is not the case. The run times can be viewed in figure 4.2 (Right). While AgileFD is slightly faster than IAFD, the difference is very small. CombiFD, which explicitly enforces the constraints [Ermon et al., 2014], has prohibitive long run times in practice, which suggests that a complete MIP encoding is both inefficient and unnecessary. These results show that IAFD can enforce all physical rules, without sacrificing much in either reconstruction error or running time.

4.5 Discussion

We propose a novel Interleaved Agile Factor Decomposition (IAFD) framework for solving the phase mapping problem, a challenging constrained matrix factorization problem in materials discovery. IAFD is a lightweight iterative approach that lazily enforces non-convex constraints. The algorithm is evaluated on several real world instances and outperforms previous solvers both in terms of run time and solution quality. IAFD's approach, based on efficient multiplicative updates from unconstrained nonnegative matrix factorization and lazily enforced constraints, performs much better compared to approaches that enforce all constraints upfront, using a large mathematical program. This approach opens up a new angle for efficiently solving more general constrained factorization problems. We anticipate deploying IAFD at the Stanford Synchrotron Radiation Lightsource in the near future to the benefit of the materials science community.

CHAPTER 5

AN AVERAGE-CASE MODEL OF NMF

5.1 Introduction

Non-negative matrix factorization (NMF) is a ubiquitous technique for data analysis, where one attempts to factorize a measurement matrix $\mathbf{X}$ into the product of non-negative matrices $\mathbf{U}, \mathbf{V}$ [Lee and Seung, 1999]. This simple problem has applications in recommender systems [Luo et al., 2014], scientific analysis [Berne et al., 2007, Trindade et al., 2017], computer vision [Gillis, 2012], internet distance prediction [Mao et al., 2006], audio processing [Schmidt et al., 2007], and many more domains. The non-negativity is often crucial for interpretability; in the context of crystallography for example, the light sources—represented as matrix factors—have non-negative intensity [Suram et al., 2016b].

Like many other non-convex optimization problems, e.g. optimizing neural networks [Blum and Rivest, 1989], finding the exact solution to NMF is NP-hard [Pardalos and Vavasis, 1991, Vavasis, 2009b]. NMF's tremendous practical success is at odds with such worst-case analysis, and simple algorithms based on gradient descent are known to find good solutions in real-world settings [Lee and Seung, 2001]. At the time when NMF was proposed, most analyses of optimization problems in machine learning focused on convex formulations such as SVMs [Cortes and Vapnik, 1995]. However, owing to the success of neural networks, non-convex optimization has experienced a resurgence in interest. While non-convex problems that can be optimized efficiently via saddle point characterization have been studied extensively [Ge et al., 2015], NMF has seen less theoretical progress. While the NMF problem is NP-hard, its empirical

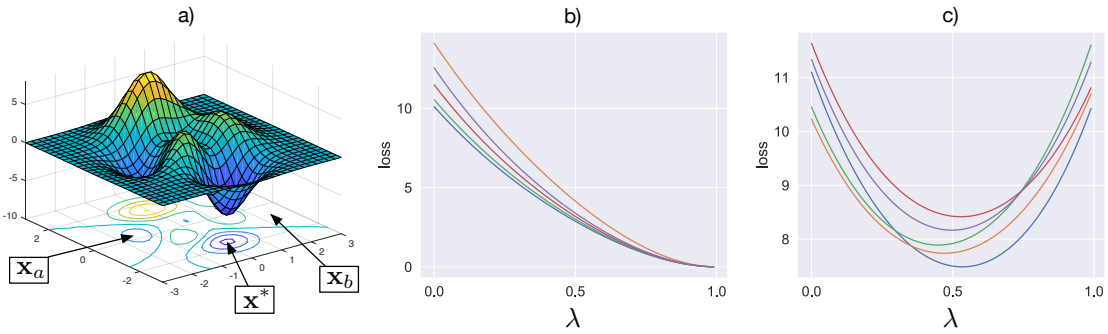

Figure 5.1: A non-convex loss surface is illustrated in a). In general, the loss will be non-convex on straight paths connecting random points $\mathbf{x}_a, \mathbf{x}_b$ and the global minimizer $\mathbf{x}^*$. We consider a model of NMF with a randomized planted solution; as shown in b), the loss is typically convex on straight paths between points $\mathbf{x}_a$ and a planted solution $\mathbf{x}^*$. Additionally, as illustrated in c), the loss is typically convex on straight paths between points $\mathbf{x}_a$ and $\mathbf{x}_b$.

experience and widespread usage suggests that the problem might be tractable in the average case, albeit not in the worst case.

In this paper, we prove theoretically and empirically that a benign convexity property called *star-convexity* typically holds in NMF. From a theoretical perspective, we consider NMF instances with planted randomized solutions, inspired by the stochastic block model for social networks [Holland et al., 1983, Decelle et al., 2011] and the planted clique problem studied in sum-of-squares literature [Barak et al., 2016]. We prove that between two random points, the loss is convex with high probability, and conclude that the loss surface is star-convex in the typical case—even if the loss is computed over unobserved data. From an empirical perspective, we verify that our theoretical results hold for an extensive collection of real-world datasets spanning collaborative filtering [Zhou et al., 2008, Kula, 2017, Harper and Konstan, 2016], signal decomposition [Zhu, 2016, Li and Ngom, 2013, Li et al., 2001, Erichson et al., 2018], and audio processing [Flenner and

Hunter, 2017]. Finally, we show that star-convex behavior becomes more likely with a growing number of parameters, suggesting that a similar result may hold in neural networks as they become wider. We provide supporting empirical evidence for this hypothesis on modern network architectures. We summarize the contributions of this paper as follows:

- We prove that the NMF loss surface has benign convexity properties in the average case, which might explain why NMF typically performs well despite being NP-hard in the worst case.
- We verify that our theoretical predictions hold in an extensive suite of real-world datasets.
- Based on our theoretical results, we hypothesize that increasing width in neural networks should improve convexity. We also provide supporting experimental evidence.

5.2 NMF and Star-Convexity

NMF aims to decompose some large measurement matrix $\mathbf{X} \in \mathcal{R}^{n \times m}$ into two *non-negative* matrices $\mathbf{U} \in \mathcal{R}_+^{n \times r}$ and $\mathbf{V} \in \mathcal{R}_+^{r \times m}$ such that $\mathbf{X} \approx \mathbf{UV}$. The canonical formulation of NMF is

$$\min_{\mathbf{U},\mathbf{V} \geqslant 0} \quad \ell(\mathbf{U}, \mathbf{V}), \text{ where } \ell(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\|\mathbf{UV} - \mathbf{X}\|_F^2. \tag{5.1}$$

Practitioners commonly use NMF in recommender systems, where an entry (i, j) of $\mathbf{X}$, for example, corresponds to the rating user i gave to movie j [Luo et al., 2014]. In such settings, data might be missing if all users did not rate all movies.

In those cases, it is common to only consider the loss over observed data [Zhang et al., 2006, Candès and Recht, 2009]. We let $\hat{1}_{(i,j)}$ be an indicator variable that is 1 if entry (i, j) is "observed" and 0 otherwise. The loss function is then

$$\min_{\mathbf{U},\mathbf{V}\geqslant 0} \quad \ell(\mathbf{U},\mathbf{V}) = \frac{1}{2}\sum_{i,j} \hat{1}_{(i,j)} \left([\mathbf{UV}]_{ij} - \mathbf{X}_{ij}\right)^2. \tag{5.2}$$

NMF's non-negative constraints prevent practitioners from applying spectral strategies, which can be otherwise used in, e.g., PCA. This restriction results in NMF's NP-hardness [Vavasis, 2009b]. Even so, previous work on the computational complexity of NMF has shown that the problem is tractable for small constant dimensions r via algebraic methods [Arora et al., 2012]. However, practitioners use simple variants of gradient descent, which are known to work reliably, rather than these algorithms [Koren et al., 2009, Lee and Seung, 2001]. This gap between theoretical hardness and practical performance is also found in deep learning. Optimizing neural networks is generally NP-hard [Blum and Rivest, 1989], but in practice, they can be optimized with simple stochastic gradient descent algorithms to outperform humans in tasks such as verifying faces [Lu and Tang, 2015] and playing Atari-games [Mnih et al., 2015]. Recent work on understanding the geometry of neural network loss surfaces has promoted the idea of convexity properties. Izmailov et al. [2018] show that the network's loss surface is convex around the local optimum, while Zhou et al. [2019] and Kleinberg et al. [2018] empirically show that the gradients during optimization typically point towards the local minima to which the network eventually converges. Of central importance in this line of work is **star-convexity**, which is a property of a function f that guarantees that f is convex along straight paths towards its optima x^*. See Figure 5.2 for an example. Formally, it is defined as

Definition 2. *A function $f : \mathcal{R}^n \to \mathcal{R}$ is **star-convex** towards $\boldsymbol{x}^*$ if for all $\lambda \in [0, 1]$ and $\boldsymbol{x} \in \mathcal{R}^n$, we have $f(\lambda \boldsymbol{x} + (1 - \lambda)\, \boldsymbol{x}^*) \leqslant \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{x}^*)$.*

Star-convex functions can be optimized in polynomial time [Lee and Valiant, 2016]. Moreover, the function only needs to be star-convex under a natural noise model [Kleinberg et al., 2018]. Since NMF is NP-hard, it is not star-convex in general; however, it is natural to conjecture that NMF is star-convex in the *typical* case. Such a property could explain the practical success of NMF on real-world datasets, which are far from worst-case. This is the working hypothesis of this paper, where the *typical* case is formalized probabilistically in Theorem 2. Indeed, one can verify numerically that NMF is typically star-convex for natural distributions and realistically sized matrices: see Figure 5.1 where we consider a rank 10 decomposition of $(100, 100)$-matrices with iid half-normal entries and a planted solution, sampled as per Assumption 2 stated in the next section. We dedicate the following sections to prove that NMF is star-convex with high probability in a planted model, and to confirm that this phenomenon generalizes to datasets from the real world, which are far from worst-case.

5.3 Proving Typical-Case Star-Convexity

Our aim now is to prove that the NMF loss-function is star-convex in the typical case for natural non-worst-case distributions of NMF instances. We consider a slightly weaker notation of star-convexity, where $f(\lambda \mathbf{x} + (1 - \lambda)\, \mathbf{x}^*) \leqslant \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{x}^*)$ holds not for all $\mathbf{x}$, but for random $\mathbf{x}$ with high probability. This is in fact the best achievable—an adversarial example of an NMF instance that isn't star-convex is simply $u_1 = 1, u^* = 0$ and $v_1 = 0, v^* = 1$. Our results show that

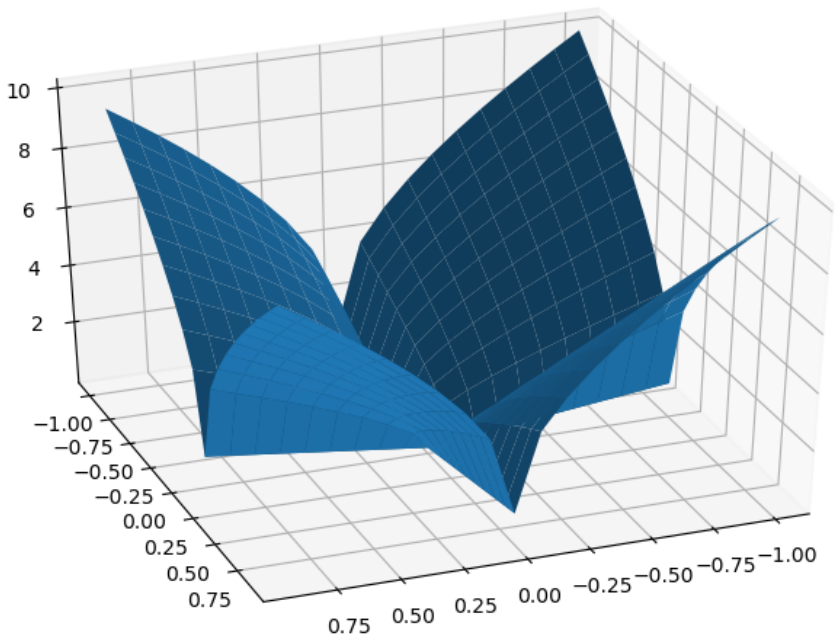

Figure 5.2: The function $(|x|^p + |y|^p)^{1/p}$ is an example of a star-convex function for $0 < p < 1$. It is non-convex in general, but convex towards $(0, 0)$.

NMF is convex on straight lines with high probability as the dimensionality of the problem increases, suggesting that the measure of such adversarial instances is small.

Inspired by the stochastic block model of social networks [Holland et al., 1983, Decelle et al., 2011] and the planted clique problem [Barak et al., 2016], we focus on a setting with a planted random solution. In the following section, we verify that the conclusions drawn from this model transfer to real-world datasets.

We assume that there is a planted optimal solution $(\mathbf{U}^*, \mathbf{V}^*)$ such that $\mathbf{X} = \mathbf{U}^*\mathbf{V}^*$, where entries of these matrices are sampled iid. This assumption follows from existing research on random input in neural networks [Li and Yuan, 2017]. Furthermore, we require matrices to be sampled from a class of distributions with good concentration properties, e.g., the half-normal distribution and bounded

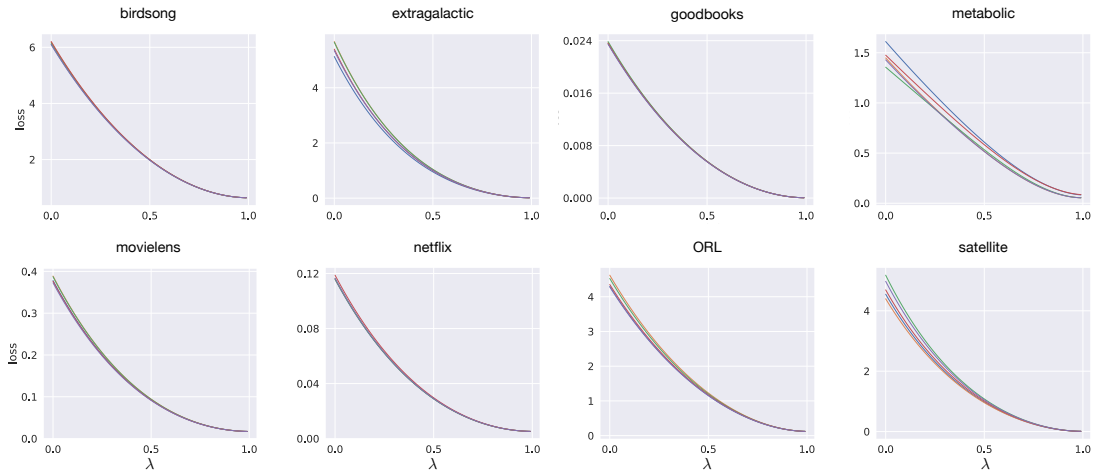

Figure 5.3: The NMF loss surface along the straight line from a random point w_0 to a local optima w^* found via gradient descent (from independent starting points). We overlap five independent lines; zoom in for detail. As our theoretical results predict, the loss surface is convex on these straight lines for all real-world datasets.

distributions. As is standard in random matrix theory [Vershynin, 2010], we develop non-asymptotic results, which hold with a probability that grows as the matrices of shapes (n, r) and (r, m) increase in size. Consequently, we specify how r and m depend on n.

Assumption 2. *For* $(\mathbf{U}, \mathbf{V}) \in R^{n \times r} \times R^{r \times m}$*, we assume that the entries of the matrices* $\mathbf{U}, \mathbf{V}$ *are sampled iid from a continuous distribution with non-negative support that either* (i) *is bounded or* (ii) *can be expressed as a* 1*-Lipschitz function of a Gaussian distribution. As* $n \to \infty$*, we assume that* r *grows as* n^γ *up to a constant factor for* $\gamma \in [1/2, 1]$*, and* m *grows as* n *up to a constant factor.*

We are now ready to state our main result: the loss function in (5.1) is convex on straight lines between points sampled as per Assumption 2, where one point can be the planted solution, with high probability. Thus, the loss satisfies our

slightly weaker notion of star-convexity, and is convex on "most" straight lines. The probability increases as the size of the problem increases, suggesting a surprising benefit of high dimensionality. We also show similar results for the loss function in (5.2) with unobserved data, under the assumption that the event of observing an entry occurs independently with constant probability p. We provide a proof sketch here.

Theorem 2. ***(Main)*** *Let matrices $\mathbf{U}_1, \mathbf{V}_1, \mathbf{U}_2, \mathbf{V}_2, \mathbf{U}^*, \mathbf{V}^*$ be sampled according to Assumption 2. Then there exists positive constants c_1, c_2, such that with probability $\geqslant 1 - c_1 \exp(-c_2 n^{1/3})$, the loss function $\ell(\mathbf{U}, \mathbf{V})$ in* (5.1) *is convex on the straight line $(\mathbf{U}_1, \mathbf{V}_1) \rightarrow (\mathbf{U}_2, \mathbf{V}_2)$. The same holds along the line $(\mathbf{U}_1, \mathbf{V}_1) \rightarrow (\mathbf{U}^*, \mathbf{V}^*)$. It also holds if any entry (i, j) is observed independently with constant probability p, but with probability $\geqslant 1 - c_1 \exp(-c_2 r^{1/3})$.*

Proof Strategy Let us parameterize the NMF solution along the line $(\mathbf{U}_2, \mathbf{V}_2) \rightarrow (\mathbf{U}_1, \mathbf{V}_1)$ as

$$\hat{\mathbf{X}}(\lambda) = [\lambda \mathbf{U}_1 + (1-\lambda)\mathbf{U}_2]\,[\lambda \mathbf{V}_1 + (1-\lambda)\mathbf{V}_2]\,.$$

For proving Theorem 2, it suffices to show that the loss function $\ell(\lambda) = \frac{1}{2}\|\hat{\mathbf{X}}(\lambda) - \mathbf{X}\|_F^2$ is convex in λ with high probability on $[0, 1]$. Our strategy is to employ a sum-of-squares lower bound on the second derivative, and then use concentration of measure from random matrix theory. For fixed matrices $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}^*, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}^*$, the function $\ell(\lambda)$ is a fourth-degree polynomial in λ, so its second derivative w.r.t. λ is a second-degree polynomial in λ. For a general second-degree polynomial $p(x) = ax^2 + bx + c$, we have $p(x) = \frac{1}{a}\left[\left(ax + \frac{b}{2}\right)^2 + \left(ac - \frac{b^2}{4}\right)\right]$. If $a > 0$, as is the case here, proving that $p(x)$ is positive for all x can be done by showing $ac \geqslant \frac{b^2}{4}$.

$\ell''(\lambda) > 0$ would imply that $\ell(\lambda)$ is convex for all λ. Thus, we need to show that

$$2\left\|\mathbf{W}_2\right\|_F^2\left(\left\|\mathbf{W}_1\right\|_F^2+2\langle\mathbf{W}_0,\mathbf{W}_2\rangle\right)\geqslant 3\left(\langle\mathbf{W}_1,\mathbf{W}_2\rangle\right)^2 \tag{5.3}$$

where the matrices $\mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2$ are given as $\mathbf{W}_0 = \mathbf{U}_2\mathbf{V}_2 - \mathbf{U}^*\mathbf{V}^*$, $\mathbf{W}_1 = (\mathbf{U}_1 - \mathbf{U}_2)\,\mathbf{V}_2 + \mathbf{U}_2\,(\mathbf{V}_1 - \mathbf{V}_2)$, $\mathbf{W}_2 = (\mathbf{U}_1 - \mathbf{U}_2)\,(\mathbf{V}_1 - \mathbf{V}_2)$. With slight abuse of notation, we have used $\langle\mathbf{A},\mathbf{B}\rangle$ to denote $\mathrm{Tr}(\mathbf{A}\mathbf{B}^T)$ for matrices $\mathbf{A}, \mathbf{B}$ of the same shape. By replacing terms in (5.3) with their means, we get

$$\begin{aligned}2(4rmn\sigma^4)\left(6rmn\sigma^4+4rmn\mu^2\sigma^2+2rmn\sigma^4\right)\\ \geqslant 3\left(-4rmn\sigma^4\right)^2.\end{aligned} \tag{5.4}$$

Here, σ^2 is the variance of the distribution of the entries in the matrices, and μ is the mean. By just counting terms of order $(rmn\sigma^4)^2$, we see that the LHS has 64 such terms while the RHS has only 48. Thus, if all matrices $\mathbf{W}_0, \mathbf{W}_1$ and $\mathbf{W}_2$ would exactly be equal to their means, the inequality in (5.3) would hold. In proving that it holds in general, we use concentration of measure results from random matrix theory to show that the terms are concentrated around their means and that large deviations are exponentially unlikely.

Concentration of Measure Consider the matrix $\mathbf{W}_2 = (\mathbf{U}_1 - \mathbf{U}_2)\,(\mathbf{V}_1 - \mathbf{V}_2)$. Given that all matrices are iid, we can center the variables so that $\mathbf{W}_2 = (\mathbf{U}_1 - \mathbf{U}_2)\,(\mathbf{V}_1 - \mathbf{V}_2) = \left(\bar{\mathbf{U}}_1 - \bar{\mathbf{U}}_2\right)\left(\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2\right)$, where the bar denotes the centered matrices. The term $\left\|\mathbf{W}_2\right\|_F^2$ can then be written as $\mathrm{Tr}\left[\left(\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2\right)^T\left(\bar{\mathbf{U}}_1 - \bar{\mathbf{U}}_2\right)^T\left(\bar{\mathbf{U}}_1 - \bar{\mathbf{U}}_2\right)\left(\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2\right)\right]$. Given that all matrix entries are independent as per Assumption 2, we would expect some concentration of measure to hold. Although Bernstein-type inequalities turn out to be too weak for our

purposes, the field of random matrix theory offers stronger results for matrices with independent sub-Gaussian entries [Ahlswede and Winter, 2002, Tropp, 2012, Meckes and Szarek, 2012]. Using concentration of measure for traces of random matrices, we achieve the following inequality.

$$P\left(\left|\|\mathbf{W}_2\|_F^2 - \mathbb{E}\left[\|\mathbf{W}_2\|_F^2\right]\right| > trn^2\right) \leqslant c_3 \exp\left(-c_4 \min(t^2, t^{1/2})n\right) \tag{5.5}$$

where c_3, c_4 are positive constants. In expressions for some terms in (5.3), however, we are not able to center all variables. For such expressions, we get similar but slightly weaker concentration results, where the exponent in the RHS of (5.5) scales as $n^{1/3}$ instead of n.

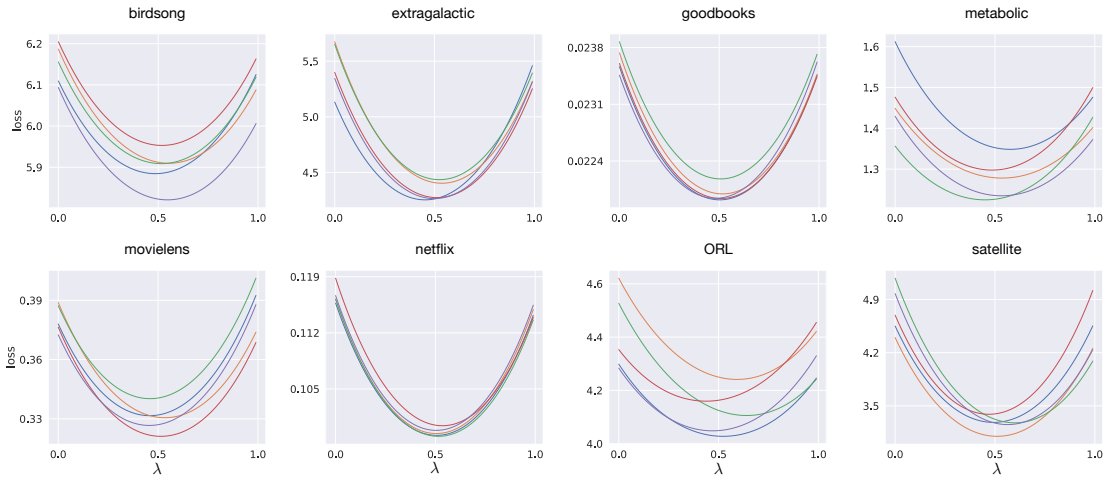

Figure 5.4: We here illustrate the NMF loss surface on straight paths connecting two random points for 8 real-world datasets. We overlap five independent lines for each dataset. Note that the curves are always convex, suggesting that the loss surface is "typically" convex as our theoretical results suggest.

Proof Sketch Given that $\mathbb{E}\left[\|\mathbf{W}_2\|_F^2\right] = 4rmn\sigma^4$, (5.5) says that the probability of $\|\mathbf{W}_2\|_F^2$ deviating from its mean by a relative factor ϵ is less than $c_3 \exp\left(-c_5\epsilon^2 n\right)$ for some small ϵ. By applying similar arguments to terms $\langle \mathbf{W}_0, \mathbf{W}_2 \rangle$ and $\langle \mathbf{W}_1, \mathbf{W}_2 \rangle$, we show that the probability of them deviating by a relative factor ϵ is less than $c_6 \exp\left(-c_7\epsilon^2 n^{1/3}\right)$. $\|\mathbf{W}_1\|_F^2$ is a problematic term, containing a term of type $\mathrm{Tr}\left(\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2\right)^T \mu_1^T \mu_1 \left(\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2\right)$, which has weak concentration properties. Even so, since matrices of type $\mathbf{A}^T\mathbf{A}$ are p.s.d. due to non-negative traces, this term is non-negative. Moreover, we can simply omit $\|\mathbf{W}_1\|_F^2$ to lower bound the convexity because the term appears on the LHS of (5.3). Using union bound, we bound the probability of at least one term deviating with a relative factor ϵ by $c_1 \exp\left(-c_8\epsilon^2 n^{1/3}\right)$ for positive constants c_1, c_8. Now, we set $\epsilon = 0.01$. If no term deviates by a factor of more than 0.01, then (5.4) still holds as $0.99^2 \cdot 64 \geqslant 1.01^2 \cdot 48$. Thus, the inequality is violated with probability at most $c_1 \exp\left(-c_2 n^{1/3}\right)$ for positive c_1, c_2. ■

Proof Sketch for Unobserved Data If the entries in (5.2) are "observed" independently with probability p, for fixed matrices $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}^*, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}^*$ such that Theorem 2 holds, we have

$$\begin{aligned}
\mathbb{E}\left[\ell''(\lambda)\right] &= \mathbb{E}\left[\sum_{ij} \hat{1}_{(i,j)} \left(\hat{\mathbf{X}'}_{ij}^2 + \hat{\mathbf{X}''}_{ij}(\hat{\mathbf{X}}_{ij} - \mathbf{X}_{ij})\right)\right] \\
&= p \sum_{ij} \left(\hat{\mathbf{X}'}_{ij}^2 + \hat{\mathbf{X}''}_{ij}(\hat{\mathbf{X}}_{ij} - \mathbf{X}_{ij})\right) \\
&\geqslant 0.
\end{aligned}$$

Thus, the expectation of $\ell(\lambda)$ is convex. To show that it is convex with high probability, we first observes that with high probability, no entry (i, j) in $\ell''(\lambda)$

is particularly large. Assuming this holds via union bound, for fixed matrices $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}^*, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}^*$ with elements that are "observed" independently with probability p, we get that $\ell''(\lambda)$ is concentrated around its convex mean via Hoeffding bounds. ■

name	description	shape (n, m, r)	sparsity	reference
birdsong	bird calls	$(5120, 1246, 88)$		[Flenner and Hunter, 2017]
extragalactic	extragalactic spectra	$(2760, 2820, 10)$		[Zhu, 2016]
goodbooks	book ratings	$(10000, 43461, 50)$	0.0022	[Kula, 2017]
metabolic	yeast activity	$(9335, 36, 3)$		[Li and Ngom, 2013]
movielens	movie ratings	$(3953, 6041, 20)$	0.0419	[Harper and Konstan, 2016]
netflix	movie ratings	$(47928, 8963, 20)$	0.0121	[Zhou et al., 2008]
ORL faces	facial images	$(400, 10304, 49)$		[Li et al., 2001].
satellite	satellite images	$(162, 94249, 4)$		[Erichson et al., 2018].

Table 5.1: Dataset details. References contain suggested rank r and previous usage.

5.4 Experiments

5.4.1 Verifying Theoretical Predictions

To verify that the conclusions from our theoretical model hold more broadly, we now empirically study real-world datasets previously used in NMF literature. A few datasets have ranks outside the scope of our theoretical model, but they still display star-convexity properties, indicating that star-convexity might be a more general phenomenon. We focus on a handful of representative datasets spanning image analysis, scientific applications, and collaborative filtering. In Table 5.3, we list these datasets together with their sparsity. We use decomposition ranks as per the values previously reported in the literature. We perform a non-negative matrix factorization via gradient descent, starting with randomly initialized data. To enable comparison between datasets, we scale all data matrices so

that the variance of observed entries is one, and divide the loss function by the number of (observed) entries. We initialize decomposition matrices using the half-normal distribution, which is scaled so that the mean matches with that of the dataset. For simplicity, we use the same learning rate of $1e{-}5$ for all datasets and run gradient descent until the rate of relative improvement in the loss falls below $1e{-}7$. This procedure gives good convergence for all datasets. As is standard in NMF, we compute the loss only over observed entries for the collaborative filtering datasets with unobserved ratings (movielens, netflix, and goodbooks) [Zhang et al., 2006]. In Figure 5.3, we plot the loss function from an initialization point to an independent local optima. In Figure 5.4, we plot the loss function between two random points drawn from the initialization distribution—observe that the loss is convex. These results agree with our theoretical model, and we conclude that many real-world matrices can be decomposed as a low-rank matrix $\mathbf{U}^*\mathbf{V}^*$ with the convexity properties our theoretical results suggest, plus a "noise term" that must have a small norm since the loss $\ell(\mathbf{U}^*, \mathbf{V}^*)$ is small.

5.4.2 Ablation Experiments

Theorem 2 suggests that, as the matrices become larger, NMF is increasingly likely to be star-convex. To test if this is the case for our real-world datasets, we perform ablation experiments by varying the dimensions of the matrices. We decrease the number of data points n by subsampling rows and columns uniformly randomly. Our measure of curvature at a point $\mathbf{x}$, given some optimal solution $\mathbf{x}^*$, is

$$\alpha(\mathbf{x}) = \min_{\lambda \in [0,1]} \ell''\left(\lambda \mathbf{x} + (1-\lambda)\mathbf{x}^*\right). \tag{5.6}$$

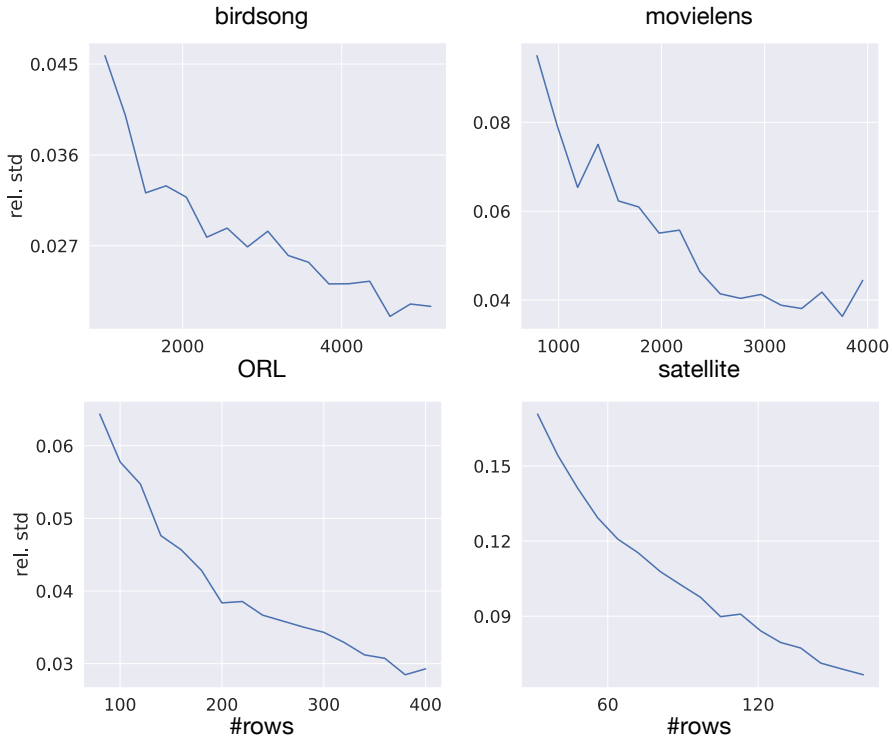

Figure 5.5: We illustrate how the relative deviation $\frac{\sigma}{\mu}$ of the curvature in (5.6) depends on the dataset's size. We normalize by μ to avoid uniform scaling. For all datasets, the relative deviations decrease with more samples, suggesting that the (positive) curvature becomes increasingly concentrated around its mean for larger matrices.

Note that $\alpha \geqslant 0$ implies star-convexity. In practice, we obtain $\mathbf{x}^*$ from gradient descent; finding the absolute minima remains a challenge. For each dataset and subsample rate, we find 50 optima and evaluate the curvature from 50 random points, thus obtaining 2500 samples of α. Figure 5.5 shows how the relative deviation $\frac{\sigma}{\mu}$ of α decreases as the dataset becomes larger. Figure 5.6 that shows the fraction of non-negative curvature as a function of input dimensionality—we confirm that the sampled curvatures typically are positive. This can also be considered as a quantitative depiction of Figure 5.3. Figures 5.5 and 5.6 together show that the curvature becomes increasingly concentrated around its positive

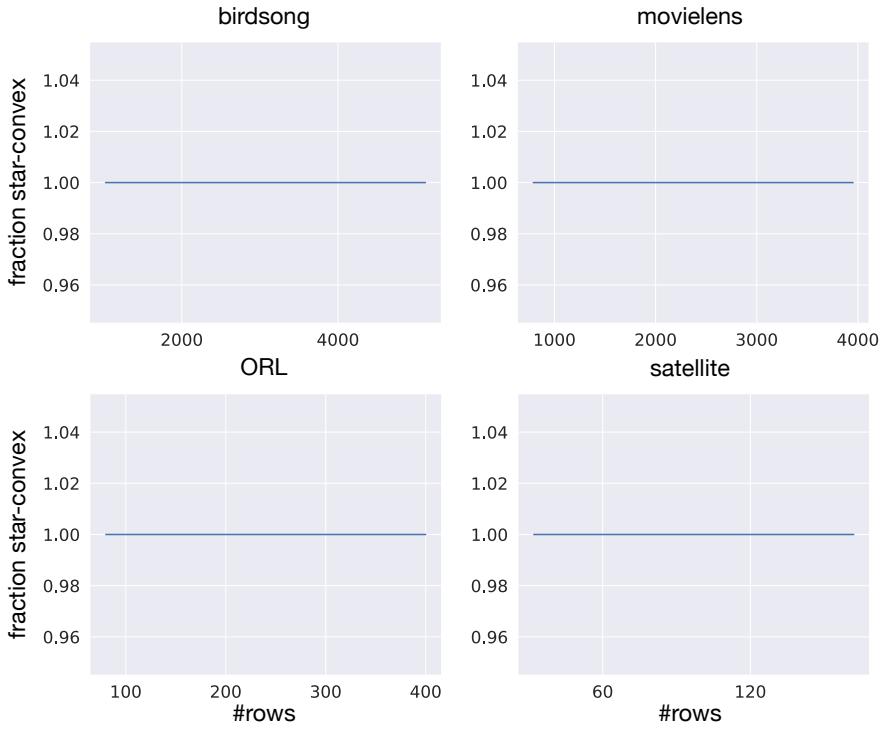

Figure 5.6: We here show the fraction of sampled curvatures (as in (5.6)) that are positive as the dimensionality of the dataset is varied. Note that it is always 1, implying that we have star-convexity even for smaller problems, even though the curvature typically fluctuates more for such problems as per Figure 5.5.

mean for larger matrices, suggesting that the star-convexity phenomenon is valid beyond our simplistic theoretical model.

5.4.3 Implications for Neural Networks

We have seen how increasing the number of parameters makes NMF problems more likely to be star-convex, while also making the curvature tend towards its positive mean, as displayed in Figure 5.5. Theorem 2 suggests that this is a result of concentration of measure, and it is natural to believe that a similar

phenomenon would occur in the context of neural networks. It has previously been observed how neural networks are locally convex [Izmailov et al., 2018], and also how overparameterization is important in deep learning [Arora et al., 2018, Frankle and Carbin, 2018]. Based on our observations in NMF, we hypothesize that a major benefit of overparameterization is in making the loss surface more convex via concentration of measure w.r.t. the weights.

To verify this hypothesis, we consider image classification on CIFAR10 with Resnet networks [He et al., 2016a] trained with standard parameters. Networks are typically only locally convex, a property we quantify as the length of subsets of random "lines" in parameter space along which the training loss function is convex. Formally, we sample a random direction $\mathbf{r}$ and then consider an interval of length one along this direction, centered around the current parameters $\mathbf{w}$, i.e., $\mathbf{w} + \lambda\mathbf{r}$ for $\lambda \in [-1/2, 1/2]$. We then define the "convexity length scale" as the

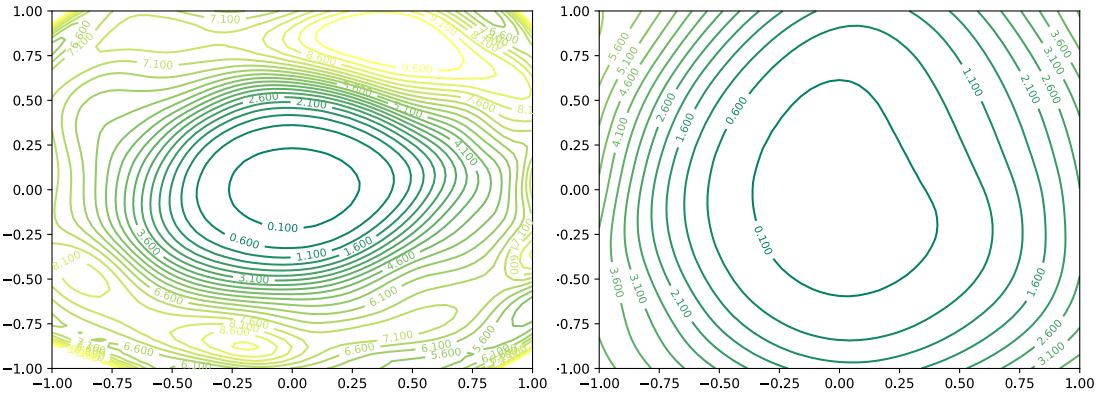

Figure 5.7: The loss landscape of a 110-layer Resnet architecture at epoch 200 along two random directions, visualized as in Li et al. [2018]. The network in the bottom image is four times as wide (i.e. has four times as many channels per layer), and its loss landscape is increasingly convex. In Table 5.2, we generalize this idea and show that the length scale of local convexity increases with network width.

	32-layers			44-layers		
epoch	k=1	k=2	k=4	k=1	k=2	k=4
0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0
100	0.77 $\pm$ 0.035	0.8 $\pm$ 0.031	**0.84** $\pm$ 0.026	0.72 $\pm$ 0.041	0.79 $\pm$ 0.037	**0.83** $\pm$ 0.028
200	0.61 $\pm$ 0.036	0.68 $\pm$ 0.036	**0.8** $\pm$ 0.031	0.66 $\pm$ 0.033	0.68 $\pm$ 0.034	**0.76** $\pm$ 0.033
300	0.55 $\pm$ 0.037	0.68 $\pm$ 0.037	**0.82** $\pm$ 0.032	0.57 $\pm$ 0.036	0.68 $\pm$ 0.036	**0.78** $\pm$ 0.032
	56-layers			68-layers		
epoch	k=1	k=2	k=4	k=1	k=2	k=4
0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0
100	0.7 $\pm$ 0.036	0.81 $\pm$ 0.03	**0.84** $\pm$ 0.03	0.71 $\pm$ 0.032	0.76 $\pm$ 0.03	**0.87** $\pm$ 0.026
200	0.63 $\pm$ 0.039	0.67 $\pm$ 0.034	**0.8** $\pm$ 0.031	0.6 $\pm$ 0.036	0.71 $\pm$ 0.031	**0.8** $\pm$ 0.029
300	0.57 $\pm$ 0.035	0.66 $\pm$ 0.035	**0.79** $\pm$ 0.033	0.58 $\pm$ 0.036	0.67 $\pm$ 0.033	**0.81** $\pm$ 0.03
	80-layers			110-layers		
epoch	k=1	k=2	k=4	k=1	k=2	k=4
0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**1.0** $\pm$ 0.0	**0.94** $\pm$ 0.016	**0.94** $\pm$ 0.018	**0.94** $\pm$ 0.016
100	0.72 $\pm$ 0.036	**0.85** $\pm$ 0.03	0.81 $\pm$ 0.027	0.79 $\pm$ 0.032	0.75 $\pm$ 0.04	**0.91** $\pm$ 0.019
200	0.59 $\pm$ 0.036	0.7 $\pm$ 0.038	**0.77** $\pm$ 0.031	0.71 $\pm$ 0.037	0.71 $\pm$ 0.036	**0.82** $\pm$ 0.03
300	0.63 $\pm$ 0.036	0.71 $\pm$ 0.036	**0.79** $\pm$ 0.03	0.63 $\pm$ 0.037	0.68 $\pm$ 0.034	**0.82** $\pm$ 0.033

Table 5.2: Typical length scales of local convexity for Resnet networks with various width (indicated by k), depth, and training. We sample 25 random "lines" of length 1 in parameter space, centered on current parameters, and report mean length of convex subset of such "lines" and the std of this statistic. Increasing width makes the loss surface increasingly locally convex.

length of the maximal sub-interval containing $\mathbf{w}$ on which $\ell(\mathbf{w} + \lambda\mathbf{r})$ is convex. Directions are sampled from Gaussian distributions and then normalized for each network filter f to have the same norm as the weights of f. Table 5.2 shows how this length scale of local convexity varies with depth, width, and training, where width is varied by multiplying the number of channels by k. Indeed, increasing width makes the landscape increasingly locally convex, supporting our hypothesis.

5.5 Related Work

As the success of deep learning has become widespread, many researchers have empirically investigated its behavior on real-world datasets [Li and Yuan, 2017, Zhang et al., 2016]. In the context of sharp vs flat local minima [Keskar et al., 2016], Li et al. [2018] illustrate how increasing the width improved flatness in a Resnet network, an observation that Table 5.2 quantifies. Our work was initially motivated by studies on local and star-convexity in neural networks due to Kleinberg et al. [2018], Izmailov et al. [2018] and Zhou et al. [2019]. Whereas such previous work empirically observes star-convexity and investigates its implications, we prove that this benign property arises simply from concentration of measure, albeit in the simpler NMF case. We intentionally focus on dense NMF problems to explain its practical success, leaving e.g., sparsity for future work [Richard and Montanari, 2014]. A common theme in non-convex optimization more generally is that functions with only saddle points and global minima can be solved via SGD [Ge et al., 2015]. We note that problems with such properties, for example, tensor decomposition, can be efficiently optimized. Our work, on the other hand, addresses an NP-hard optimization problem, utilizing statistical assumptions on the input to achieve positive results. There is extensive work on non-worst-case analyses of algorithms and machine learning models, and on what problem distributions can guarantee tractability [Bilu and Linial, 2012, Ackerman and Ben-David, 2009, Afshani et al., 2017]. On the positive side, Arora et al. [2012] have proposed an exact algorithm for NMF that runs in polynomial time for small constant r, and there are positive results for so-called "separable" NMF [Donoho and Stodden, 2004]. Our work is also related to the analysis of algorithms where instances have "planted" solutions, for instance, the planted

clique problem [Barak et al., 2016] and the stochastic block model [Holland et al., 1983, Decelle et al., 2011].

5.6 Discussion

This paper revisits NMF, a non-convex optimization problem in machine learning. We have shown that NMF is typically star-convex, provably for a natural average-case model and empirically on an extensive set of real-world datasets. Additionally, we have shown how network width improves local convexity of neural networks. Our results support the counter-intuitive observation that optimization might sometimes be *easier* in higher dimensions due to concentration of measure.

CHAPTER 6

A CLOSER LOOK AT BATCH NORMALIZATION

6.1 Introduction

Normalizing the input data of neural networks to zero-mean and constant standard deviation has been known for decades LeCun et al. [1998] to be beneficial to neural network training. With the rise of deep networks, Batch Normalization (BN) naturally extends this idea across the intermediate layers within a deep network Ioffe and Szegedy [2015a], although for speed reasons the normalization is performed across mini-batches and not the entire training set. Nowadays, there is little disagreement in the machine learning community that BN accelerates training, enables higher learning rates, and improves generalization accuracy Ioffe and Szegedy [2015a] and BN has successfully proliferated throughout all areas of deep learning Huang et al. [2017], He et al. [2016b], Silver et al. [2017], Ba et al. [2016]. However, despite its undeniable success, there is still little consensus on why the benefits of BN are so pronounced. In their original publication Ioffe and Szegedy [2015a] Ioffe and Szegedy hypothesize that BN may alleviate "internal covariate shift" – the tendency of the distribution of activations to drift during training, thus affecting the inputs to subsequent layers. However, other explanations such as improved stability of concurrent updates Goodfellow et al. [2016] or conditioning Salimans and Kingma [2016] have also been proposed.

Inspired by recent empirical insights into deep learning Zhang et al. [2016], Keskar et al. [2016], Neyshabur et al. [2017], in this paper we aim to clarify these vague intuitions by placing them on solid experimental footing. We show that the activations and gradients in deep neural networks without BN tend

to be heavy-tailed. In particular, during an early on-set of divergence, a small subset of activations (typically in deep layer) "explode". The typical practice to avoid such divergence is to set the learning rate to be sufficiently small such that no steep gradient direction can lead to divergence. However, small learning rates yield little progress along flat directions of the optimization landscape and may be more prone to convergence to sharp local minima with possibly worse generalization performance Keskar et al. [2016].

BN avoids activation explosion by repeatedly correcting all activations to be zero-mean and of unit standard deviation. With this "safety precaution", it is possible to train networks with large learning rates, as activations cannot grow incrontrollably since their means and variances are normalized. SGD with large learning rates yields faster convergence along the flat directions of the optimization landscape and is less likely to get stuck in sharp minima.

We investigate the interval of viable learning rates for networks with and without BN and conclude that BN is much more forgiving to very large learning rates. Experimentally, we demonstrate that the activations in deep networks without BN grow dramatically with depth if the learning rate is too large. Finally, we investigate the impact of random weight initialization on the gradients in the network and make connections with recent results from random matrix theory that suggest that traditional initialization schemes may not be well suited for networks with many layers — unless BN is used to increase the network's robustness against ill-conditioned weights.

6.1.1 The Batch Normalization Algorithm

As in Ioffe and Szegedy [2015a], we primarily consider BN for convolutional neural networks. Both the input and output of a BN layer are four dimensional tensors, which we refer to as $I_{b,c,x,y}$ and $O_{b,c,x,y}$, respectively. The dimensions corresponding to examples within a batch b, channel c, and two spatial dimensions x, y respectively. For input images the channels correspond to the RGB channels. BN applies the same normalization for all activations in a given channel,

$$O_{b,c,x,y} \leftarrow \gamma_c \frac{I_{b,c,x,y} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c \qquad \forall\ b, c, x, y. \tag{6.1}$$

Here, BN subtracts the mean activation $\mu_c = \frac{1}{|\mathcal{B}|} \sum_{b,x,y} I_{b,c,x,y}$ from all input activations in channel c, where $\mathcal{B}$ contains all activations in channel c across all features b in the entire mini-batch and all spatial x, y locations. Subsequently, BN divides the centered activation by the standard deviation σ_c (plus ϵ for numerical stability) which is calculated analogously. During testing, running averages of the mean and variances are used. Normalization is followed by a channel-wise affine transformation parametrized through γ_c, β_c, which are learned during training.

6.1.2 Experimental Setup

To investigate batch normalization we will use an experimental setup similar to the original Resnet paper He et al. [2016b]: image classification on CIFAR10 Krizhevsky and Hinton [2009] with a 110 layer Resnet. We use SGD with momentum and weight decay, employ standard data augmentation and image preprocessing techniques and decrease learning rate when learning plateaus, all

as in He et al. [2016b] and with the same parameter values. The original network can be trained with initial learning rate 0.1 over 165 epochs, however which fails without BN. We always report the best results among initial learning rates from $\{0.1, 0.003, 0.001, 0.0003, 0.0001, 0.00003\}$ and use enough epochs such that learning plateaus.

6.2 Disentangling the benefits of BN

Without batch normalization, we have found that the initial learning rate of the Resnet model needs to be decreased to $\alpha = 0.0001$ for convergence and training takes roughly 2400 epochs. We refer to this architecture as an unnormalized network. As illustrated in Figure 6.1 this configuration does not attain the accuracy of its normalized counterpart. Thus, seemingly, batch normalization yields faster training, higher accuracy and enable higher learning rates. To disentangle how these benefits are related, we train a batch normalized network using the learning rate and the number of epochs of an unnormalized network, as well as an initial learning rate of $\alpha = 0.003$ which requires 1320 epochs for training. These results are also illustrated in Figure 6.1, where we see that a batch normalized networks with such a low learning schedule performs no better than an unnormalized network. Additionally, the train-test gap is much larger than for normalized networks using lr $\alpha = 0.1$, indicating more overfitting. A learning rate of $\alpha = 0.003$ gives results in between these extremes. This suggests that it is the higher learning rate that BN enables, which mediates the majority of its benefits; it improves regularization, accuracy and gives faster convergence. Similar results can be shown for variants of BN.

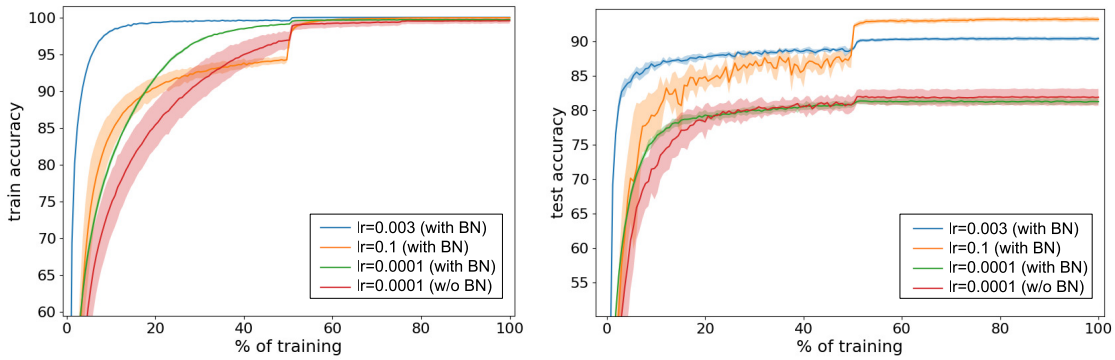

Figure 6.1: The training (*left*) and testing (*right*) accuracies as a function of progress through the training cycle. We used a 110-layer Resnet with three distinct learning rates 0.0001, 0.003, 0.1. The smallest, 0.0001 was picked such that the network without BN converges. The figure shows that with matching learning rates, both networks, with BN and without, result in comparable testing accuracies (red and green lines in right plot). In contrast, larger learning rates yield higher test accuracy for BN networks, and diverge for unnormalized networks (not shown). All results are averaged over five runs with std shown as shaded region around mean.

6.2.1 Learning rate and generalization

To explain these observations we consider a simple model of SGD; the loss function $\ell(x)$ is a sum over the losses of individual examples in our dataset $\ell(x) = \frac{1}{N}\sum_{i=1}^{N} \ell_i(x)$. We model SGD as sampling a set B of examples from the dataset with replacements, and then with learning rate α estimate the gradient step as $\alpha\nabla_{SGD}(x) = \frac{\alpha}{|B|}\sum_{i\in B}\nabla\ell_i(x)$. If we subtract and add $\alpha\nabla\ell(x)$ from this expression we can restate the estimated gradient $\nabla_{SGD}(x)$ as the true gradient, and a noise term

$$\alpha\nabla_{SGD}(x) = \underbrace{\alpha\nabla\ell(x)}_{\text{gradient}} + \underbrace{\frac{\alpha}{|B|}\sum_{i\in B}\left(\nabla\ell_i(x) - \nabla\ell(x)\right)}_{\text{error term}}.$$

We note that since we sample uniformly we have $\mathbb{E}\Big[\frac{\alpha}{|B|}\sum_{i\in B}\big(\nabla\ell_i(x)-\nabla\ell(x)\big)\Big]=0$. Thus the gradient estimate is unbiased, but will typically be noisy. Let us define an architecture dependent noise quantity C of a single gradient estimate such that $C=\mathbb{E}\big[\|\nabla\ell_i(x)-\nabla\ell(x)\|^2\big]$. Using basic linear algebra and probability theory, see the Apppendix, we can upper-bound the noise of the gradient step estimate given by SGD as

$$\mathbb{E}\big[\|\alpha\nabla\ell(x)-\alpha\nabla_{SGD}(x)\|^2\big]\leqslant\frac{\alpha^2}{|B|}C. \tag{6.2}$$

Depending on the tightness of this bound, it suggests that the noise in an SGD step is affected similarly by the learning rate as by the inverse mini-batch size $\frac{1}{|B|}$. This has indeed been observed in practice in the context of parallelizing neural networks Goyal et al. [2017], Smith et al. [2017] and derived in other theoretical models Jastrzkebski et al. [2017]. It is widely believed that the noise in SGD has an important role in regularizing neural networks Zhang et al. [2016], Chaudhari and Soatto [2017]. Most pertinent to us is the work of Keskar et al. Keskar et al. [2016], where it is empirically demonstrated that large mini-batches lead to convergence in sharp minima, which often generalize poorly. The intuition is that larger SGD noise from smaller mini-batches prevents the network from getting "trapped" in sharp minima and therefore bias it towards wider minima with better generalization. Our observation from (6.2) implies that SGD noise is similarly affected by the learning rate as by the inverse mini-bath size, suggesting that a higher learning rate would similarly bias the network towards wider minima. We thus argue that the better generalization accuracy of networks with BN, as shown in Figure 6.1, can be explained by the higher learning rates that BN enables.

6.3 Batch Normalization and Divergence

So far we have provided empirical evidence that the benefits of batch normalization are primarily caused by higher learning rates. We now investigate why BN facilitates training with higher learning rates in the first place. In our experiments, the maximum learning rates for unnormalized networks have been limited by the tendency of neural networks to *diverge* for large rates, which typically happens in the first few mini-batches. We therefore focus on the gradients at initialization. When comparing the gradients between batch normalized and unnormalized networks one consistently finds that the gradients of comparable parameters are larger and distributed with heavier tails in unnormalized networks. Representative distributions for gradients within a convolutional kernel are illustrated in Figure 6.2.

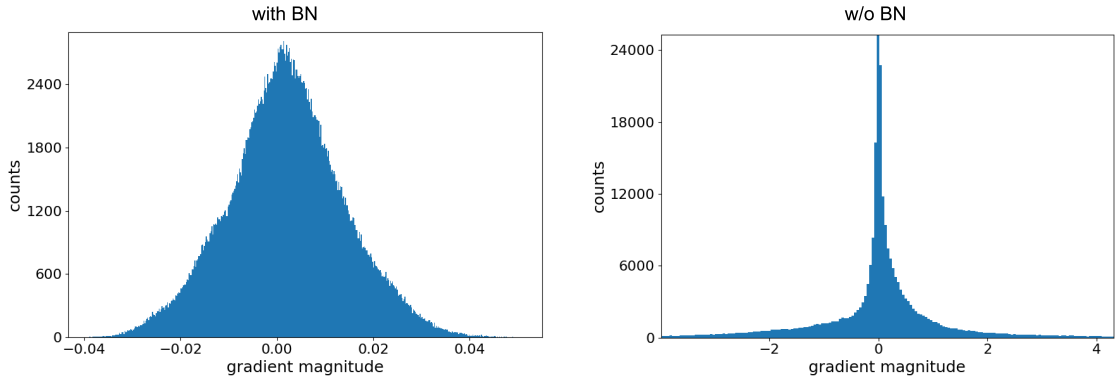

Figure 6.2: Histograms over the gradients at initialization for (midpoint) layer 55 of a network with BN (*left*) and without (*right*). For the unnormalized network, the gradients are distributed with heavy tails, whereas for the normalized networks the gradients are concentrated around the mean. (Note that we have to use different scales for the two plots because the gradients for the unnormalized network are almost two orders of magnitude larger than for the normalized on.)

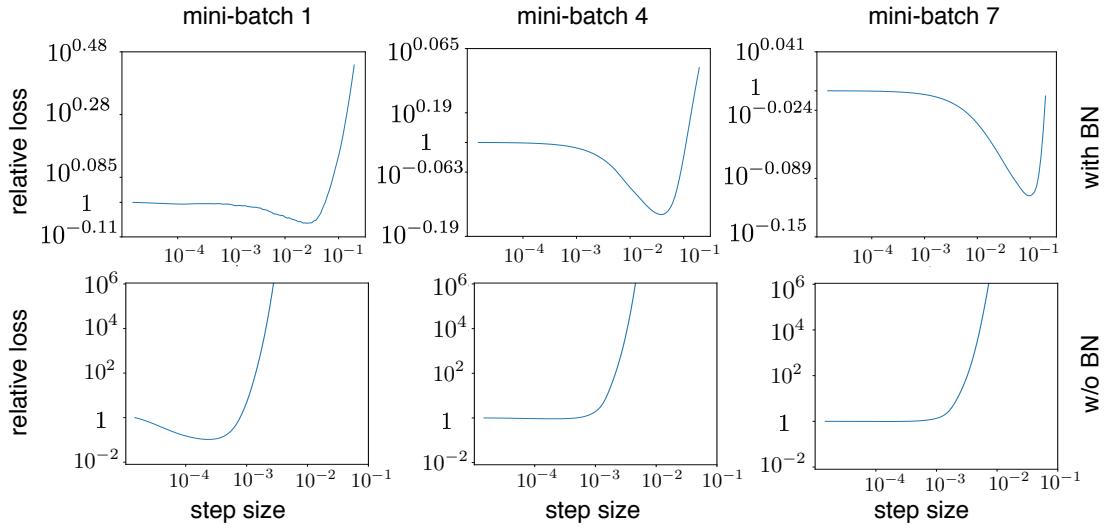

Figure 6.3: Illustrations of the relative loss over a mini-batch as a function of the step-size (normalized by the loss before the gradient step). Several representative batches and networks are shown, each one picked at the start of the standard training procedure. Throughout all cases the network with BN (bottom row) is far more forgiving and the loss decreases over larger ranges of α. Networks without BN show divergence for larger step sizes.

A natural way of investigating divergence is to look at the loss landscape along the gradient direction during the first few mini-batches that occur with the normal learning rate (0.1 with BN, 0.0001 without). In Figure 6.3 we compare networks with and without BN in this regard. For each network we compute the gradient on individual batches and plot the relative change in loss as a function of the step-size (i.e. new_loss/old_loss). (Please note the different scales along the vertical axes.) For unnormalized networks only small gradient steps lead to reductions in loss, whereas networks with BN can use a far broader range of learning rates.

Let us define *network divergence* as the point when the loss of a mini-batch increases beyond 10^3 (a point from which networks have never managed to

recover to acceptable accuracies in our experiments). With this definition, we can precisely find the gradient update responsible for divergence. It is interesting to see what happens with the means and variances of the network activations along a 'diverging update'. Figure 6.4 shows the means and variances of channels in three layers (8,44,80) during such an update (without BN). The color bar reveals that the scale of the later layer's activations and variances is orders of magnitudes higher than the earlier layer. This seems to suggest that the divergence is caused by activations growing progressively larger with network depth, with the network output "exploding" which results in a diverging loss. BN successfully mitigates this phenomenon by correcting the activations of each channel and each layer to zero-mean and unit standard deviation, which ensures that large activations in lower levels cannot propagate uncontrollably upwards. We argue that this is the primary mechanism by which batch normalization enables higher learning rates – no matter how large of a gradient update is applied, the network will always "land safely" in a region without activations growing with network depth. Our explanation would suggest that the mean and variance μ, σ^2 used for normalization needs to be updated every batch for this "security guarantee" to hold. This explanation is also consistent with the general folklore observations that shallower networks allow for larger learning rates. In shallower networks there aren't as many layers in which the activation explosion can propagate.

6.4 Batch Normalization and Gradients

Figure 6.4 shows that the moments of unnormalized networks explode during network divergence and Figure 6.5 depicts the moments as a function of the layer

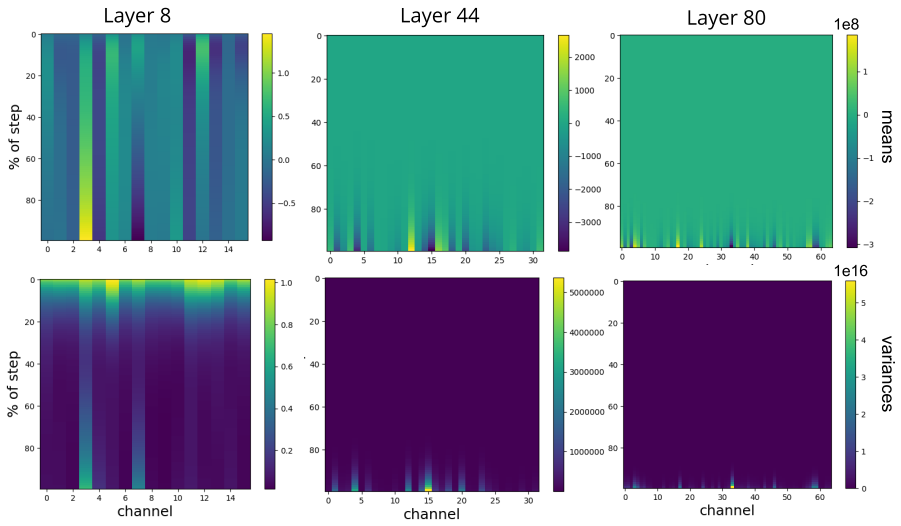

Figure 6.4: Heatmap of channel means and variances during a diverging gradient update (without BN). The vertical axis denote what percentage of the gradient update has been applied, 100% corresponds to the endpoint of the update. The moments explode in the higher layer (note the scale of the color bars).

depth after initialization (without BN) in log-scale. The means and variances of channels in the network tend to increase with the depth of the network even at initialization time — suggesting that a substantial part of this growth is data independent. In Figure 6.5 we also note that the network transforms normalized inputs into an output that reaches scales of up to 10^2 for the largest output channels. It is natural to suspect that such a dramatic relationship between output and input are responsible for the large gradients seen in Figure 6.2. To test this intuition, we train a Resnet that uses one batch normalization layer only at the very last layer of the network, normalizing the output of the last residual block but no intermediate activation. Such an architecture allows for learning rates up to 0.03 and yields a final test accuracy of 90.1% — capturing two-thirds of the overall BN improvement (see Figure 6.1). This suggests that normalizing the final layer of a deep network may be one of the most important contributions of BN.

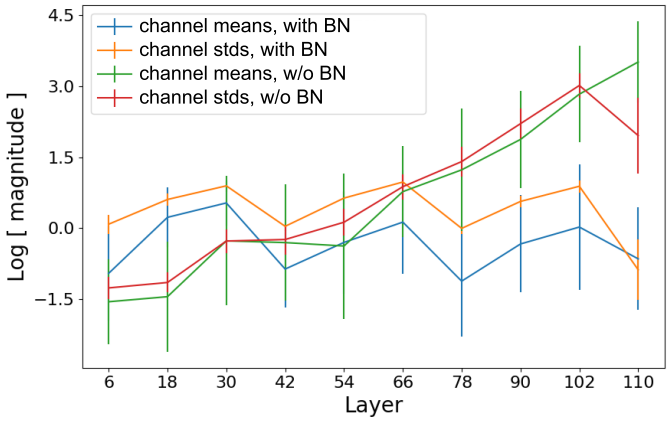

Figure 6.5: Average channel means and variances as a function of network depth at initialization (error bars show standard deviations) on log-scale for networks with and without BN. The batch normalized network the mean and variances stays relatively constant throughout the network. For an unnormalized network, they seem to grow almost exponentially with depth.

For the final output layer corresponding to the classification, a large channel mean implies that the network is biased towards the corresponding class. In Figure 6.6 we created a heatmap of $\frac{\partial L_b}{\partial O_{b,j}}$ after initialization, where L_b is the loss for image b in our mini-batch, and activations j corresponds to class j at the final layer. A yellow entry indicates that the gradient is positive, and the step along the negative gradient would decrease the prediction strength of this class for this particular image. A dark blue entry indicates a negative gradient, indicating that this particular class prediction should be strengthened. Each row contains one dark blue entry, which corresponds to the true class of this particular image (as initially all predictions are arbitrary). A striking observation is the distinctly yellow column in the left heatmap (network without BN). This indicates that after initialization the network tends to almost always predict the same (typically wrong) class, which is then corrected with a strong gradient update. In contrast, the network with BN does not exhibit the same behavior,

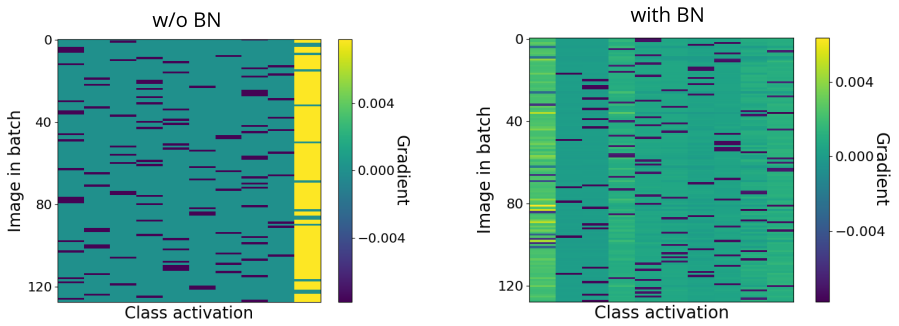

Figure 6.6: A heat map of the output gradients in the final classification layer after initialization. The columns correspond to a classes and the rows to images in the mini-batch. For an unnormalized network (*left*), it is evident that the network consistently predicts one specific class (very right column), irrespective of the input. As a result, the gradients are highly correlated. For a batch normalized network, the dependence upon the input is much larger.

instead positive gradients are distributed throughout all classes. Figure 6.6 also sheds light onto why the gradients of networks without BN tend to be so large in the final layers: the rows of the heatmap (corresponding to different images in the mini-batch) are highly correlated. Especially the gradients in the last column are positive for almost all images (the only exceptions being those image that truly belong to this particular class label). The gradients, summed across all images in the minibatch, therefore consist of a sum of terms with matching signs and yield large absolute values. Further, these gradients differ little across inputs, suggesting that most of the optimization work is done to rectify a bad initial state rather than learning from the data.

6.4.1 Gradients of convolutional parameters

We observe that the gradients in the last layer can be dominated by some arbitrary bias towards a particular class. Can a similar reason explain why the gradients for convolutional weights are larger for unnormalized networks. Let us consider a convolutional weight $K_{o,i,x,y}$, where the first two dimensions correspond to the outgoing/ingoing channels and the two latter to the spatial dimensions. For notational clarity we consider 3-by-3 convolutions and define $S = \{-1,0,1\} \times \{-1,0,1\}$ which indexes into K along spatial dimensions. Using definitions from section 6.1.1 we have

$$O_{b,c,x,y} = \sum_{c'} \sum_{x',y' \in S} I_{b,c',x+x',y+y'} K_{c,c',x',y'} \tag{6.3}$$

Now for some parameter $K_{o,i,x,y}$ inside the convolutional weight K, its derivate with respect to the loss is given by the backprop equation Rumelhart et al. [1986] and (6.3) as

$$\frac{\partial L}{\partial K_{o,i,x',y'}} = \sum_{b,x,y} d^{bxy}_{o,i,x',y'}, \qquad \text{where} \qquad d^{bxy}_{o,i,x',y'} = \frac{\partial L}{\partial\, O_{b,o,x,y}} I_{b,i,x+x',y+y'}. \tag{6.4}$$

The gradient for $K_{o,i,x,y}$ is the sum over the gradients of examples within the mini-batch, and over the convoluted spatial dimensions. We investigate the signs of the summands in (6.4) across both network types and probe the sums at initialization in Table 6.1. For an unnormalized networks the absolute value of (6.4) and the sum of the absolute values of the summands generally agree to within a factor 2 or less. For a batch normalized network, these expressions differ by a factor of 10^2, which explains the stark difference in gradient magnitude

	$a = \sum_{bxy} \lvert d^{bxy}_{c_o c_i i j} \rvert$	$b = \lvert \sum_{bij} d^{bxy}_{c_o c_i i j} \rvert$	a/b
Layer 18, with BN	7.5e-05	3.0e-07	**251.8**
Layer 54, with BN	1.9e-05	1.7e-07	**112.8**
Layer 90, with BN	6.6e-06	1.6e-07	**40.7**
Layer 18, w/o BN	6.3e-05	3.6e-05	**1.7**
Layer 54, w/o BN	2.2e-04	8.4e-05	**2.6**
Layer 90, w/o BN	2.6e-04	1.2e-04	**2.1**

Table 6.1: Gradients of a convolutional kernel as described in (6.4) at initialization. The table compares the absolute value of the sum of gradients, and the sum of absolute values. Without BN these two terms are similar in magnitude, suggesting that the summands have matching signs throughout and are largely data independent. For a batch normalized network, those two differ by about two orders of magnitude.

between normalized and unnormalized networks observed in Figure 6.2. These results suggest that for an unnormalized network, the summands in (6.4) are similar across both spatial dimensions and examples within a batch. They thus encode information that is neither input-dependent or dependent upon spatial dimensions, and we argue that the learning rate would be limited by the large input-independent gradient component and that it might be too small for the input-dependent component.

Table 6.1 suggests that for an unnormalized network the gradients are similar across spatial dimensions and images within a batch. It's unclear however how they vary across the input/output channels i, o. To study this we consider the matrix $\mathbf{M}_{i,o} = \sum_{xy} \lvert \sum_{bxy} d^{bxy}_{oixy} \rvert$ at initialization, which intuitively measures the average gradient magnitude of kernel parameters between input channel i and output channel o. Representative results are illustrated in Figure 6.7. The heatmap shows a clear trend that some channels constantly are associated with larger gradients while others have extremely small gradients by comparison. Since some channels have large means, we expect in light of (6.4) that weights

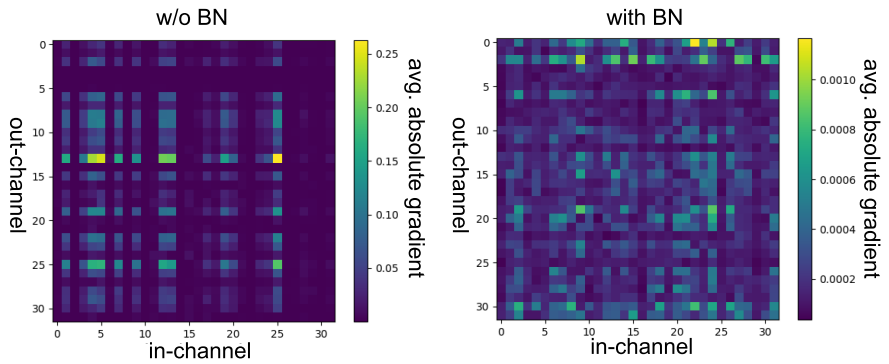

Figure 6.7: Average absolute gradients for parameters between in and out channels for layer 45 at initialization. For an unnormalized network, we observe a dominant low-rank structure. Some in/out-channels have consistently large gradients while others have consistently small gradients. This structure is less pronounced with batch normalization (*right*).

outgoing from such channels would have large gradients which would explain the structure in Figure 6.7.

6.5 Random initialization

In this section argue that the gradient explosion in networks without BN is a natural consequence of random initialization. This idea seems to be at odds with the trusted Xavier initialization scheme [Glorot and Bengio, 2010] which we use. Doesn't such initialization guarantee a network where information flows smoothly between layers? These initialization schemes are generally derived from the desiderata that the variance of channels should be constant when randomization is taken over random weights. We argue that this condition is too weak. For example, a pathological initialization that sets weights to 0 or 100 with some probability could fulfill it. In Glorot and Bengio [2010] the authors make simplifying assumptions that essentially result in a linear neural

network. We consider a similar scenario and connect them with recent results in random matrix theory to gain further insights into network generalization. Let us consider a simple toy model: a linear feed-forward neural network where $A_t \ ... \ A_2 A_1 x = y$, for weight matrices $A_1, A_2 ... A_n$. While such a model clearly abstracts away many important points they have proven to be valuable models for theoretical studies [Glorot and Bengio, 2010, Yun et al., 2017, Hardt and Ma, 2016, Lu and Kawaguchi, 2017]. CNNs can, of course, be flattened into fully-connected layers with shared weights. Now, if the matrices are initialized randomly, the network can simply be described by a product of random matrices. Such products have recently garnered attention in the field of random matrix theory, from which we have the following recent result due to Liu et al. [2016].

Theorem 3. ***Singular value distribution of products of independent Gaussian matrices [Liu et al., 2016].*** *Assume that $X = X_1 X_2 ... X_M$, where X_i are independent $N \times N$ Gaussian matrices s.t. $\mathbb{E}[X_{i,jk}] = 0$ and $\mathbb{E}[X_{i,jk}^2] = \sigma_i^2/N$ for all matrices i and indices j, k. In the limit $N \to \infty$, the expected singular value density $\rho_M(x)$ of X for $x \in \left(0, (M+1)^{M+1}/M^M\right)$ is given by*

$$\rho_M(x) = \frac{1}{\pi x} \frac{\sin((M+1)\varphi)}{\sin(M\varphi)} \sin\varphi, \qquad \textit{where} \qquad x = \frac{\left(\sin((M+1)\varphi)\right)^{M+1}}{\sin\varphi (\sin(M\varphi))^M} \tag{6.5}$$

Figure 6.8 illustrates some density plots for various values of M and θ. A closer look at (6.5) reveals that the distribution blows up as $x^{-M/(M+1)}$ nears the origin, and that the largest singular value scales as $O(M)$ for large matrices. In Figure 6.9 we investigate the singular value distribution for practically sized matrices. By multiplying more matrices, which represents a deeper linear network, the singular values distribution becomes significantly more heavy-tailed. Intuitively this means that the ratio between the largest and smallest singular value (the

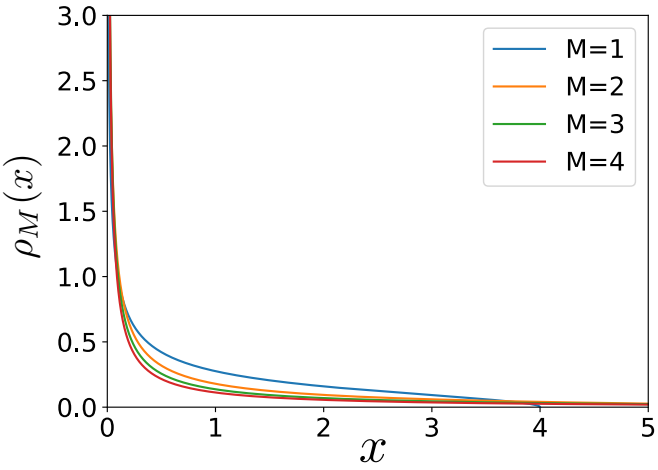

Figure 6.8: Distribution of singular values according to theorem 3 for some M. The theoretical distribution becomes increasingly heavy-tailed for more matrices, as does the empirical distributions of Figure 6.9

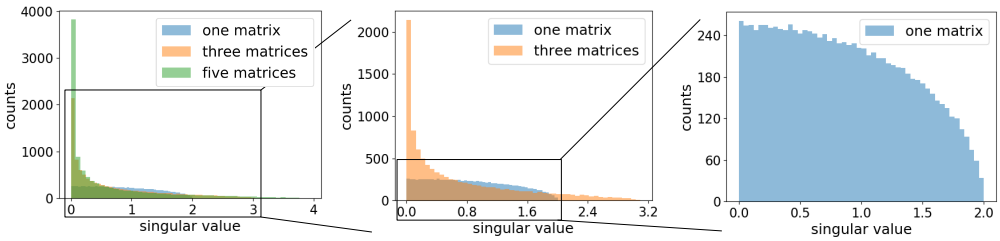

Figure 6.9: An illustration of the distributions of singular values of random square matrices and product of independent matrices. The matrices have dimension N=1000 and all entries independently drawn from a standard Gaussian distribution. Experiments are repeated ten times and we show the total number of singular values among all runs in every bin, distributions for individual experiments look similar. The left plot shows all three settings. We see that the distribution of singular values becomes more heavy-tailed as more matrices are multiplied together.

condition number) will increase with depth.

Consider $\min_{A_i,\, i=1,2\,\ldots\,t} \|A_t \ldots A_2 A_1 x - y\|^2$, this problem is similar to solving a linear system $\min_x \|Ax - y\|^2$ if one only optimizes over a single weight matrix A_i. It is well known that the complexity of solving $\min_x \|Ax - y\|$ via gradient descent

can be characterized by the condition number κ of A, the ratio between largest σ_{max} and smallest singular value σ_{min}. Increasing κ has the following effects on solving a linear system with gradient descent: **1)** convergence becomes slower, **2)** a smaller learning rate is needed, **3)** the ratio between gradients in different subspaces increases [Bertsekas and Scientific, 2015]. There are many parallels between these results from numerical optimization, and what is observed in practice in deep learning. Based upon Theorem 3, we expect the conditioning of a linear neural network at initialization for more shallow networks to be better which would allow a higher learning rate. And indeed, for an unnormalized Resnet one can use a much larger learning if it has only few layers. An increased condition number also results in different subspaces of the linear regression problem being scaled differently, although the notion of subspaces are lacking in ANNs, Figure 6.5 and 6.7 show that the scale of channels differ dramatically in unnormalized networks. The Xavier [Glorot and Bengio, 2010] and Kaming initialization schemes [He et al., 2015a] amounts to a random matrix with iid entries that are all scaled proportionally to $n^{-1/2}$, the same exponent as in Theorem 3, with different constant factors. Theorem 3 suggests that such an initialization will yield ill-conditioned matrices, independent of these scale factors. If we accept these shortcomings of Xavier-initialization, the importance of making networks robust to initialization schemes becomes more natural.

6.6 Related Work

The original batch normalization paper posits that internal covariate explains the benefits of BN Ioffe and Szegedy [2015a]. We do not claim that internal covariate shift does not exist, but we believe that the success of BN can be

explained without it. We argue that a good reason to doubt that the primary benefit of BN is eliminating internal covariate shift comes from results in Mishkin and Matas [2015], where an initialization scheme that ensures that all layers are normalized is proposed. In this setting, internal covariate shift would not disappear. However, the authors show that such initialization can be used instead of BN with a relatively small performance loss. Another line of work of relevance is Smith and Topin [2017] and Smith [2018], where the relationship between various network parameters, accuracy and convergence speed is investigated, the former article argues for the importance of batch normalization to facilitate a phenomenon dubbed 'super convergence'.

6.7 Discussion

We have investigated batch normalization and its benefits, showing how the latter are mainly mediated by larger learning rates. We argue that the larger learning rate increases the implicit regularization of SGD, which improves generalization. Our experiments show that large parameter updates to unnormalized networks can result in activations whose magnitudes grow dramatically with depth, which limits large learning rates. Additionally, we have demonstrated that unnormalized networks have large and ill-behaved outputs, and that this results in gradients that are input independent. Via recent results in random matrix theory, we have argued that the ill-conditioned activations are natural consequences of the random initialization.

CHAPTER 7

CONCLUSION

In this thesis, I have applied using machine learning in several high-dimensional scientific domains. For the problem of detecting invasive species habitats from remote sensing images, we have introduced a method for unsupervised feature learning. Furthermore, we have demonstrated how this method allows us to inexpensively combine sparse expert labels with abundant remote sensing data. In the domain of bio-acoustics, I have introduced a novel data-driven compression framework for acoustic signals. At last, we have considered demixing x-ray crystallography datasets to accelerate materials science. In this domain, we have demonstrated how interleaving continuous and discrete optimization procedures can lead to high-quality solutions which respect physical rules. In all these domains we have demonstrated how machine learning methods can perform well once we incorporate our prior knowledge into the learning mechanisms – be it through combinatorial optimization steps or using geography to define learning objectives. Thus, while machine learning has recently demonstrated impressive gains in everyday domains such as basic reading comprehension, we have shown that it can also be used in scientific domains and accelerate sustainability solutions.

In scientific domains, one needs to know *when* and *why* machine learning models can be expected to work well. Inspired by our practical problems we have taken steps towards a principled understanding of machine learning. On the theoretical side, we have introduced an average-case model for matrix factorization and proved that it exhibits certain convexity properties despite matrix factorization being NP-hard in the worst case. Our proof highlights of high-dimensional

data can sometimes be beneficial, despite machine learning practitioners often referring to the curse of dimensionality. On the empirical side, we have investigated batch normalization – a ubiquitous technique in deep learning whose utility has not been well understood. In contrast with earlier researchers' hypothesis that it eliminates internal covariate shift, we have demonstrated that it improves conditioning which allows for larger learning rates and more regularization. Our work presents early steps towards machine learning as a scientifically grounded domain, rather than a set of best practices acquired through trial and error.

This work was made possible through the Computational Sustainability research network, via our collaboration with the Joint Center for Artificial Photosynthesis (JCAP) at Caltech, the Cornell Lab of Ornithology, and researchers at the Department of Natural Resources at Cornell. We were also fortunate to test our materials science methods at JCAP. We would also like to thank the Elephant listening project and its volunteers.

BIBLIOGRAPHY

M. Ackerman and S. Ben-David. Clusterability: A theoretical study. In *Artificial Intelligence and Statistics*, pages 1–8, 2009.

P. Afshani, J. Barbay, and T. M. Chan. Instance-optimal geometric algorithms. *Journal of the ACM (JACM)*, 64(1):3, 2017.

E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *NIPS*, 2017.

R. Ahlswede and A. Winter. Strong converse for identification via quantum channels. *IEEE Transactions on Information Theory*, 48(3):569–579, 2002.

J. C. Aker and I. M. Mbiti. Mobile phones and economic development in africa. *Journal of Economic Perspectives*, 2010.

S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM, 2012.

S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018.

G. P. Asner, D. E. Knapp, T. Kennedy-Bowdoin, M. O. Jones, R. E. Martin, J. Boardman, and R. F. Hughes. Invasive species detection in hawaiian rainforests using airborne imaging spectroscopy and lidar. *Remote Sensing of Environment*, 112(5):1942–1955, 2008.

P. Atkins and J. De Paula. Atkins' physical chemistry. *New York*, page 77, 2006.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545, 2019.

J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

B. Barak, S. B. Hopkins, J. Kelner, P. Kothari, A. Moitra, and A. Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 428–437. IEEE, 2016.

R. Bardeli, D. Wolff, F. Kurth, M. Koch, K.-H. Tauchert, and K.-H. Frommolt. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. *Pattern Recognition Letters*, 2010.

M. F. Baumgartner and S. E. Mussoline. A generalized baleen whale call detection and classification system. *The Journal of the Acoustical Society of America*, 2011.

O. Berne, C. Joblin, Y. Deville, J. Smith, M. Rapacioli, J. Bernard, J. Thomas, W. Reach, and A. Abergel. Analysis of the emission of very small dust particles from spitzer spectro-imagery data using blind signal separation methods. *Astronomy & Astrophysics*, 469(2):575–586, 2007.

D. P. Bertsekas and A. Scientific. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

Y. Bilu and N. Linial. Are stable instances easy? *Combinatorics, Probability and Computing*, 21(5):643–660, 2012.

M. Bittle and A. Duncan. A review of current marine mammal detection and classification algorithms for use in automated passive acoustic monitoring. In *Proceedings of Acoustics*, 2013.

J. Bjorck, Y. Bai, X. Wu, Y. Xue, M. Whitmore, and C. Gomes. Scalable relaxations of sparse packing constraints: Optimal biocontrol in predator-prey networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A. Blum and R. L. Rivest. Training a 3-node neural network is np-complete. In *Advances in neural information processing systems*, pages 494–501, 1989.

A. L. Blum and R. L. Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

H. A. Bourlard and N. Morgan. *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 2012.

İ. E. Büyüktahtakın, Z. Feng, and F. Szidarovszky. A multi-objective optimization approach for invasive species control. *Journal of the Operational Research Society*, 65(11):1625–1635, 2014.

A. Campos-Arceiz and S. Blake. Megagardeners of the forest–the role of elephants in seed dispersal. *Acta Oecologica*, 2011.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

P. Chaudhari and S. Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *arXiv preprint arXiv:1710.11029*, 2017.

D. Chen and C. P. Gomes. Bias reduction via end-to-end shift learning: Application to citizen science. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 493–500, 2019.

Q. Chen, X. Song, H. Yamada, and R. Shibasaki. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

N. Department of Environmental Conservation. *New York State Invasive Species Comprehensive Management Plan*, 2018. URL `https://www.dec.ny.gov/docs/lands_forests_pdf/iscmpfinal.pdf`.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, 2004.

O. Dufour, T. Artieres, H. Glotin, and P. Giraudet. Clusterized mel filter cepstral coefficients and support vector machines for bird song identification. In *Soundscape Semiotics-Localization and Categorization*. 2014.

F. Erbs, S. H. Elwen, and T. Gridley. Automatic classification of whistles from coastal dolphins of the southern african subregion. *The Journal of the Acoustical Society of America*, 2017.

N. B. Erichson, A. Mendible, S. Wihlborn, and J. N. Kutz. Randomized nonnegative matrix factorization. *Pattern Recognition Letters*, 104:1–7, 2018.

S. Ermon, R. Le Bras, C. P. Gomes, B. Selman, and R. B. Van Dover. Smt-aided combinatorial materials discovery. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 172–185. Springer, 2012.

S. Ermon, R. L. Bras, S. K. Suram, J. M. Gregoire, C. Gomes, B. Selman, and R. B. Van Dover. Pattern decomposition with complex combinatorial constraints: Application to materials discovery. *arXiv preprint arXiv:1411.7441*, 2014.

J. Flenner and B. Hunter. A deep non-negative matrix factorization neural network. *Semantic Scholar*, 2017.

J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

O. Fried, S. Avidan, and D. Cohen-Or. Patch2vec: Globally consistent image patch representation. In *Computer Graphics Forum*, volume 36, pages 183–194. Wiley Online Library, 2017.

T. Ganchev and I. Potamitis. Automatic acoustic identification of singing insects. *Bioacoustics*, 2007.

C. F. Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. sumtibus Frid. Perthes et IH Besser, 1809.

R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, 2015.

S. Gholami, L. Xu, S. Mc Carthy, B. Dilkina, A. Plumptre, M. Tambe, R. Singh, M. Nsubuga, J. Mabonga, M. Driciru, et al. Stay ahead of poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations. *arXiv preprint arXiv:1903.06669*, 2019.

N. Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13(Nov):3349–3386, 2012.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

C. Gomes, T. Dietterich, C. Barrett, J. Conrad, B. Dilkina, S. Ermon, F. Fang, A. Farnsworth, A. Fern, X. Fern, et al. Computational sustainability: Computing for a better world and a sustainable future. *Communications of the ACM*, 62 (9):56–65, 2019.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, 2014.

R. M. Gray and D. L. Neuhoff. Quantization. *IEEE transactions on information theory*, 1998.

A. Gupta, M. Farajtabar, B. Dilkina, and H. Zha. Discrete interventions in hawkes processes with applications in invasive species management. In *IJCAI*, pages 3385–3392, 2018.

M. Hardt and T. Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015a.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015b. URL `http://arxiv.org/abs/1512.03385`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016a.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016b.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

D. Hedwig, M. DeBellis, and P. H. Wrege. Not so far: attenuation of low-frequency vocalizations in a rainforest environment suggests limited acoustic mediation of social interaction in african forest elephants. *Behavioral Ecology and Sociobiology*, 2018.

P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994.

E. Hoffer and N. Ailon. Deep metric learning using triplet network. In A. Feragen, M. Pelillo, and M. Loog, editors, *Similarity-Based Pattern Recognition*, pages 84–92, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24261-3.

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

T. P. Holmes, E. A. Murphy, and D. D. Royle. The economic impacts of hemlock woolly adelgid on residential landscape values: Sparta, new jersey case study. In *In: Onken, B.; Reardon, R. comps., eds. Proceedings of the 3rd symposium on hemlock woolly adelgid in the eastern United States, FHTET-2005-01. Morgantown, WV: US Department of Agriculture, Forest Service, Forest Health Technology Enterprise Team: 15-24.*, 2005.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411 – 430, 2000. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(00)00026-5. URL `http://www.sciencedirect.com/science/article/pii/S0893608000000265`.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015a.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015b.

P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

B. Jagadeesh and B. S. Kumar. Psychoacoustic model-1 implementation for mpeg audio encoder using wavelet packet decomposition. In *Emerging Research in Electronics, Computer Science and Technology*. 2014.

S. Jastrzkebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

N. Jean, S. Wang, A. Samar, G. Azzari, D. Lobell, and S. Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3967–3974, 2019.

J. R. Jensen. *Remote sensing of the environment: An earth resource perspective 2/e*. Pearson Education India, 2009.

N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017.

B.-H. Juang, S. Levinson, and M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.). *IEEE Transactions on Information Theory*, 1986.

J. Kang, R. Fernandez-Beltran, P. Duan, S. Liu, and A. J. Plaza. Deep unsupervised embedding for remotely sensed images based on spatially augmented momentum contrast. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

S. Kankanahalli. End-to-end optimized speech coding with deep neural networks. *arXiv preprint arXiv:1710.09064*, 2017.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

R. Kleinberg, Y. Li, and Y. Yuan. An alternative view: When does sgd escape local minima? *arXiv preprint arXiv:1802.06175*, 2018.

H. V. Koops, J. Van Balen, and F. Wiering. Automatic segmentation and deep learning of bird sounds. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, 2015.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/`

4824-imagenet-classification-with-deep-convolutional-neural-network
pdf.

M. Kula. Mixture-of-tastes models for representing users with diverse interests. *arXiv preprint arXiv:1711.08379*, 2017.

E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. 2001.

N. D. Lane, P. Georgiev, and L. Qendro. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015.

R. Le Bras, T. Damoulas, J. M. Gregoire, A. Sabharwal, C. P. Gomes, and R. B. Van Dover. Constraint reasoning and kernel clustering for pattern decomposition with scaling. In *International Conference on Principles and Practice of Constraint Programming*, pages 508–522. Springer, 2011.

R. Le Bras, R. Bernstein, S. K. Suram, J. M. Gregoire, B. Selman, C. P. Gomes, and R. B. van Dover. A computational challenge problem in materials discovery: Synthetic problem generator and real-world datasets. 2014.

Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

D.-T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980.

J. C. Lee and P. Valiant. Optimizing star-convex functions. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–614. IEEE, 2016.

L. B. Lentile, Z. A. Holden, A. M. Smith, M. J. Falkowski, A. T. Hudak, P. Morgan, S. A. Lewis, P. E. Gessler, and N. C. Benson. Remote sensing techniques to assess active fire characteristics and post-fire effects. *International Journal of Wildland Fire*, 15(3):319–345, 2006.

H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. *CVPR (1)*, 207:212, 2001.

Y. Li and A. Ngom. The non-negative matrix factorization toolbox for biological data mining. *Source code for biology and medicine*, 8(1):10, 2013.

Y. Li and Y. Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

D.-Z. Liu, D. Wang, L. Zhang, et al. Bulk and soft-edge universality for singular values of products of ginibre random matrices. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 52, pages 1734–1762. Institut Henri Poincaré, 2016.

C. Long, J. Hattrick-Simpers, M. Murakami, R. Srivastava, I. Takeuchi, V. L. Karen, and X. Li. Rapid structural mapping of ternary metallic alloy systems using the combinatorial approach and cluster analysis. *Review of Scientific Instruments*, 78(7):072217, 2007.

C. Long, D. Bunker, X. Li, V. Karen, and I. Takeuchi. Rapid identification of structural phases in combinatorial thin-film libraries using x-ray diffraction and non-negative matrix factorization. *Review of Scientific Instruments*, 80(10): 103902, 2009.

C. Lu and X. Tang. Surpassing human-level face verification performance on lfw with gaussianface. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

H. Lu and K. Kawaguchi. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*, 2017.

X. Luo, M. Zhou, Y. Xia, and Q. Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

O. Mac Aodha, R. Gibb, K. E. Barlow, E. Browning, M. Firman, R. Freeman, B. Harder, L. Kinsey, G. R. Mead, S. E. Newson, et al. Bat detective—deep learning tools for bat acoustic signal detection. *PLoS computational biology*, 2018.

Y. Mao, L. K. Saul, and J. M. Smith. Ides: An internet distance estimation service

for large networks. *IEEE Journal on Selected Areas in Communications*, 24(12): 2273–2284, 2006.

M. A. McDonald and C. G. Fox. Passive acoustic methods applied to fin whale population density estimation. *The Journal of the Acoustical Society of America*, 1999.

M. Meckes and S. Szarek. Concentration for noncommutative polynomials in random matrices. *Proceedings of the American Mathematical Society*, 140(5): 1803–1813, 2012.

P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaev, G. Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

D. Mishkin and J. Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

A.-r. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, 2009.

R. Morelle. Slow birth rate found in african forest elephants. 2016.

N. Nangia and S. R. Bowman. Human vs. muppet: A conservative estimate of human performance on the glue benchmark. *arXiv preprint arXiv:1905.10425*, 2019.

NatureServe. *iMapInvasives: NatureServe's online data system supporting strategic invasive species management*, 2020. URL `http://www.imapinvasives.org`.

B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

S. Nienhuis, T. J. Haxton, and T. C. Dunkley. An empirical analysis of the consequences of zebra mussel invasions on fisheries in inland, freshwater lakes in southern ontario. *Management of Biological Invasions*, 5(3):287, 2014.

A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

K. L. Oten, S. A. Merkle, R. M. Jetton, B. C. Smith, M. E. Talley, and F. P. Hain. Understanding and developing resistance in hemlocks to the hemlock woolly adelgid. *Southeastern Naturalist*, 13(6):147–167, 2014.

T. Painter and A. Spanias. Perceptual coding of digital audio. 2000.

P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is np-hard. *Journal of Global Optimization*, 1(1):15–22, 1991.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

R. Piiroinen, F. E. Fassnacht, J. Heiskanen, E. Maeda, B. Mack, and P. Pellikka. Invasive tree species detection in the eastern arc mountains biodiversity hotspot using one class classification. *Remote Sensing of Environment*, 218:119–131, 2018.

D. Pimentel, R. Zuniga, and D. Morrison. Update on the environmental and economic costs associated with alien-invasive species in the united states. *Ecological economics*, 52(3):273–288, 2005.

G. Pleiss, P. H. Wrege, and C. Gomes. Unpublished technical report. 2016.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

E. Richard and A. Montanari. A statistical model for tensor pca. In *Advances in Neural Information Processing Systems*, pages 2897–2905, 2014.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

M. Sahidullah and G. Saha. Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition. *Speech Communication*, 2012.

T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015.

H. Sak, A. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

J. Salamon, J. P. Bello, A. Farnsworth, and S. Kelling. Fusing shallow and deep learning for bioacoustic bird species classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

M. N. Schmidt, J. Larsen, and F.-T. Hsiao. Wind noise reduction using non-negative sparse coding. In *2007 IEEE workshop on machine learning for signal processing*, pages 431–436. IEEE, 2007.

T. Setiyono, A. Nelson, and F. Holecz. Remote sensing-based crop yield monitoring and forecasting. *Crop monitoring for improved food security*, 2014.

B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning Series. Morgan & Claypool, 2012. ISBN 978-1-60845-725-0. URL `https://books.google.com/books?id=z7toC3z_QjQC`.

A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799. ACM, 2005.

K. Shea and H. P. Possingham. Optimal release strategies for biological control agents: an application of stochastic dynamic programming to population management. *Journal of Applied ecology*, 37(1):77–86, 2000.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

A. Širović, J. A. Hildebrand, and S. M. Wiggins. Blue and fin whale call source levels and propagation range in the southern ocean. *The Journal of the Acoustical Society of America*, 2007.

P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *International Conference on Independent Component Analysis and Signal Separation*, pages 494–499. Springer, 2004.

L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. arxiv e-prints, page. *arXiv preprint arXiv:1708.07120*, 2017.

S. L. Smith, P.-J. Kindermans, and Q. V. Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

G. Spencer. Robust cuts over time: Combatting the spread of invasive species with unreliable biological control. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

S. K. Suram, Y. Xue, J. Bai, R. L. Bras, B. Rappazzo, R. Bernstein, J. Bjorck, L. Zhou, R. B. van Dover, C. P. Gomes, et al. Automated phase mapping with agilefd and its application to light absorber discovery in the v-mn-nb oxide system. *arXiv preprint arXiv:1610.02005*, 2016a.

S. K. Suram, Y. Xue, J. Bai, R. Le Bras, B. Rappazzo, R. Bernstein, J. Bjorck, L. Zhou, R. B. van Dover, C. P. Gomes, et al. Automated phase mapping with agilefd and its application to light absorber discovery in the v–mn–nb oxide system. *ACS combinatorial science*, 19(1):37–46, 2016b.

M. A. Taleghan, T. G. Dietterich, M. Crowley, K. Hall, and H. J. Albers. Pac optimal mdp planning with application to invasive species management. *The Journal of Machine Learning Research*, 16(1):3877–3903, 2015.

A. Thode, J. Bonnel, M. Thieury, A. Fagan, C. Verlinden, D. Wright, C. Berchok, and J. Crance. Using nonlinear time warping to estimate north pacific right whale calling depths in the bering sea. *The Journal of the Acoustical Society of America*, 2017.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. page 13, 1999.

G. F. Trindade, M.-L. Abel, and J. F. Watts. Non-negative matrix factorisation of large mass spectrometry datasets. *Chemometrics and Intelligent Laboratory Systems*, 163:76–85, 2017.

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

H. K. Turesson, S. Ribeiro, D. R. Pereira, J. P. Papa, and V. H. C. de Albuquerque.

Machine learning algorithms for automatic classification of marmoset vocalizations. 2016.

S. Ustin, D. DiPietro, K. Olmstead, E. Underwood, and G. Scheer. Hyperspectral remote sensing for invasive species detection and mapping. volume 3, pages 1658 – 1660 vol.3, 07 2002. ISBN 0-7803-7536-X. doi: 10.1109/IGARSS.2002.1026212.

S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009a.

S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009b.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

A. X. Wang, C. Tran, N. Desai, D. Lobell, and S. Ermon. Deep transfer learning for crop yield prediction with remote sensing data. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–5, 2018.

C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online, Aug. 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.acl-long.80`.

P. H. Wrege, E. D. Rowland, S. Keen, and Y. Shiu. Acoustic monitoring for

conservation in tropical forests: examples from forest elephants. *Methods in Ecology and Evolution*, 2017.

M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon. Transfer learning from deep features for remote sensing and poverty mapping. *arXiv preprint arXiv:1510.00098*, 2015.

W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke. The microsoft 2017 conversational speech recognition system. arxiv. *arXiv preprint arXiv:1708.06073*, 2017.

Y. Xue, J. Bai, R. Le Bras, B. Rappazzo, R. Bernstein, J. Bjorck, L. Longpre, S. Suram, B. van Dover, J. Gregoire, and C. Gomes. Phase mapper: An ai platform to accelerate high throughput materials discovery. *Twenty-ninth international conference on innovative applications of artificial intelligence*.

Y. Xue, X. Wu, D. Morin, B. Dilkina, A. Fuller, J. A. Royle, and C. P. Gomes. Dynamic optimization of landscape connectivity embedding spatial-capture-recapture information. In *AAAI*, pages 4552–4558, 2017.

M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi. Deep learning for person re-identification: A survey and outlook. *arXiv preprint arXiv:2001.04193*, 2020.

C. Yun, S. Sra, and A. Jadbabaie. Global optimality conditions for deep neural networks. *arXiv preprint arXiv:1707.02444*, 2017.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 549–553. SIAM, 2006.

Z. Zhang and P. R. White. A blind source separation approach for humpback whale song separation. *The Journal of the Acoustical Society of America*, 2017.

Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pages 337–348. Springer, 2008.

Y. Zhou, J. Yang, H. Zhang, Y. Liang, and V. Tarokh. Sgd converges to global minimum in deep learning via star-convex path. *arXiv preprint arXiv:1901.00451*, 2019.

G. Zhu. Nonnegative matrix factorization (nmf) with heteroscedastic uncertainties and missing data. *arXiv preprint arXiv:1612.06037*, 2016.