Relationship between sample size and architecture for the estimation of Sobolev functions using deep neural networks

Stéphane Chrétien

University of Lyon 2
ERIC Laboratory

MLOMA, Clermont Ferrand 21 décembre 2023

Introduction

- Modern machine learning and statistics deal with the problem of learning from data:
  - given a training dataset (y_i, x_i) $i \in I$ where
    - $x_i \in \mathbb{R}^d$ is the input
    - $y_i \in \mathbb{R}$ is the output,

    one seeks a function $f : \mathbb{R}^d \mapsto \mathbb{R}$ from a certain function class $\mathcal{F}$ that has good prediction performance on test data (y_t, x_t), $t \in T$, i.e. which has small testing error

$$\sum_{t \in T} \ell(y_t, f(x_t)) \tag{1}$$

- For this purpose, we often solve the (possibly penalised) following problem on the training dataset

$$\sum_{i=1}^{n} \ell(y_t, f(x_t)) + pen(f) \tag{2}$$

and hope that the estimator $\hat{f}$ will generalise well on unobserved data.

- This problem is of fundamental significance and finds applications in numerous scenarios.
  - For instance, in image recognition,
    - the input x corresponds to the raw image
    - the output y is the image category

    and the goal is to find a mapping f that can classify new images with acceptable accuracy.
- Decades of research efforts in statistical machine learning have been devoted to developing methods to find f efficiently with provable guarantees.

- Prominent examples include
  - linear classifiers (e.g., linear / logistic regression, linear discriminant analysis),
  - kernel methods (e.g., support vector machines),
  - tree-based methods (e.g., decision trees, random forests),
  - nonparametric regression (e.g., nearest neighbors, local kernel smoothing), etc.
- Roughly speaking, each aforementioned method corresponds to a different function class $\mathcal{F}$ from which the final classifier f is chosen.

- Deep learning, in its simplest form, consists in looking for functions of the form

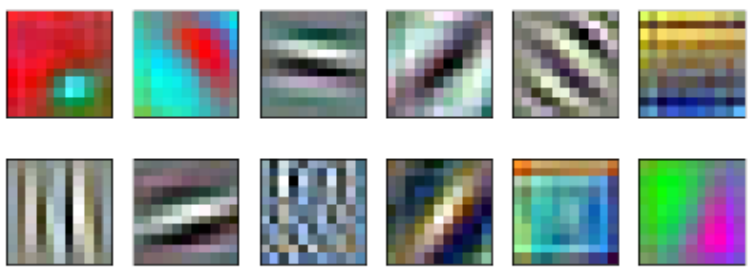
$$\mathcal{F} = \left\{ f(x, \theta) = W_L(\sigma_L(W_{L-1}(\sigma_{L-1}(\cdots \sigma_2(W_1(x)))))) \right\}.$$

  where σ_l is a non-linear function which applies componentwise and W_l is an affine operator, $l = 1, \ldots, L$.

- Note that this general architecture does not work without specific tweaks and tricks

  (convolutions, initialisation, dropout, batch normalisation, layerwise normalisation, etc . . .).

- Deep learning is able to approximate complicated nonlinear maps through composing many simple nonlinear functions.
- The motivation for the multilayer architecture is that there are different levels of features and the layers might be able to properly account for these different levels independently.
- Here, we sample and visualize weights from a pre-trained AlexNet model.



- This can be used to generate new images using for instance, Generative Adversarial Networks or Diffusion models.

- Evolution of the performances for "old" architectures . . .

Model	Year	# Layers	# Params	Top-5 error
Shallow	< 2012	—	—	$> 25\%$
AlexNet	2012	8	61M	16.4%
VGG19	2014	19	144M	7.3%
GoogleNet	2014	22	7M	6.7%
ResNet-152	2015	152	60M	3.6%

- It is widely acknowledged that two indispensable factors contribute to the success of deep learning, namely
  - huge datasets that often contain millions of samples and
  - immense computing power resulting from clusters of graphics processing units (GPUs).
- Admittedly, these resources are only recently available.

- However, these two alone are not sufficient to explain the mystery of deep learning:
  - Why is over-parametrization not a problem ?
    - overparametrisation should lead to overfitting,

    - BUT ... this is not what we always observe in practice !

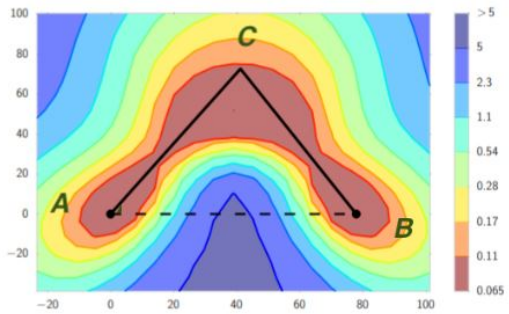
- and
  - nonconvexity does not seem to be a problem: even with the help of GPUs, training deep learning models is still NP-hard in the worst case due to the highly nonconvex loss function to minimize.
    - Nevertheless, standard incremental algorithms (Stochastic Gradient Descent, etc) often reach good minimisers of the Empirical Risk
- A lot remains to be understood ! ...

Why overparametrise, to begin with ?

Why overparametrise, to begin with ?

- It is often observed that depth helps efficiently extract features at different scales from the inputs,
- recent studies found that *increasing both depth and width in a shallow model leads to very nice continuous limits, where PDE tools can be put to work...*
- Networks with wide layers (larger than sample size) enjoy connectivity of the minimisers (Nguyen 2019)

> See Figure 1 below from Garipov et al. 2018 for an illustration. Solutions A and B have low cost but the line connecting them goes through solutions with high cost. But we can find C of low cost such that paths AC and CB only pass through low-cost region.



Why overparametrise, to begin with ?

- Recent interesting work by Bubeck and Sellke states that overparametrisation is key to robustness

---

A Universal Law of Robustness via Isoperimetry

Sébastien Bubeck
Microsoft Research
sebubeck@microsoft.com

Mark Sellke
Stanford University
msellke@stanford.edu

Abstract

Classically, data interpolation with a parametrized model class is possible as long as the number of parameters is larger than the number of equations to be satisfied. A puzzling phenomenon in deep learning is that models are trained with many more parameters than what this classical theory would suggest. We propose a theoretical explanation for this phenomenon. We prove that for a broad class of data distributions and model classes, overparametrization is *necessary* if one wants to interpolate the data *smoothly*. Namely we show that *smooth* interpolation requires d times more parameters than mere interpolation, where d is the ambient data dimension. We prove this universal law of robustness for any smoothly parametrized function class with polynomial size weights, and any covariate dis-

Potentially bad consequences of overparametrisation ?

What are the bad consequences of overparametrisation ?

- When some of the layers are not wide, *over-parametrization* usually entails existence of many local minimisers with potentially different statistical performance.
  - Common practice advises to runs stochastic gradient descent with random initialization and converges to parameters with *very good practical prediction accuracy*.
    - Why is this simple approach actually often working ?
- Overfitting should take place in full generality
  - Does the optimisation algorithm help find better networks ?

The goal of current research is to resolve these paradoxes !

Generalisation bounds

- "Old" but we interesting work by Bartlett, Barron and others ...

Theorem (B., Foster, Telgarsky, 2017)

With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every f_W with $R_W \leq r$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

Here, f_W is computed in a network with L layers and parameters $W_1, \ldots, W_L$:

$$f_W(x) := \sigma_L(W_L \sigma_{L-1}(W_{L-1} \cdots \sigma_1(W_1 x) \cdots)),$$

where the σ_i are 1-Lipschitz, and we measure the scale of f_W using a product of norms of the matrices W_i,
for example, $r := \prod_{i=1}^{L} \|W_i\|_* \left(\sum_{i=1}^{L} \frac{\|W_i\|_{2,1}^{2/3}}{\|W_i\|_*^{2/3}}\right)^{3/2}$.

- New trends involve PAC-Bayes bounds and compression

Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data

Gintare Karolina Dziugaite
Department of Engineering
University of Cambridge

Daniel M. Roy
Department of Statistical Sciences
University of Toronto

Abstract

One of the defining properties of deep learning is that models are chosen to have many more parameters than available training data. In light of this capacity for overfitting, it is remarkable that simple algorithms like SGD reliably return solutions with low test error. One roadblock to explaining these phenomena in for trained neural networks in the modern deep learning regime where the number of network parameters eclipses the number of training examples.

The bounds we compute are data dependent, incorporating millions of components optimized numerically to identify a large region in weight space with low average empirical error around the solution obtained by stochastic gradient descent (SGD). The data dependence is essential: indeed, the [illegible]nension of neural networks is typically bounded

- New trends involve PAC-Bayes bounds and compression

Stronger generalization bounds for deep nets via a compression approach

Sanjeev Arora* Rong Ge† Behnam Neyshabur‡ Yi Zhang§

Abstract

Deep nets generalize well despite having more parameters than the number of training samples. Recent works try to give an explanation using PAC-Bayes and Margin-based analyses, but do not as yet result in sample complexity bounds better than naive parameter counting. The current paper shows generalization bounds that're orders of magnitude better in practice. These rely upon new succinct reparametrizations of the trained net — a compression that is explicit and efficient. These yield generalization bounds via a simple compression-based framework introduced here. Our results also provide some theoretical justification for widespread empirical success in compressing deep nets.

Analysis of correctness of our compression relies upon some newly identified "noise stability"properties of trained deep nets, which are also experimentally verified. The study of these properties and resulting generalization bounds are also extended to convolutional nets, which had eluded earlier attempts on proving generalization.

- New trends involve PAC-Bayes bounds and compression

Generalization Bounds: Perspectives from Information Theory and PAC-Bayes

Fredrik Hellström
University College London
f.hellstrom@ucl.ac.uk

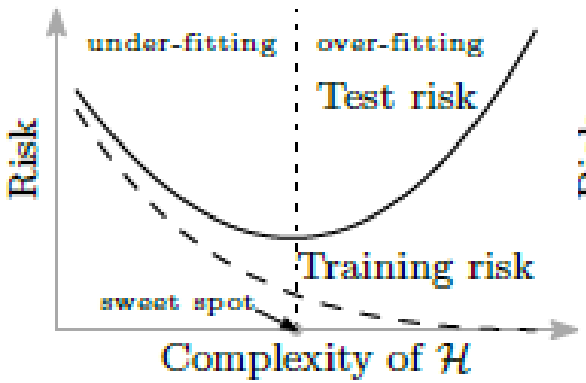
Giuseppe Durisi
Chalmers University of Technology
durisi@chalmers.se

Benjamin Guedj
Inria and University College London
benjamin.guedj@inria.fr

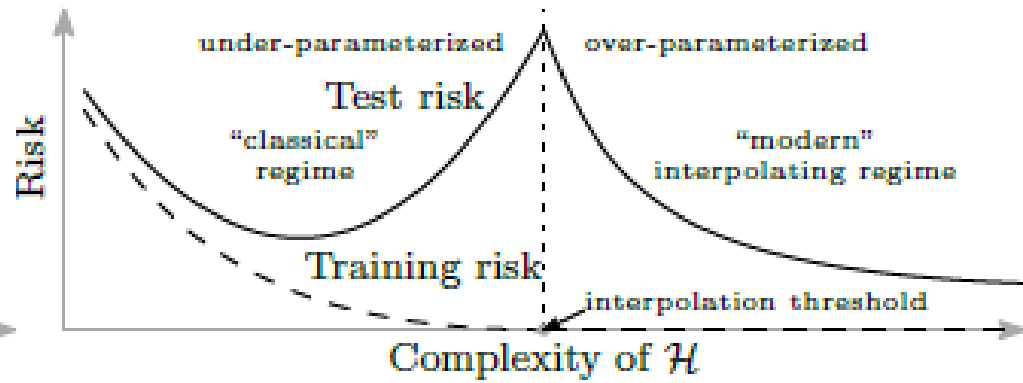
Maxim Raginsky
University of Illinois

Double Descent and Benign overfitting

- Overparametrisation might work although it contradicts the intuition that "overfitting hurst generalisation"
- Now, one often speaks of "interpolation" and one hopes that it does not hurt generalisation;
- When this holds, one speaks of Benign Overfitting.



(a) U-shaped "bias-variance" risk curve



(b) "double descent" risk curve

- Montanari et al. resolved this paradox ... for the linear model ! (Uses a lot of random matrix theory in the asymptotic regime)

Surprises in High-Dimensional Ridgeless Least Squares Interpolation

Trevor Hastie Andrea Montanari Saharon Rosset Ryan J. Tibshirani

Abstract

Interpolators—estimators that achieve zero training error—have attracted growing attention in machine learning, mainly because state-of-the art neural networks appear to be models of this type. In this paper, we study minimum ℓ_2 norm ("ridgeless") interpolation in high-dimensional least squares regression. We consider two different models for the feature distribution: a linear model, where the feature vectors $x_i \in \mathbb{R}^p$ are obtained by applying a linear transform to a vector of i.i.d. entries, $x_i = \Sigma^{1/2} z_i$ (with $z_i \in \mathbb{R}^p$); and a nonlinear model, where the feature vectors are obtained by passing the input through a random one-layer neural network, $x_i = \varphi(W z_i)$ (with $z_i \in \mathbb{R}^d$, $W \in \mathbb{R}^{p \times d}$ a matrix of i.i.d. entries, and φ an activation function acting componentwise on $W z_i$). We recover—in a precise quantitative way—several phenomena that have been observed in large-scale neural networks and kernel machines, including the "double descent" behavior of the prediction risk, and the potential benefits of overparametrization.

1 Introduction

Modern deep learning models involve a huge number of parameters. In nearly all applications of these models, current practice suggests that we should design the network to be sufficiently complex so that the model (as trained, typically, by gradient descent) interpolates the data, i.e., achieves zero training error. Indeed, in a thought-provoking experiment, Zhang et al. (2016) showed that state-of-the-art deep neural network architectures can be trained to interpolate the data even when the actual labels are replaced by entirely random ones.

Despite their enormous complexity, deep neural networks are frequently seen to generalize well, in meaningful practical problems. At first sight, this seems to defy conventional statistical wisdom: interpolation (vanishing training

- Benign overfitting already occurs in traditional statistics

Does data interpolation contradict statistical optimality?

Mikhail Belkin
The Ohio State University

Alexander Rakhlin
MIT

Alexandre B. Tsybakov
CREST, ENSAE

Abstract

We show that learning methods interpolating the training data can achieve optimal rates for the problems of nonparametric regression and prediction with square loss.

1 Introduction

In this paper, we exhibit estimators that interpolate the data, yet achieve optimal rates of convergence for the problems of nonparametric regression and prediction with square loss. This curious observation goes against the usual (or, folklore?) intuition that a good statistical procedure should forego the exact fit to data in favor of a more smooth representation. The family of estimators we consider do exhibit a bias-variance trade-off with a tuning parameter, yet this "regularization" co-exists in harmony with data interpolation.

Motivation for this work is the recent focus within the machine learning community on the out-of-sample performance of neural networks. These flexible models are typically trained to fit the data exactly (either in their sign or in the actual value), yet they predict well on unseen data. The conundrum has served both as a source of excitement about the "magical" properties of neural networks, as well as a call for the development of novel statistical techniques to resolve it.

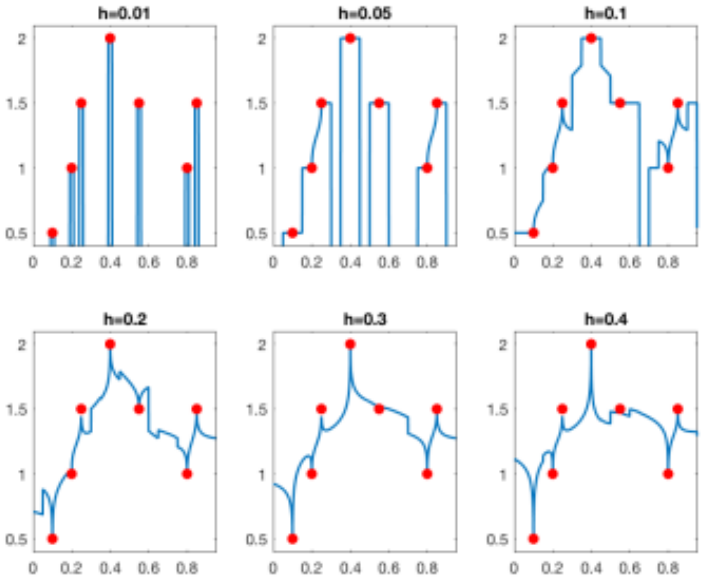
The aim of this short note is to show that not only can interpolation be a good statistical procedure, but it can even be optimal in a minimax sense. To the best of our knowledge, such optimality has not been exhibited before.

Let (X, Y) be a random pair on $\mathbb{R}^d \times \mathbb{R}$ with distribution P_{XY}, and let $f(x) = \mathbb{E}[Y|X = x]$ be the regression function. A goal of nonparametric estimation is to construct an estimate f_n of f, given a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ drawn independently from P_{XY}. A classical approach to

- Benign overfitting already occurs in traditional statistics

The Nadaraya-Watson estimator for a singular kernel K is defined as

$$f_n(x) = \begin{cases} Y_i & \text{if } x = X_i \text{ for some } i = 1, \ldots, n, \\ 0 & \text{if } \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) = 0, \\ \frac{\sum_{i=1}^n Y_i K\left(\frac{x - X_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right)} & \text{otherwise.} \end{cases}$$



Interpolation with $K(u) = \|u\|^{-a} \mathbf{I}\{\|u\| \leq 1\}$, $a = 0.49$, and various values of h.

Figure: Singular Kernel estimators that interpolate !

Interpolation and statistics

The Nadaraya-Watson estimator for a singular kernel K is defined as

$$f_n(x) = \begin{cases} Y_i & \text{if } x = X_i \text{ for some } i = 1, \ldots, n, \\ 0 & \text{if } \sum_{i=1}^n K\left(\frac{x-X_i}{h}\right) = 0, \\ \frac{\sum_{i=1}^n Y_i K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right)} & \text{otherwise.} \end{cases}$$

Theorem (Belkin, Rahklin and Tsybakov (2019))

The Nadaraya Watson estimator achieves the minimax pointwise risk over certain Holder classes. (And thus, you cannot do essentially better !)

What do we know about the bassin of attraction of deep learning minimisers ?

- Flat minimisers (borrowed from T. Goldstein)

Understanding Generalization through Visualizations

W. Ronny Huang
University of Maryland
wrhuang@umd.edu

Zeyad Emam
University of Maryland
zeyad@math.umd.edu

Micah Goldblum
University of Maryland
goldblum@math.umd.edu

Liam Fowl
University of Maryland
lfowl@math.umd.edu

Justin K. Terry
University of Maryland
justinkterry@gmail.com

Furong Huang
University of Maryland
furongh@cs.umd.edu

Tom Goldstein
University of Maryland
tomg@cs.umd.edu

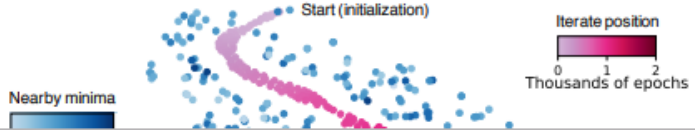
Abstract

The power of neural networks lies in their ability to generalize to unseen data, yet the underlying reasons for this phenomenon remain elusive. Numerous rigorous attempts have been made to explain generalization, but available bounds are still quite loose, and analysis does not always lead to true understanding. The goal of this work is to make generalization more intuitive. Using visualization methods, we discuss the mystery of generalization, the geometry of loss landscapes, and how the curse (or, rather, the blessing) of dimensionality causes optimizers to settle into minima that generalize well.

1 Introduction

Neural networks are a powerful tool for solving classification problems. The power of these models is due in part to their expressiveness; they have many parameters that can be efficiently optimized to fit nearly any finite training set. However, the real power of neural network models comes from their ability to *generalize;* they often make accurate predictions on test data that were not seen during training, provided the test data is sampled from the same distribution as the training data.

The generalization ability of neural networks is seemingly at odds with their expressiveness. Neural network training algorithms work by minimizing a loss function that measures model performance using only training data. Because of their flexibility, it is possible to find parameter configurations



- Flat minimisers (borrowed from T. Goldstein)

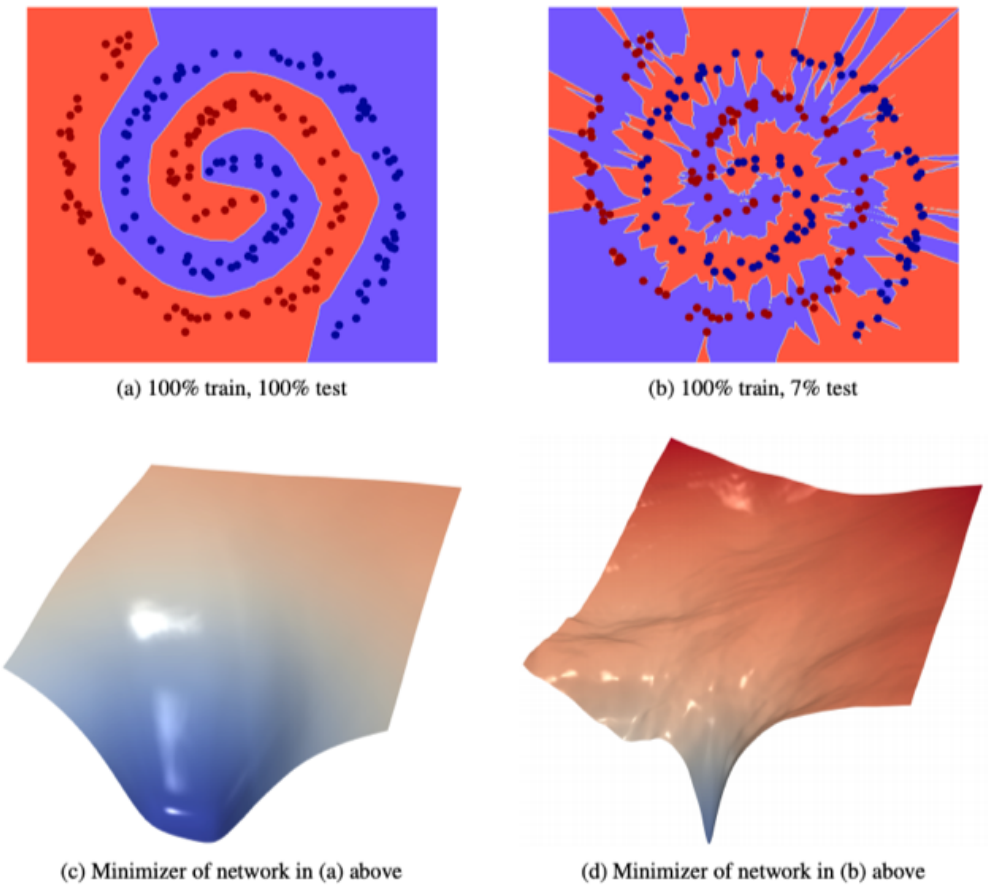


Figure: Flat minimiser (left) and steep minimiser (right)

Flat minimisers (borrowed from T. Goldstein)

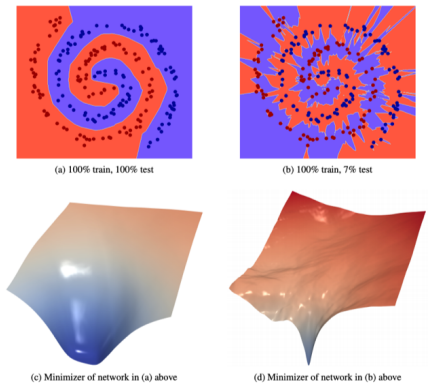


Figure: Flat minimiser (left) and steep minimiser (right)

Empirical discovery

On a set of experimental problems Tom Goldstein and collaborators have empirically discovered that the basins surrounding good minima have a volume at least 10,000 orders of magnitude larger than that of bad minima (!!), rendering it impossible to stumble upon bad minima in practice..

Do local optimisers communicate with each other ?

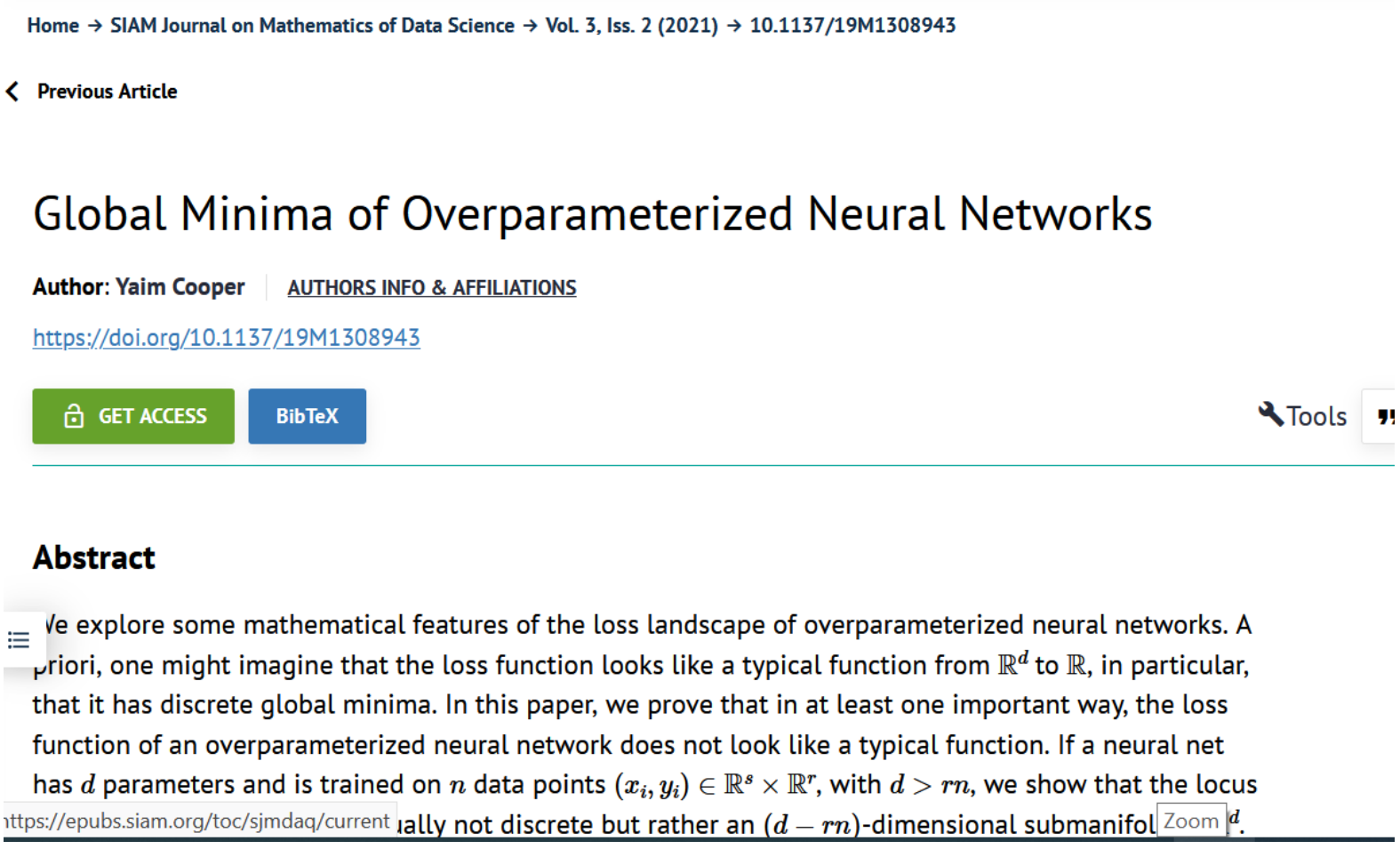


Figure: A neat result by Cooper in SIAM Journal Data Science

Landscape connectivity (borrowed from blog post by R. Ge)

- A big mystery about deep learning is how, in a highly nonconvex loss landscape, gradient descent often finds near-optimal solutions those with training cost almost zero-even starting from a random initialization.

- This can be explained by this very counterintuitive phenomenon:

Empirical discovery

(Freeman and Bruna, 2016, Garipov et al. 2018, Draxler et al. 2018) All pairs of low-cost solutions found via gradient descent can actually be connected by simple paths in the parameter space, such that every point on the path is another solution of almost the same cost. In fact the low-cost path connecting two near-optima can be piecewise linear with two line-segments, or a Bezier curve.

Landscape connectivity (borrowed from blog post by R. Ge)

Explaining Landscape Connectivity of Low-cost Solutions for Multilayer Nets

Rohith Kuditipudi
Duke University
rohith.kuditipudi@duke.edu

Xiang Wang
Duke University
xwang@cs.duke.edu

Holden Lee
Princeton University
holdenl@princeton.edu

Yi Zhang
Princeton University
y.zhang@cs.princeton.edu

Zhiyuan Li
Princeton University
zhiyuanli@cs.princeton.edu

Wei Hu
Princeton University
huwei@cs.princeton.edu

Sanjeev Arora
Princeton University and Institute for Advanced Study
arora@cs.princeton.edu

Rong Ge
Duke University
rongge@cs.duke.edu

Abstract

Mode connectivity (Garipov et al., 2018; Draxler et al., 2018) is a surprising phenomenon in the loss landscape of deep nets. Optima—at least those discovered by gradient-based optimization—turn out to be connected by simple paths on which the loss function is almost constant. Often, these paths can be chosen to be piece-wise linear, with as few as two segments.

We give mathematical explanations for this phenomenon, assuming generic properties (such as dropout stability and noise stability) of well-trained deep nets, which have previously been identified as part of

Landscape connectivity (borrowed from blog post by R. Ge)

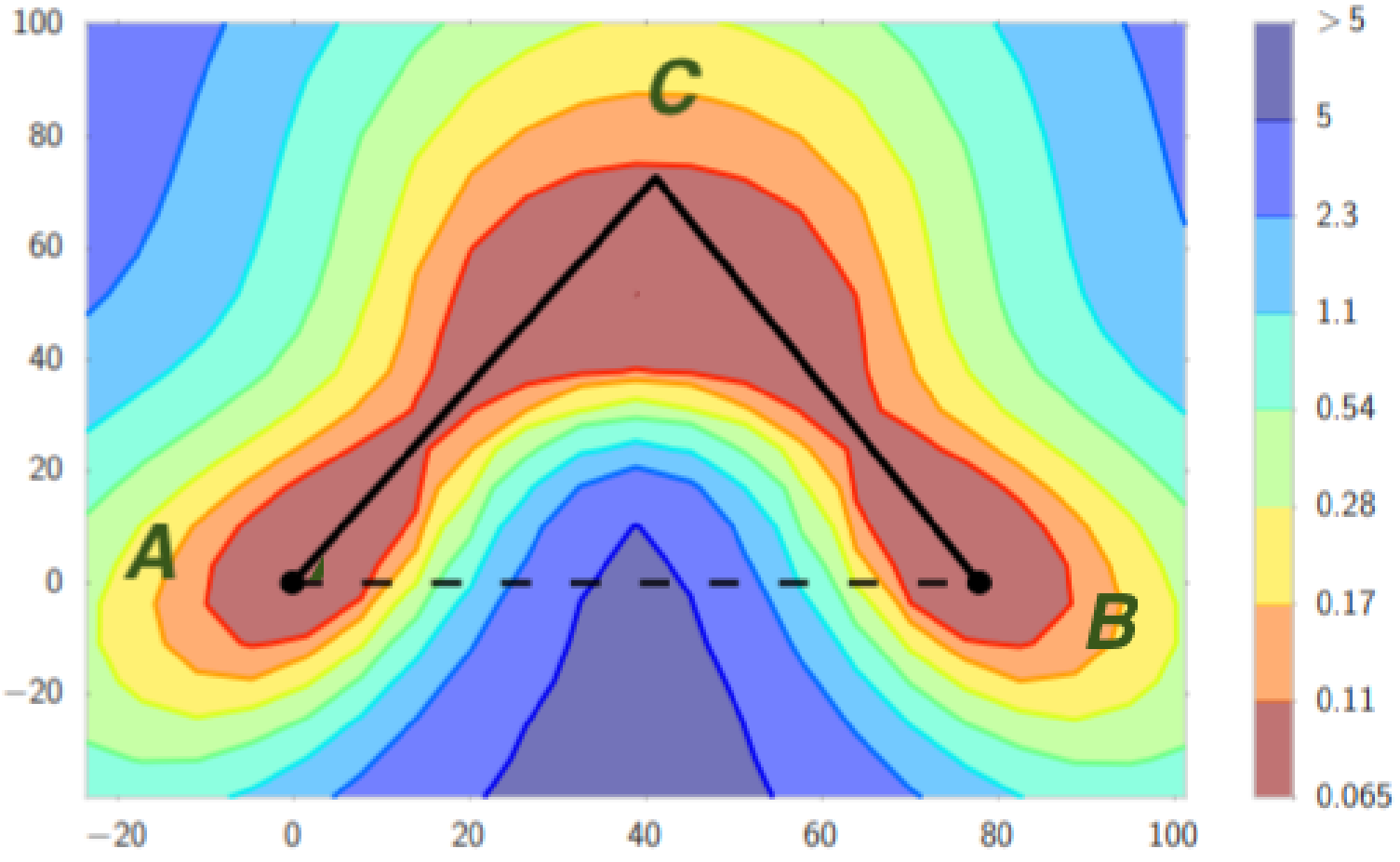


Figure: Landscape connectivity !

Landscape connectivity

Theorem

If two trained multilayer ReLU nets with the same architecture are ϵ-dropout stable, then they can be connected in the loss landscape via a piece-wise linear path in which the number of linear segments is linear in the number of layers, and the loss of every point on the path is at most ϵ higher than the loss of the two end points.

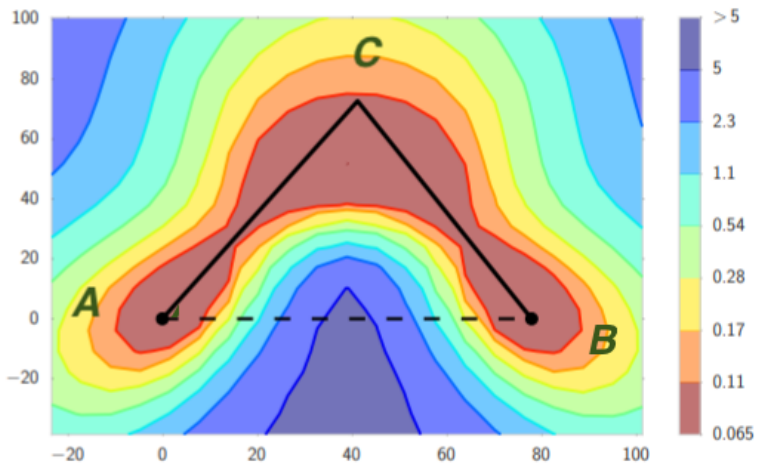


Figure: Connectivity via dropout-stability !

Landscape connectivity (borrowed from R. Ge's blog post)

Theorem

If two trained multilayer ReLU nets with the same architecture are ϵ-noise stable, then they can be connected in the loss landscape via a piece-wise linear path with at most 10 segments, and the loss of every point on the path is at most ϵ higher than the loss of the two end points.

Striking properties of stochastic gradient descent

- For some time, width was thought to only help simplify the analysis because minimizers are connected

- For some time, width was thought to help simplify the analysis because minimizers are connected
- . . . but more is actually happening ! An asymptitical phenomenon shows that :

Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent

---

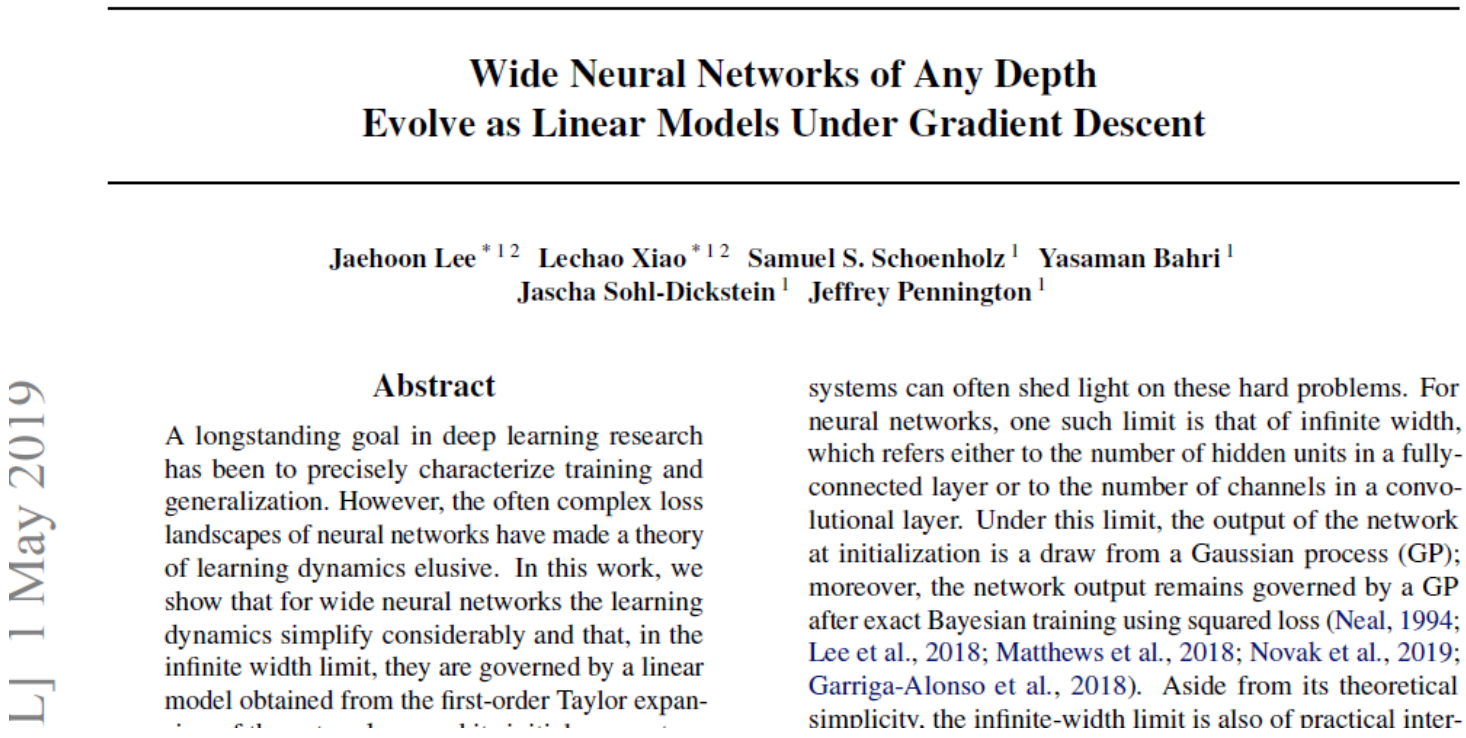
Jaehoon Lee [*1 2] Lechao Xiao [*1 2] Samuel S. Schoenholz [1] Yasaman Bahri [1]
Jascha Sohl-Dickstein [1] Jeffrey Pennington [1]

Abstract

A longstanding goal in deep learning research has been to precisely characterize training and generalization. However, the often complex loss landscapes of neural networks have made a theory of learning dynamics elusive. In this work, we show that for wide neural networks the learning dynamics simplify considerably and that, in the infinite width limit, they are governed by a linear model obtained from the first-order Taylor expan-

systems can often shed light on these hard problems. For neural networks, one such limit is that of infinite width, which refers either to the number of hidden units in a fully-connected layer or to the number of channels in a convolutional layer. Under this limit, the output of the network at initialization is a draw from a Gaussian process (GP); moreover, the network output remains governed by a GP after exact Bayesian training using squared loss (Neal, 1994; Lee et al., 2018; Matthews et al., 2018; Novak et al., 2019; Garriga-Alonso et al., 2018). Aside from its theoretical simplicity, the infinite-width limit is also of practical inter-

- . . . but more is actually happening ! An asymptotical phenomenon shows that :



Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent

Jaehoon Lee [*1 2] Lechao Xiao [*1 2] Samuel S. Schoenholz [1] Yasaman Bahri [1]
Jascha Sohl-Dickstein [1] Jeffrey Pennington [1]

Abstract

A longstanding goal in deep learning research has been to precisely characterize training and generalization. However, the often complex loss landscapes of neural networks have made a theory of learning dynamics elusive. In this work, we show that for wide neural networks the learning dynamics simplify considerably and that, in the infinite width limit, they are governed by a linear model obtained from the first-order Taylor expan-

systems can often shed light on these hard problems. For neural networks, one such limit is that of infinite width, which refers either to the number of hidden units in a fully-connected layer or to the number of channels in a convolutional layer. Under this limit, the output of the network at initialization is a draw from a Gaussian process (GP); moreover, the network output remains governed by a GP after exact Bayesian training using squared loss (Neal, 1994; Lee et al., 2018; Matthews et al., 2018; Novak et al., 2019; Garriga-Alonso et al., 2018). Aside from its theoretical simplicity, the infinite-width limit is also of practical inter-

- Gradient descent is then shown to converge exponentially quickly to a closeby interpolator.

- Width helps simplify the analysis
- Recent works of Jaquot et al. says it then reduces reduces Deep Neural Networks to a tangent kernel method whose feature map corresponds to the gradient of the network at a random initialisation . . .

Neural Tangent Kernel:
Convergence and Generalization in Neural Networks

Arthur Jacot
École Polytechnique Fédérale de Lausanne
arthur.jacot@netopera.net

Franck Gabriel
Imperial College London and École Polytechnique Fédérale de Lausanne
franckrgabriel@gmail.com

Clément Hongler
École Polytechnique Fédérale de Lausanne
clement.hongler@gmail.com

- Width helps simplify the analysis
- Recent works of Jaquot et al. Chizat, Oyallon and Bach says that this phenomenon of exponential speed is a phenomenon that takes place in a much larger setting . . .

On Lazy Training in Differentiable Programming

Lénaïc Chizat
CNRS, Université Paris-Sud
Orsay, France
lenaic.chizat@u-psud.fr

Edouard Oyallon
CentraleSupelec, INRIA
Gif-sur-Yvette, France
edouard.oyallon@centralesupelec.fr

Francis Bach
INRIA, ENS, PSL Research University
Paris, France
francis.bach@inria.fr

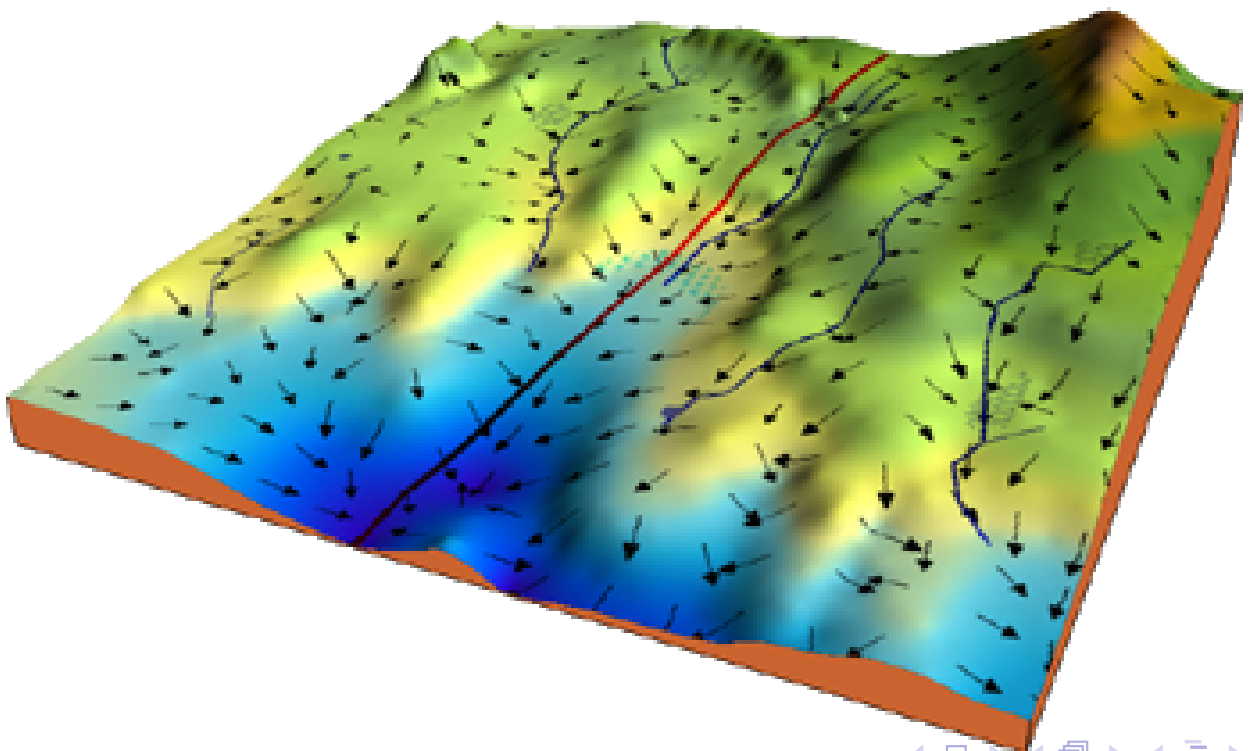
Abstract

In a series of recent theoretical works, it was shown that strongly over-parameterized neural networks trained with gradient-based methods could converge exponentially fast to zero training loss, with their parameters hardly varying. In this work, we show that this "lazy training" phenomenon is not specific to over-parameterized neural networks, and is due to a choice of scaling, often implicit, that makes the model behave as its linearization around the initialization, thus

Other interesting property of gradient descent : implicit bias

- Implicit biais of gradient descent
  - For minimising a function $F(\theta)$, one can use the gradient method :

$$\theta^{(l+1)} = \theta^{(l)} - \eta_l \, \nabla F(\theta^{(l)}) \tag{3}$$



- if there is a unique global minimizer θ_*, then the goal of optimization algorithms is to find this minimizer,

- when there are multiple minimizers (thus for a function which cannot be strongly convex), one can easily show that

$$F(\theta_t) - \inf_{\theta \in \mathbb{R}^d} F(\theta) \tag{4}$$

is converging to zero.

Implicit biais of gradient descent

- With some extra assumptions, we can show that the algorithm is converging to one of the multiple minimizers of F
  - note that when F is convex, this set is also convex.
- But . . . which one ?

Implicit biais of gradient descent

- This is what is referred to as the implicit regularization property of certain optimization algorithms, and in particular, gradient descent and its variants.
    - This is interesting in overparametrised machine learning because there usually are many minimizers
- In a nutshell, gradient descent usually leads to minimum ℓ_2-norm solutions.
    - This shows that the chosen empirical risk minimizer is not arbitrary !

A slightly more general nonlinear regression setup: ridge functions
(work with Emmanuel Caron, Univ. Avignon, France)

Mathematical Model

Let $Z_i = (X_i, Y_i)$ in $\mathbb{R}^{d+1} \times \mathbb{R}$, $i = 1, \ldots, n$ be observations drawn from the following model

$$Y_i = f^*(X_i) + \varepsilon_i \tag{5}$$

$i = 1, \ldots, n$, where we assume that

- the vectors X_i, $i = 1, \ldots, n$ are random and i.i.d., taking values in $\mathbb{R}^d$
- the values ε_i, $i = 1, \ldots, n$ are the random observation errors.

The goal is to estimate f^* based on the observation $Z_1, \ldots, Z_n$.

The estimation of f^* will be based on restricting the search to a subset $\mathcal{F}$ of functions of a Banach space $\mathcal{B}$.

In order to generalise, the estimator should be chosen in the set of stationary points of the empirical version of the risk $\mathcal{R} : \mathcal{F} \to \mathbb{R}$ defined by

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(X))\right],$$

where $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ satisfies

- $\ell(y, y) = 0$ for all $y \in \mathbb{R}$ and
- $\ell(y, \cdot)$: $\mathbb{R} \mapsto \mathbb{R}$ is a strictly convex twice continuously differentiable nonnegative function

Let $\hat{R}_n(f)$ denote the empirical risk defined by

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i)). \tag{6}$$

Then, the Empirical Risk Minimizer $\hat{f}^{ERM}$ will be a solution to

$$\hat{f}^{ERM} \in \operatorname{argmin}_{f \in \mathcal{F}} \hat{R}_n(f). \tag{7}$$

Let us start with ridge type functions

Ridge type functions

We consider a statistical model of the form

$$\mathbb{E}[Y_i \mid X_i] = f(X_i^t \theta^*), \qquad i = 1, \ldots, n, \tag{8}$$

where

- $\theta^* \in \mathbb{R}^p$
- the function f: $\mathbb{R} \mapsto \mathbb{R}$ is assumed increasing

Ridge type functions

- the data $X_1, \ldots, X_n$ will be assumed isotropic and subGaussian
- the matrix

$$X^\top = [X_1, \ldots, X_n] \tag{9}$$

  is full rank with probability one.
- for all $i = 1, \ldots, n$, the random vectors X_i are assumed
  - to have a second moment matrix $\mathbb{E}[X_i X_i^t] = I_p$,
  - to have ℓ_2-norm equal[1] to $\sqrt{p}$.
- the errors $\epsilon_i = Y_i - \mathbb{E}[Y_i]$ are independent subGaussian centered random variables with ψ_2-norm upper bounded by K_ϵ.

[1] notice that this is different from the usual regression model, where the columns are assumed to be normalised

Ridge type functions

In order to estimate θ^*, the Empirical Risk Minimizer $\hat{\theta}$ is defined as a solution to the following optimisation problem

$$\hat{\theta} \in \operatorname{argmin}_{\theta \in \Theta} \hat{R}_n(\theta) \tag{10}$$

with

$$\hat{R}_n(\theta) = \operatorname{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i - f(X_i^t \theta)). \tag{11}$$

Moreover, we assume that $\ell'(0) = 0$ and ℓ'' is upper bounded by a constant $C_{\ell''} > 0$.

- Let us concentrate on Ridge type functions

Theorem

Assume that $\hat{\theta}$ is near interpolating, i.e. $|Y_i - f(X_i^\top \hat{\theta})| \leq \varepsilon$. Let $\hat{\theta}^\circ$ denote the minimum norm near interpolating solution to the ERM problem, i.e.

$$\textit{argmin}_\theta \ \|\theta\|_2 \quad \textit{subject to } \varepsilon_{\min} \leq |Y_i - f(X_i^\top \hat{\theta})| \leq \varepsilon_{\max}, \tag{12}$$

$i = 1, \ldots, n$, for some $\varepsilon_{\min}$, $\varepsilon_{\max} \geq 0$.
Assume that f^{-1} is $C_{f^{-1}}$-Lipschitz on the set

$$\{z \mid \varepsilon_{\min} \leq |Y_i - f(z)| \leq \varepsilon_{\max}, i = 1, \ldots, n\}.$$

Theorem

Then, under technical assumptions, for any constant $\gamma > 0$, we have

$$|f(X_{n+1}^\top \hat{\theta}^\circ) - f(X_{n+1}^\top \theta^*)| \leq \frac{2n\ C_f C_{f^{-1}}}{\sigma_{\min}(X)} \varepsilon_{\max} + tK_X \|\theta^*\|_2 + t\ \frac{C\ K_X K_\varepsilon (\sqrt{n}+1)}{((1+\alpha)\sqrt{p} - C_{K_X}\sqrt{n})} + tK_\varepsilon + t\ \frac{6\sqrt{C}\, C_{\ell''} C_{f'} K_\varepsilon \sqrt{n}}{\delta(r)((1-\alpha)\sqrt{p} - C_{K_X}},$$

with probability at least

$$1 - 2\exp\left(-c_{K_X}\alpha^2 p\right) - \exp\left(-n/2\right) - \exp(-c_{K_X} p) - 3\exp(-t^2/2),$$

Theorem

where r is a solution of

$$r = \frac{C_{(\ell'', f', \varepsilon)}\sqrt{n}}{\delta(r)\left((1-\alpha)\sqrt{p} - C_{K_X}\sqrt{n}\right)}. \tag{13}$$

A handy result from Neuberger about the distance of the solution of a zero finding problem, i.e. consisting in solving

$$F(\hat{f}) = 0,$$

to the initial guess f^*.

The Continuous Newton's Method, Inverse Functions, and Nash-Moser

J. W. Neuberger

1. INTRODUCTION. The conventional Newton's method for finding a zero of a function $F : R^n \to R^n$, assuming that $(F'(y))^{-1}$ exists for at least some y in R^n, is the familiar iteration: pick z_0 in R^n and define

$$z_{k+1} = z_k - (F'(z_k))^{-1} F(z_k) \quad (k = 0, 1, 2, \dots),$$

hoping that $z_1, z_2, \dots$ converges to a zero of F. What can stop this process from finding a zero of F? For one thing, there might not *be* a zero of F. For another, the process might terminate for some integer k in the event that $F'(z_k)$ does not have an inverse.

A domain of attraction corresponding to a given root of F consists of the set of all

Theorem (Neuberger's theorem)

Suppose that $r > 0$, *that* $\theta^* \in \mathbb{R}^p$ *and that the map* F *is continuous on* $\overline{B_r}(\theta^*)$, *with the property that for each* θ *in* $B_r(\theta^*)$ *there exists a vector* d *in* $\overline{B_r}(0)$ *such that,*

$$\lim_{t \downarrow 0} \frac{F(\theta + td) - F(\theta)}{t} = -F(\theta^*). \tag{14}$$

Then there exists u *in* $\overline{B_r}(\theta^*)$ *such that* $F(u) = 0$.

Since the loss is twice differentiable, the empirical risk $\hat{R}_n$ is itself twice differentiable. The Gradient of the empirical risk is given by

$$\begin{aligned}\nabla \hat{R}_n(\theta) &= -\frac{1}{n}\sum_{i=1}^{n} \ell'(Y_i - f(X_i^t\theta))\ f'(X_i^t\theta)X_i \\ &= -\frac{1}{n}X^t D(\nu)\ l'(\epsilon)\end{aligned}$$

where $\ell'(\epsilon)$ is to be understood componentwise, and

$$\nu_i = f'(X_i^t\theta) \tag{15}$$

and the Hessian is given by

$$\begin{aligned}\nabla^2 \hat{R}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \Big(&\ell''(Y_i - f(X_i^t\theta))\ f'(X_i^t\theta)^2 \\ &- \ell'(Y_i - f(X_i^t\theta))\ f''(X_i^t\theta)\Big)X_iX_i^t.\end{aligned} \tag{16}$$

The Deep Neural Network case

Assumption

The sample satisfies the following separation

$$\min_{i,i'=1}^{n} \|X_i - X_{i'}\|_2 \geq cn^{-1/\nu} \tag{17}$$

with probability larger than or equal to $1-\delta$*, for some positive constants* c*,* ν *and for* $\delta \in (0,1)$*.*

The Holder exponent ν is usually interpreted as a surrogate for the intrinsic dimension of the data manifold. E.g., this intrinsic dimension was estimated to be less than 20 for the MNIST dataset

.

Intrinsic Dimensionality Estimation of Submanifolds in $\mathbb{R}^d$

Matthias Hein MH@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Jean-Yves Audibert AUDIBERT@CERTIS.ENPC.FR
CERTIS, ENPC, Paris, France

Here is a Banach space version of the Neuberger theorem.

Theorem (Neuberger's theorem)

Suppose that $\mathcal{B}$, $\mathcal{J}$, and $\mathcal{K}$ are three Banach spaces and that $\mathcal{B}$ is compactly embedded in $\mathcal{J}$.
Suppose that $F : \mathcal{B} \to \mathcal{K}$ is continuous with respect to the topologies of $\mathcal{J}$ and $\mathcal{K}$.
Suppose that $f \in \mathcal{B}$, that $r > 0$, and that for each g in $B_r(f)$, there is an h in $\bar{B}_r(0)$ such that

$$\lim_{t \to 0^+} \frac{1}{t}(F(g + th) - F(g)) = -F(f).$$

Then there is $\hat{f}$ in $\bar{B}_r(f)$ such that $F(\hat{f}) = 0$.

For $r > 0$ and u in $\mathcal{B}$, $B_r(u)$ and $\bar{B}_r(u)$ will denote the open and closed balls in $\mathcal{B}$, respectively, with center u and radius r.

Theorem (Neuberger's theorem for ERM)

Suppose that $r > 0$, *that* $\theta^* \in \mathbb{R}^p$ *and that the Jacobian* $D\hat{R}_n(\cdot)$ *is a continuous map on* $\mathcal{B}(\theta^*, r)$ *with the property that for each* θ *in* $\mathcal{B}(\theta^*, r)$ *there exists a vector* d *in* $\overline{\mathcal{B}(0, r)}$ *such that,*

$$\lim_{t \downarrow 0} \frac{D\hat{R}_n(\theta + td) - D\hat{R}_n(\theta)}{t} = -D\hat{R}_n(\theta^*). \tag{18}$$

Then there exists u *in* $\overline{\mathcal{B}(\theta^*, r)}$ *such that* $D\hat{R}_n(u) = 0$.

- We recall that $f \in \mathcal{F}$, and $d' \in \mathcal{B}$ such that $\mathcal{F} \subset \mathcal{B}$. Let us compute the directional derivative of $\hat{R}_n$

$$D\hat{R}_n(f) \cdot h' = \lim_{t \to 0} \frac{\hat{R}_n(f + th') - \hat{R}_n(f)}{t}$$

$$= \lim_{t \to 0} \frac{\frac{1}{n} \sum_{i=1}^n \ell(Y_i, f(X_i) + t\ h'(X_i)) - \ell(Y_i, f(X_i))}{t}$$

$$= \lim_{t \to 0} \frac{\frac{1}{n} \sum_{i=1}^n \partial_2 \ell(Y_i, f(X_i))\ t\ h'(X_i) + c\ \partial_2^2 \ell(Y_i, f(X_i))\ t^2\ h'^2(X_i)}{t}$$

with $c \in [0, 1]$, and thus

$$D\hat{R}_n(f) \cdot h' = \frac{1}{n} \sum_{i=1}^n \partial_2 \ell(Y_i, f(X_i)) h'(X_i).$$

- In the same spirit, we get

$$D^2 \hat{R}_n(f) \cdot (h', h) = \frac{1}{n} \sum_{i=1}^{n} \partial_2^2 \, \ell(Y_i, f(X_i)) \; h'(X_i) h(X_i).$$

- Based on these computations, Neuberger's theorem resorts to obtaining a bound on the norm of an appropriate solution h to the following linear system

$$\frac{1}{n}\sum_{i=1}^{n} \partial_2^2 \, \ell(Y_i, f(X_i)) \; h'(X_i) h(X_i)$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \partial_2 \ell(Y_i, f^*(X_i)) h'(X_i)$$

for all $f \in B_r(f^*)$ and for all $h' \in \mathcal{B}$.

- The idea is now to decouple the problem and
  - first solve it in a Sobolev space, and then
  - approximate the solution by a deep neural network

Simultaneous Neural Network Approximation for Smooth Functions

Sean Hon

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR

Haizhao Yang

Department of Mathematics, Purdue University, IN 47907, USA

Abstract

We establish in this work approximation results of deep neural networks for smooth functions measured in Sobolev norms, motivated by recent development of numerical solvers for partial differential equations using deep neural networks. Our approximation results are nonasymptotic in the sense that the error bounds are explicitly characterized in terms of both the width and depth of the networks simultaneously with all involved constants explicitly determined. Namely, for $f \in C^s([0,1]^d)$, we show that deep ReLU networks of

Theorem

Suppose that $f^* \in C^s\left([0,1]^d\right)$ *with* $s > 1 \in \mathbb{N}^+$ *satisfies* $\|\partial^\alpha f\|_{L^\infty\left([0,1]^d\right)} < 1$ *for any* $\alpha \in \mathbb{N}^d$ *with* $|\alpha| \leq k$.

Theorem

Suppose that $f^* \in C^s\left([0,1]^d\right)$ *with* $s > 1 \in \mathbb{N}^+$ *satisfies* $\|\partial^\alpha f\|_{L^\infty\left([0,1]^d\right)} < 1$ *for any* $\alpha \in \mathbb{N}^d$ *with* $|\alpha| \leq k$.

Let $\hat{f}$ *denote any estimator of* f^*.

Theorem

Suppose that $f^* \in C^s\left([0,1]^d\right)$ *with* $s > 1 \in \mathbb{N}^+$ *satisfies* $\|\partial^\alpha f\|_{L^\infty([0,1]^d)} < 1$ *for any* $\alpha \in \mathbb{N}^d$ *with* $|\alpha| \leq k$.

Let $\hat{f}$ *denote any estimator of* f^*.

Then there exists a neural network $f_{\hat{W}}$ *such that*

$$\|f_{\hat{W}} - \hat{f}\|_{W^{k,p}(\mathcal{D})} \leq 3(\kappa+1)^d \ 8^{\kappa-k} \beta_{width}^{-2(\kappa-k)/d} \beta_{depth}^{-2(\kappa-k)/d} \cdot \left(1 + n^{\frac{k}{\nu}} \max_{|\alpha|\leq K} \|\partial^\alpha \psi\|_{L^\infty([0,1]^d)}\right) + 6\left(\frac{c}{2}\right)^{d/p-k} K_\epsilon \ n^{(1+\frac{k-d/p}{\nu})} \|\psi\|_{W^{k,p}([0,1]^d)}.$$

Theorem

Suppose that $f^* \in C^s\left([0,1]^d\right)$ *with* $s > 1 \in \mathbb{N}^+$ *satisfies* $\|\partial^\alpha f\|_{L^\infty([0,1]^d)} < 1$ *for any* $\alpha \in \mathbb{N}^d$ *with* $|\alpha| \leq k$.

Thus there is a deep network which does as well as any estimator $\hat{f}$ *of* f^*.

The architecteure constraints on this network are

width

$$16\kappa^{d+1} d(\beta_{width} + 2)\log_2(8\beta_{width})$$

and depth

$$27\kappa^2(\beta_{depth} + 2)\log_2(4\beta_{width}).$$

Sketch of the proof

Notice that for all $f \in B_s(f_{W^*})$, we have

$$\frac{1}{n}\sum_{i=1}^{n} \frac{\partial \ell}{\partial_2}(Y_i, f(X_i))\ h'(X_i) = -\frac{1}{n}\sum_{i=1}^{n} (Y_i - f(X_i))\ \ h'(X_i),$$

and that

$$\frac{1}{n}\sum_{i=1}^{n} \frac{\partial^2 \ell}{\partial_2^2}(Y_i, f(X_i))\ h'(X_i)h(X_i) = \frac{1}{n}\sum_{i=1}^{n} h'(X_i)h(X_i).$$

Then, using the fact that ℓ is the ℓ_2^2 loss, Neuberger's condition reads

$$\frac{1}{n}\sum_{i=1}^{n} h'(X_i)h(X_i) = \frac{1}{n}\sum_{i=1}^{n} h'(X_i)\,(Y_i - f_{W^*}(X_i)).$$

One possible solution can be obtained by setting

$$h(X_i) = Y_i - f_{W^*}(X_i) = \varepsilon_i$$

$i = 1, \ldots, n$, i.e. using a noise interpolating solution.

One simple option is to take

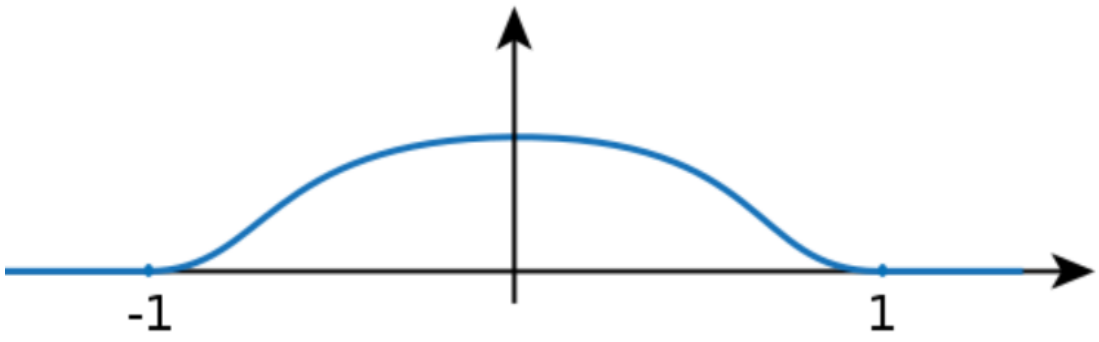
$$h(x) = \sum_{i=1}^{n} \epsilon_i \, \psi\left(\frac{x - X_i}{\sigma}\right)$$

where $\psi : \mathbb{R}^p \to \mathbb{R}$ is a kernel function and $\sigma > 0$ is a bandwidth.

Here, ψ denotes the bump function

$$\psi(x) = \begin{cases} \exp\left(1 - \frac{1}{1-\|x\|_2^2}\right) & \text{if } \|x\|_2^2 \leq 1, \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

and let $\psi_\sigma = \psi(\cdot/\sigma)$.



Let $\psi_\sigma = \psi(\cdot/\sigma)$.

- Now, observe that, based on Assumption 1, the functions $\psi((x - X_i)/\sigma)$, and their successive derivatives up to k, $i = 1, \ldots, n$, have disjoint supports for with probability larger than or equal to $1 - \delta$ as long as $\sigma \leq cn^{-1/\nu}$.
- We thus obtain that

$$\|h\|_{\mathcal{B}} \leq \|\epsilon\|_1 \ \|\psi_\sigma\|_{\mathcal{B}}$$

- Moreover, as is well known for subGaussian vectors, the norm is controlled by

$$\|\epsilon\|_2 \leq 6K_\epsilon n.$$

with probability at least $1 - \exp(-n)$, combining the conclusion of Theorem 13 follows from Neuberger's Theorem 8.

The proof for the deep neural network case is completed by using the approximation result of Hon and Wang.

Simultaneous Neural Network Approximation for Smooth Functions

Sean Hon

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR

Haizhao Yang

Department of Mathematics, Purdue University, IN 47907, USA

Abstract

We establish in this work approximation results of deep neural networks for smooth functions measured in Sobolev norms, motivated by recent development of numerical solvers for partial differential equations using deep neural networks. Our approximation results are nonasymptotic in the sense that the error bounds are explicitly characterized in terms of both the width and depth of the networks simultaneously with all involved constants explicitly determined. Namely, for $f \in C^s([0,1]^d)$, we show that deep ReLU networks of

- The number of layers may have to increase logarithmically with the number of samples
- The total number of parameters blows up polynomially in the number of samples and exponentially in the dimension of the problem

Conclusion and perspectives

- This simple exercice in using quantitative zero finding theorems such as Neuberger's theorem shows that we can easily prove results that do not blow up with the number of layers with interpolating networks
- We can easily study local minimisers as well using the same technique

- We would need to explore approximation theory in unusual/non standard directions:
  - improve the Hon and Wang theorem by introducing the constraint that the network be a flat minimiser
    - This would explain that Stochastic Gradient methods can find the correct approximation with large probability (?)

Biblio

Some papers:

- A finite sample analysis of the double descent phenomenon for ridge function estimation, Emmanuel Caron and Stephane Chretien: arXiv preprint arXiv:2007.12882
- On the problem of estimating a Sobolev function using deep neural networks, Hadrien Bigot-Balland, Emmanuel Caron and Stephane Chretien (soon on Arxiv)