# The Mathematics of Large Machine Learning Models

Andrea Montanari

Stanford University

August 11, 2025

# Prelude: The role of theory

# Current state of Machine Learning

I started working in high-dim statistics and ML and around 2008

- **2008**: $\approx 1.5 \cdot 10^3$ papers submitted each year in top 3 venues
- **2024**: $\approx 3 \cdot 10^4$ papers submitted each year in top 3 venues
- Major discoveries
  - Machines can learn language
  - Learn from examples during conversations
  - Exhibit step-by-step problem-solving

# Current state of Machine Learning

I started working in high-dim statistics and ML and around 2008

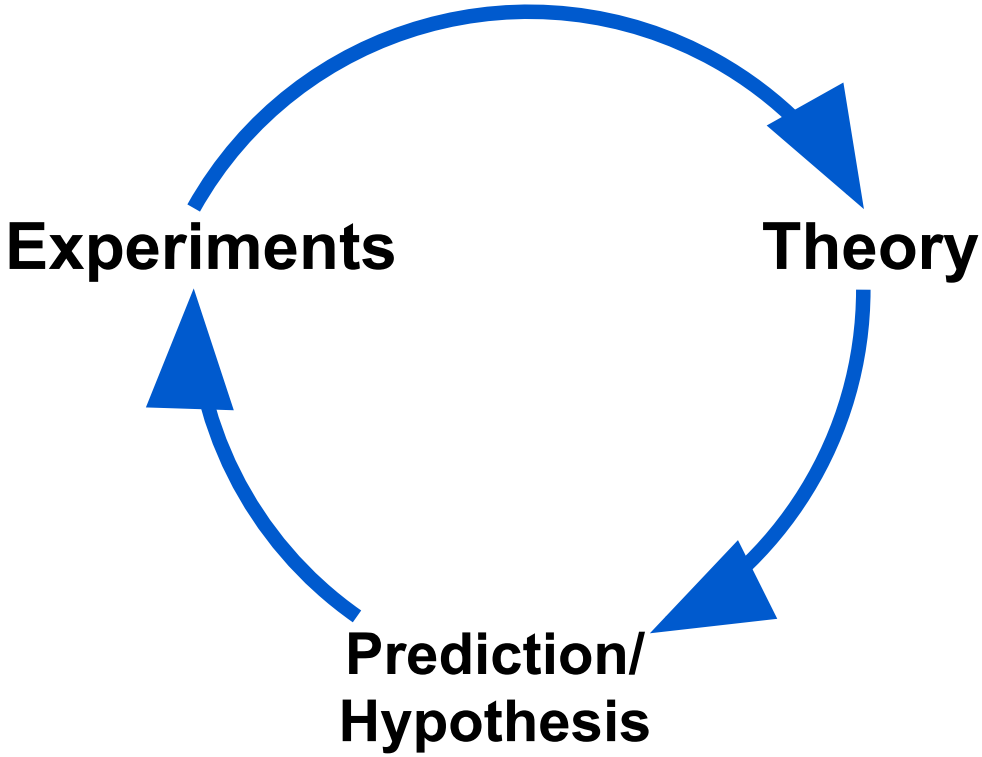
- **2008**: $\approx 1.5 \cdot 10^3$ papers submitted each year in top 3 venues
- **2024**: $\approx 3 \cdot 10^4$ papers submitted each year in top 3 venues
- Major discoveries
  - Machines can learn language
  - Learn from examples during conversations
  - Exhibit step-by-step problem-solving

# Current state of Machine Learning

I started working in high-dim statistics and ML and around 2008

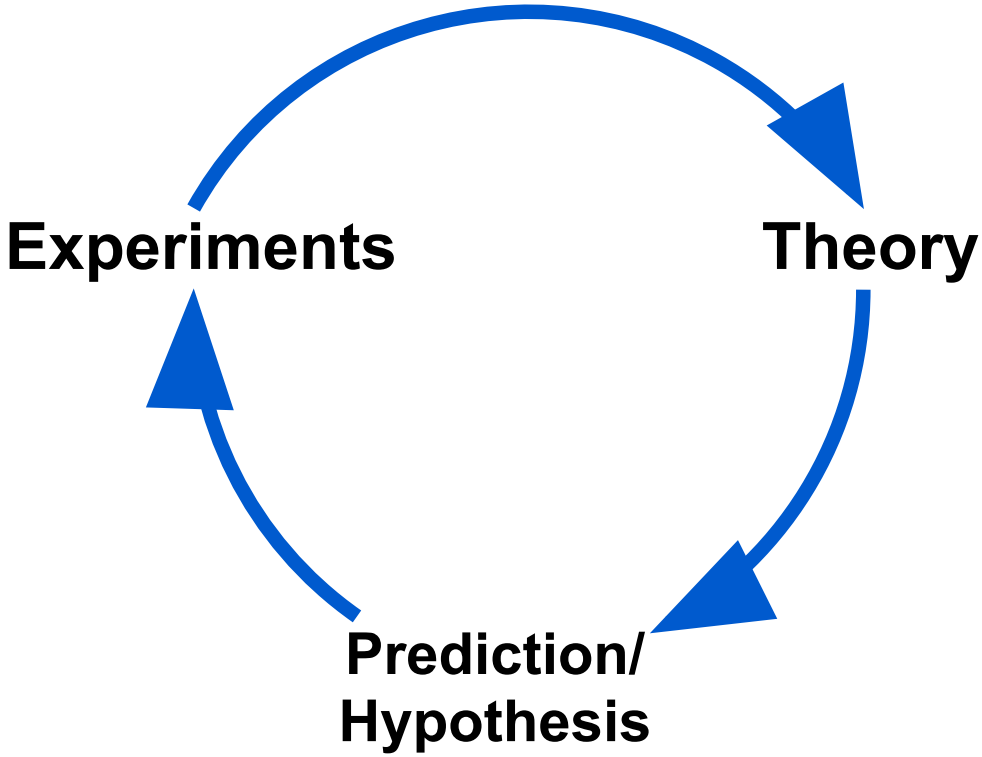
- **2008**: $\approx 1.5 \cdot 10^3$ papers submitted each year in top 3 venues
- **2024**: $\approx 3 \cdot 10^4$ papers submitted each year in top 3 venues
- Major discoveries
  - Machines can learn language
  - Learn from examples during conversations
  - Exhibit step-by-step problem-solving

# A success of the scientific method?



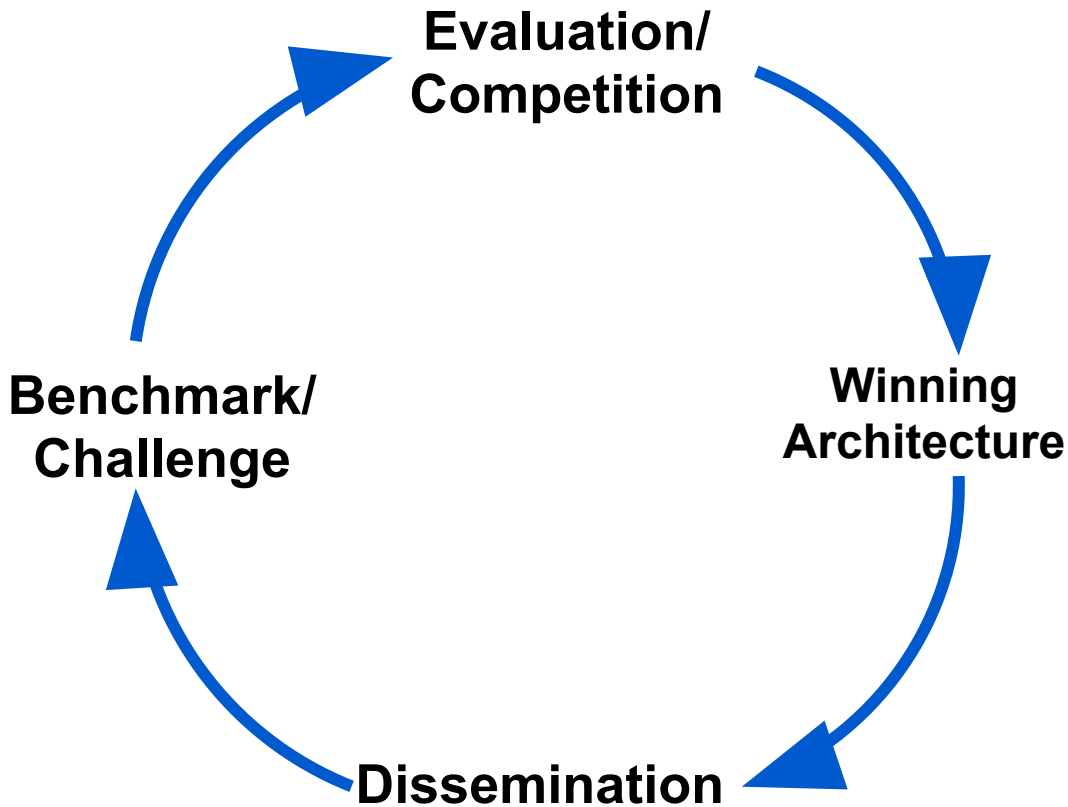

Not really!

# A success of the scientific method?

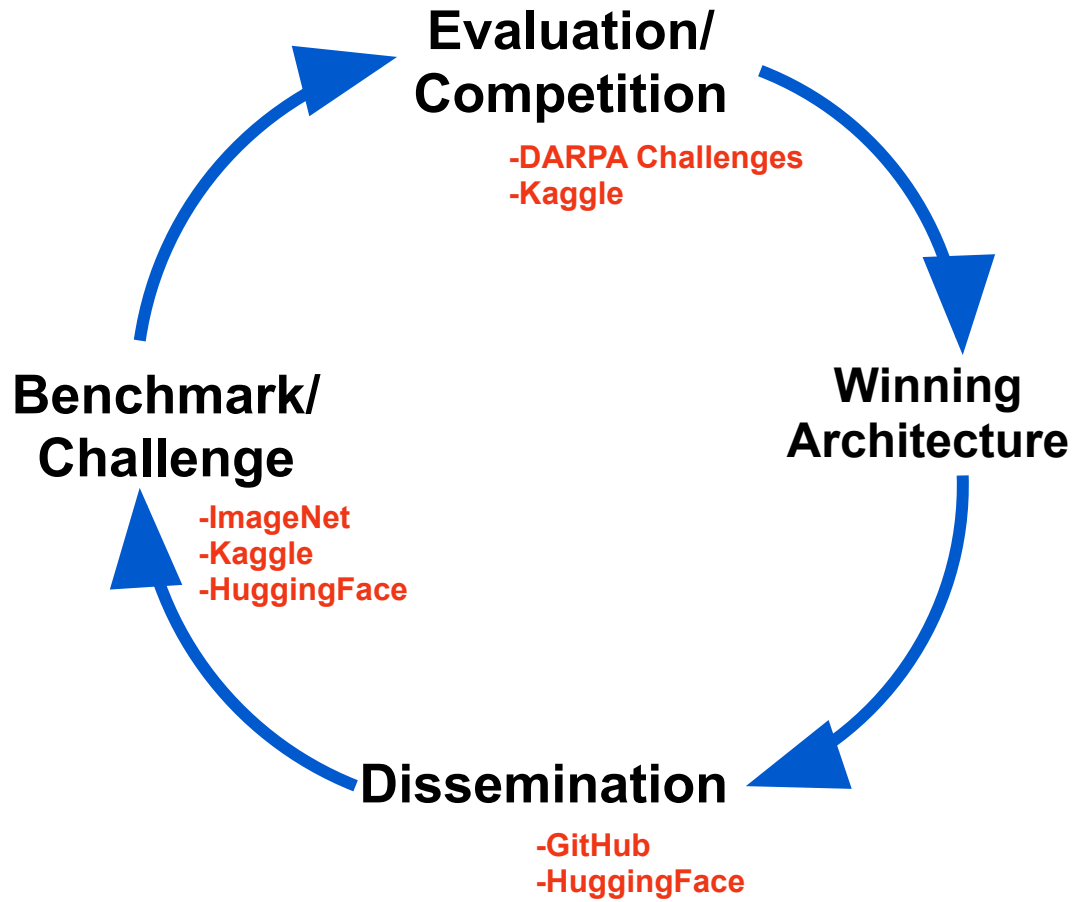



Not really!

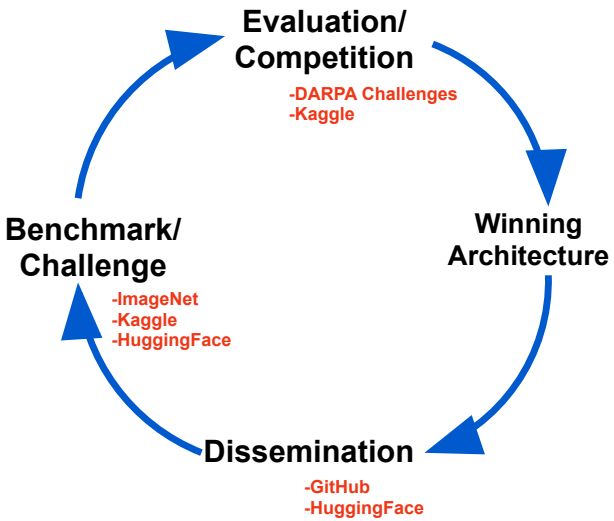
# The 'AI research method'

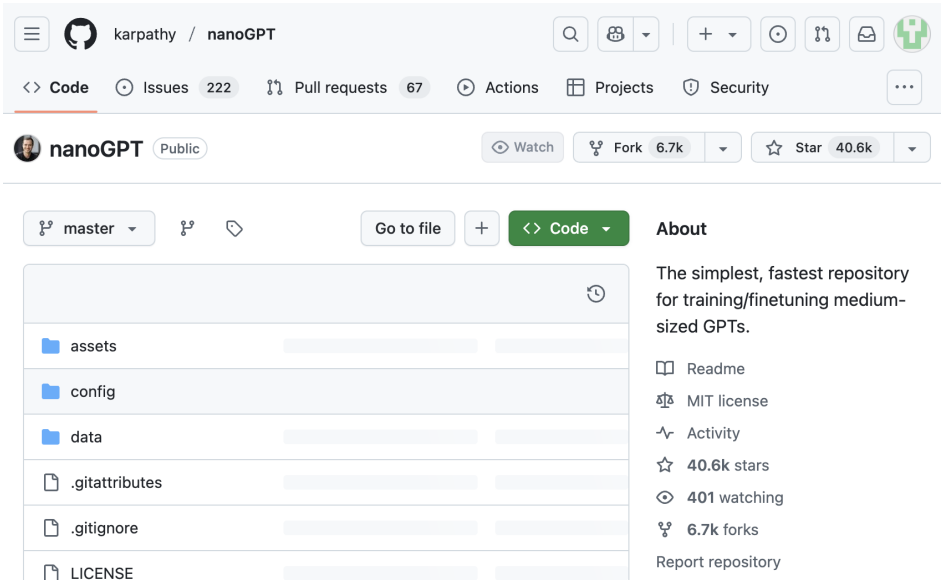


# The AI research method and its institutions

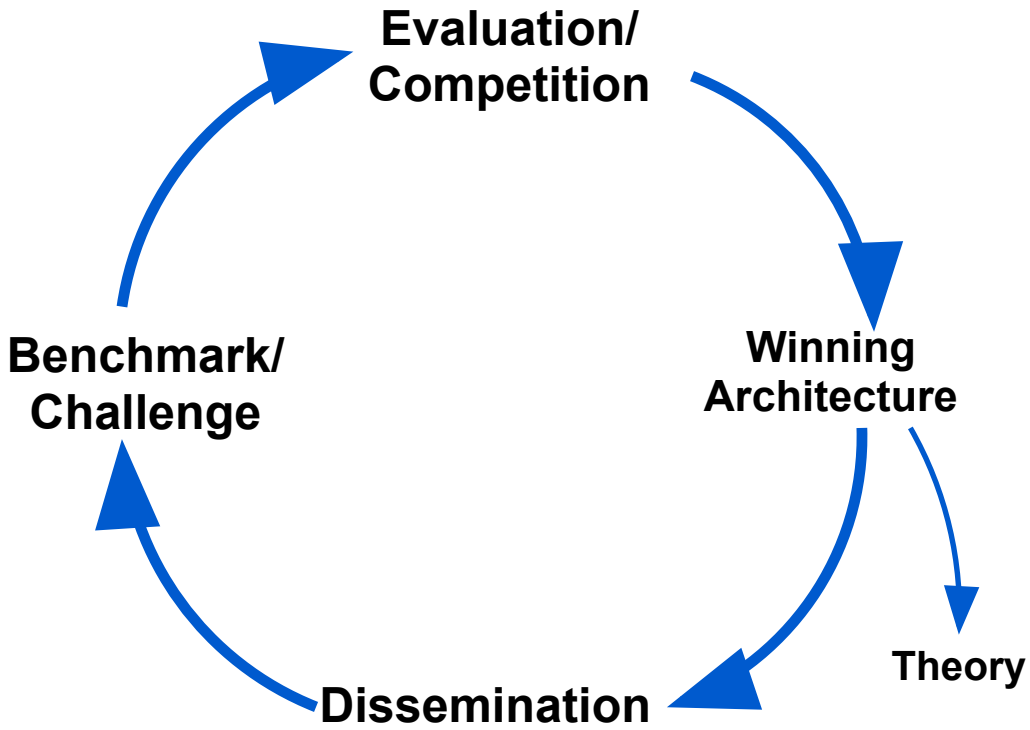


# The AI research method and its institutions



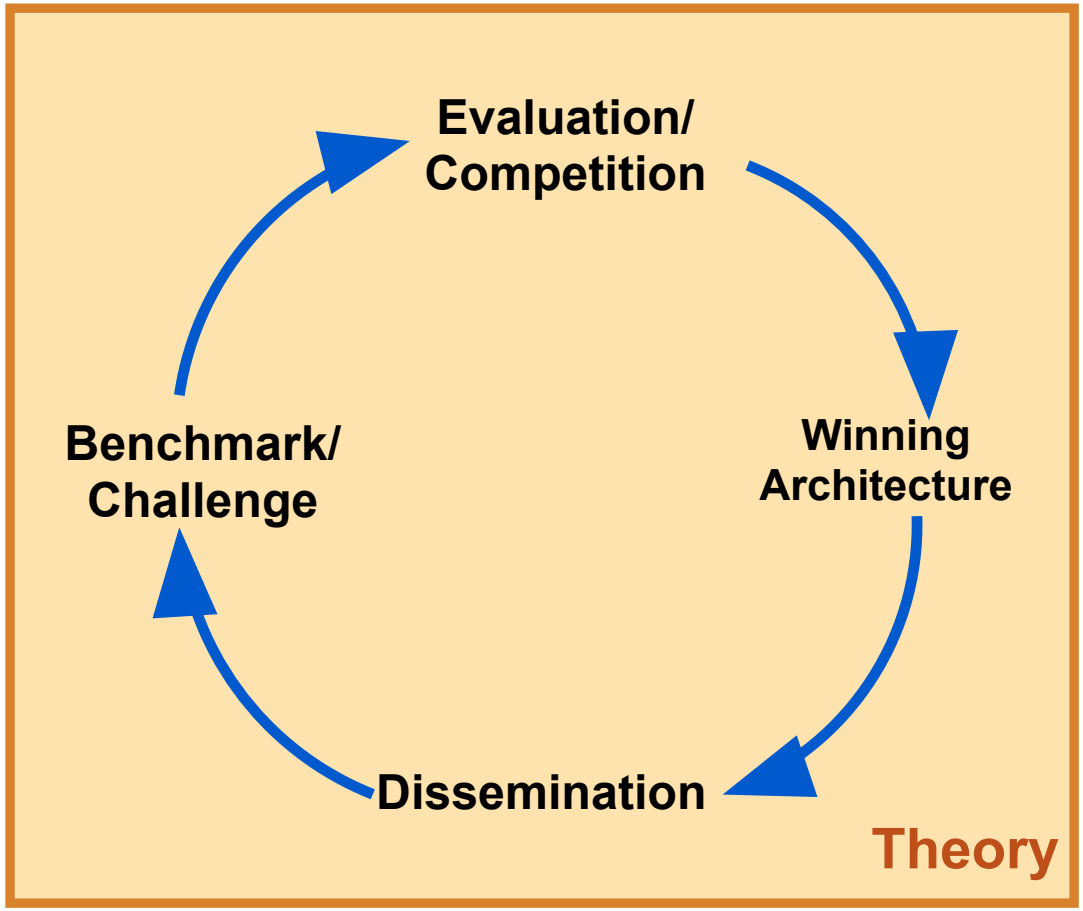



# And theory?




- Explain why architecture X is currently winning the race

# And theory?




- Ask very fundamental questions about very general phenomena
  - Does more data always help? How much?
  - Do more complex models always help? How much?

# A few theoretical successes

- Overfitting and generalization
- Scaling
- Generic phenomena in (stochastic) gradient descent
- Benchmarking/uncertainty quantification
- ...

# A few theoretical successes

- **Overfitting and generalization**
- Scaling
- Generic phenomena in (stochastic) gradient descent
- Benchmarking/uncertainty quantification
- ...

# Outline

# The generalization problem

# History

**Before 1960:**

- A program defines the input-output relation

**Rosenblatt, 1960:**

- Input-output pairs $(x_i, y_i)$, $i \leq n$
- Let the machine **learn** the relation
- Proof of concept: 'Perceptron'

# History

**Before 1960:**

- A program defines the input-output relation

**Rosenblatt, 1960:**

- Input-output pairs $(\mathbf{x}_i, y_i)$, $i \leq n$
- Let the machine **learn** the relation
- Proof of concept: 'Perceptron'

# History

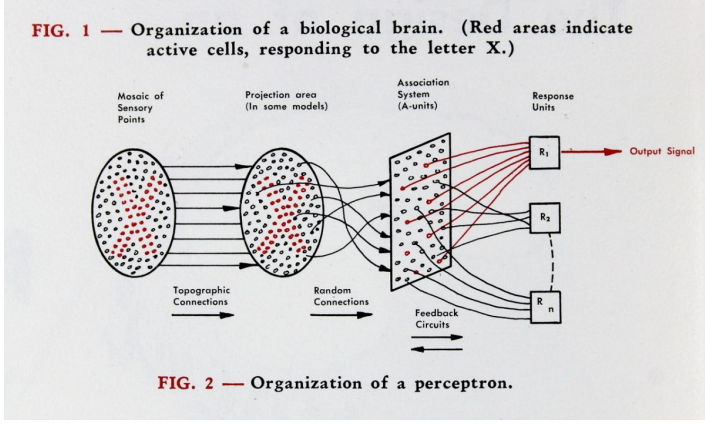
**Before 1960:**

- A program defines the input-output relation

**Rosenblatt, 1960:**

- Input-output pairs $(\mathbf{x}_i, y_i)$, $i \leq n$
- Let the machine **learn** the relation
- Proof of concept: 'Perceptron'

Frank Rosenblatt




Perceptron

# History

**Before 1960:**

- A program defines the input-output relation

**Rosenblatt, 1960:**

- Input-output pairs $(\mathbf{x}_i, y_i)$, $i \leq n$
- Let the machine **learn** the relation
- Proof of concept: 'Perceptron'

Why does it work? What does it mean to learn?

# History

**Before 1960:**

- A computer is programmed
- A program needs to determine the input-output relation

**Rosenblatt 1960:**

- Show the machine input-output pairs $(\mathbf{x}_i, y_i)$, $i \leq n$
- Let the machine **learn** the relation
- Proof of concept: Perceptron

**Vapnik–Chervonenkis, 1969-1974:**

- Learning is statistical learning

---

Vapnik studied in Samarkand: closer to Bangalore than Stanford is to MIT.

# History

**Before 1960:**

- A computer is programmed
- A program needs to determine the input-output relation

**Rosenblatt 1960:**

- Show the machine input-output pairs $(\mathbf{x}_i, y_i)$, $i \leq n$
- Let the machine **learn** the relation
- Proof of concept: Perceptron

**Vapnik–Chervonenkis, 1969-1974:**

- Learning is statistical learning

---

Vapnik studied in Samarkand: closer to Bangalore than Stanford is to MIT.

# Statistical learning

- **Data:** $(x_1, y_1), \ldots, (x_n, y_n) \sim_{\text{iid}} \mathbb{P}$
- **Input-output relations:** $f(\cdot; \theta_1), f(\cdot; \theta_2), \ldots, f(\cdot; \theta_M)$.

# Statistical learning

- **Data:** $(x_1, y_1), \ldots, (x_n, y_n) \sim_{\text{iid}} \mathbb{P}$
- **Models:** $f(\cdot; \theta_1), f(\cdot; \theta_2), \ldots, f(\cdot; \theta_M)$.
- **Learning:** $\mathscr{R}(\theta) := \mathbb{E}\{\text{dist}(y_{n+1}, f(x_{n+1}; \theta))\}$:
  - $\hat{\theta}$ selected from data $(x_1, y_1), \ldots, (x_n, y_n)$
  - Minimize test error $\mathscr{R}(\theta)$!

---

Statisticians: 'This is *just* statistics'

# Statistical learning

- **Data:** $(x_1, y_1), \ldots, (x_n, y_n) \sim_{\text{iid}} \mathbb{P}$
- **Models:** $f(\cdot; \theta_1), f(\cdot; \theta_2), \ldots, f(\cdot; \theta_M)$.
- **Learning:** $\mathscr{R}(\theta) := \mathbb{E}\{\text{dist}(y_{n+1}, f(x_{n+1}; \theta))\}$:
  - $\hat{\theta}$ selected from data $(x_1, y_1), \ldots, (x_n, y_n)$
  - Minimize test error $\mathscr{R}(\theta)$!

---

Statisticians: 'This is *just* statistics'

# Statistical learning

- **Data:** $(x_1, y_1), \ldots, (x_n, y_n) \sim_{\text{iid}} \mathbb{P}$
- **Models:** $f(\cdot; \theta_1), f(\cdot; \theta_2), \ldots, f(\cdot; \theta_M)$.
- **Learning:** $\mathscr{R}(\theta) := \mathbb{E}\{\text{dist}(y_{n+1}, f(x_{n+1}; \theta))\}$:
  - $\hat{\theta}$ selected from data $(x_1, y_1), \ldots, (x_n, y_n)$
  - Minimize test error $\mathscr{R}(\theta)$!

---

Statisticians: 'This is *just* statistics'

# Statistical learning

| $\mathrm{dist}(y_i, f(x_i;\theta))$ | $f(\cdot;\theta_1)$ | $f(\cdot;\theta_2)$ | $\cdots$ | $f(\cdot;\theta_M)$ |
|---|---|---|---|---|
| $(y_1, x_1)$ | 0 | 1 | $\cdots$ | 0 |
| $(y_2, x_2)$ | 1 | 0 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $(y_n, x_n)$ | 0 | 1 | $\cdots$ | 1 |
| $(y_{n+1}, x_{n+1})$ | 1 | 0 | $\cdots$ | 1 |

| | $f(\cdot;\theta_1)$ | $f(\cdot;\theta_2)$ | $\cdots$ | $f(\cdot;\theta_M)$ |
|---|---|---|---|---|
| empirical risk | $\widehat{\mathscr{R}}_n(\theta_1)$ | $\widehat{\mathscr{R}}_n(\theta_2)$ | $\cdots$ | $\widehat{\mathscr{R}}_n(\theta_M)$ |
| population risk | $\mathscr{R}(\theta_1)$ | $\mathscr{R}(\theta_2)$ | $\cdots$ | $\mathscr{R}(\theta_M)$ |

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \mathrm{dist}(y_i, f(x_i;\theta)), \qquad \mathscr{R}(\theta) = \mathbb{E}\,\mathrm{dist}(y_{n+1}, f(x_{n+1};\theta)).$$

# Statistical learning

| $\mathrm{dist}(y_i, f(x_i;\theta))$ | $f(\cdot;\theta_1)$ | $f(\cdot;\theta_2)$ | $\cdots$ | $f(\cdot;\theta_M)$ |
|---|---|---|---|---|
| $(y_1, x_1)$ | 0 | 1 | $\cdots$ | 0 |
| $(y_2, x_2)$ | 1 | 0 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $(y_n, x_n)$ | 0 | 1 | $\cdots$ | 1 |
| $(y_{n+1}, x_{n+1})$ | 1 | 0 | $\cdots$ | 1 |

| | $f(\cdot;\theta_1)$ | $f(\cdot;\theta_2)$ | $\cdots$ | $f(\cdot;\theta_M)$ |
|---|---|---|---|---|
| empirical risk | $\widehat{\mathscr{R}}_n(\theta_1)$ | $\widehat{\mathscr{R}}_n(\theta_2)$ | $\cdots$ | $\widehat{\mathscr{R}}_n(\theta_M)$ |
| population risk | $\mathscr{R}(\theta_1)$ | $\mathscr{R}(\theta_2)$ | $\cdots$ | $\mathscr{R}(\theta_M)$ |

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \mathrm{dist}(y_i, f(x_i;\theta)), \qquad \mathscr{R}(\theta) = \mathbb{E}\,\mathrm{dist}(y_{n+1}, f(x_{n+1};\theta)).$$

# Statistical learning

| $\mathrm{dist}(y_i, f(x_i;\theta))$ | $f(\cdot;\theta_1)$ | $f(\cdot;\theta_2)$ | $\cdots$ | $f(\cdot;\theta_M)$ |
|---|---|---|---|---|
| $(y_1, x_1)$ | 0 | 1 | $\cdots$ | 0 |
| $(y_2, x_2)$ | 1 | 0 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $(y_n, x_n)$ | 0 | 1 | $\cdots$ | 1 |
| $(y_{n+1}, x_{n+1})$ | 1 | 0 | $\cdots$ | 1 |

| | $f(\cdot;\theta_1)$ | $f(\cdot;\theta_2)$ | $\cdots$ | $f(\cdot;\theta_M)$ |
|---|---|---|---|---|
| empirical risk | $\widehat{\mathscr{R}}_n(\theta_1)$ | $\widehat{\mathscr{R}}_n(\theta_2)$ | $\cdots$ | $\widehat{\mathscr{R}}_n(\theta_M)$ |
| population risk | $\mathscr{R}(\theta_1)$ | $\mathscr{R}(\theta_2)$ | $\cdots$ | $\mathscr{R}(\theta_M)$ |

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \mathrm{dist}(y_i, f(x_i;\theta))\,, \qquad \mathscr{R}(\theta) = \mathbb{E}\,\mathrm{dist}(y_{n+1}, f(x_{n+1};\theta))\,.$$

# Statistical learning

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \operatorname{dist}(y_i, f(x_i))\,, \qquad \mathscr{R}(\theta) = \mathbb{E}\operatorname{dist}(y_i, f(x_i;\theta))\,.$$

- Central limit theorem: $\left|\mathscr{R}(\theta_i) - \widehat{\mathscr{R}}_n(\theta_i)\right| = O(1/\sqrt{n})$
- Vapnik–Chervonenkis, 1969-1974: $\mathcal{F} := \{f(\,\cdot\,;\theta) : \ \theta \in \mathcal{S}\}$

$$\max_{\theta\in\mathcal{S}} \left|\widehat{\mathscr{R}}_n(\theta) - \mathscr{R}(\theta)\right| \le \text{const.}\sqrt{\frac{d_{VC}(\mathcal{F})}{n}}$$

# Statistical learning

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \operatorname{dist}(y_i, f(x_i))\,, \qquad \mathscr{R}(\theta) = \mathbb{E}\operatorname{dist}(y_i, f(x_i;\theta))\,.$$

- Central limit theorem: $\left|\mathscr{R}(\theta_i) - \widehat{\mathscr{R}}_n(\theta_i)\right| = O(1/\sqrt{n})$
- Vapnik–Chervonenkis, 1969-1974: $\mathcal{F} := \{f(\,\cdot\,;\theta) : \ \theta \in \mathcal{S}\}$

$$\max_{\theta\in\mathcal{S}} \left|\widehat{\mathscr{R}}_n(\theta) - \mathscr{R}(\theta)\right| \le \text{const.}\sqrt{\frac{d_{VC}(\mathcal{F})}{n}}$$

# Statistical learning

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \operatorname{dist}(y_i, f(x_i))\,, \qquad \mathscr{R}(\theta) = \mathbb{E}\operatorname{dist}(y_i, f(x_i;\theta))\,.$$

- Central limit theorem: $\left|\mathscr{R}(\theta_i) - \widehat{\mathscr{R}}_n(\theta_i)\right| = O(1/\sqrt{n})$
- Vapnik–Chervonenkis, 1969-1974: $\mathcal{F} := \{f(\,\cdot\,;\theta) :\ \theta \in \mathcal{S}\}$

$$\max_{\theta\in\mathcal{S}} \left|\widehat{\mathscr{R}}_n(\theta) - \mathscr{R}(\theta)\right| \le \text{const.}\sqrt{\frac{d_{VC}(\mathcal{F})}{n}}$$

# Statistical learning

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \operatorname{dist}(y_i, f(x_i))\,, \qquad \mathscr{R}(\theta) = \mathbb{E}\operatorname{dist}(y_i, f(x_i;\theta))\,.$$

- Central limit theorem: $\left|\mathscr{R}(\theta_i) - \widehat{\mathscr{R}}_n(\theta_i)\right| = O(1/\sqrt{n})$
- Vapnik–Chervonenkis, 1969-1974: $\mathcal{F} := \{f(\,\cdot\,;\theta) : \ \theta \in \mathcal{S}\}$

$$\max_{\theta\in\mathcal{S}} \left|\widehat{\mathscr{R}}_n(\theta) - \mathscr{R}(\theta)\right| \lesssim \sqrt{\frac{d_{VC}(\mathcal{F})}{n}}$$

# Key insight: Uniform CLT-style bounds

$$\max_{\theta \in \mathcal{S}} \left| \widehat{\mathscr{R}}_n(\theta) - \mathscr{R}(\theta) \right| \lesssim \sqrt{\frac{d_{VC}(\mathcal{F})}{n}}$$

- Simultaneously for infinitely many $\theta$'s!
- Bound does not depend on the number of parameters $p$ ($\theta \in \mathbb{R}^p$)
- Only depends on the function class $\mathcal{F}$ via $d_{VC}(\mathcal{F})$.

# Key insight: Uniform CLT-style bounds

$$\max_{f \in \mathcal{F}} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \sqrt{\frac{d_{VC}(\mathcal{F})}{n}}$$

- Simultaneously for infinitely many $\theta$'s!
- Bound does not depend on the number of parameters $p$ ($\theta \in \mathbb{R}^p$)
- Only depends on the function class $\mathcal{F}$ via $d_{VC}(\mathcal{F})$.

# A couple of examples

**Example 1:** Linear model

$$\mathcal{F}_{\text{lin}}(\rho) := \left\{ f(x;\theta) = \langle \theta, x \rangle \ : \ \ \|\theta\| \le \rho \right\},$$

$$\max_{f \in \mathcal{F}_{\text{lin}}(\rho)} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \rho \sqrt{\frac{d}{n}}.$$

**Example 2:** 2-layer networks [Bartlett 1996]

$$\mathcal{F}_{\text{2lnn}}(\gamma) := \left\{ f(x;\theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle) \ : \ \ \|w_i\| = 1 \forall i \in [N], \|a\|_1 \le \gamma \right\},$$

$$\max_{f \in \mathcal{F}_{\text{2lnn}}(\gamma)} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \gamma \sqrt{\frac{d}{n}}. \qquad \text{Independent of } N!$$

# A couple of examples

**Example 1:** Linear model

$$\mathcal{F}_{\text{lin}}(\rho) := \left\{ f(x;\theta) = \langle \theta, x \rangle \ : \ \ \|\theta\| \le \rho \right\},$$

$$\max_{f \in \mathcal{F}_{\text{lin}}(\rho)} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \rho \sqrt{\frac{d}{n}}.$$

**Example 2:** 2-layer networks [Bartlett 1996]

$$\mathcal{F}_{\text{2lnn}}(\gamma) := \left\{ f(x;\theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle) \ : \ \ \|w_i\| = 1 \forall i \in [N], \|a\|_1 \le \gamma \right\},$$

$$\max_{f \in \mathcal{F}_{\text{2lnn}}(\gamma)} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \gamma \sqrt{\frac{d}{n}}.$$

Independent of N!

# A couple of examples

**Example 1:** Linear model

$$\mathcal{F}_{\text{lin}}(\rho) := \left\{ f(x;\theta) = \langle \theta, x \rangle \ : \ \ \|\theta\| \le \rho \right\},$$

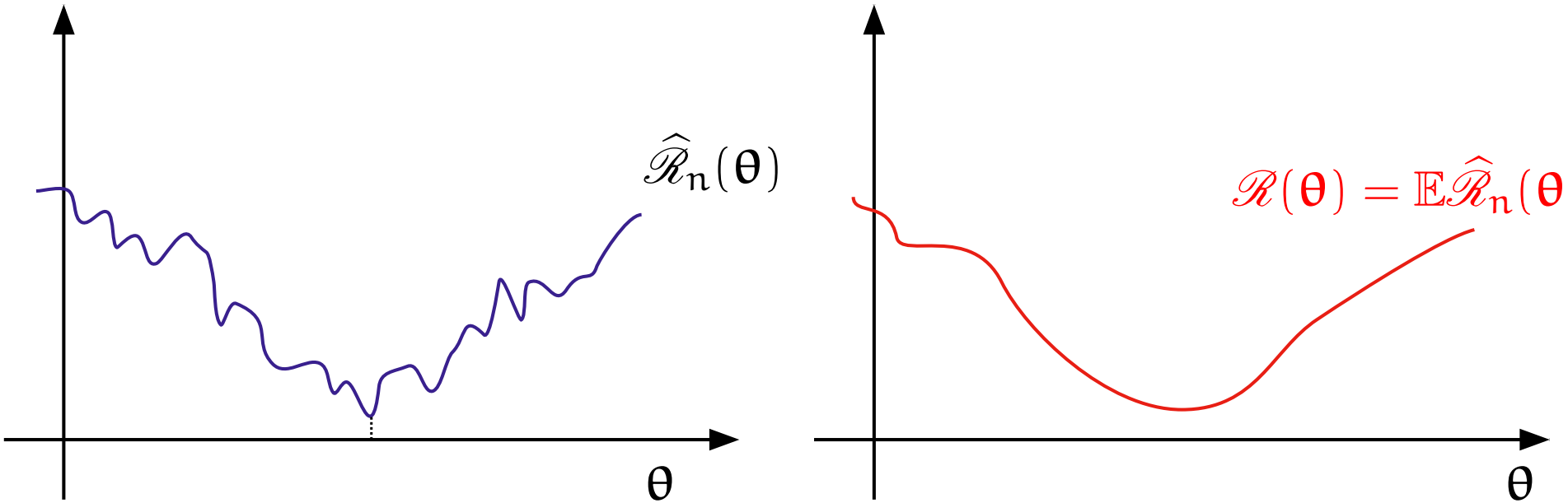
$$\max_{f \in \mathcal{F}_{\text{lin}}(\rho)} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \rho \sqrt{\frac{d}{n}}.$$

**Example 2:** 2-layer networks [Bartlett 1996]

$$\mathcal{F}_{\text{2lnn}}(\gamma) := \left\{ f(x;\theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle) \ : \ \ \|w_i\| = 1 \forall i \in [N], \|a\|_1 \le \gamma \right\},$$

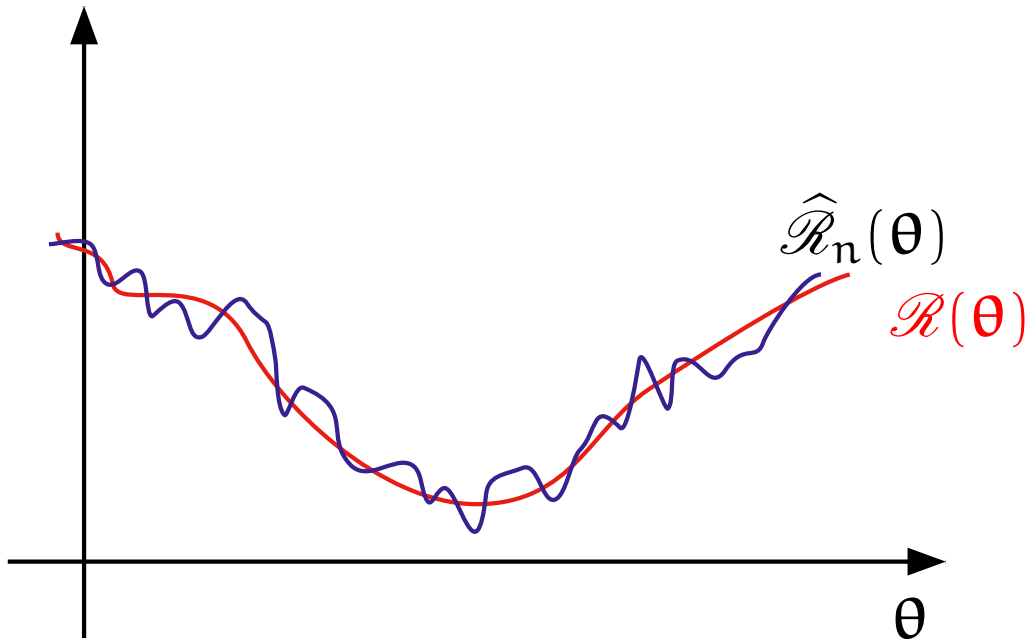
$$\max_{f \in \mathcal{F}_{\text{2lnn}}(\gamma)} \left| \widehat{\mathscr{R}}_n(f) - \mathscr{R}(f) \right| \lesssim \gamma \sqrt{\frac{d}{n}}. \qquad \text{Independent of } N!$$

# Statistical learning: Implications



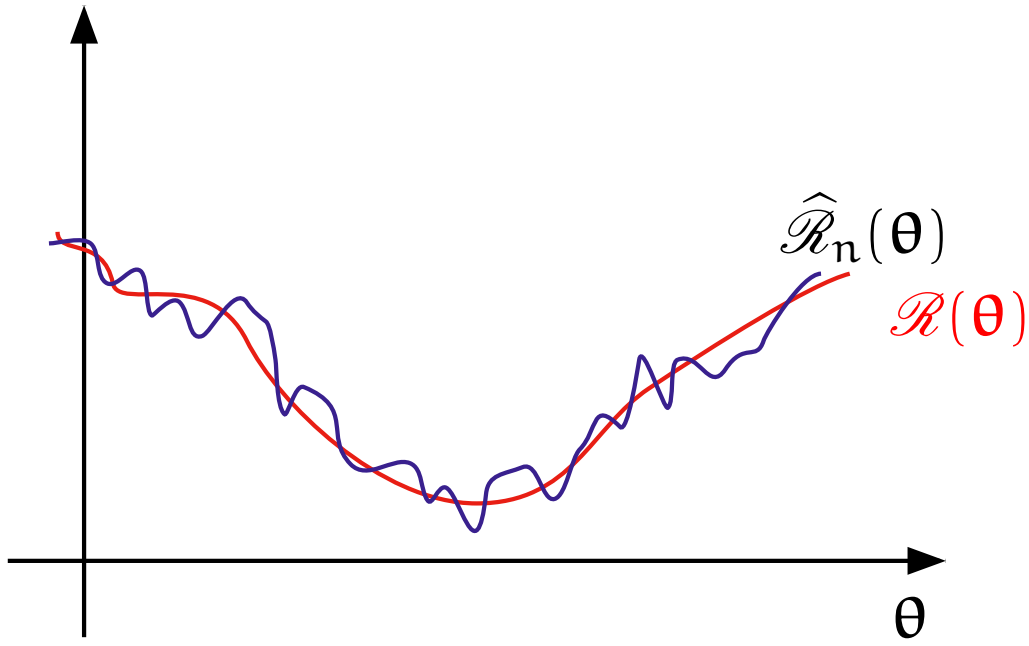

- Want to minimize $\mathscr{R}(\theta) := \mathbb{E}\{\ell(y_{n+1}, f(x_{n+1}; \theta))\}$ $\quad \ell = \text{dist}$
- We have access to $\widehat{\mathscr{R}}_n(\theta) := n^{-1} \sum_{i \le n} \ell(y_i, f(x_i; \theta))$

# Statistical learning: Implications




- Vapnik–Chervonenkis

# Statistical learning: Implications

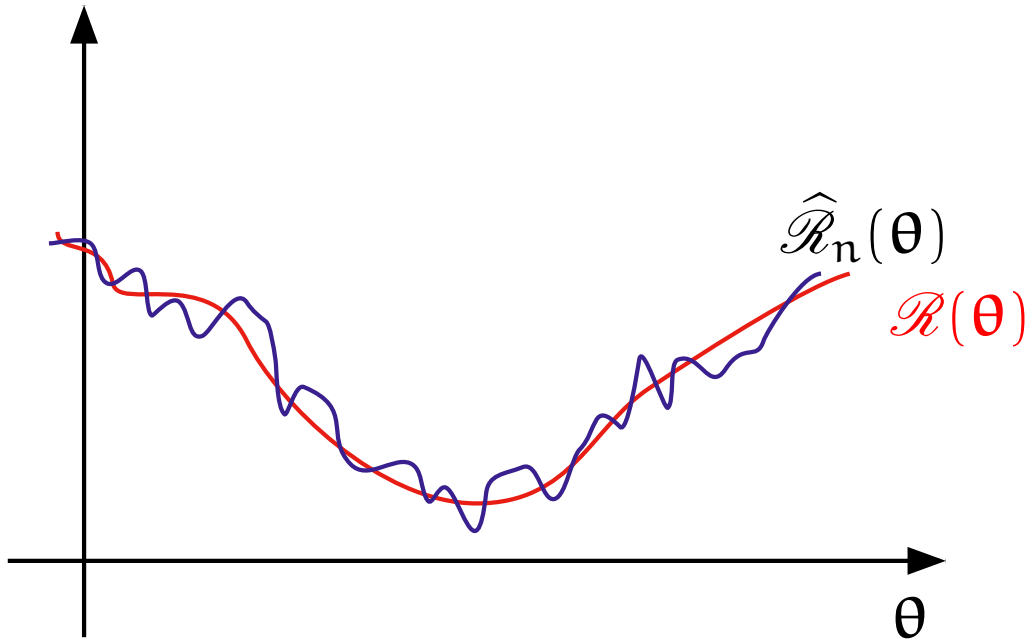



- Vapnik–Chervonenkis:

$$\max_{\theta \in \mathcal{S}} \left| \mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta) \right| = O\left( \sqrt{\frac{d_{VC}(\mathcal{F})}{n}} \right)$$

- $\Rightarrow$ Can minimize $\widehat{\mathscr{R}}_n(\theta)$ instead of $\mathscr{R}(\theta)$
- Choose $\mathcal{S}$ so that $d_{VC}(\mathcal{S}) \ll n$

# Statistical learning: Implications

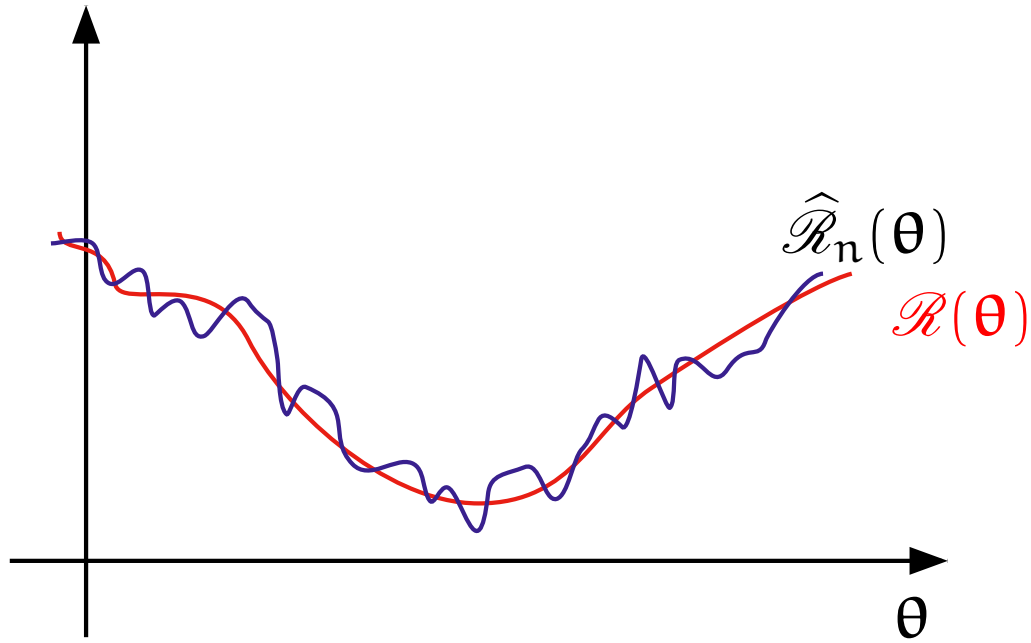



- Vapnik–Chervonenkis:

$$\max_{\theta \in \mathcal{S}} \left| \mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta) \right| = O\left( \sqrt{\frac{d_{VC}(\mathcal{F})}{n}} \right)$$

- $\Rightarrow$ Can minimize $\widehat{\mathscr{R}}_n(\theta)$ instead of $\mathscr{R}(\theta)$
- Choose $\mathcal{S}$ so that $d_{VC}(\mathcal{S}) \ll n$

# Statistical learning: Implications

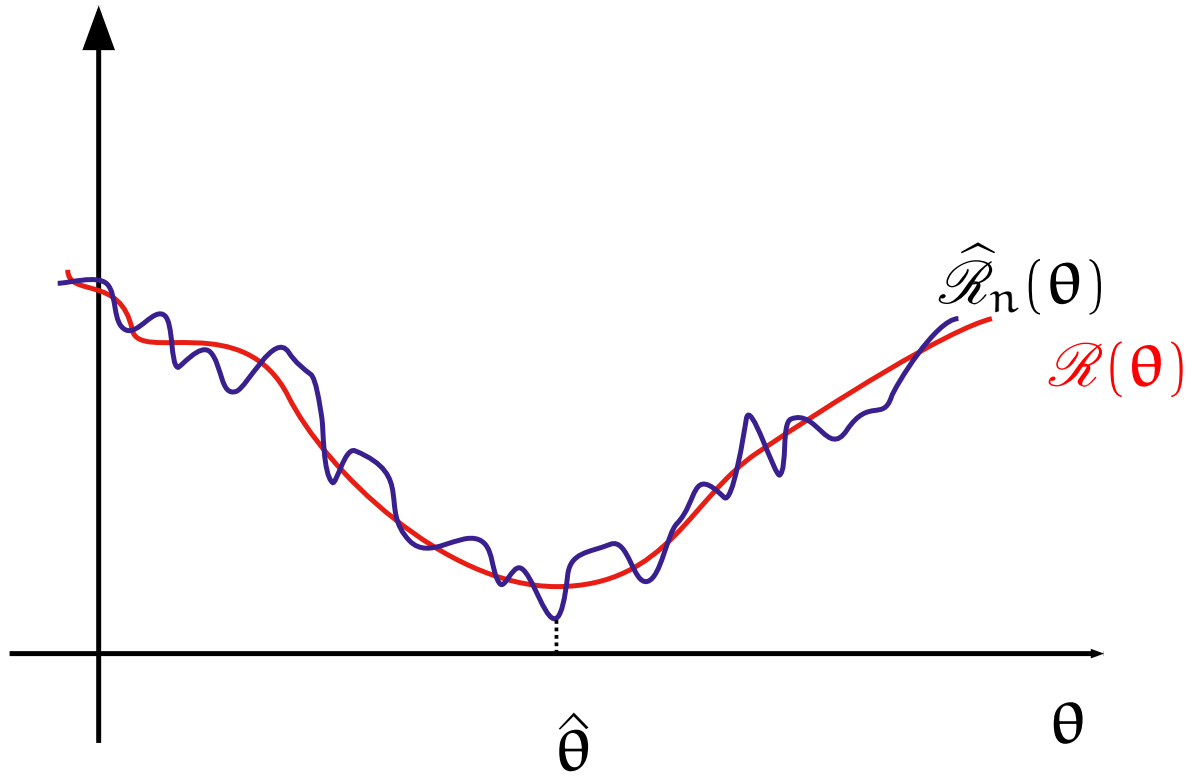



- Vapnik–Chervonenkis:

$$\max_{\theta \in \mathcal{S}} \left| \mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta) \right| = O\left( \sqrt{\frac{d_{VC}(\mathcal{F})}{n}} \right)$$

- $\Rightarrow$ Can minimize $\widehat{\mathscr{R}}_n(\theta)$ instead of $\mathscr{R}(\theta)$
- Choose $\mathcal{S}$ so that $d_{VC}(\mathcal{S}) \ll n$

# Statistical learning

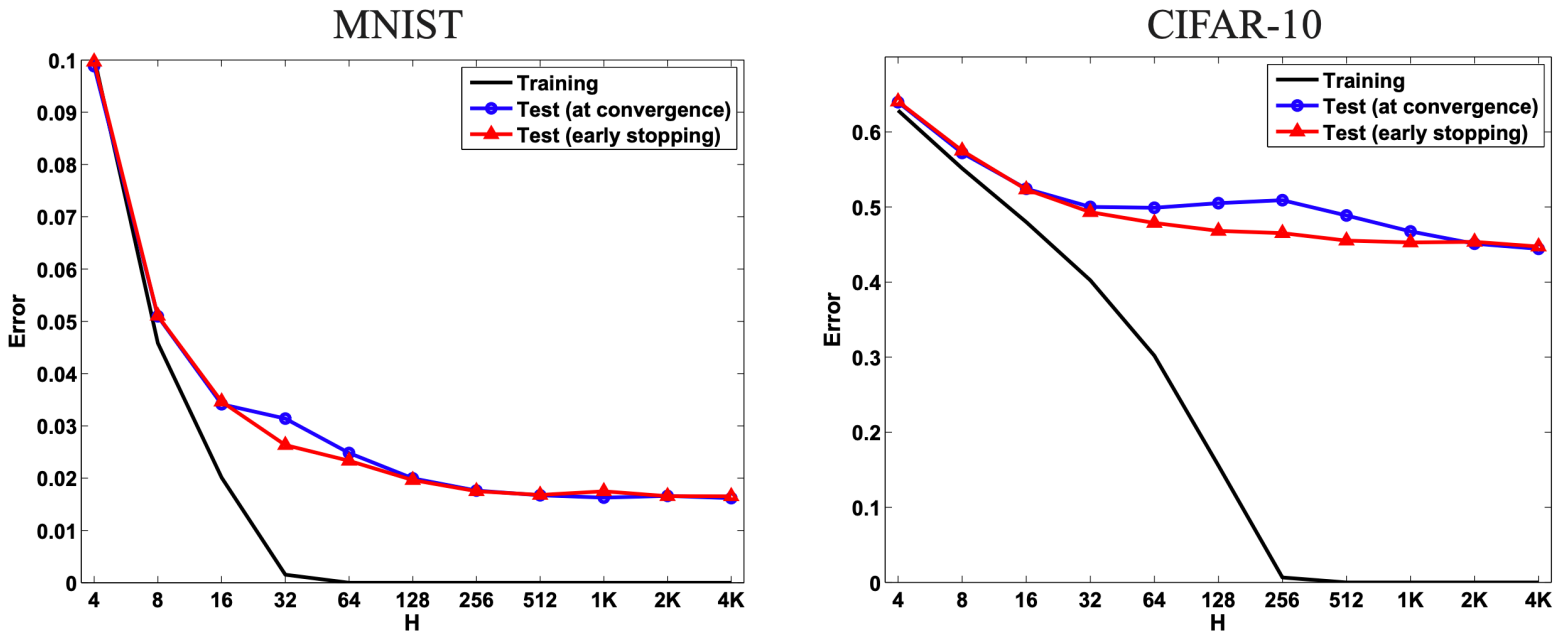



- Learned model $\hat{\theta}$

$$\text{Generalization Error} := \mathscr{R}(\hat{\theta}) - \widehat{\mathscr{R}}_n(\hat{\theta})$$

## Enters modern machine learning

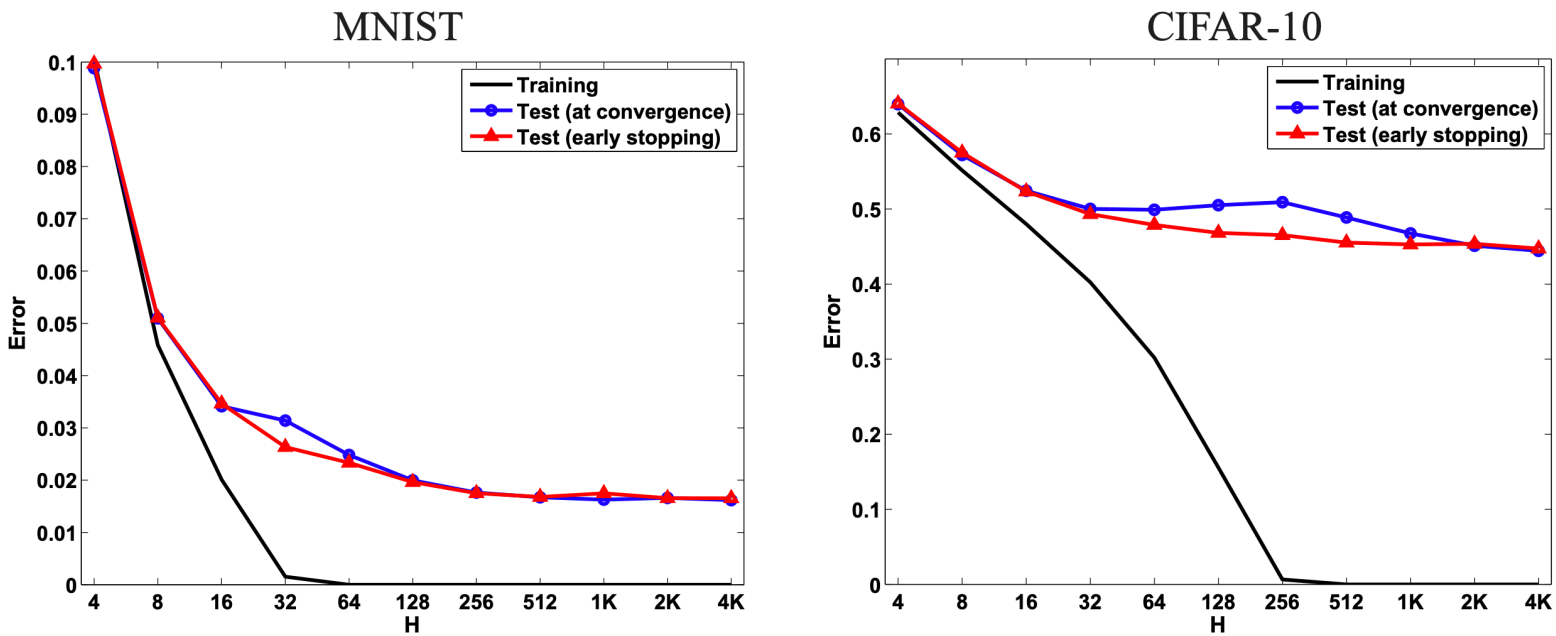
# A small experiment




- x-axis $\propto$ Number of parameters
- Converges to vanishing training error
- Resulting weights depend on the initialization

MNIST: 50,000 images 10 classes; 2-layers. Cross-entropy [Neyshabur, Tomioka, Srebro, 2014]

# A small experiment

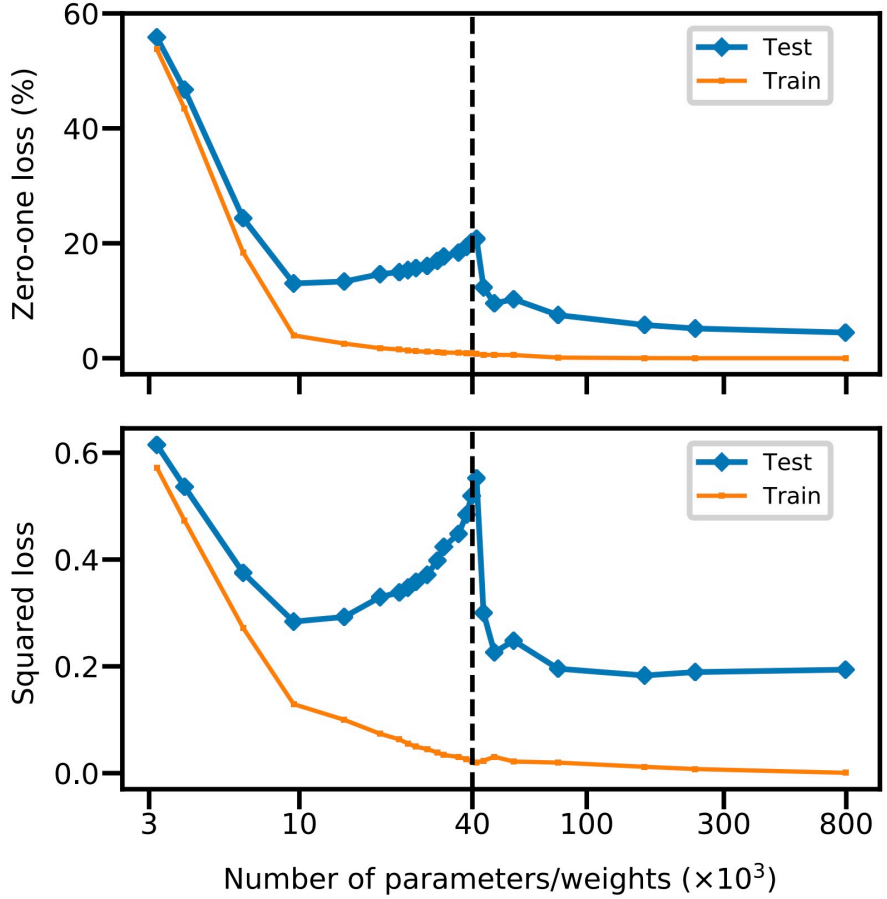



- x-axis $\propto$ Number of parameters
- Converges to vanishing training error
- Resulting weights depend on the initialization

---

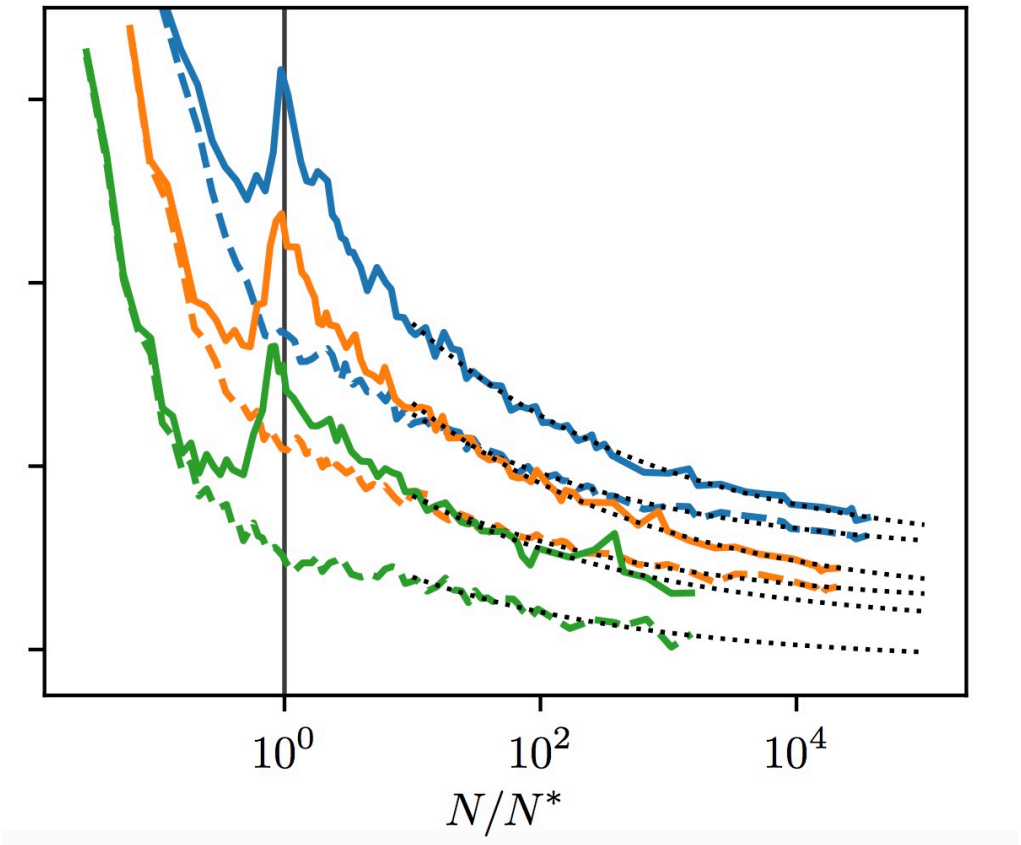
MNIST: 50,000 images 10 classes; 2-layers. Cross-entropy [Neyshabur, Tomioka, Srebro, 2014]

# Another version of the same experiment



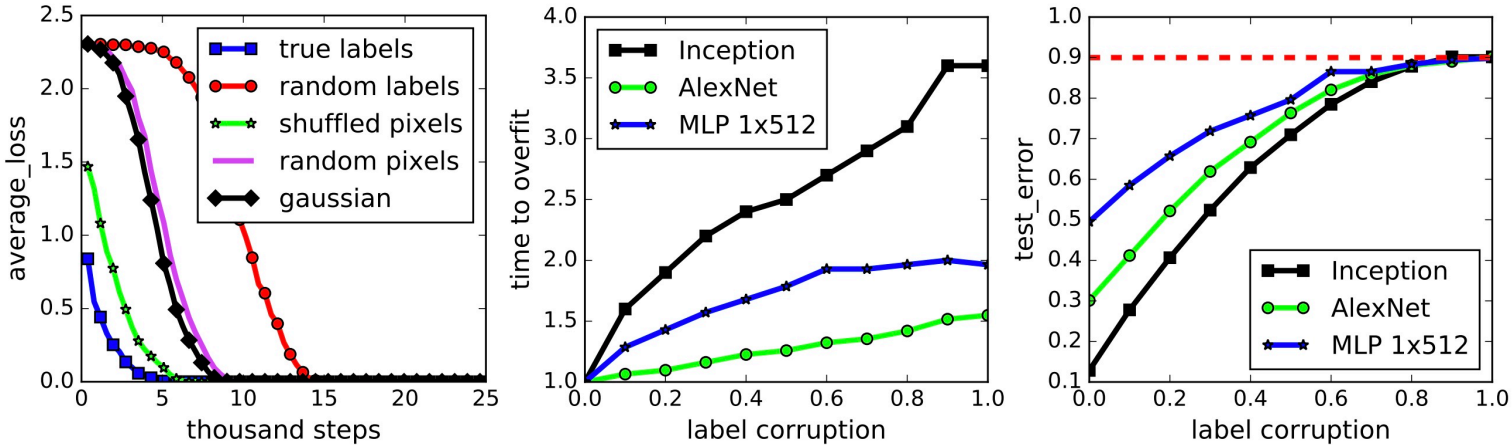

MNIST: 4,000 images 10 classes; 2-layers. Square loss. Belkin, Hsu, Ma, Mandal, 2018

# Yet another version




MNIST: 50,000 images in 2 different classes. 5-layers Neural Net. Quadratic hinge loss. Spigler, Geiger, d'Ascoli, Sagun, Biroli, Wyart, 2018

# A larger scale experiment

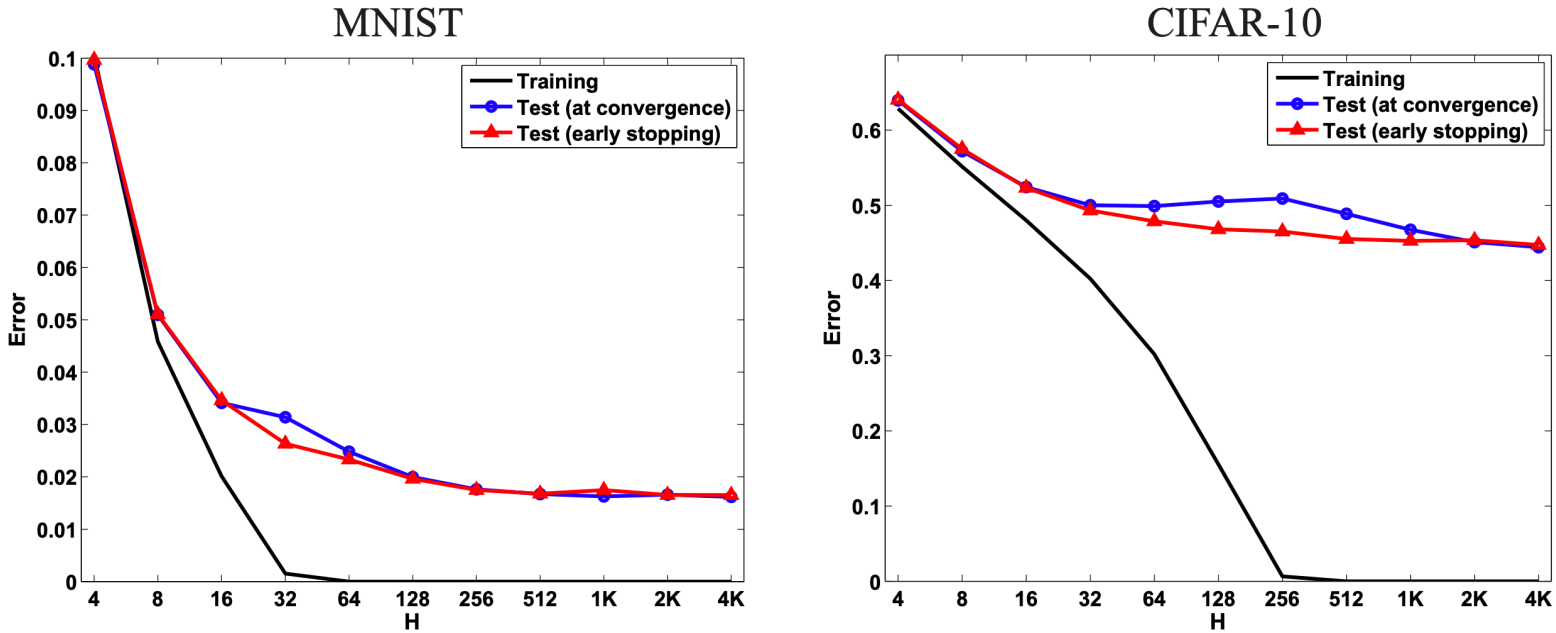



(a) learning curves (b) convergence slowdown (c) generalization error growth

- Model complex enough to 'interpolate' random labels
- Despite this, does well on uncorrupted test samples
- Test error $\gg$ Train error $\approx 0$

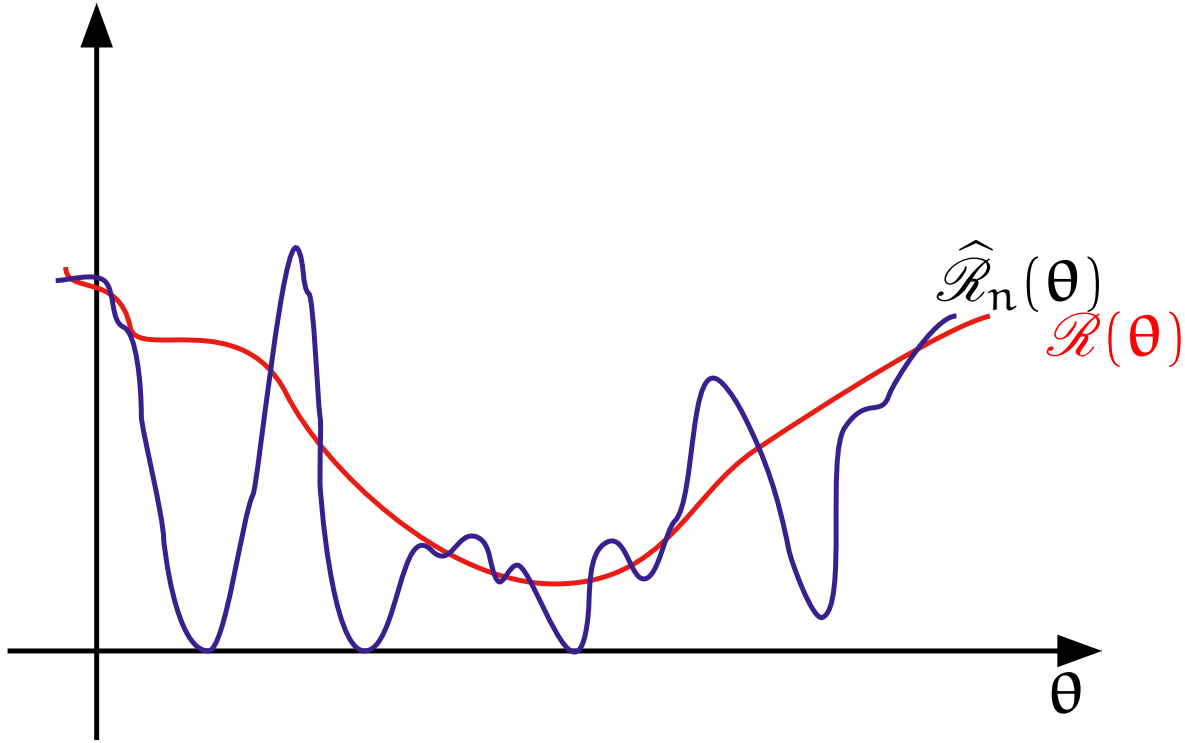
CIFAR-10: 60,000 32x32 images in 10 classes; Inception network. Cross-entropy [Zhang et al., 2016]

# What does this tell us about the landscape?




- Many (near-)global minima with $\widehat{\mathscr{R}}_n(\theta) \approx 0$.
- Close to a random initialization
- At global minima $0 \approx \widehat{\mathscr{R}}_n(\theta) \ll \mathscr{R}(\theta)$.

# The actual landscape: A cartoon

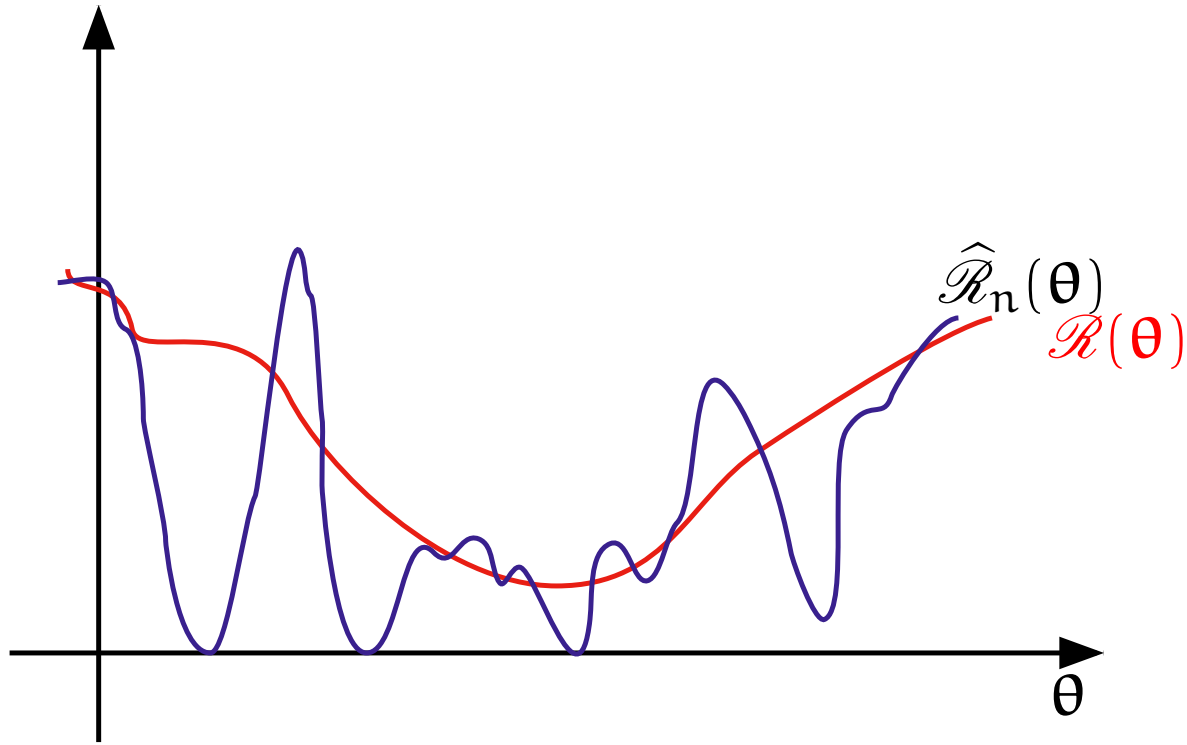



- How does generalization work?
- Why does minimizing $\widehat{\mathscr{R}}_n(\theta)$ yields models with small $\mathscr{R}(\theta)$?

Many 'bad' global optima

Explanation must involve the optimization dynamics

# The actual landscape: A cartoon

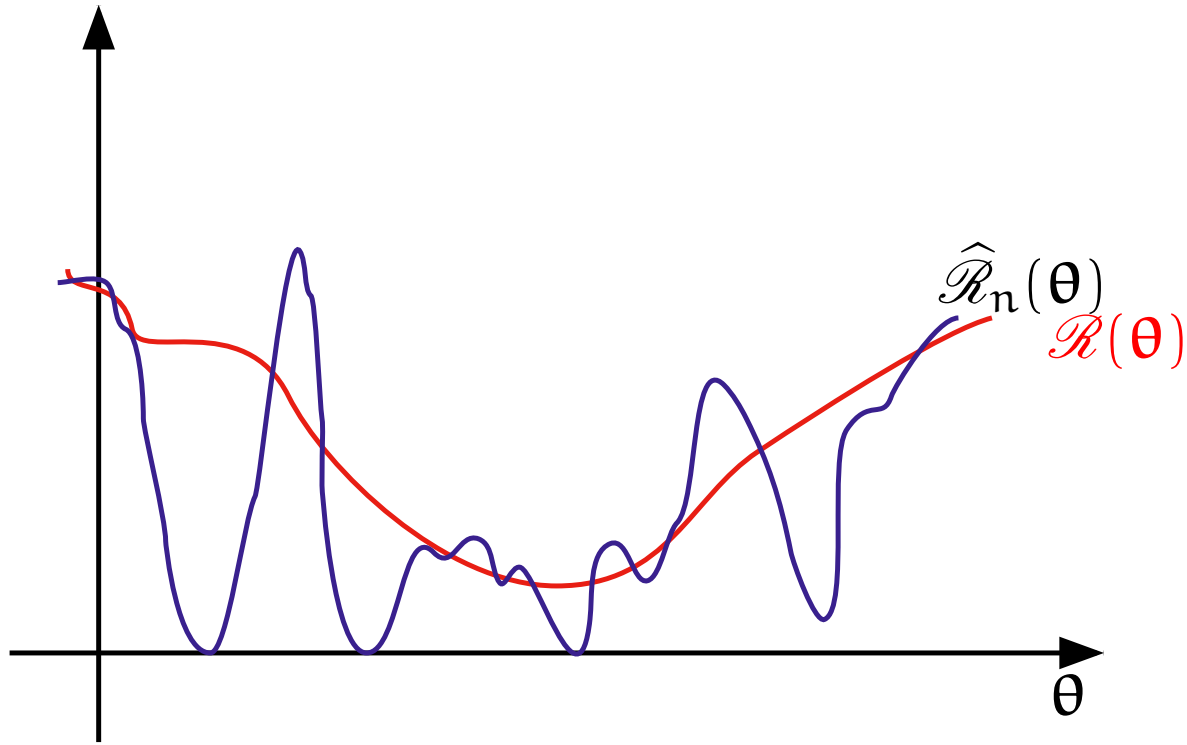



- How does generalization work?
- Why does minimizing $\widehat{\mathscr{R}}_n(\theta)$ yields models with small $\mathscr{R}(\theta)$?

Many 'bad' global optima

Explanation must involve the optimization dynamics

# The actual landscape: A cartoon

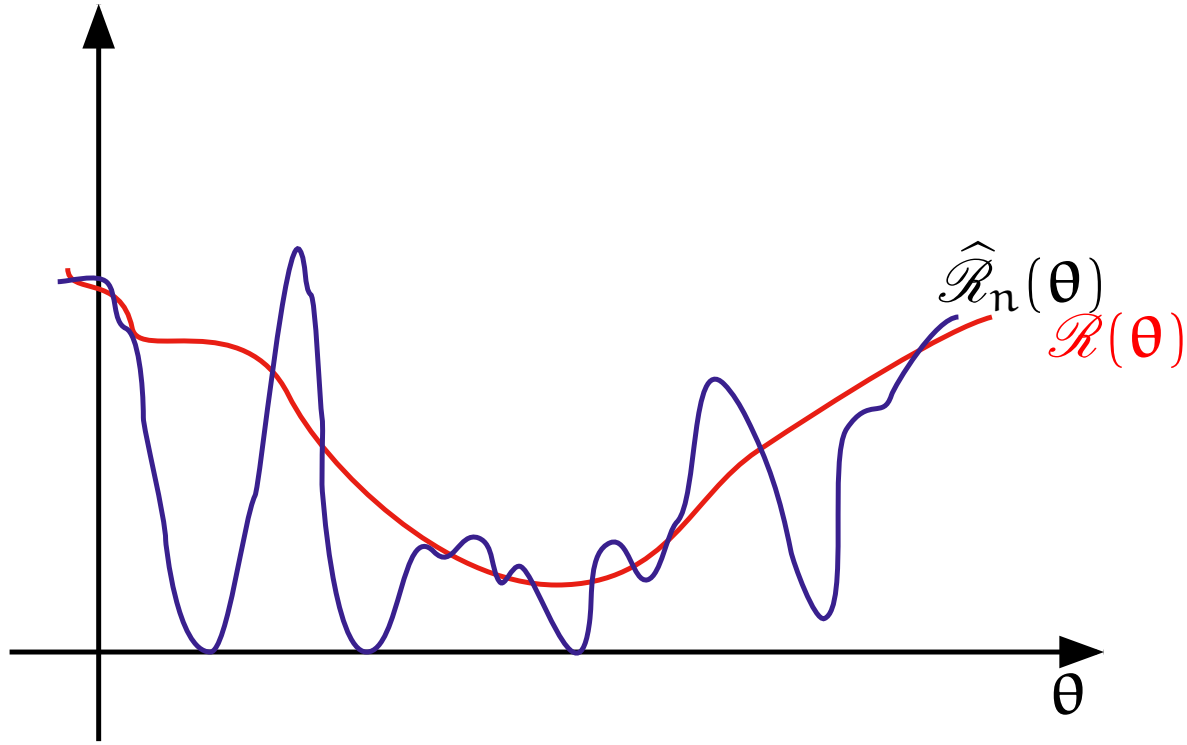



- How does generalization work?
- Why does minimizing $\widehat{\mathscr{R}}_n(\theta)$ yields models with small $\mathscr{R}(\theta)$?

Many 'bad' global optima

Explanation must involve the optimization dynamics

# The actual landscape




## Two theories

**Neural tangent theory:**

Global minima near a random initialization behave well

**Feature learning:**

Few steps of gradient descent sufficient to converge close to $\operatorname{argmin}_\theta \mathscr{R}(\theta)$. (Overfitting cannot take place so quickly.)

# Two-layer neural networks

$$f(x;\theta) = \sum_{j=1}^{N} a_j \sigma(\langle w_j, x\rangle)\,, \quad \theta = ((a_i)_{i\le N}, (w_i)_{i\le N}) \in \mathbb{R}^N \times (\mathbb{S}^{d-1})^N\,.$$

- $p = Nd$
- Square loss train error:

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{2n}\sum_{i=1}^{n} \big(y_i - f(x_i;\theta)\big)^2\,.$$

- Square loss test error:

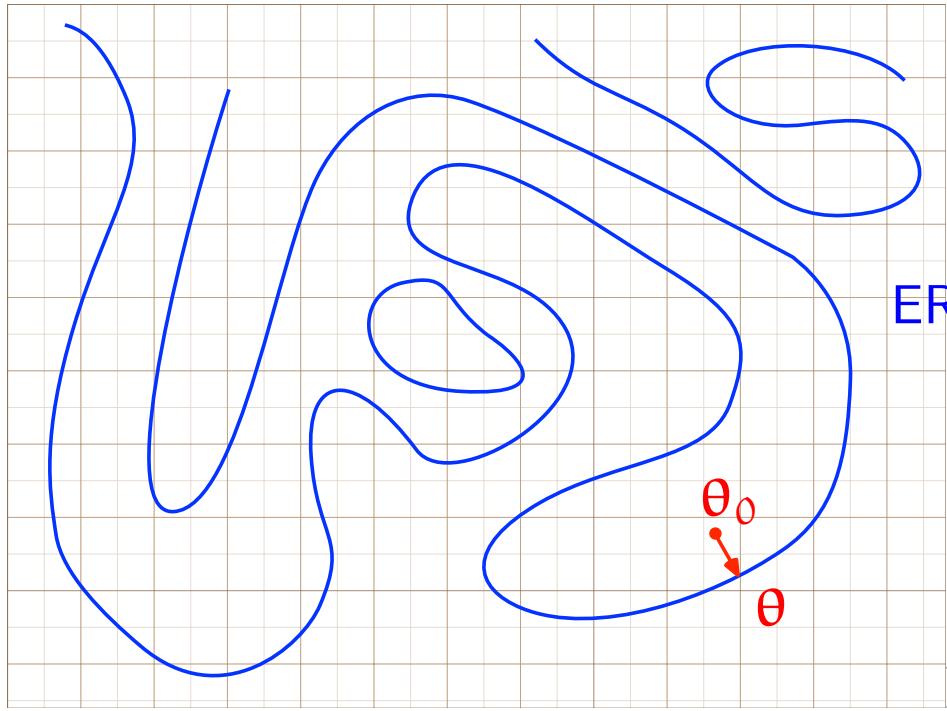
$$\mathscr{R}(\theta) := \frac{1}{2n}\mathbb{E}\big\{\big(y - f(x;\theta)\big)^2\big\}\,.$$

# Single index model ('hydrogen atom'?)

Data $\{(x_i, y_i) : i \le n\}$ iid, $\varepsilon_i \sim N(0, \tau^2)$

$$x_i \sim N(0, I_d), \quad y_i = \varphi(\langle w_*, x_i \rangle) + \varepsilon_i$$

# 'Neural tangent' theory




$\mathsf{ERM}_{n,d} := \{\theta : \widehat{\mathscr{R}}_n(\theta) = 0\}$

$\mathbb{R}^p$ (parameter space)

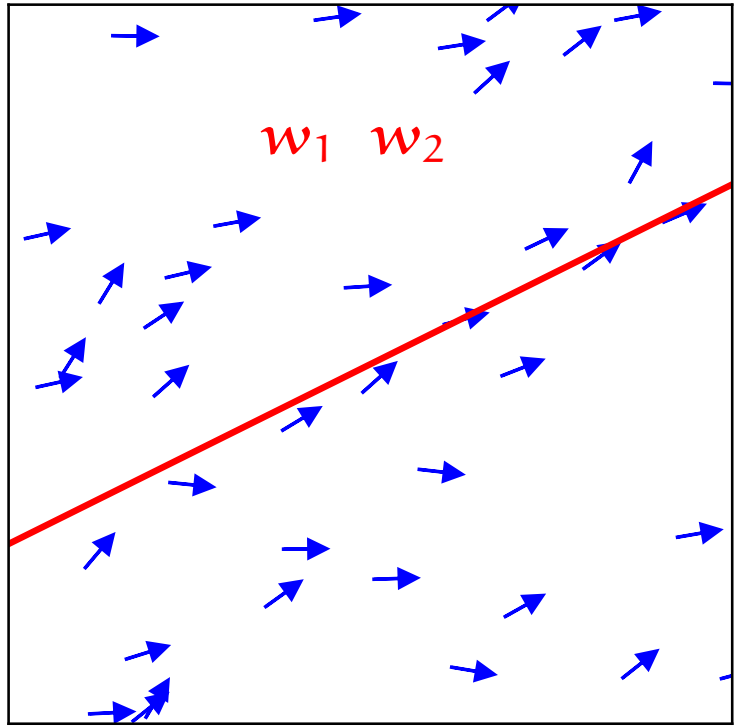
- Fast convergence to $\widehat{\mathscr{R}}_n(\hat{\theta}) \approx 0$
- Overfitting $\widehat{\mathscr{R}}_n(\hat{\theta}) \ll \mathscr{R}(\hat{\theta})$
- Fits min-norm kernel regression (see next)
- Statistically suboptimal.

---

Jacot, Gabriel, Hongler, 2018; Du, Zhai, Poczos, Singh 2018; Allen-Zhu, Li, Song 2018; Chizat, Bach, 2019; Oymak, Soltanolkotabi, 2019; ...

# Feature learning

$$f(x;\theta) = \sum_{j=1}^{N} a_j \sigma(\langle w_j, x\rangle)$$




$\mathbb{R}^d$ ($x$-space $\simeq$ $w$-space)

- Converges (ideally) to Bayes error $\widehat{\mathscr{R}}_n(\hat{\theta}) \approx \tau^2/2$.
- No overfitting $\widehat{\mathscr{R}}_n(\hat{\theta}) \approx \mathscr{R}(\hat{\theta})$
- Learns latent direction $w_*$: nonlinear.
- Expected to be statistically optimal.

---

Ba et al 2022; Barak et al 2022; Abbe, Boix Adsera, Misiakiewicz, 2022; Damian, Lee, Soltanolkotabi., 2022; Arnaboldi et al. 2023; Bethier, M, Zhou, 2024; ...

# Two theories

**Neural tangent theory:**

Global minima near a random initialization behave well

**Feature learning:**

Few steps of GD sufficient to converge to $\operatorname{argmin}_{\boldsymbol{\theta}} \mathscr{R}(\boldsymbol{\theta})$.

**Punchline:** Each is correct in a different dynamical regime

# Two theories

**Neural tangent theory:**

Global minima near a random initialization behave well

**Feature learning:**

Few steps of GD sufficient to converge to $\operatorname{argmin}_{\boldsymbol{\theta}} \mathscr{R}(\boldsymbol{\theta})$.

**Punchline:** Each is correct in a different dynamical regime

## Two previews

**Preview #1:** Generalization in the neural tangent regime

# Neural tangent model for 2-layer networks

**Linearize around initialization $W^0$**

$$f(x; a, W) = \sum_{j=1}^{N} a_j \sigma(\langle w_j, x \rangle)$$

$$f_{NT}(x; \overline{b}, b) = \sum_{j=1}^{N} \langle b_j, x \rangle \sigma'(\langle w_j^0, x \rangle) + \sum_{j=1}^{N} \overline{b}_j \sigma(\langle w_j^0, x \rangle) \, .$$

**Gradient descent $\longrightarrow$ closest interpolant**

$$\widehat{b} = \operatorname{argmin} \Big\{ \|b\| : \;\; y_i = f_{NT}(x_i; b) \;\; \forall i \leq n \Big\}$$

# Neural tangent model for 2-layer networks

**Linearize around initialization $W^0$**

$$f(x; a, W) = \sum_{j=1}^{N} a_j \sigma(\langle w_j, x \rangle)$$

$$f_{NT}(x; \overline{b}, b) = \sum_{j=1}^{N} \langle b_j, x \rangle \sigma'(\langle w_j^0, x \rangle) + \cancel{\sum_{j=1}^{N} \overline{b}_j \sigma(\langle w_j^0, x \rangle)}\,.$$

**Gradient descent $\longrightarrow$ closest interpolant**

$$\widehat{b} = \operatorname{argmin}\Big\{ \|b\| : \;\; y_i = f_{NT}(x_i; b) \;\; \forall i \leq n \Big\}$$

# Neural tangent model for 2-layer networks

**Gradient descent $\longrightarrow$ closest interpolant**

$$\hat{\mathbf{b}} = \operatorname{argmin}\Big\{ \|\mathbf{b}\| : \;\; y_i = f_{\mathrm{NT}}(\mathbf{x}_i; \mathbf{b}) \;\; \forall i \leq n \Big\}$$

**Equivalently**

$$z_i = \boldsymbol{\Phi}(\boldsymbol{x}_i) := \left(\boldsymbol{x}_i^{\mathsf{T}} \sigma'(\langle \boldsymbol{w}_1^0, \boldsymbol{x}_i \rangle), \boldsymbol{x}_i^{\mathsf{T}} \sigma'(\langle \boldsymbol{w}_2^0, \boldsymbol{x}_i \rangle), \ldots, \boldsymbol{x}_i^{\mathsf{T}} \sigma'(\langle \boldsymbol{w}_N^0, \boldsymbol{x}_i \rangle)\right)^{\mathsf{T}},$$

$$\hat{\mathbf{b}}_\lambda = \operatorname{argmin}_{\mathbf{b}} \left\{ \|\mathbf{y} - \mathbf{Z}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_2^2 \right\}, \qquad \lambda \downarrow 0 \, .$$

# Neural tangent model for 2-layer networks

**Gradient descent $\longrightarrow$ closest interpolant**

$$\hat{\mathbf{b}} = \operatorname{argmin}\Big\{ \|\mathbf{b}\| : \;\; y_i = f_{\mathrm{NT}}(\mathbf{x}_i; \mathbf{b}) \;\; \forall i \leq n \Big\}$$

**Equivalently**

$$\mathbf{z}_i = \boldsymbol{\phi}(\mathbf{x}_i) := \big(\mathbf{x}_i^{\mathsf{T}} \sigma'(\langle \boldsymbol{w}_1^0, \mathbf{x}_i \rangle), \mathbf{x}_i^{\mathsf{T}} \sigma'(\langle \boldsymbol{w}_2^0, \mathbf{x}_i \rangle), \ldots, \mathbf{x}_i^{\mathsf{T}} \sigma'(\langle \boldsymbol{w}_N^0, \mathbf{x}_i \rangle)\big)^{\mathsf{T}},$$

$$\hat{\mathbf{b}}_\lambda = \operatorname{argmin}_{\mathbf{b}} \Big\{ \|\mathbf{y} - \mathbf{Z}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_2^2 \Big\}, \qquad \lambda \downarrow 0 \, .$$

# Simpler setting: Random features model

$$\min_{\mathbf{b}} \Big\{ \sum_{i=1}^{n} \big(y_i - f_{\mathrm{RF}}(\boldsymbol{x}_i; \mathbf{b})\big)^2 + \lambda \|\mathbf{b}\|_2^2 \Big\}, \quad f_{\mathrm{RF}}(\boldsymbol{x}; \mathbf{b}) = \sum_{i=1}^{N} b_i \, \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle) \, .$$

**Theorem** (Mei, M. 2019)

*Assume* $n, N, d \to \infty$

$$\frac{N}{d} \to \psi_1, \qquad \frac{n}{d} \to \psi_2.$$

*Decompose* $\sigma(x) = \sigma_0 + \sigma_1 x + \sigma^{\mathrm{NL}}(x)$ *where (for* $G \sim \mathsf{N}(0,1)$*)* $\mathbb{E}[G\sigma^{\mathrm{NL}}(G)] = \mathbb{E}[\sigma^{\mathrm{NL}}(G)] = 0$, $\zeta^2 := \frac{\sigma_1^2}{\mathbb{E}[\sigma^{\mathrm{NL}}(G)^2]}$. *Then there are explicit functions* $\mathscr{B}(\zeta, \psi_1, \psi_2, \overline{\lambda})$, $\mathscr{V}(\zeta, \psi_1, \psi_2, \overline{\lambda})$, *such that*

$$R(\widehat{f}_\lambda) = F_1^2 \mathscr{B}(\zeta, \psi_1, \psi_2, \lambda) + (\tau^2 + F_*^2) \mathscr{V}(\zeta, \psi_1, \psi_2, \lambda) + F_*^2 + o_d(1) \, .$$

# Simpler setting: Random features model

$$\min_{b}\Big\{\sum_{i=1}^{n}\big(y_i - f_{RF}(x_i;b)\big)^2 + \lambda\|b\|_2^2\Big\}, \quad f_{RF}(x;b) = \sum_{i=1}^{N} b_i\,\sigma(\langle w_i, x\rangle)\,.$$

**Theorem** (Mei, M. 2019)

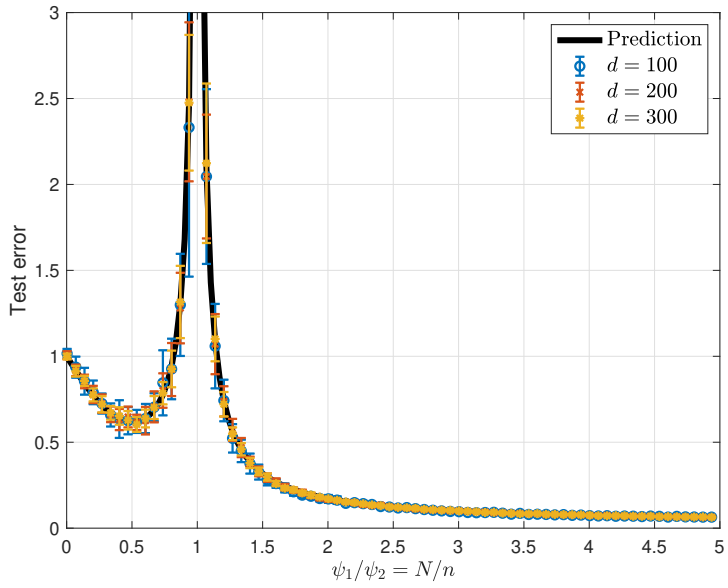
*Assume* $n, N, d \to \infty$

$$\frac{N}{d} \to \psi_1, \qquad \frac{n}{d} \to \psi_2.$$

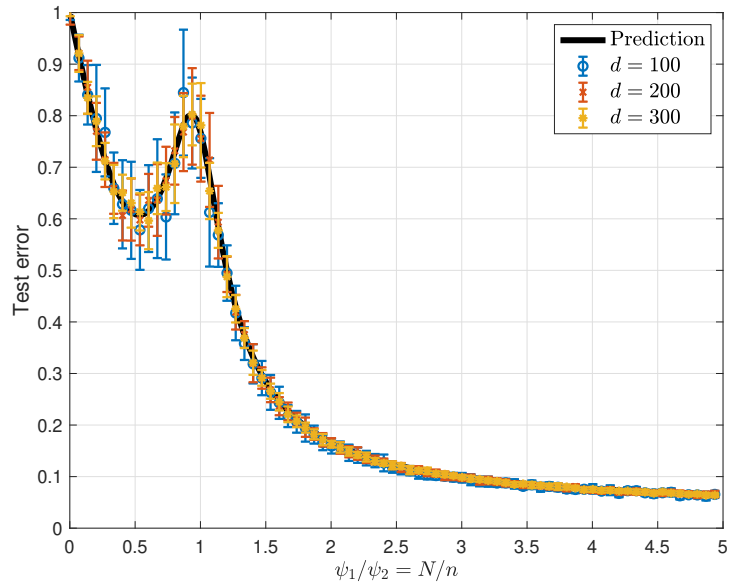
*Decompose* $\sigma(x) = \sigma_0 + \sigma_1 x + \sigma^{NL}(x)$ *where (for* $G \sim N(0,1)$*)* $\mathbb{E}[G\sigma^{NL}(G)] = \mathbb{E}[\sigma^{NL}(G)] = 0$, $\zeta^2 := \frac{\sigma_1^2}{\mathbb{E}[\sigma^{NL}(G)^2]}$. *Then there are explicit functions* $\mathscr{B}(\zeta,\psi_1,\psi_2,\bar{\lambda})$, $\mathscr{V}(\zeta,\psi_1,\psi_2,\bar{\lambda})$, *such that*

$$R(\hat{f}_\lambda) = F_1^2\mathscr{B}(\zeta,\psi_1,\psi_2,\lambda) + (\tau^2 + F_*^2)\mathscr{V}(\zeta,\psi_1,\psi_2,\lambda) + F_*^2 + o_d(1)\,.$$

# Insight: $N/n \gg 1$ optimal




$\lambda = 0+$




$\lambda > 0$

- Solid line: Theoretical prediction
- Mathematics tool: Random matrix theory

**Preview #2:** Dynamical decoupling of feature learning and overfitting

# Gradient flow dynamics

**Gradient flow**

$$\dot{\theta}(t) = -\frac{n}{d}\nabla\widehat{\mathscr{R}}_n(\theta(t))\,.$$

**Square loss train error**

$$\widehat{\mathscr{R}}_n(\theta) := \frac{1}{2n}\sum_{i=1}^{n}\big(y_i - f(x_i;\theta)\big)^2, \quad \theta = (a, W)\,.$$

**Two-layer network**

$$f(x;\theta) = \frac{1}{N}\sum_{j=1}^{N} a_j\sigma(\langle w_j, x\rangle)\,.$$

# Dynamical approach

**Gradient flow**

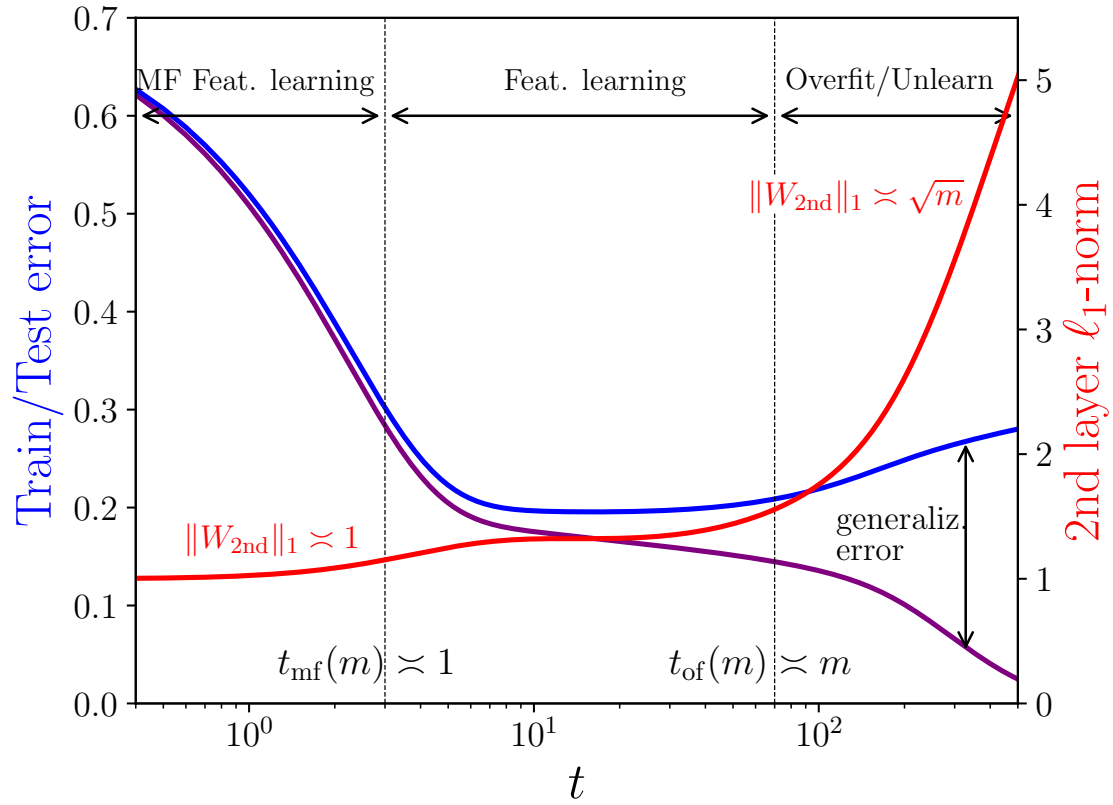
$$\dot{\theta}(t) = -\frac{n}{d}\nabla\widehat{\mathscr{R}}_n(\theta(t))\,.$$

1. Dynamics in a random landscape

2. Exact asymptotics as $n, d \to \infty$, $n/d \to \overline{\alpha}$ by rigorizing tools from theoretical physics
   [DMFT, dynamical mean-field theory: Celentano, Cheng, M, 2022]

3. Large width limit $N, \overline{\alpha} \to \infty$, $\overline{\alpha}/N = \alpha$:
   [M, Urbani, 2025]

# Dynamical decoupling: Learning → Overfitting




t ≍ 1:

- Test error ≈ Train error
- Learns latent direction $\boldsymbol{w}_*$
- Mean field asymptotically correct (Mei, M, Nguyen, 2018; Chizat, Bach, 2018; Rotskoff, Vanden Eijnden, 2018)

t ≍ m ≡ N:

- Test error ≫ Train error; Train error → 0
- *Unlearns* latent direction $\boldsymbol{w}_*$
- Neural tangent kernel learning

# Conclusion

# Conclusion

- Important to ask fundamental questions about AI.
- What does it mean that 'AI models learn'?
- Requires to understand training dynamics

Thank you!

# Conclusion

- Important to ask fundamental questions about AI.
- What does it mean that 'AI models learn'?
- Requires to understand training dynamics

Thank you!