Fit without Fear: Remarkable Mathematical Phenomena of Deep Learning through the Prism of Interpolation

by Mikhail Belkin

Carlos Gonzalez

University of Oxford, Department of Economics

March, 2023

Motivation

- Recent success of Deep Learning and Neural Networks: From ChatGPT-3 to Protein Folding
- However, standard mathematical frameworks in ML (PAC learning, VC-Dimensionality, ULLN...) fail to explain their success
- **Key Question:** Why do huge over-parameterized models which interpolate (completely overfit) data generalize to unseen data?
- **Key Question:** Why gradient-based algorithms (like GD or SGD) work in highly non-convex manifolds?

Notation and Problem Characterization

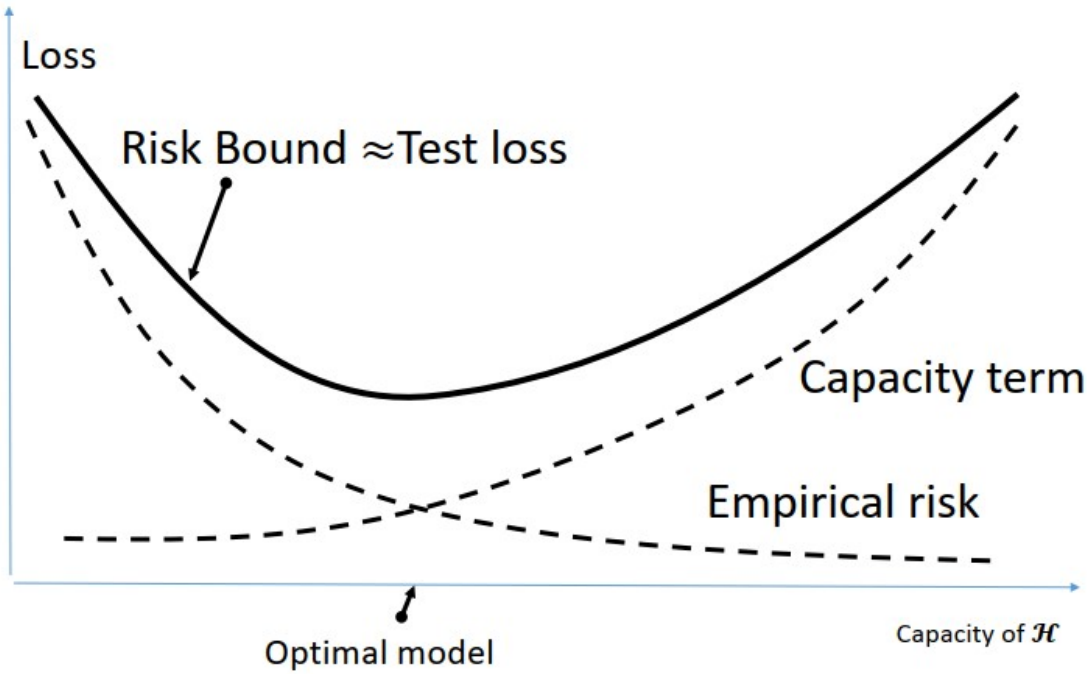
- Traditional "Image Classification" framework $\{(\boldsymbol{x}_i, y_i), \boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d, y_i \in \{-1, 1\}, i = 1, \ldots, n\}$
- We assume that $(\boldsymbol{x}_i, y_i)$ is sampled *iid* from a joint distribution P (big restriction?)
- Define the Bayes optimal classifier as $f^* = \arg\min_{f:\mathbb{R}^d \to \mathbb{R}} \mathbb{E}_{P(\boldsymbol{x},y)} l(f(\boldsymbol{x}), y)$
- where $l(\cdot, \cdot)$ is a loss function (like $\mathbb{1}_{f(\boldsymbol{x}) \neq y}$ or quadratic-loss)
- **Goal:** Infer an f (from data OR induction!) with small expected loss compared to the Bayes optimal

Classical Under-parameterized Regimes

- Classical frameworks are based on WYSIWYG
- In particular, we need that $\mathcal{R}_{\text{emp}}(f) = \frac{1}{n}\sum_i l(f(\boldsymbol{x_i}), y_i)$ is a good predictor of population risk for all f in the class $\mathcal{H}$
- This bound is of the form $O^*\left(\sqrt{\frac{\text{cap}(\mathcal{H})}{n}}\right)$, where $\text{cap}(\mathcal{H})$ is measured through the VC-dim of $\mathcal{H}$
- Free lunch theorem tells us that we need to restrict the capacity of $\mathcal{H}$
- **Intuition:** If our function perfectly predicts the data points in the training data, then we are learning no deeper "population patterns" which will generalize to unseen data

Classical Under-parameterized Regimes

- A tension in $\mathcal{H}$ immediately emerges. A class which is too small will (most likely) fail to include an $f : \mathcal{R}(f) \approx \mathcal{R}(f^*)$
- A class which is too big, will suffer from overfitting considerations



A Solution to Modern Regimes within the Classical Paradigm: Margins Theory

- So . . . why do huge classes for $\mathcal{H}$ still do well?
- **Partial Solution:** No need to enforce bounds between empirical and population risk uniformly across all $f \in \mathcal{H}$. We only need to worry about those f which are good empirically $\mathcal{H}_\epsilon = \{f \in \mathcal{H} : \mathcal{R}_{\text{emp}}(f) \leq \epsilon\}$.
- This allows us to derive data-dependent bounds based on $\text{cap}(\mathcal{H}_\epsilon, X) << \text{cap}(\mathcal{H})$. This moves the optimal crossing in Figure above to the right
- This theory has not been universally accepted

Breaking the Classic Paradigm

- Consider a deterministic process $P :\rightarrow \mathbb{R} \times \{-1, 1\}$ and its associated Bayes Optimal $f_P^*(x) = y$
- Define P_q as a distribution such that $y = f(x)$ with prob $(1 - q)$ and it is decided randomly with prob q
- Certainly, $f_P^* = f_{P_q}^*$ but $\mathcal{R}(f_P^*) = q/2$
- Empirically some interpolated regimes like kernel machines achieve good generalization (even with very high q)!
- But remember, $\mathcal{R}_{\text{emp}}(f_{\text{ker}}) = 0$ while $\mathcal{R}(f_{\text{ker}}) \approx \frac{q}{2}$
- **Punchline:** Generalization is produced regardless no apparent connection between Empirical and Population Risk

Modern Over-parameterized Regimes

- The elephant in the room: Why on earth do modern ML-models which completely overfit data work well in practice?
- And more importantly: Why **some** generalize to unseen data while others don't? If empirical risk minimization cannot lead our decisions anymore, what's the nature of the inductive bias which allows us to select the correct interpolated regime?
- Finally, in interpolated regimes, global minima are frequently captured in manifolds $\implies$ optimization is non-convex, so is ML also alien to (gradient-based) optimization theory?

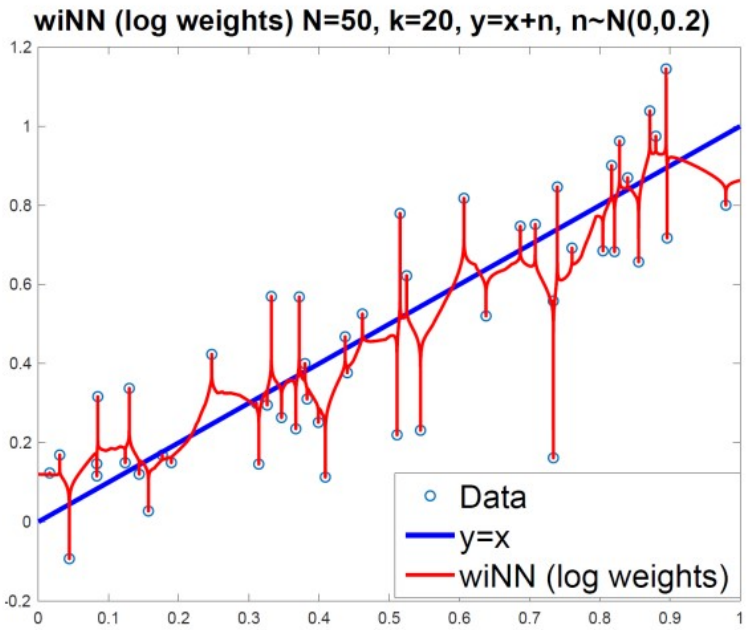
Theoretical Guarantees in Interpolated Regimes

- First step: Do we have any regret guarantees for interpolating algorithms?
- **1-NN** with risk bounded by $2 \cdot \mathcal{R}(f^*)$ so not ideal
- **Simplical interpolation**
  - Triangulate data using $\boldsymbol{x}$ as vertices
  - Consider f_{simp} a function which is linear within each simplex
  - Cool result $\mathcal{R}(f_{\text{simp}}) - \mathcal{R}(f^*) = O(\frac{1}{\sqrt{d}})$
  - Not only near optimal, but excess risk decreases with dimension
  - Why near optimal despite fitting noisy data? Noise concentrates only around corrupted vertices. Example

Theoretical Guarantees in Interpolated Regimes

- Unfortunately, f_{simp} fails to converge to optimal
- But other (kernel based) NN algorithms do like

$$f_{\text{sing}}(\boldsymbol{x}) = \frac{\sum_i K(\boldsymbol{x}, \boldsymbol{x}_{(i)}) y_{(i)}}{\sum_i K(\boldsymbol{x}, \boldsymbol{x}_{(i)})}$$



Inductive Bias

- So now, we've seen that there exist risk-optimal estimators which are foreign to ULLN and Empirical Risk Minimization
- However modern architectures are not Nearest Neighbours (direct) based algorithms, but optimization based algorithms. Hopefully, I've convinced you about the needs and possibilities of building theory for interpolated regimes, but we need to dig a bit more

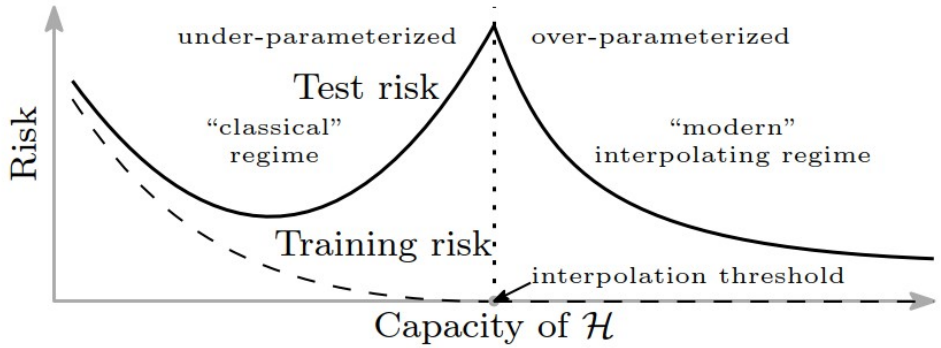
Inductive Bias

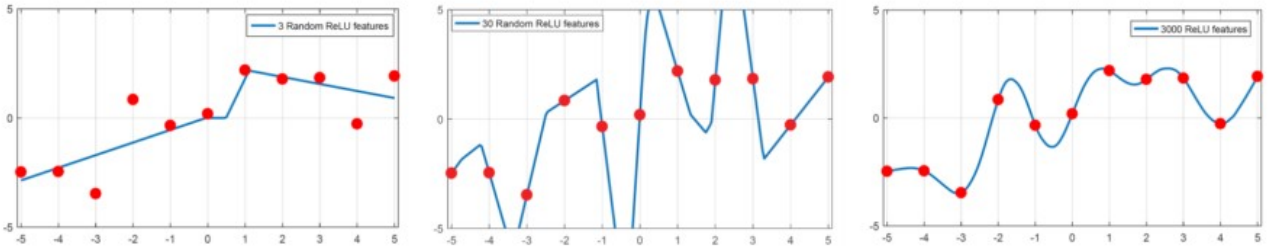
- Need to go beyond risk minimization $\implies$ Smoothness of f within the class of interpolating functions S is a good candidate
- Motivation, kernel machines
  $f_{\text{ker}}(\boldsymbol{x}) = \arg\min_{f(\boldsymbol{x}_i)=y_i \forall i} || f ||_{\mathcal{H}_K}$ ("the smoothest function" within a RKHS)
- If this inductive bias is correct, once you go beyond the interpolating threshold the extension of $\mathcal{H}$ is beneficial provided

$$\min_{f \in S_2} || f ||_s \leq \min_{f \in S_1} || f ||_s \tag{1}$$

- **Key Message:** When going beyond the threshold, regret undergoes a qualitative change: **Double Descent**

Double Descent





- Random Fourier Features as an empirical example of Double Descent (convergence to kernel machines as number of parameters m go to ∞)

So . . . is Inductive Bias about Smoothness?

- Imagine (corrupted) standard linear regime $y_i = \langle \boldsymbol{\beta}^*, \boldsymbol{x}_i \rangle + \epsilon_i$
- Set $d > n$ and define $\boldsymbol{\beta}_{\text{int}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^d, \langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle = y_i \forall i} || \boldsymbol{\beta} ||$
- Evidence shows very good performance of this type of algorithms (even compared to best under-parameterized / regularized alternatives)
- Inductive bias based on norm minimization performs nice even without smoothness interpretation
- **Final Remark:** In linear over-parameterized and kernel machines (S)GD converge to $\boldsymbol{\beta}_{\text{int}}$ despite lack of convexity (!!)

Proof Convergence of SGD

- Set $\mathcal{T} = \text{Span}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. Observe $\boldsymbol{\beta}_{\text{int}} \in \mathcal{T}$. Why? If $\boldsymbol{\beta}_{\text{int}} \notin \mathcal{T}$, an orthogonal projection from $\boldsymbol{\beta}_{\text{int}} \to \mathcal{T}$ will certainly have smaller norm, contradicting $\boldsymbol{\beta}_{\text{int}}$
- The space of interpolating predictors $\mathcal{S}$ is by definition orthogonal to $\mathcal{T} \implies \boldsymbol{\beta}_{\text{int}} \in \mathcal{S} \cap \mathcal{T}$
- Observe that GD converges to the minimizer of the loss function
- We can show that the gradient of the loss function at any point is in the span of the training sample thus optimization path lies entirely in $\mathcal{T}$
- If we initialize $\boldsymbol{\beta}_0 \in \mathcal{T}$ (i.e. $\boldsymbol{\beta}_0 = 0$) GD converges to $\boldsymbol{\beta}_{\text{int}}$ $\square$

What about Neural Networks?

- Yes, smoothness / transition to linearity seemed to do the trick with Kernel machines and some special architectures like RFF. But Neural Networks $f(\boldsymbol{w};\boldsymbol{x})$ are complicated, highly non-linear systems. Surely they do not become "linear" with m
- In other words, surely Neural Networks cannot be characterized as Kernel Machines. i.e. there is no Reproducible Kernel Hilbert Space whose associated norm is minimized by a NN
- Let me introduce the **Tangent Kernel**

$$K_{(\boldsymbol{x},\boldsymbol{z})}(\boldsymbol{w}) = \left\langle \nabla_{\boldsymbol{w}} f(\boldsymbol{w};\boldsymbol{x}), \nabla_{\boldsymbol{w}} f(\boldsymbol{w};\boldsymbol{z}) \right\rangle$$

What about Neural Networks?

- Define $\phi_{\boldsymbol{w}}(\boldsymbol{x}) = \nabla_{\boldsymbol{w}} f(\boldsymbol{w}; \boldsymbol{x})$
- Technically, it can be shown that for infinitely wide NN, $\phi_{\boldsymbol{w}}(\boldsymbol{x})$ is independent of $\boldsymbol{w}$ in a ball around the initialization point $\boldsymbol{w}_0$, which is equivalent to the linearity of $f(\boldsymbol{w}; \boldsymbol{x})$
- Some further intuitions. Consider the Taylor expansion of $f(\boldsymbol{w}; \boldsymbol{x})$ around $\boldsymbol{w}_0$. Difference between $f(\boldsymbol{w}; \boldsymbol{x})$ and its first order linear approximation is bounded by the spectral norm of the Hessian, which decreases with m
- As usual, reality is a bit more complicated, but remains fascinating that large classes of very complex functions turn out to be Kernel Machines and hence, linear in parameters

PL* Condition

- **Problem:** Over-parameterized non-linear optimization is (generally) non-convex, not even locally. How can they be a solution to GD based methods?
- We show that these spaces satisfy (a version of) Polyak-Lojasiewicz condition which turns out to be sufficient for gradient optimization despite non-convexity
- Some notation. Consider of finding $\boldsymbol{w} : f(\boldsymbol{w}, \boldsymbol{x}_i) = y_i$. Equivalently $F(\boldsymbol{w}) = \boldsymbol{y}$
- If such solution exists solving the equation above is equivalent to find $\arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}) = \| F(\boldsymbol{w}) - \boldsymbol{y} \|^2$
- A sufficient condition for gradient optimization. PL condition: $\mathcal{L}(\boldsymbol{w})$ is μ-PL if for a ball of radius $O(1/\mu)$

$$\frac{1}{2} \| \nabla \mathcal{L}(\boldsymbol{w}) \|^2 \geq \mu \cdot (\mathcal{L}(\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^*)) \tag{2}$$

PL* Condition

- But we don't know $\mathcal{L}(\boldsymbol{w}^*)$. So what if simply $\frac{1}{2} \mid\mid \nabla \mathcal{L}(\boldsymbol{w}) \mid\mid^2 \geq \mu \mathcal{L}(\boldsymbol{w})$
- This is known as PL* Condition
- Note: PL condition does not bite in under-parameterized regimes. Why?
- Why do over-parameterized regimes satisfy PL*? Time to come back to the Tangent Kernel K

Tangent Kernel and PL* Condition in NN

- It can be shown that for the square loss $\mathcal{L}(\boldsymbol{w})$ satisfies PL* with $\mu = \lambda_{\text{min}}(K)$. What implies that we need to verify that $\lambda_{\text{min}}(K) > \mu \geq 0$ for balls of radius $1/\mu$. How can we verify this analytic condition for NN?
- Two solutions
  - Hessian Control can be used when our system undergoes transition to linearity
  - Transformation Control based on controlling our condition number by expressing the system as a composition of two or more well-conditioned maps

Efficiency of SGD

- Empirically, SGD (even with fixed step-sizes!) does too well compared to standard GD in NN optimization. Why?
- Once again, interpolation is at the center
- Intuition: The closer we get to interpolation, loss becomes arbitrarily close to zero, thus variance induced by mini-batching decreases. This is known as "Automatic Variance Reduction"
- Consider two regimes
  - Linear Scaling: One iteration of SGD on $m < m^*$ observations is equivalent to m iterations of SGD with 1 iteration
  - Saturation: One iteration of SGD on $m > m^*$ observations is equivalent to 1 iteration of full GD
- Some numbers $m^* \approx 10$ with $n = 10^6$. Truly remarkable improvement

Conclusion

	Classical regime	Modern Regime
Generalization curve	U-Shaped	Double Descent
Optimal Model	Bottom of the U	Any Large Model
Optimization	Locally Convex	PL* condition
(S)GD	GD conv to local min	Exp conv of SGD

Thanks!