Generalization and Optimization in Deep Learning: Overparameterization and Interpolation [1]

Thanh Nguyen-Tang

August 10, 2021

[1] Based on Belkin 2021, "Fit without fear: remarkable mathematical phenomenon of deep learning through the prism of interpolation"

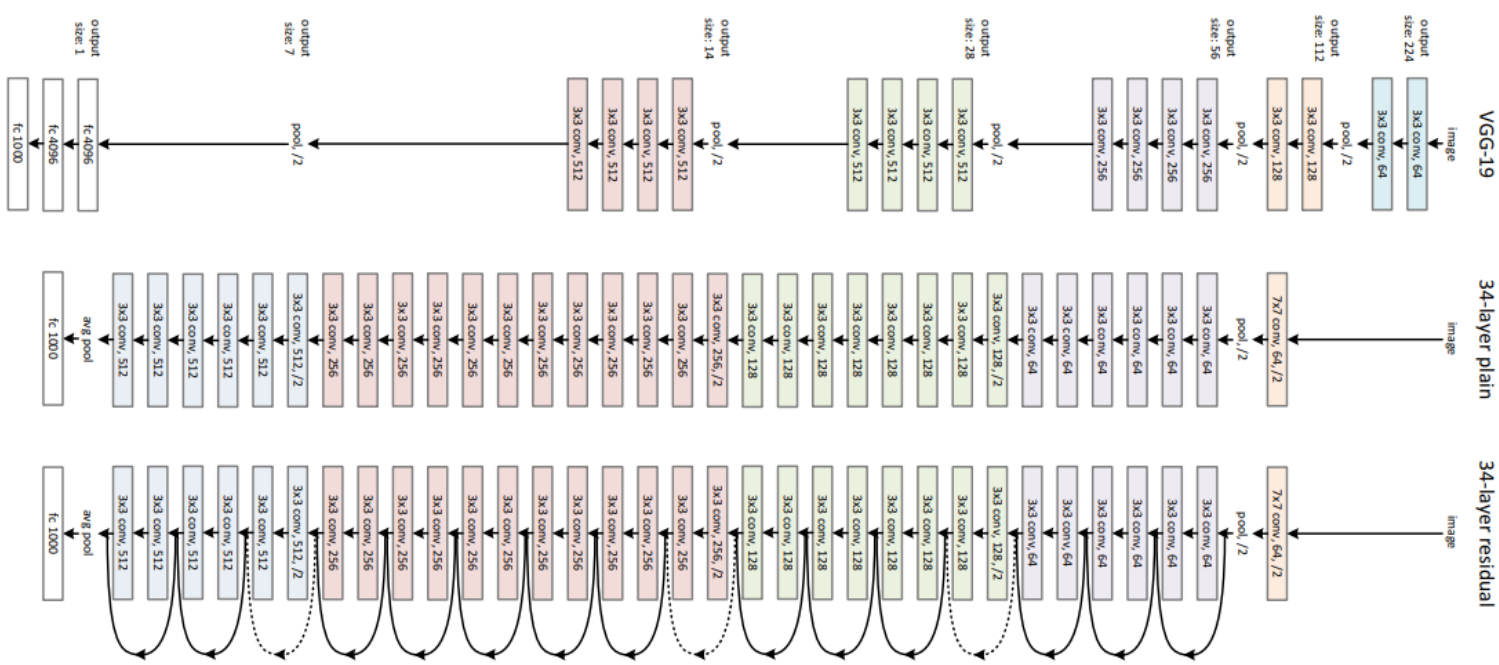
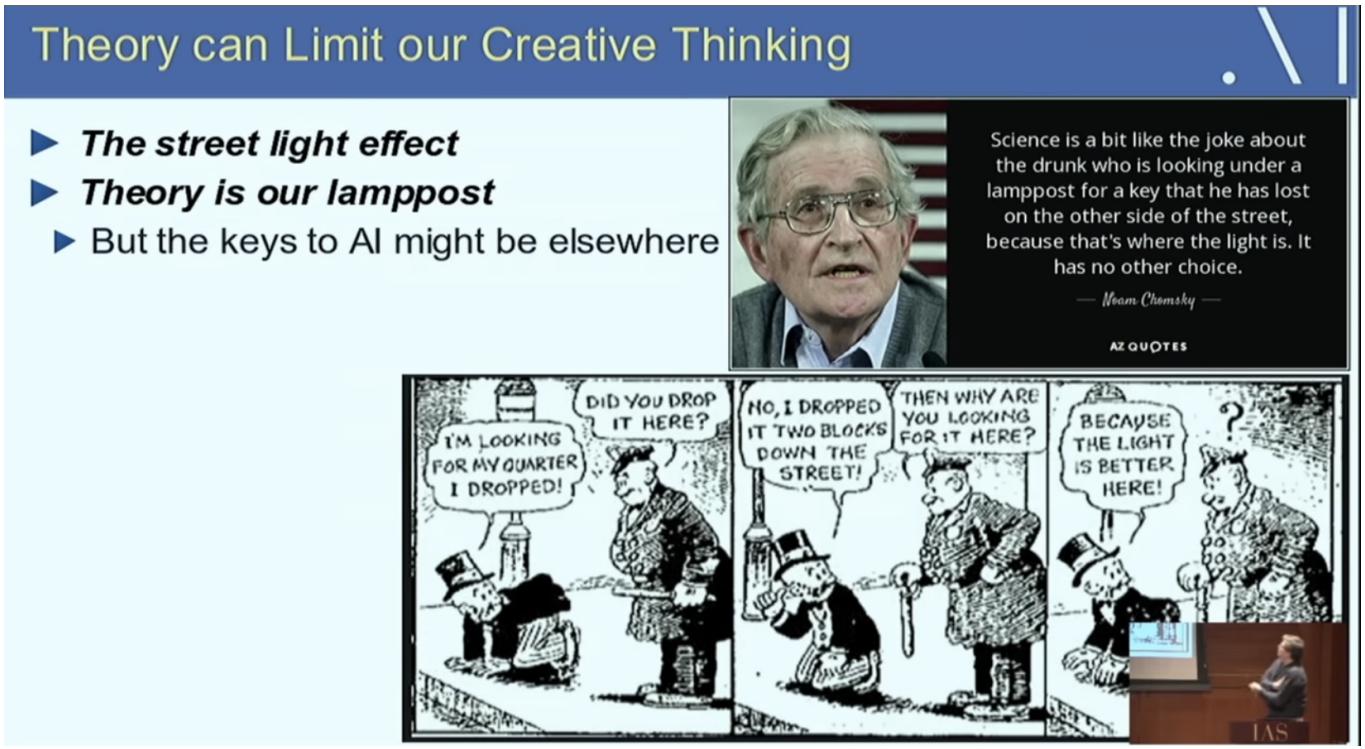


Figure: Resnet18 has around 11 million trainable parameters while the "largest" image benchmark ImageNet has "only" up to ≈ 1.3 million images!

Deep neural nets: A model with millions of parameters still achieve remarkable generalization using simple first-order optimization
$\rightarrow$ Classical statistical learning is unable to explain this phenomenon, let alone suggesting any new algorithm design
$\rightarrow$ Crisis of ML theory

Crisis of ML Theory

- "Machine learning has become alchemy" (A. Rahimi, B. Recht, NIPS 2017) https://youtu.be/x7psGHgatGM?t=722
- ML theory "looking for lost keys under a lamp post because that's where the light is" (Yann Lecun, 2018) https://youtu.be/gG5NCkMerHU?t=3189



Overview

Main questions:

- **Generalization**: Why do deep nets generalize to unseen data?
- **Optimization**: Why can non-convex objective functions be optimized

Interpolation = fitting the noisy training data precisely (zero training error)
In this talk,

- Review classical statistical learning theorem and why it fails to explain generalization in "**interpolating**" functions
- How **interpolation** and **over-parameterization** can explain these generalization and optimization phenomenon in deep learning

Supervised learning setting

- Given data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $(\boldsymbol{x}_i, y_i) \in \mathcal{X}^d \times \{-1, 1\}$ are i.i.d. samples from a data distribution P on $\mathcal{X}^d \times \{-1, 1\}$.
- **Goal**: To learn a function $f : \mathbb{R}^d \to \{-1, 1\}$ that generalizes to new unseen data sampled from P.
- The Bayes optimal classifier:

$$f^* = \underset{f:\mathbb{R}^d \to \{-1,1\}}{\arg\min} \; \mathbb{E}_{P(\boldsymbol{x},y)} \left[l(f(\boldsymbol{x}), y) \right],$$

where $l(f(\boldsymbol{x}), y) = 1_{f(\boldsymbol{x}) \neq y}$.

Empirical and Structural Risk Minimization (ERM) by V. Vapnik

- It approximates the optimal f^* using data as a proxy to the unknown data distribution and a restrictive hypothesis space:

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(\boldsymbol{x}_i), y_i)$$

$$f_{emp} = \arg\min_{f \in \mathcal{H}} R_{emp}(f)$$

- The relationship between f^* and f_{emp} depends on the choice of the hypothesis

Two key ingredients to ERM

- ULLN (Uniform Law of Large Numbers):

$$\forall f \in \mathcal{H}, R(f) = \mathbb{E}_{P(\boldsymbol{x},y)}[l(f(\boldsymbol{x}), y)] \approx R_{emp}(f)$$

- CC (Capacity Control): Control the capacity of the space $\mathcal{H}$, i.e., $cap(\mathcal{H})$ such as VC dim and covering number

Generalization of ERM

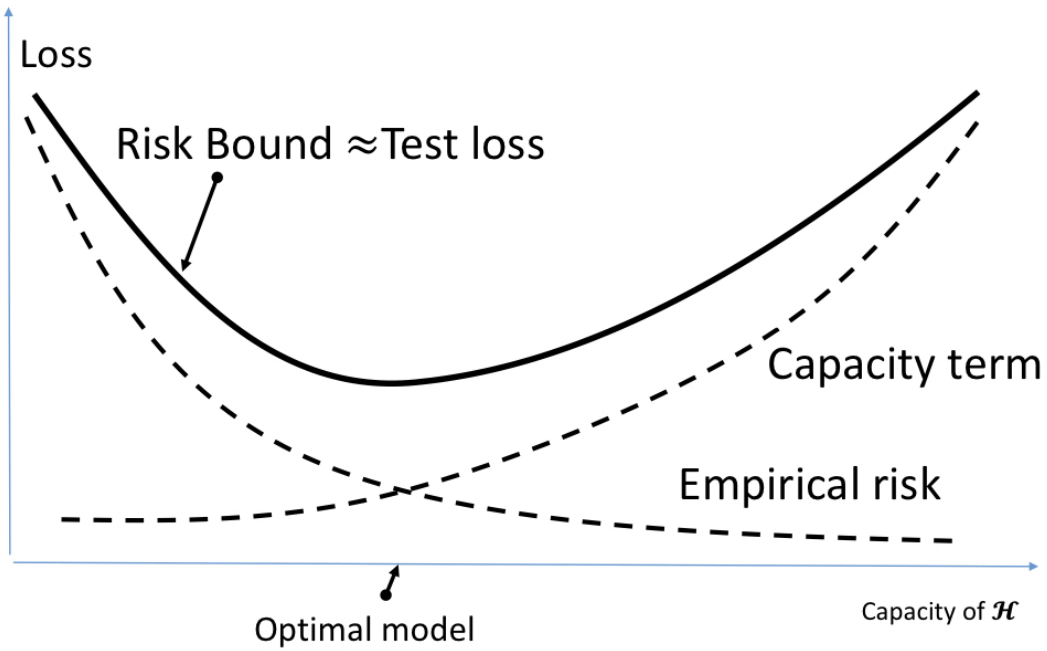
- Guarantee form:

$$\forall f \in \mathcal{H}, \quad \underbrace{R(f)}_{\text{expected risk}} - \underbrace{R_{emp}(f)}_{\text{empirical risk}} < \underbrace{\tilde{\mathcal{O}}\left(\sqrt{\frac{cap(\mathcal{H})}{n}}\right)}_{\text{capacity term}}$$

- Generalization error:

$$R(f_{emp}) - \min_{f \in \mathcal{H}} R(f) < \tilde{\mathcal{O}}\left(\sqrt{\frac{cap(\mathcal{H})}{n}}\right),$$

$\rightarrow$ the true risk of f_{emp} is nearly optimal as long as $cap(\mathcal{H}) \ll n$



Margin theory and data-dependent capacity

- The previous bound applies for all $f \in \mathcal{H} \rightarrow$ too conservative and we care about only a subset of $\mathcal{H}$
- Increasing model complexity for over-fitting does not necessarily lead to drop in performance as predicted in the previous theory (e.g., boosting)
- A refined theory: $\forall f \in \mathcal{H}, R(f) - R_{emp}(f) < \tilde{\mathcal{O}}\left(\sqrt{\frac{cap(\mathcal{H},X)}{n}}\right)$, where $cap(\mathcal{H}, X)$ depends on training data X and can be significantly smaller than $cap(\mathcal{H})$ (e.g., $cap(\mathcal{H}, X)$ could be Rademacher complexity)

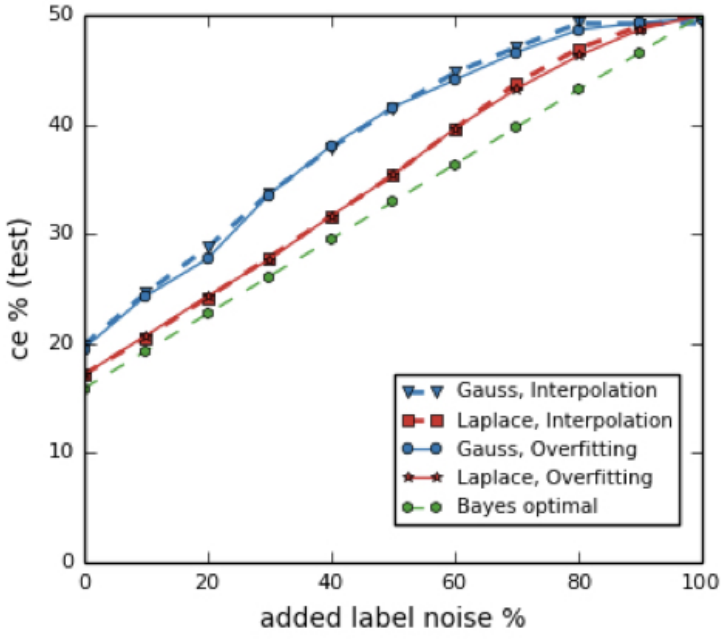
What you see is not what you get

- The previous theories are "what you see is what you get" (WYSIWYG): the empirical risk in the training data is well predictive of the true risk in the unseen data with the difference controlled by the model complexity.
- Yet, this is not true for **interpolating** classifiers/predictors
  - An **interpolating** classifier f_{in}: fits the noisy training data precisely, i.e., $f_{in}(\boldsymbol{x}_i) = y_i, \forall i \in [n]$, i.e., $R_{emp}(f_{in}) = 0$.
  - Deep nets can have zero training error (and fit to the noise) but still generalize very well [2]
  - Yet, obviously many functions that interpolates the noisy data can generalize arbitrarily bad

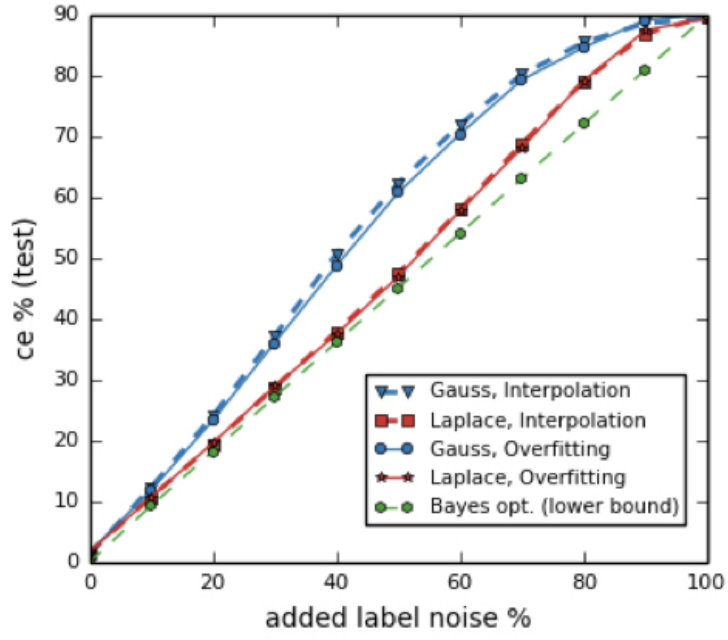
[2] Zhang et al. "Understanding deep learning requires rethinking generalization"

Classical learning theory yields vacuous bounds for **interpolating** predictors: kernel machines as an example

- Consider a kernel machine: $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$ where K is a positive definite kernel (e.g., Gaussian kernels or Laplace kernels)
- There is a unique predictor f_{ker} of the above form that interpolates the data, i.e., $f_{ker}(\boldsymbol{x}_i) = y_i, \forall i \in [n]$
- P_q: with probability q the label for any $\boldsymbol{x}$ is assigned to $\{-1, 1\}$ with equal probability, with probability $1 - \delta$, the label is assigned according to the original P.



(a) Synthetic, 2-class problem



(b) MNIST, 10-class

Figure 2: (From [12]) Interpolated (zero training square loss), "overfitted" (zero training classification error), and Bayes error for datasets with added label noise. y axis: test classification error.

Via simple arguments, we have

- $R_{P_q}(f) = \frac{q}{2} + R_P(f), \forall f$
- Assume $P(y|\boldsymbol{x})$ is deterministic (e.g., MNIST case) $\rightarrow$ $R_P(f_P^*) = 0$
- We have

$$R_{P_q}(f_{ker,q}) - \underbrace{R_{emp}(f_{ker,q})}_{=0} = R_{P_q}(f_{ker,q}) \geq \frac{q}{2}$$

- We must have

$$\frac{q}{2} \underbrace{\leq}_{\text{correct}} \tilde{\mathcal{O}}\left(\sqrt{\frac{cap(\mathcal{H}, X)}{n}}\right) \underbrace{\leq}_{\text{nontrivial}} \frac{1}{2}$$

We remark:

- No such capacity bound is known
- Strong generalization of classifiers that interpolate noisy data do not follow WYSIWYG and independent of the model capacity

Questions:

- What rationale for an interpolating classifier (that fits the noise in the training data as well) to generalize well?
- As model capacity cannot explain the strong generalization of interpolating classifiers, what else does?

$\rightarrow$ A case study in k-NN with singular weighting schemes

A concrete example for an interpolating classifier that generalizes well and a clue for why

- 1-NN is perhaps the simplest interpolating classifier
  $R_{emp}(\text{1-NN}) = 0$, but it has a high excess risk

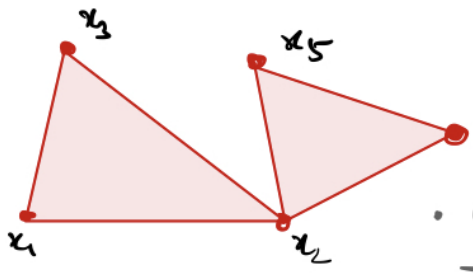
  $R(f^*) \leq R(\text{1-NN}) \leq 2R(f^*)$ as $n \to \infty$[Cover and Hart, 1967]

- A simple modification to 1-NN obtains an interpolating classifier with near-optimal excess risk $\to$ simplicial interpolation

Simplical interpolation and the blessing of dimensionality

Consider a triangulation of the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ that is a partition of the convex hull of the data into a set of d-dimensional simplices:

$\begin{cases} f_{simp}(x_i) = y_i, \forall i \\ f_{simp} \text{ is linear within each simplex} \end{cases}$



A simplicial 2-simplex

- one dimensional simplical interpolation $=$ 1-NN

1. Vertices of each simplex are data points.
2. For any data point $\mathbf{x}_i$ and simplex s, $\mathbf{x}_i$ is either a vertex of s or does not belong to s.

Blessing of dimensionality: $R(f_{simp}) - R(f^*) = \mathcal{O}(\frac{1}{\sqrt{d}})$

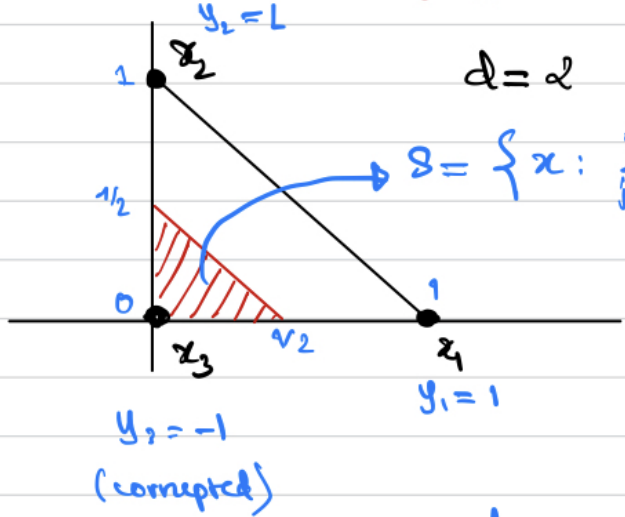
Q: How can an interpolating function be near optimal despite it fits the noisy data? Why does increasing dimension help?

$\rightarrow$ **key insight**: the incorrect predictions are localized in the neighborhood of noisy points and this neighborhood gets smaller with dimension.

Training data: $x_1, x_2, \ldots, x_d, x_{d+1}$ where

$$\begin{cases} x_i = (0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0) & \forall i \in [d] \\ x_{d+1} = (0, \ldots, 0) \end{cases}$$

$y_i = 1 \quad \forall i \in [d]$, $y_{d+1} = -1$ (corrupted by noise)



$$f_{simp}(x) = \text{sign}\left(2 \sum_{j=1}^{d} [x]_j - 1\right)$$

f_{simp} agrees w/ f^* except on S

$$\text{vol}(S) = \frac{1}{2^d} \text{vol}(\text{simplex})$$

k-NN with singular weighing scheme:

- achieves *consistency* as compared to simplical interpolation
- optimal for both regression and classification

$$f_{sing}(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} K(\boldsymbol{x}, \boldsymbol{x}_i) y_i}{\sum_{i=1}^{k} K(\boldsymbol{x}, \boldsymbol{x}_i)} \text{ where } K(\boldsymbol{x}, \boldsymbol{z}) = \|\boldsymbol{x} - \boldsymbol{z}\|^{-\alpha}, \alpha > 0$$

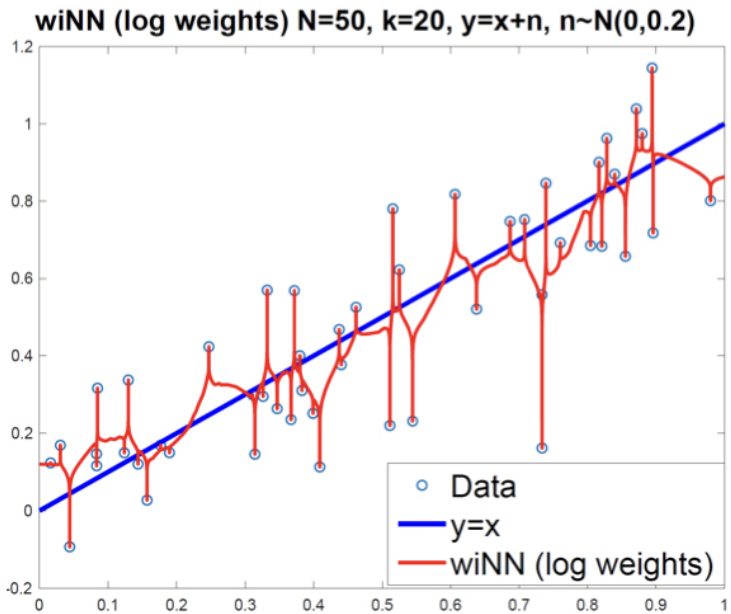


Figure 4: Singular kernel for regression. Weighted and interpolated nearest neighbor (wiNN) scheme. Figure credit: Partha Mitra.

Key insights so far

- Fitting noisy training data exactly (i.e., interpolation) does not necessarily result in poor generalization
- Uniform law of large numbers and using empirical risk to approximate the true risk does not work for interpolating functions

$\rightarrow$ New "modern" theory beyond the classical statistical learning theory is required to explain generalization of interpolating classifiers/predictors!

Double Descent

Double Descent:

- high risk at interpolating threshold
- increasing the number of parameters leads to improved generalization

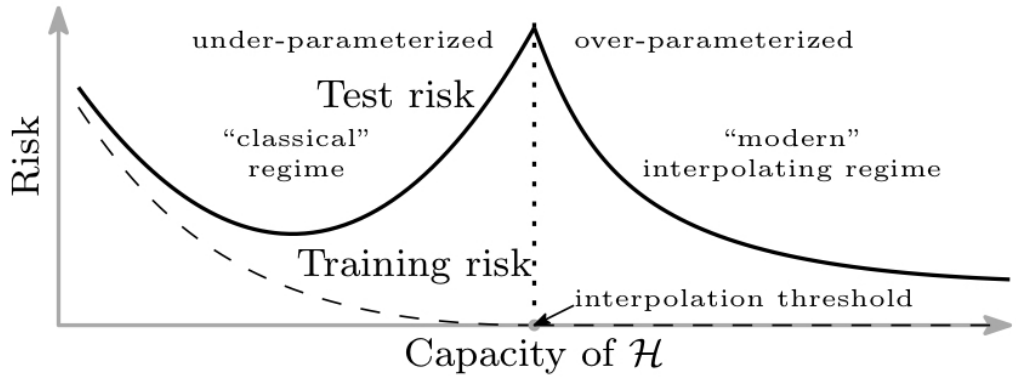


Figure 5: Double descent generalization curve (figure from [9]). Modern and classical regimes are separated by the interpolation threshold.

- "Classical" under-parameterized regimes: Typically no $f \in \mathcal{H}$ such that $R_{emp}(f) = 0$
- "Modern" over-parameterized regimes: A typicall large set S of predictors that interpolate the (noisy) training data: $S = \{f \in \mathcal{H} : R_{emp}(f) = 0\}$

Inductive Biases

- An interpolating learning algorithm $\mathcal{A}$ selects $f_{\mathcal{A}} \in \mathcal{S}$
- An fundamental issue of inductive bias: Why do solutions such as those obtained by kernel machines and neural networks, generalize well while other possible solutions do not?

$\rightarrow$ Among all the functions that interpolates the training data, select the one that maximizes some notion of functional smoothness $\|f\|_s$:

$$f_{in} = \underset{f \in \mathcal{H}: f(\boldsymbol{x}_i) = y_i, \forall i \in [n]}{\arg\min} \|f\|_s$$

Inductive Biases (con't)

- Generalization tends to correlate with minimum norm (a measure of functional smoothness)
- Predictors from richer classes tend to perform better as it has smaller functional norm

Simple argument: Consider two hypothesis spaces $\mathcal{H}_1 \subset \mathcal{H}_2$ and their corresponding manifolds of interpolating predictors $\mathcal{S}_1 \subset \mathcal{H}_1$ and $\mathcal{S}_2 \subset \mathcal{H}_2$, then $\mathcal{S}_1 \subset \mathcal{S}_2$: $\min_{f \in \mathcal{S}_2} \|f\|_s \leq \min_{f \in \mathcal{S}_1} \|f\|_2$

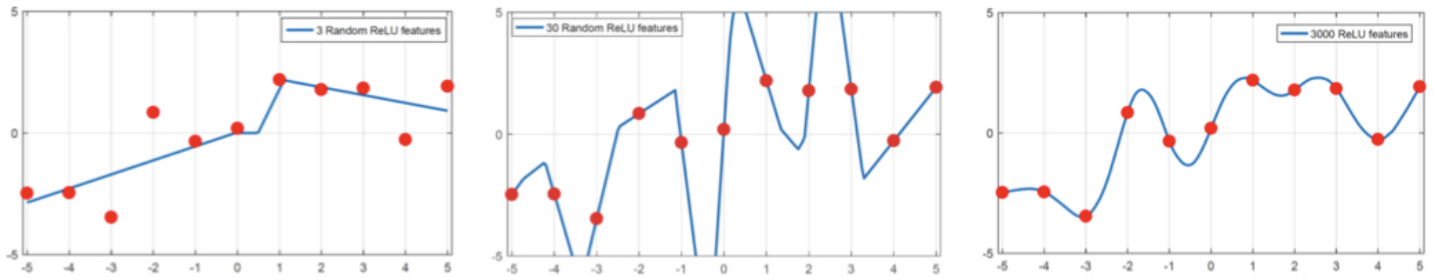


Figure 6: Illustration of double descent for Random ReLU networks in one dimension. Left: Classical under-parameterized regime (3 parameters). Middle: Standard over-fitting, slightly above the interpolation threshold (30 parameters). Right: "Modern" heavily over-parameterized regime (3000 parameters).

Example: Random Fourier Feature (RFF)

- RFF model $\mathcal{H}_m$: $f(\boldsymbol{w}, \boldsymbol{x}) = \sum_{k=1}^{m} w_k e^{\sqrt{-1}\langle \boldsymbol{v}_k, \boldsymbol{x}\rangle}$, where $\{\boldsymbol{v}_k\}_{k=1}^{m}$ are fixed weights sampled independently from the standard normal distribution on $\mathbb{R}^d$, and $\boldsymbol{w} = (w_1, \ldots, w_m) \in \mathcal{C}^m$ are trainable parameters.
- RFF model = a neural net with one hidden layer of size m and fixed first layer weights
- Given data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, in the over-parameterized regime, linear regression is given via

$$f_m = \underset{f \in \mathcal{H}_m : f(\boldsymbol{w}, \boldsymbol{x}_i) = y_i, \forall i \in [n]}{\arg\min} \|\boldsymbol{w}\|$$

- Representer Theorem:

$$\lim_{m \to \infty} f_m(\boldsymbol{x}) = \underset{f \in \mathcal{S} \subset \mathcal{H}_K}{\arg\min} \|f\|_{\mathcal{H}_K} =: f_{ker},$$

where $\mathcal{H}_K$ is the RKHS of the Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\|\boldsymbol{x} - \boldsymbol{z}\|^2)$ and $\mathcal{S}$ is the manifold of the interpolating functions in $\mathcal{H}_K$

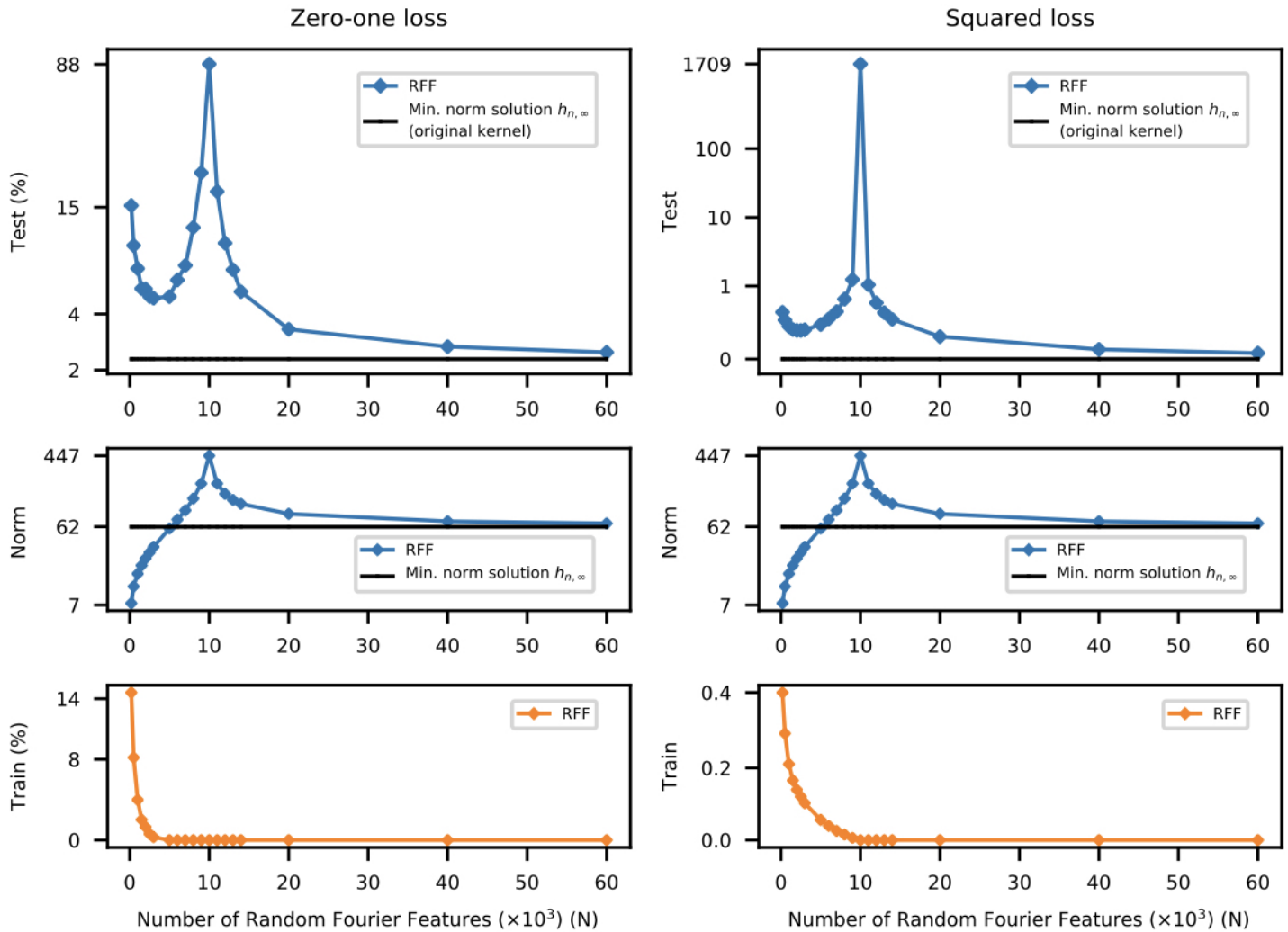


Figure 7: Double descent generalization curves and norms for Random Fourier Features on a subset of MNIST (a 10-class hand-written digit image dataset). Figure from [9].

Wide Neural Networks: A transition to (near-)linearity

- Why are complicated non-linear systems with large numbers of parameters able to generalize to unseen data?
  $\rightarrow$ It turns out neural nets behave very much like kernel machines when the wide is sufficiently large
- Neural network functions:

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{\sqrt{m}} W^{(L)} \sigma\left(W^{(L-1)} \sigma\left(\ldots \sigma\left(W^{(1)} \sigma(\boldsymbol{x})\right) \ldots\right)\right),$$

  where $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$, $d_0 = d$, $m = \min_l d_l$, and w is a concatenation of $W^{(l)}$.
- Transition to linearity: when the wide becomes sufficiently large, the neural network becomes nearly a linear function of their parameters
- The linearity behaviour can be characterized via **Neural Tangent Kernel (NTK)** [Jacot et al. 2018]

Neural Tangent Kernels

- The tangent kernel at $\boldsymbol{w} \in \mathbb{R}^M$ is defined as

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

- Feature map: $\phi_{\boldsymbol{w}}(\boldsymbol{x}) = \nabla_{\boldsymbol{w}} f(\boldsymbol{w}, \boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^M$
- NTK Theorem: When the network width m is sufficiently large, in a ball around a random initialization point $\boldsymbol{w}_0$, the neural network function is nearly a kernel machine:

$$f(\boldsymbol{w}, \boldsymbol{x}) \approx \langle \boldsymbol{w} - \boldsymbol{w}_0, \phi_{\boldsymbol{w}_0}(\boldsymbol{x}) \rangle + f(\boldsymbol{w}_0, \boldsymbol{x})$$

A sketch proof for (near-)linearity transition in NTK

- Let $\mathcal{B}(\boldsymbol{w}_0, R) \subset \mathbb{R}^M$ be a ball of radius R around $\boldsymbol{w}_0$
- Taylor expansion with the Lagrangian remainder: For any $\boldsymbol{w} \in \mathcal{B}(\boldsymbol{w}_0, R)$, there is $\xi \in \mathcal{B}(\boldsymbol{w}_0, R)$ such that:

$$f(\boldsymbol{w}, \boldsymbol{x}) = \langle \boldsymbol{w} - \boldsymbol{w}_0, \phi_{\boldsymbol{w}_0}(\boldsymbol{x}) \rangle + f(\boldsymbol{w}_0, \boldsymbol{x}) + \frac{1}{2} \langle \boldsymbol{w} - \boldsymbol{w}_0, \boldsymbol{H}(\xi)(\boldsymbol{w} - \boldsymbol{w}_0) \rangle,$$

where $\boldsymbol{H}(\xi)$ is the Hessian at ξ.

- Bound the difference by the spectral norm of the Hessian

$$\sup_{x \in \mathcal{B}(\boldsymbol{w}_0, R)} \left| f(\boldsymbol{w}, \boldsymbol{x}) - f(\boldsymbol{w}_0, \boldsymbol{x}) - \langle \boldsymbol{w} - \boldsymbol{w}_0, \phi_{\boldsymbol{w}_0}(\boldsymbol{x}) \rangle \right| \leq \frac{R^2}{2} \sup_{x \in \mathcal{B}(\boldsymbol{w}_0, R)} \|\boldsymbol{H}(\xi)\|$$

- It can be shown that:

$$\sup_{x \in \mathcal{B}(\boldsymbol{w}_0, R)} \|\boldsymbol{H}(\xi)\| = \tilde{\mathcal{O}}(\frac{1}{\sqrt{m}})$$

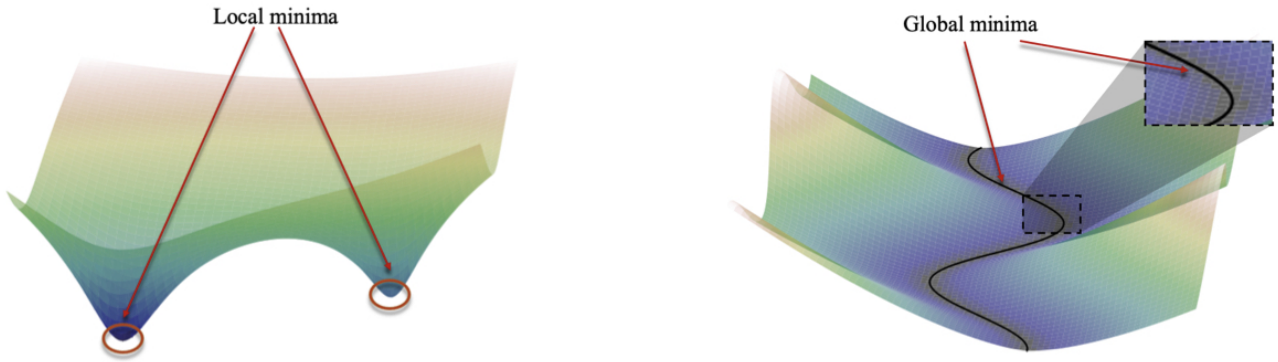
Key insights

- **Over-parameterization leads to near-linear behaviour**: Neural networks with sufficiently wide width behaves nearly linear in terms of its parameters around the initialization
- Near-linearity of deep neural nets is a property of the model and is unrelated to training procedures.

Optimization in deep neural networks

- The loss landscape of deep nets is non-convex, not even locally (why?)
- Why are simple gradient-based optimization such as (S)GD able to reliably find a globale minimum in such a non-convex optimization in deep nets?

Loss landscape of over-parameterized models



(a) Loss landscape of under-parameterized models (b) Loss landscape of over-parameterized models

Figure 1: Panel (a): Loss landscape is locally convex at local minima. Panel (b): Loss landscape incompatible with local convexity as the set of global minima is not locally linear.

- Every local minima is global and the manifold of minimizers $\mathcal{S}$ has positive dimension
- The loss is locally non-convex (otherwise, the manifold must be locally linear which is hardly the case)

Q: What mathematical property that allows such a loss landscape to be optimized efficiently by (S)GD?
$\rightarrow$ A simple sufficient condition: Polyak-Lojasiewicz (PL)-condition that is satisfied due to over-parameterization.

μ-PL condition

- For $\mu > 0$, a loss function $\mathcal{L}(\boldsymbol{w})$ is said to satisfy the μ-PL condition on a ball $\mathcal{B}$ if

$$\frac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{w})\|^2 \geq \mu \mathcal{L}(\boldsymbol{w}), \forall \boldsymbol{w} \in \mathcal{B}$$

- **Theorem** [3]: PL-condition in a ball of sufficiently large radius implies both existing of an interpolating solution within the ball and an exponential convergence of (S)GD.

[3] Liu et al., "Loss landscapes and optimization in over-parameterized non-linear systems and neural networks", 2021.

- Convergence of GD:

Theorem 6 (Local PL* condition $\Rightarrow$ existence of a solution + fast convergence). *Suppose the system $\mathcal{F}$ is $L_{\mathcal{F}}$-Lipschitz continuous and $\beta_{\mathcal{F}}$-smooth. If the square loss $\mathcal{L}(\mathbf{w})$ satisfies the μ-PL* condition in the ball $B(\mathbf{w}_0, R) := \{\mathbf{w} \in \mathbb{R}^m : \|\mathbf{w} - \mathbf{w}_0\| \leq R\}$ with $R = \frac{2L_{\mathcal{F}}\|\mathcal{F}(\mathbf{w}_0)-\mathbf{y}\|}{\mu}$. Then we have the following:*
(a) Existence of a solution: There exists a solution (global minimizer of $\mathcal{L}$) $\mathbf{w}^* \in B(\mathbf{w}_0, R)$, such that $\mathcal{F}(\mathbf{w}^*) = \mathbf{y}$.
(b) Convergence of GD: Gradient descent with a step size $\eta \leq 1/(L_{\mathcal{F}}^2 + \beta_{\mathcal{F}}\|\mathcal{F}(\mathbf{w}_0) - \mathbf{y}\|)$ converges to a global solution in $B(\mathbf{w}_0, R)$, with an exponential (a.k.a. linear) convergence rate:

$$\mathcal{L}(\mathbf{w}_t) \leq \left(1 - \eta\mu\right)^t \mathcal{L}(\mathbf{w}_0). \tag{26}$$

- Convergence of SGD:

Theorem 7. *Assume each $\ell_i(\mathbf{w})$ is β-smooth and $\mathcal{L}(\mathbf{w})$ is γ-smooth satisfies the μ-PL* condition in the ball $B(\mathbf{w}_0, R)$ with $R = \frac{2n\sqrt{2\beta\mathcal{L}(\mathbf{w}_0)}}{\mu\delta}$ where $\delta > 0$. Then, with probability $1 - \delta$, SGD with mini-batch size $s \in \mathbb{N}$ and step size $\eta \leq \frac{ns\mu}{\gamma(\beta + \gamma(s-1))}$ converges to a global solution in the ball $B(\mathbf{w}_0, R)$, with an exponential convergence rate:*

$$\mathbb{E}[\mathcal{L}(\mathbf{w}_t)] \leq \left(1 - \frac{\mu s\eta}{n}\right)^t \mathcal{L}(\mathbf{w}_0). \tag{27}$$

μ-PL for overparameterized models

- Let $F(\boldsymbol{w}) = (f(\boldsymbol{w}, \boldsymbol{x}_1), \ldots, f(\boldsymbol{w}, \boldsymbol{x}_n))$, and $\boldsymbol{y} = (y_1, \ldots, y_n)$
- An interpolating solution $f(\boldsymbol{w}, \boldsymbol{x})$ is a solution for a single equation: $F(\boldsymbol{w}) = \boldsymbol{y}, F : \mathbb{R}^M \to \boldsymbol{R}^n$
- It is equivalent to minimizing the squared loss: $\mathcal{L}(\boldsymbol{w}) = \frac{1}{2}\|F(\boldsymbol{w}) - \boldsymbol{y}\|^2$
- Consider the tangent kernel:

$$K(\boldsymbol{w}) = \nabla F(\boldsymbol{w})^T \cdot \nabla F(\boldsymbol{w}) \in \mathbb{R}^{n \times n} \text{ where } \nabla F(\boldsymbol{w}) \in \mathbb{R}^{M \times n}$$

- Simple calculation leads to:

$$\|\nabla \mathcal{L}(\boldsymbol{w})\|^2 = \langle F(\boldsymbol{w}) - \boldsymbol{y}, K(\boldsymbol{w})(F(\boldsymbol{w}) - \boldsymbol{y}) \rangle$$

- Thus, we have

$$\frac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{w})\|^2 \geq \lambda_{\min}(K(\boldsymbol{w})) \cdot \mathcal{L}(\boldsymbol{w})$$

Basin of non-singular tangent kernels

- Singular Tangent Kernel: $\{\boldsymbol{w} \in \mathbb{R}^M : \lambda_{\min}(K(\boldsymbol{w})) = 0\}$
- $\lambda_{\min}(K(\boldsymbol{w})) > 0$ iff the columns of $\nabla F(\boldsymbol{w})$ spans $\mathbb{R}^n$
- As $M \gg n$, we expect the set of non-singular tangent kernels to be large!

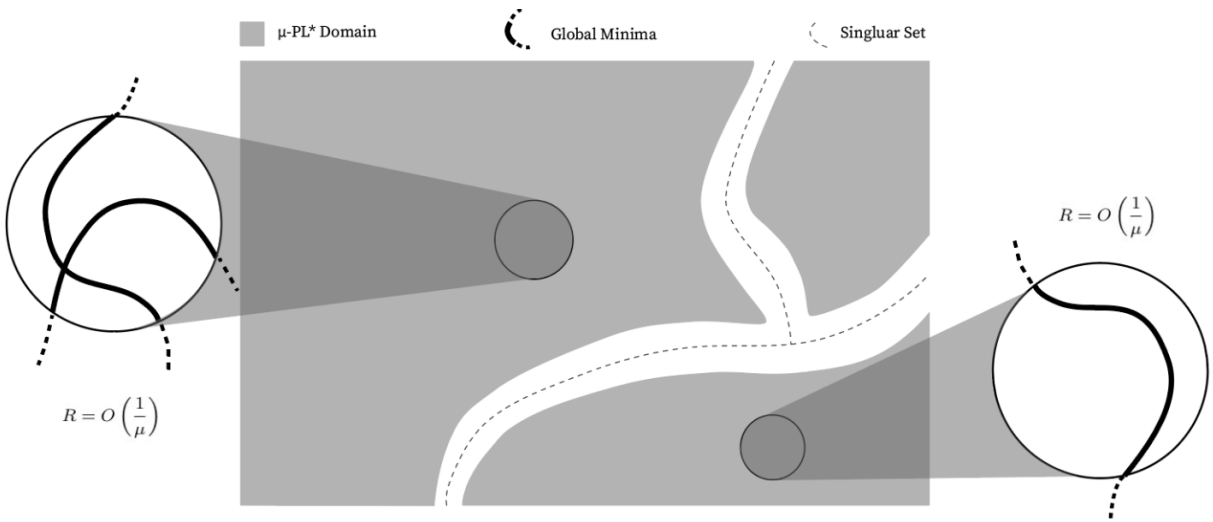


Figure 9: The loss function $\mathcal{L}(\mathbf{w})$ is μ-PL* inside the shaded domain. Singular set correspond to parameters $\mathbf{w}$ with degenerate tangent kernel $K(\mathbf{w})$. Every ball of radius $O(1/\mu)$ within the shaded set intersects with the set of global minima of $\mathcal{L}(\mathbf{w})$, i.e., solutions to $F(\mathbf{w}) = \mathbf{y}$. Figure credit: [51].

Hessian control

- The distance of the tangent kernel on a ball to its initialization is bounded by the spectral norm of the Hessian [4]:

$$\forall \boldsymbol{w} \in \mathcal{B}(\boldsymbol{w}_0, R), \|K(\boldsymbol{w}) - K(\boldsymbol{w}_0)\| \leq \mathcal{O}\left(R \max_{\xi \in \mathcal{B}(\boldsymbol{w}_0, R)} \|\boldsymbol{H}(\xi)\|\right)$$

- Thus, $\forall \boldsymbol{w} \in \mathcal{B}(\boldsymbol{w}_0, R)$, we have

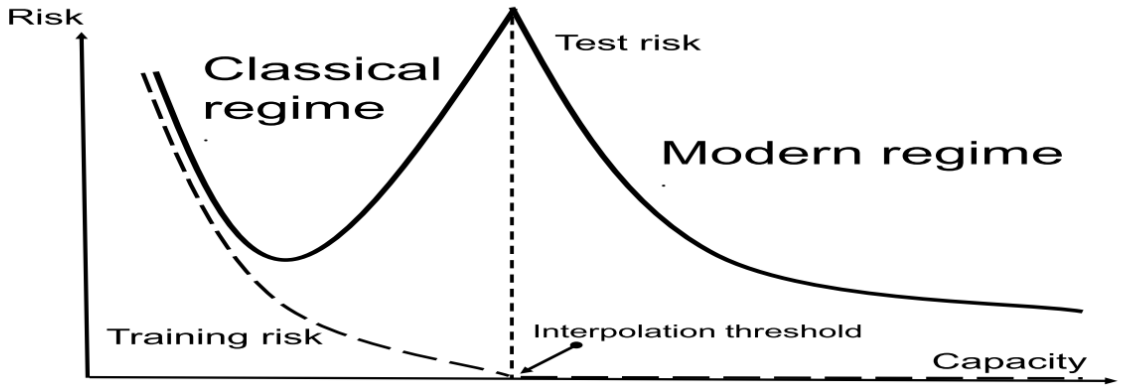
$$\|\lambda_{\min}(K(\boldsymbol{w})) - \lambda_{\min}(K(\boldsymbol{w}_0))\| \leq \mathcal{O}\left(R \max_{\xi \in \mathcal{B}(\boldsymbol{w}_0, R)} \|\boldsymbol{H}(\xi)\|\right)$$

- Note that $\sup_{x \in \mathcal{B}(\boldsymbol{w}_0, R)} \|\boldsymbol{H}(\xi)\| = \tilde{\mathcal{O}}(\frac{1}{\sqrt{m}})$ and $\lambda_{\min}(K(\boldsymbol{w}_0)) = \mathcal{O}(1)$ [5] independent of m
- Thus, $\lambda_{\min}(K(\boldsymbol{w}))$ is away from 0 for sufficiently wide network

[4] Liu et al., "On the linearity of large nonlinear models: when and why the tangent kernel is constant", 2020.

[5] Du et al., "GD provably optimizes over-parameterized NNs", '18.

Take-home insights



	Classical (under-parameterized)	Modern (over-parameterized)
Generalization curve	U-shaped	Descending
Optimal model	Bottom of U (hard to find)	Any large model (easy to find)
Optimization landscape:	Locally convex Minimizers locally unique	Not locally convex Manifolds of minimizers Satisfies PL* condition
GD/SGD convergence	GD converges to local min SGD w. fixed learning rate does not converge	GD/SGD converge to global min SGD w. fixed learning rate converges exponentially
Adversarial examples	?	Unavoidable
Transition to linearity		Wide networks w. linear last layer