Thesis advisors: Professors Boaz Barak and Madhu Sudan Preetum Nakkiran

Towards an Empirical Theory of Deep Learning

Abstract

In this thesis, we take an empirical approach to the theory of deep learning. We treat deep learning systems as black boxes, with inputs we can control (train samples, architecture, model size, optimizer, etc.) and outputs we can observe (the neural network function, its test error, its parameters, etc.). Our goal is to characterize how our choice of inputs affects the outputs. As an empirical theory, we aim to describe this behavior quantitatively, if not prove it rigorously. We hope for theories that are as general and universal as possible, applying in a wide range of deep learning settings, including those in practice.

We present three empirical theories towards this goal. (1) Deep Double Descent demonstrates that the relationship between inputs and outputs in deep learning is not always monotonic in natural ways: there is a predictable "critical regime" where, for example, training on more data can actually hurt performance, but models are well-behaved outside this regime. (2) The Deep Bootstrap Framework shows that to understand the *generalization* of the output network, it is sufficient to understand *optimization* aspects of our input choices. (3) Distributional Generalization takes a closer look at the output network, and finds that trained models actually "generalize" in a much broader sense than we classically expect. We introduce a new kind of generalization to capture these behaviors.

Our results shed light on existing topics in learning theory (especially generalization, overparameterization, interpolation), and also reveal new phenomena which require new frameworks to capture. In some cases, our study of deep learning has exposed phenomena that hold even for non-deep methods. We thus hope the results of this thesis will eventually weave into a general theory of learning, deep and otherwise.

Mathematical theory is not critical to the development of machine learning. But scientific inquiry is.

Leo Breiman, 1995.

1
Introduction

The motivation for this thesis is to develop a theory of deep learning. First, we discuss the definition of deep learning, and why we need a theory. We then describe obstacles to a fully rigorous mathematical theory, and outline our empirical approach.

Briefly, we will take an "empirical science" approach to theory, which is closest to the approach taken in physics. We will treat deep learning as an empirical phenomena– an aspect of Nature that we can observe and experiment with, and whose behavior we want to characterize and understand.

We will try to identify universal phenomena in deep learning in the real world (i.e. in practice). We will try to capture these phenomena in conjectures as formally as possible, and then test these conjectures through experiment. In some cases, our conjectures suggest entirely new phenomena, which we then validate in the real world.

1.1 What is Deep Learning?

There is no formal definition of deep learning (DL) which captures all of its current and potential future instantiations. What the Turing machine did for defining computation, we do not yet have for defining deep learning— and this is one goal of theory. But informally, "deep learning" refers not to a single method, but to a collection of *ingredients* which can be combined in various ways to produce learning systems for certain classes of learning problems. These ingredients typically include certain parametric function families ("neural network architectures"), optimization algorithms (typically Stochastic Gradient Descent and variants), objective functions, and datasets (typically high-dimensional "natural" distributions, such as images or language). We cannot formally specify which sets of ingredients are allowed, which ways they can be combined, or which classes of problems they can solve. However, in practice we have a growing set of "recipes," which are successful ways of combining ingredients that yield desirable behavior in the real world.

This paradigm of deep learning has been enormously successful in practice: it has addressed long-standing learning problems which posed obstacles to prior methods (e.g. in image classification Krizhevsky, Sutskever, and Hinton (2012)), and it has made progress on entirely new types of problems, which we were not even attempting to solve before. Its success is now fairly robust: it is clear that DL is not simply an "accidental" success in a few settings, but rather a versatile toolbox that has hope of solving many real-world learning problems, given enough data and computational resources.

The Need for Theory Despite the tremendous empirical success of deep learning, we are very far from a scientific understanding of the field. The state-of-the-art in deep learning steadily advances, but in unpredictable ways: it is unclear if a new advance will be due to a new architecture, dataset, optimizer, training methodology, etc. And we continue to discover new capabilities and application domains of deep learning— we do not yet know which problems deep learning can or cannot solve. Practitioners and theorists alike are regularly *surprised* by advances in deep learning: cases where changing the *ingredients* lead to unexpected gains in the final system. One goal of theory in deep learning is to eliminate such surprises from the practice of deep learning. In doing so, we expect to gain a more thorough understanding of the nature of computational learning. This motivates our guiding question below.

1.2 Guiding Questions

In developing a theory, we must decide: what is the goal of the theory? What questions should it answer? In this thesis, the following question will guide our work:

"How does what we *do* affect what we *get*?"

"Doing" deep learning involves many moving parts– choices of architecture, dataset, sample size, initialization, loss function, pretraining, etc. At the end, we "get" a trained model, which we can study under various lenses (its test error, its parameters, its robustness, etc). All of the different inputs affect the model produced, but in highly coupled ways, which we do not adequately understand. This is not only a matter of theoretical discomfort: Practitioners are regularly surprised when jointly changing the inputs to deep learning systems lead to unexpected capabilities (e.g. ViT Dosovitskiy, Beyer, Kolesnikov, Weissenborn, Zhai, Unterthiner, Dehghani, Minderer, Heigold, Gelly, Uszkoreit, and Houlsby (2021), GPT-3 Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, Agarwal, Herbert-Voss, Krueger, Henighan, Child, Ramesh, Ziegler, Wu, Winter, Hesse, Chen, Sigler, Litwin, Gray, Chess, Clark, Berner, McCandlish, Radford, Sutskever, and Amodei (2020a), CLIP Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, Krueger, and Sutskever (2021), AlphaFold2 ?).

Existing frameworks from learning theory are essentially attempts to answer the above question in certain settings. For example, consider the "uniform convergence" framework for supervised learning. If we learn with a hypothesis class that satisfies uniform convergence of its error, then we are guaranteed that: no matter what we do, if we output a model with small train error, then it will also have small test error. In this sense, what we *do* on the train set is exactly what we *get* on the test set (w.r.t. error).

Unfortunately, this particular framework does not always apply for deep learning– but the situation is subtle. There are settings in deep learning which are well-captured by "uniform convergence" (e.g. when data is large and models are small). But there are closely related settings where uniform convergence fails severely (e.g. with small data or large models). For example, we can train large *over-parameterized* models which have essentially arbitrary behavior on their train sets, regardless of their test performance. A large network with 0% error on its train set could have anywhere between 0-100% error at test time, depending on what we *do*. This example reflects the tantalizing state of DL theory: existing theories of learning often capture deep learning behaviors in *some* regimes, but fail in others. This motivates the search for better frameworks to reason about deep learning in a unified way[*].

Finally, deep learning is especially interesting because we often *get* much more than we asked for. We simply ask for an empirical-risk-minimizer on the train set, but we get a function with structured internal representations (*representation learning*), which improves sample-complexity on related learning problems (*transfer learning*), and can be combined modularly with other neural networks,

[*] The following works survey the state of the fully rigorous approach to deep learning theory: Bartlett, Montanari, and Rakhlin (2021); Berner, Grohs, Kutyniok, and Petersen (2021); Belkin (2021).

trained in different ways. We will see a new example of getting behaviors "for free" later in this thesis, in Distributional Generalization (Chapter 6).

1.2.1 Objects of Study

For the works presented in this thesis, we will simplify the above guiding question in the following ways:

First, we will consider deep learning only in the context of *supervised classification*. This is both convenient and appropriate: supervised classification was one of the first "breakthrough applications" of deep learning, and thus methods for supervised classification are among the most robust and well-understood in practice. Moreover, supervised learning in practice is the closest setting to problems studied in theory– and so this is a natural first step to bridge theory and practice. Here, we can at least define the problem formally (if not the solution). Finally, many of the other applications or behaviors in DL reduce to supervised learning as a special case (e.g. reinforcement learning, semi-supervised learning, self-supervised learning)– and so we hope that understanding supervised learning is the key to understanding many other settings of deep learning.

Second, we will consider the models we *get* only in terms of their black-box and on-distribution behavior. Deep learning systems output *parameters* θ, which determine a function f_θ. We will only consider the input-output behavior of this function f, and not open the black-box of parameters θ. Moreover, we will not consider arbitrary inputs x, but only inputs $x \sim D$ drawn from the distribution D on which f was trained. This is in some sense the simplest non-trivial level of abstraction, but there is already much to be said about behaviors at this level.

1.3 Our Approach

To address our guiding question, the next step is to decide *what type* of theory is appropriate for deep learning. There are many types of theories in the sciences, from quantitative theories about mathematical objects (e.g. theory of computation, or of differential manifolds), to quantitative theories about the real world (often called "laws" in physics), to qualitative theories about the real world (e.g. evolution, or the central dogma in biology). We will search for a *quantitative* theory about deep learning as it is used in the real world. Many of the objects involved have quantitative descriptions— model architecture, optimizer, test error, etc— and so we hope their interactions can be described quantitatively as well. Among quantitative theories, there are fully rigorous theories, and then there are empirical theories.

1.3.1 Fully Rigorous Theory

The fully rigorous mathematical theory is the "gold standard" of theory: a theorem with assumptions which hold in the real world, and whose conclusions capture behaviors we care about. The field of *compressed sensing* provides an example of such a theory (Candès and Tao, 2006; Tropp and Gilbert, 2007). The theory here states that, *if* real world signals satisfy a certain mathematical property (e.g. approximate sparsity in the Haar wavelet basis), then a certain explicit mathematical procedure (ℓ_1-minimization with Gaussian measurements) will "work" in a formal sense (i.e. will recover the signal, up to quantified error bounds). The sparsity assumption in this theorem can in principle be verified, by making sufficient observations about the real world.

It is tempting to pursue a fully rigorous theory of deep learning. However, this approach quickly runs into obstacles.

1.3.2 Obstacles to a Fully Rigorous Theory

The primary obstacle to a fully rigorous theory is definitional: we cannot yet formally define the methods we use, nor formally define the problems we expect them to solve. It is not even a matter of proofs being too difficult— we are not even able to formally state conjectures which capture real world behaviors.

It is instructive to explicitly encounter these obstacles as we try to formalize a candidate theorem. For example, we could hope to at least describe the existing practical success of deep learning (and perhaps even predict its future successes), via a theorem of the following form:

"Deep neural nets work on natural distributions, when trained on enough data"

To do this formally, we must formally define each object. What do we mean by "deep neural nets"? We mean some family of architectures (MLPs, CNNs, etc) coupled with some family of optimizers (SGD, Adam, etc). Certainly not all architectures can be allowed: Too restrictive a family of architectures is not interesting (we do not want a theory of one particular neural network; we want a general theory of neural networks). And too general a family is also not interesting: Stochastic Gradient Descent on a sufficiently contrived architecture can simulate an arbitrary circuit (Abbe and Sandon, 2020). There is no existing definition of the set of "reasonable architectures"— which should include all the ones that are successful in practice now, and which could be successful in the future.

We encounter similar problems in specifying each object (the set of allowable optimizers, activation functions, loss functions, etc). There are many moving parts in deep learning, and learning systems are robust to many choices of these parts, but we cannot precisely specify which choices are allowed. Moreover, we may not be working at the right level of abstraction: what we think of as "deep learning" today may be a subset of some much larger unifying object which we have not yet discovered (e.g. "locally-optimizable online learning systems").

Next, what do we mean by "natural distributions"? Deep learning is not an appropriate tool for *every* learning problem. Non-deep methods are still used in many places in practice, and theory gives us explicit examples of problems which are "hard" for deep learning but easy for other methods (Shalev-Shwartz, Shamir, and Shammah, 2017). Yet, there is a certain family of problems for which deep methods tend to be more successful than other methods. These often share characteristics such as: high-dimensional inputs, no semantic input coordinates, poorly understood generative processes, and large datasets. Representative examples are many vision problems (image classification, segmentation, generation, etc.) and certain natural language processing (NLP) problems (Krizhevsky et al., 2012; He, Zhang, Ren, and Sun, 2016a; Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio, 2014; Devlin, Chang, Lee, and Toutanova, 2018; Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin, 2017b). But we have no explicit definition of a set of "natural distributions" for which deep learning works. Such a definition, to be relevant, must include both certain image distributions and certain natural-language distributions— two a priori very different kinds of inputs— which highlights the difficulty in definition.

However, just because we cannot yet formally define these objects does not mean they do not exist. The state of practice strongly suggests that some version of the statement "deep learning works" is true, with an appropriate choice of definitions. We have a certain intuitive sense of what a "reasonable architecture" on a "natural distribution" means— and we can allow ourselves to use these informal terms in our conjectures now, in the hopes that they can be formalized in the future.

1.3.3 The Way Forward

In light of the above obstacles, there are essentially two ways to proceed:

1. "Bottom up": We start with a simple mathematical theory which we fully understand (but

is unrealistic), and gradually build up complexity to better approximate the real world. This approach is being pursued by many scientists, and has lead to richer theories of regression, interpolation, kernel methods, non-convex optimization, infinite-width limits, etc. However, such theories are always at risk of describing objects that are detached from the deep learning in the real world— and may or may not have similar behaviors. In fact, we will see several examples where this approach to theory has suggested incorrect intuitions about the real world.

2. "Top Down": We start with the real world, and gradually abstract away pieces of it which we can describe (if not prove) mathematically. This is the empirical theory approach: we attempt to *quantitatively describe* behaviors in the real world, as formally and generally as possible, even if we cannot derive them from first-principles. This approach is underutilized in machine learning, but has a long and fruitful history in physics, where it often leads to "laws" (Kepler's laws, Ampère's law, Boyle's law, etc). Such examples in physics abound, where behaviors that are universal and empirically characterized often lead to deeper understanding, and eventually to unified mechanistic theories.

The former approach lacks realism; the latter lacks mathematical justification. We chose to do the latter, because (1) We are primarily interested in the real world, and (to us) formally describing it is almost as good as having proofs, (2) It is an underutilized approach in the current scientific community, and (3) In our experience, models for deep learning constructed from the "bottom up" approach rarely hold beyond the specific phenomena they were designed to explain. However, models constructed from the "top down" approach often hold far more generally, and can suggest *new* deep learning phenomena beyond what they were designed to capture.

But it is always useful to have an eye for "both ends", in order to eventually join them.

1.4 Our Results

Informally, the three works presented in this thesis address our guiding question by showing:

1. The relationship between what we *do* and what we *get* is not always a continuous and well-behaved one. There is "critical regime" where models become very sensitive to their inputs, and behave poorly— but outside this regime, behavior is monotonic and predictable (Deep Double Descent, Nakkiran, Kaplun, Bansal, Yang, Barak, and Sutskever (2020)).

2. With respect to test error, what we *do* matters essentially only through two factors: its online optimization speed (when training on an infinite stream of fresh samples) and its offline optimization speed (when training to fit a fixed train set). This reduces understanding generalization to understanding certain aspects of optimization (Deep Bootstrap Framework, Nakkiran, Neyshabur, and Sedghi (2021a)).

3. Although what we *do* is designed to minimize a single metric (test error), the models we *get* have much richer structure beyond just their test error. Our work reveals, for example, that even models with high test error can have certain "good behavior" at a coarser scale. These behaviors are captured by a new, broader, definition of generalization (Distributional Generalization, Nakkiran and Bansal (2020)).

1.4.1 Overparameterization and Interpolation

Much of our work will focus on understanding *interpolating* classifiers– that is, classifiers which fit their train sets exactly (with 0% train error). This setting presents perhaps the largest gap in our knowledge, since it is in conflict with the spirit of many existing theories of learning. Many theories (including empirical risk minimization, and kernel methods) suggest that we need some form of "capacity control" or "regularization" in order to learn. Without such explicit regularization, learning methods may “overfit” and perform poorly, especially in the presence of label noise (Belkin, Ma, and Mandal, 2018b; Bartlett et al., 2021). In contrast, deep neural nets can be trained to interpolation, without explicit regularization, and still achieve good test performance (Zhang, Bengio, Hardt, Recht, and Vinyals, 2017b; Belkin, Hsu, Ma, and Mandal, 2019a). The issues of overparameterization and interpolation are related, since interpolation is often only possible with large overparameterized networks.

This disconnect motivates a rich body of work on overparameterized and interpolating methods, including the “implicit bias” and “benign overfitting” programs (Zhang et al., 2017b; Belkin, Hsu, and Mitra, 2018a; Belkin et al., 2018b, 2019a; Liang and Rakhlin, 2018; Hastie, Montanari, Rosset, and Tibshirani, 2019; Schapire, Freund, Bartlett, Lee, et al., 1998; Breiman, 1995; Mei and Montanari, 2019; Gunasekar, Lee, Soudry, and Srebro, 2018a; Soudry, Hoffer, Nacson, and Srebro, 2018; Bartlett, Long, Lugosi, and Tsigler, 2020). It motivates us as well. In this context, the works in this thesis will:

1. Highlight “pathologies” of models when passing between underparameterized and overparameterized regimes (Deep Double Descent, Nakkiran et al. (2020)).

2. Demonstrate a close connection between underparameterized and overparameterized regimes (Deep Bootstrap Framework, Nakkiran et al. (2021a)). In a certain sense, models which are overparameterized (fitting their finite train sets) “behave as though” they were underparameterized (trained on effectively infinite data). This converts questions of “implicit bias” into questions about *explicit* properties of optimization.

3. Demonstrate a significant *difference* between underparameterized and overparameterized regimes (Distributional Generalization, Nakkiran and Bansal (2020)). Along the way, we will show realistic settings where interpolation is *not* “benign,” but rather “malignant” in predictable ways. In particular, our conjectures imply that interpolating neural networks are

not consistent in settings with nonzero Bayes risk.

1.5 Organization

In the remainder of this thesis, we will focus only on the three representative papers above.

We will first describe necessary background and notation (Chapter 2). Then we briefly summarize the contributions of these papers (Chapter 3), in relation to our guiding question and areas of overparameterization & interpolation. The three subsequent chapters are based on the content of each of the papers. We conclude with reflections and open questions.

1.5.1 Relations to Published Work

Chapter 4 is based on:

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I. (2020). Deep Double Descent: Where bigger models and more data hurt. *In International Conference on Learning Representations.*

Chapter 5 is based on:

Nakkiran, P., Neyshabur, B., Sedghi, H. (2021). The deep bootstrap framework: Good online learners are good offline generalizers. *In International Conference on Learning Representations.*

Chapter 6 is based on:

Nakkiran, P. & Bansal, Y. (2020). Distributional generalization: A new kind of generalization. *ArXiv, abs/2009.08092.*

7

Conclusion

In this thesis, we have developed several empirical laws about deep learning “in the wild.” Our investigations shed light on existing questions in learning theory (generalization, overparameterization, interpolation), and also revealed entirely new behaviors which required new frameworks to state (Distributional Generalization).

Deep Double Descent taught us that even modern networks may have pathological behavior in a “critical regime”, but are well-behaved outside of it. The Deep Bootstrap demonstrated that although we train models on finite train sets in practice, they behave as if we trained them on an infinite stream of samples. And Distributional Generalization implies that although we think we are training *classifiers*, the objects we get are better thought of as *samplers*.

We hope these results will be helpful as conceptual tools: they abstract away the complexity of "deep learning" down to its essential objects, and demonstrate robust behaviors that occur even at this high level of abstraction. Abstracting away all irrelevant details makes our theory both simpler and plausibly more universal: we hope our results hold not only for the neural networks of today, but for whatever "deep learning" means 10+ years from now.

In fact, we hope our results will eventually weave into a general theory of *learning*, beyond just deep learning. This has already happened to some extent: double descent was previously observed to hold in many non-deep learning systems as well (Belkin et al., 2019a). And in Distributional Generalization, the conjectures we derived to explain behaviors of neural networks turned out to also apply to other methods (decision trees and kernels) – revealing new behaviors of these methods. We hope that the Deep Bootstrap Framework will be similarly universal: instead of applying only to deep networks trained via SGD or variants, it may apply generically anytime a "reasonable" online optimizer is used in an offline setting.

It is remarkable that it is even possible to state claims that hold for deep networks just as well as for decision trees— two a priori very different methods, used on very different kinds of tasks. This gives us hope that there exists a unified theory of learning, which captures important behaviors of many modern models – deep or otherwise.

What can we learn, deeply or otherwise?

7.1 Open Questions

Here we give a partial list of open questions that we believe may be relevant to the future of deep learning theory.

1. What distinguishes deep learning from other methods? Why was deep learning so successful in settings where prior methods were not? Are they information-theoretic factors (e.g. sample complexity), computational factors (e.g. optimization time and space), or other factors (e.g. ability to integrate auxiliary data, representation learning, etc).
2. Is there a "minimal set of assumptions" for deep learning theory? Is there some set of conjectures/laws/assumptions which, if we assume, would explain all other phenomena in deep learning? (Can we do for deep learning theory what one-way-functions did for cryptography theory?)
3. What learning methods would we use if we relax certain constraints: If we had infinite computation time, or infinite space, or infinite samples? Would we still use deep learning in its current form?
4. Is there an "axiomatic definition" of deep learning? E.g. can deep learning be defined as "the unique system which satisfies properties X, Y, Z" for certain values of X, Y, Z? (These properties may be related to online learning, robustness, ability to incorporate auxiliary data, etc.)
5. Can we formally define broader notions of learning or generalization, which capture the observed behavior of large models such as GPT-3 and CLIP? These behaviors go beyond the classical notions of generalization, and even beyond our notion of Distributional Generalization.
6. In deep learning, we often get "more than we asked for." That is, we only minimize an on-distribution error/loss, but we get networks with many other interesting properties (from good internal representations, to unexpected kinds of off-distribution generalization, etc).

How should we think of these auxiliary behaviors, and how can we predict them?

7. What is *representation learning* formally? Why do trained deep networks have structured internal representations, and what is this structure?
8. Why can deep learning systems often be effectively *composed* with other learning systems (deep and non-deep)?
9. The "definitional question": Can we formally define "deep learning" in a way that captures all current and future evolutions of the term? (But not too broad a definition to be trivial).
10. The "natural distributions question": Can we formally understand the set of tasks for which deep learning systems "work"?