Lecture 1/2 What is Machine Learning?

IEMS 402 Statistical Learning

Northwestern

Logistics

Logistics

- Course Website: https://2prime.github.io/teaching/2025-Statistical-Learning
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)

max(HW1,HW8)+max(HW2,HW3)+max(HW4,HW5)+max(HW6,HW7).

- [Homework 1] Review of Probability and Optimization — Review of technical basic
- [Homework 2] Bias and Variance Trade-off 1
- [Homework 3] Bias and Variance Trade-off 2
- [Homework 4] Asymptotic Theory 1
- [Homework 5] Asymptotic Theory 2
- [Homework 6] Non-Asymptotic Theory 1
- [Homework 7] Non-Asymptotic Theory 2
- [Homework 8] Advanced Topics — Advanced research in OR

Start early!

- Latex and overleaf (not required)

Logistics

- Course Website: https://2prime.github.io/teaching/2025-Statistical-Learning
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)

max(HW1,HW8)+max(HW2,HW3)+max(HW4,HW5)+max(HW6,HW7).

- [Homework 1] Review of Probability and Optimization
- [Homework 2] Bias and Variance Trade-off 1 — Easy
- [Homework 3] Bias and Variance Trade-off 2
- [Homework 4] Asymptotic Theory 1 — Easy
- [Homework 5] Asymptotic Theory 2
- [Homework 6] Non-Asymptotic Theory 1 — Easy
- [Homework 7] Non-Asymptotic Theory 2
- [Homework 8] Advanced Topics

} Start early!

- Latex and overleaf (not required)

Logistics

- Course Website: https://2prime.github.io/teaching/2025-Statistical-Learning
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)

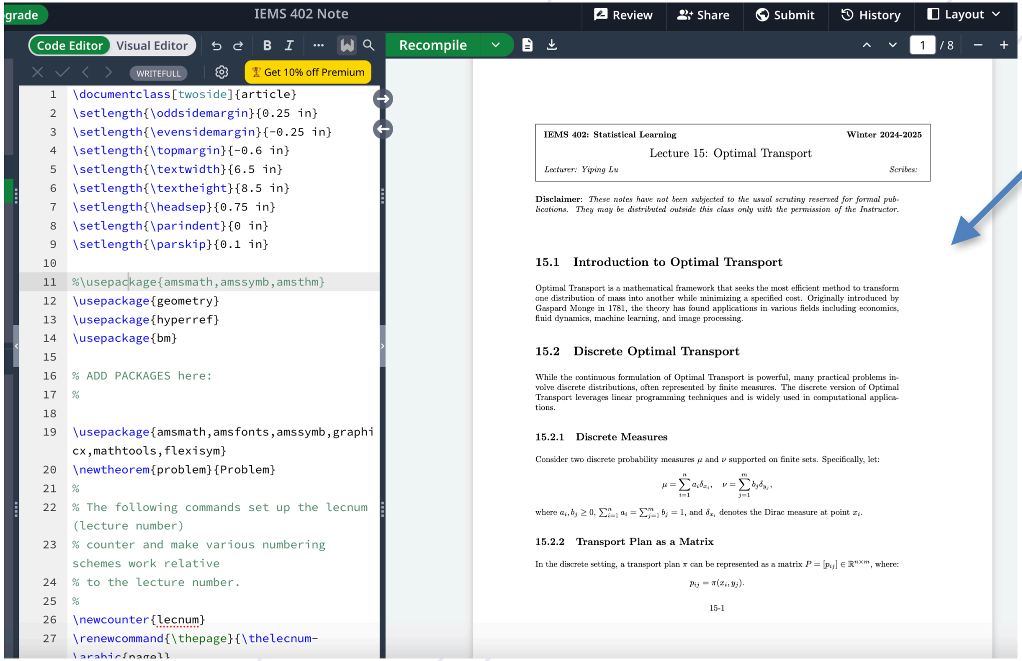
Exams

The same technique as the exam

- [Practice Mid-Term Exam]
  - Modern Machine Learning Concepts, Bias and Variance Trade-off
  - Kernel Smoothing, Asymptotic Theory, Influnce Function Concentration Inequality, Uniform Bound
- [Practice Final Exam]
  - Rademacher complexity, Covering Number, Dudley's theorem
  - RKHS, Optimal Transport, Robust Learning

Logistics

- Course Website: https://2prime.github.io/teaching/2025-Statistical-Learning
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)



Refine my note

Logistics

- Course Website: https://2prime.github.io/teaching/2025-Statistical-Learning
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)
- Textbook: Bach, Francis. Learning theory from first principles. MIT press, 2024.
  - https://www.di.ens.fr/~fbach/ltfp_book.pdf

Gradescope

Campuswire

ChatGPT Tutor!

Late Work Policy

- For your first late assignment within 12 hours after the deadline (as indicated on Gradescope), no point deductions.
- All subsequent assignments submitted within 12 hours after the deadline will convert to a zero at the end of semester.
- In all cases, work submitted 12 hours or more after the deadline will not be accepted.

Prelinminary

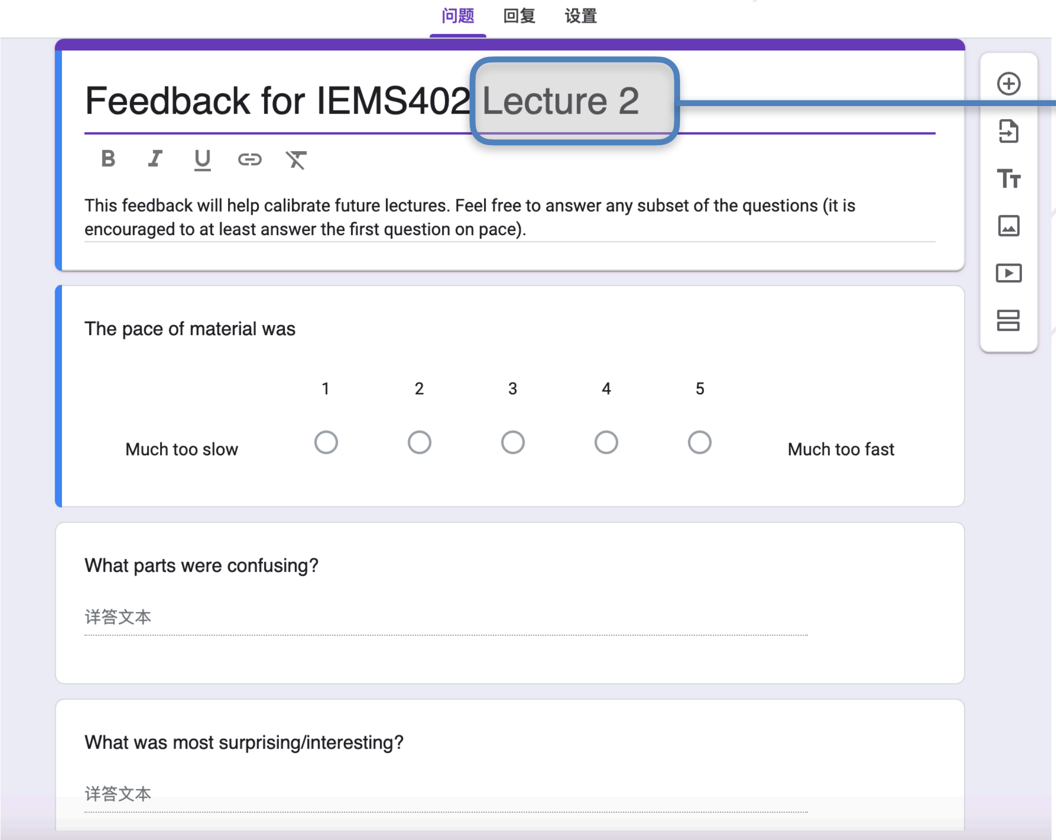
Review Document:
https://2prime.github.io/files/IEMS402/IEMS402ProbOptReview.pdf

Calculus, Linear Algebra
IEMS 302 Probability Probability and Statistics: Strong Law of Large Numbers, Central Limit Theorem, Big-O, little-o notation,
Optimization Theory: **Lagrangian Duality Theory IEMS 450-2: Mathematical Optimization II**
(Interestingly, IEMS 450-1 is not required)

You need to know
- Law of strong numbers, Central Limit Theorem, Continuous Map Theorem, Slutsky Theorem, Markov's Inequality

You don't need to distinguish Convergence in Probability/Covergence in distribution, you just need to write →

Online Calibration with Human Feedback



Feedback for each lecture

Other Course

Stats 300b - Stanford

1. Introduction
2. Convergence of random variables (January 14)
3. Delta method (January 14)
4. Basics of asymptotic normality (January 18 and 20)
5. Moment method (January 20)
6. Uniform laws of large numbers (January 26)
7. Basics of concentration (January 28 and February 2)
8. Sub Gaussian processes and chaining (February 2 and February 4)
9. VC Dimension (February 4)
10. Uniform central limit theorems and convergence in distribution (February 9 and February 11)
11. Applications of Uniform Central Limit Theorems (February 16 and February 18)
12. Relative efficiency and basic tests (February 18 and February 23)
13. Asymptotic level and relative efficiency in testing (February 23 and 25)
14. Contiguity and Asymptotics (February 25)
15. Local Asymptotic Normality (March 2 and 4)
16. Regular estimators and consequences (March 8 and 10)
17. U statistics (March 11 and 16)
18. Parting thoughts (March 18)

Stats 705 - CMU

Date	**Lecture Topic**
August 31	Review
September 2	Concentration Inequalities
September 4	Concentration Inequalities
September 7	**No Class (Labor Day)**
September 9	Convergence
September 11	Convergence
September 14	Central Limit Theorem
September 18	Uniform Laws and Empirical Process Theory
September 18	Uniform Laws and Empirical Process Theory
September 21	Uniform Laws and Empirical Process Theory
September 23	Review
September 25	**TEST 1**
September 28	Likelihood and Sufficiency
September 30	Point Estimation (MLE)
October 2	Point Estimation (Method of Moments, Bayes)
October 5	Decision Theory
October 7	Decision Theory
October 9	Asymptotic Theory
October 12	Asymptotic Theory
October 14	Hypothesis Testing
October 16	NO CLASS (Community Engagement)
October 19	Goodness-of-fit, two-sample, independence
October 21	Multiple testing
October 23	**NO CLASS (Mid-Semester Break)**
October 26	Multiple testing
October 28	Confidence Intervals
October 30	Confidence Intervals
November 2	Confidence Intervals
November 4	Review
November 6	**TEST 2**
November 9	Bootstrap
November 11	Bootstrap
November 13	Bayesian Inference
November 16	Bayesian Inference
November 18	Linear Regression
November 20	Non-parametric Regression
November 23	NO CLASS
November 25	NO CLASS (Thanksgiving)
November 27	NO CLASS
November 30	Minimax Lower Bounds
December 2	Minimax Lower Bounds
December 4	High-dimensional Statistics
December 7	High-dimensional Statistics
December 9	Model Selection
December 11	Model Selection

Other Course

Stanford: Stats 300b/ CS229T

Berkeley: Stats 241/Stats 241B

MIT IDS.160/9.521/18.656/6.S988

CMU Stat705, 10-072

Princeton COS 511

Cornell CS6783, ORIE 7790

Umich EECS598, UW Madison CS 839, UofT STA3000F

Good machine learning courses are open source!

Why IEMS402?

Nature
https://www.nature.com › articles · 翻译此页

Highly accurate protein structure prediction with AlphaFold

作者： J Jumper · 2021 · 被引用次数： 30628 — AlphaFold greatly improves the accuracy of structure prediction by incorporating novel neural network architectures and t

Nature
https://www.nature.com › articles · 翻译此页

Magnetic control of tokamak plasmas through deep ...

作者： J Degrave · 2022 · 被引用次数： 897 — In this work, we introduce a previously undescribed architecture for t

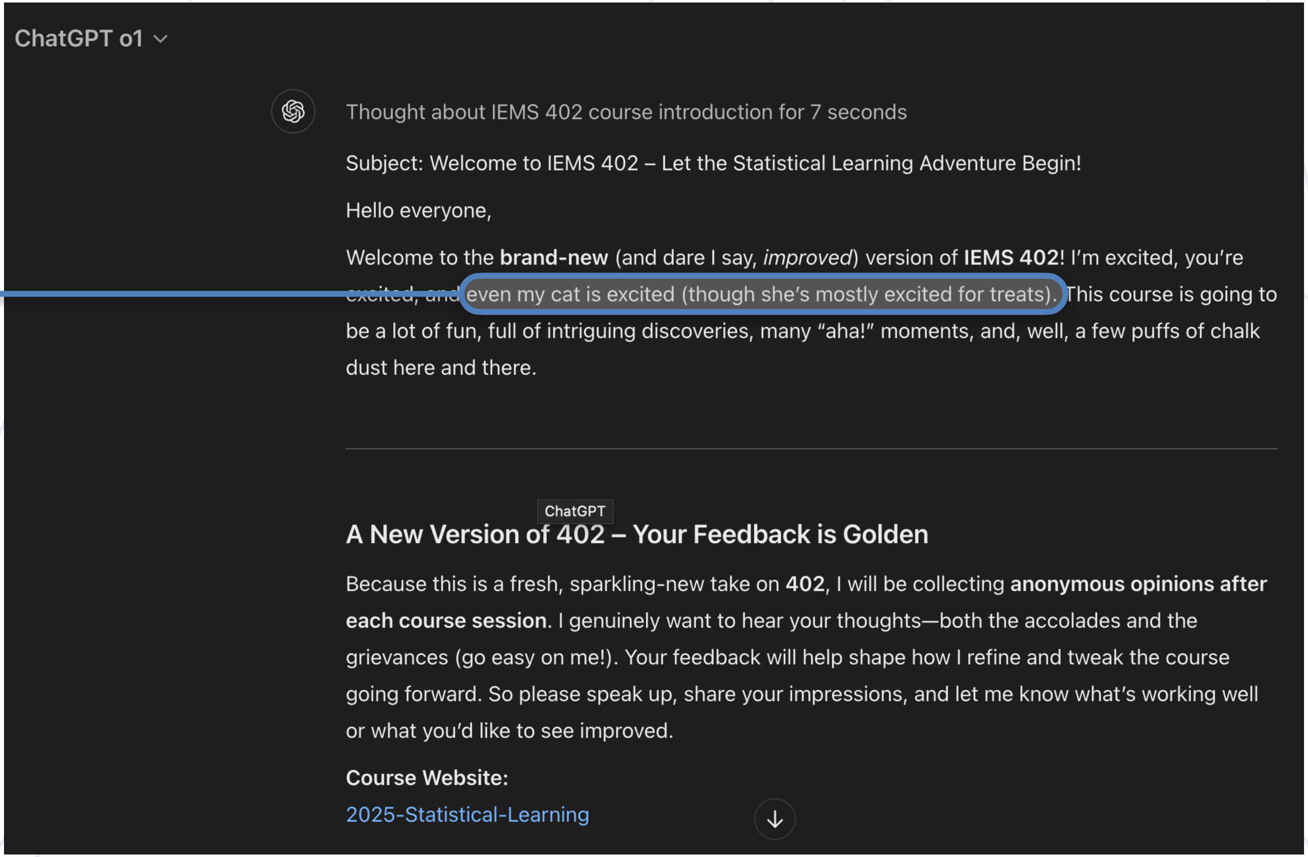
Nature
https://www.nature.com › news · 翻译此页

DeepMind AI weather forecaster beats world-class system

2024年12月4日 — Google DeepMind has developed the first artificial intelligence (AI) model of its kind to predict the weather more accurately than the best ...

Why IEMS402?

I Don’t have a cat


ChatGPT o1

Thought about IEMS 402 course introduction for 7 seconds

Subject: Welcome to IEMS 402 – Let the Statistical Learning Adventure Begin!

Hello everyone,

Welcome to the **brand-new** (and dare I say, *improved*) version of **IEMS 402**! I’m excited, you’re excited, and even my cat is excited (though she’s mostly excited for treats). This course is going to be a lot of fun, full of intriguing discoveries, many “aha!” moments, and, well, a few puffs of chalk dust here and there.

ChatGPT

A New Version of 402 – Your Feedback is Golden

Because this is a fresh, sparkling-new take on **402**, I will be collecting **anonymous opinions after each course session**. I genuinely want to hear your thoughts—both the accolades and the grievances (go easy on me!). Your feedback will help shape how I refine and tweak the course going forward. So please speak up, share your impressions, and let me know what’s working well or what you’d like to see improved.

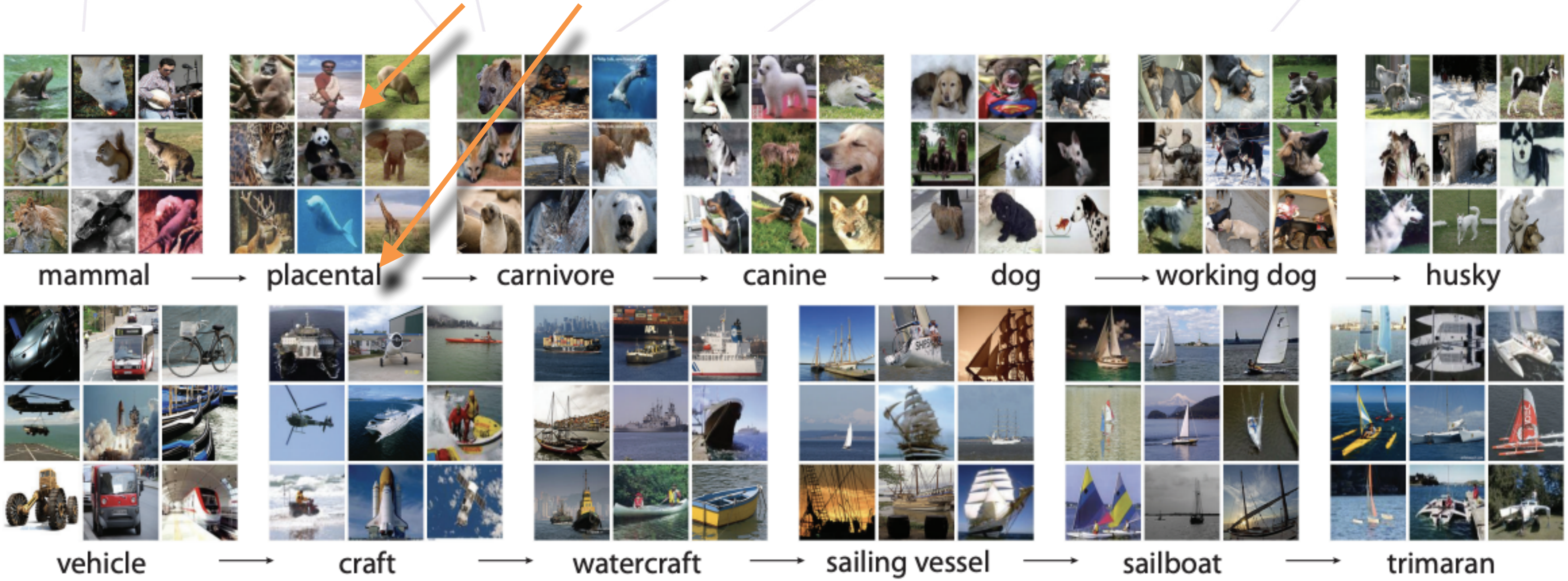
Course Website:

2025-Statistical-Learning

Supervised Learning

Supervised Learning

- Aim: learn a predictor $f : \mathscr{X} \rightarrow \mathscr{Y}$



Probably Approximately Correct

PAC Learning Model

- **Input: Training Data.** $S = \{(x_1, y_1), \dots (x_m, y_m)\}$ is a finite set of pairs in $\chi \times \mathcal{Y}$. This is the *input* that the learner has access to. Such labeled examples are also referred to as *training examples* or *labeled sample set*. The size of the sample set m is the *sample size*. We will generally assume that the sample S was generated by drawing m IID samples from the distribution D.
- **Output: Hypothesis.** A Hypothesis class consists of a subset of target functions $\mathcal{H} = \{h : h : \chi \to \mathcal{Y}\}$ that turns unlabeled samples to labels. Each learning algorithm outputs a hypothesis, the class of hypotheses the learner may return is the algorithms hypothesis class.

ML Alg: input: m data output: a function $h: X \to Y$

Probably Approximately Correct

PAC Learning Model

- **Input: Training Data.** $S = \{(x_1, y_1), \dots (x_m, y_m)\}$ is a finite set of pairs in $\chi \times \mathcal{Y}$. This is the *input* that the learner has access to. Such labeled examples are also referred to as *training examples* or *labeled sample set*. The size of the sample set m is the *sample size*. We will generally assume that the sample S was generated by drawing m IID samples from the distribution D.

- **Output: Hypothesis.** A Hypothesis class consists of a subset of target functions $\mathcal{H} = \{h : h : \chi \to \mathcal{Y}\}$ that turns unlabeled samples to labels. Each learning algorithm outputs a hypothesis, the class of hypotheses the learner may return is the algorithms hypothesis class.

"sample complexity"

Aim: know m_C^A

Our Goal

Definition 1.1 ((realizable) PAC Learning). *A concept class $\mathcal{C}$ of target functions is PAC learnable (w.r.t to $\mathcal{H}$) if there exists an algorithm A and function $m_{\mathcal{C}}^A : (0,1)^2 \to \mathbb{N}$ with the following property:*

Assume $S = ((x_1, y_1), \dots, (x_m, y_m))$ is a sample of IID examples generated by some arbitrary distribution D such that $y_i = h(x_i)$ for some $h \in C$ almost surely. If S is the input of A and $m > m_{\mathcal{C}}^A(\epsilon, \delta)$ then the algorithm returns a hypothesis $h_S^A \in \mathcal{H}$ such that, with probability $1 - \delta$ (over the choice of the m training examples):

$$err(h_S^A) < \epsilon$$

How to define error?

The function $m_{\mathcal{C}}^A(\epsilon, \delta)$ is referred to as the sample complexity *of algorithm A.*

Supervised Learning

- Aim: learn a predictor $f : \mathcal{X} \to \mathcal{Y}$
- What is a good predictor? -> evaluation criteria

Assume data sample from a distribution p

$$\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x))dp(x, y) .$$

Evaluate the error of label and predicition

Ex. Classification 0-1 loss

$$\ell(y, f(x)) = \begin{cases} 0 & y = f(x) \\ 1 & y \neq f(x) \end{cases}$$

Ex. Regression

$$\ell(y, f(x)) = (f(x) - y)^2$$

$y \in \mathbb{R}$, $f(x) \in \mathbb{R}$

Supervised Learning

- Aim: learn a predictor $f : \mathcal{X} \to \mathcal{Y}$
- What is a good predictor? -> evaluation criteria

$$\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x))dp(x, y)\,.$$

Evaluate the error of label and predicition

If I want to know the risk, I need to have all the data in the univers?

Good habit: "^" means empirical

Empirical Risk: $\hat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$, where $\{(x_i, y_i)\}_{i=1}^{n}$ is a collected dataset

Conditional Risk

$$\mathscr{R}(f) = \mathbb{E}_{x' \sim p}\left[\mathbb{E}\left[\ell(y, f(x')) \middle| x = x'\right]\right] = \int_{\mathscr{X}} \mathbb{E}\left[\ell(y, f(x')) \middle| x = x'\right] dp(x') \,.$$

Conditional Risk: $r(z \,|\, x') = \mathbb{E}\left[\ell(y, z) \middle| x = x'\right]$

* means the best

- Bayes Predictor: $f^*(x') \in \arg\min_{z \in \mathscr{Y}} \mathbb{E}\left[\ell(y, z) \middle| x = x'\right] = \arg\min_{z \in \mathscr{Y}} r(z \,|\, x') \,.$

Conditional Risk

$$\mathcal{R}(f) = \mathbb{E}_{x' \sim p}\left[\mathbb{E}\left[\ell(y, f(x'))\,\middle|\, x = x'\right]\right] = \int_{\mathcal{X}} \underbrace{\mathbb{E}\left[\ell(y, f(x'))\,\middle|\, x = x'\right]}\, dp(x')\,.$$

Conditional Risk: $r(z\,|\,x') = \mathbb{E}\left[\ell(y,z)\,\middle|\, x = x'\right]$

$r(z|x') = \mathbb{E}[(y-z)^2 | x = x'] \Leftarrow z^* = \mathbb{E}[y | x = x']$

- Bayes Predictor: $f^*(x') \in \arg\min_{z \in \mathcal{Y}} \mathbb{E}\left[\ell(y,z)\,\middle|\, x = x'\right] = \arg\min_{z \in \mathcal{Y}} r(z\,|\,x')\,.$

What is the Bayes Predictor of ℓ_2 loss or ℓ_1 loss?

→ mean

→ median

Homework, pinball loss → quantile.

How to design a loss function

- Method 1: Know what is your Bayes Predictor! Homework 1 Question 1.

How to design a loss function

- Method 1: Know what is your Bayes Predictor! Homework 1 Question 1.
- Method 2: Use Max likelihood
  - Step 1: understand what is your $p(y|x)$, e.g. Gaussian, heavy tail distribution
  - Step 2: What is the log-likelihood of dataset $\{(x_i, y_i)\}_{i=1}^n$?

How to design a loss function

- Method 1: Know what is your Bayes Predictor! Homework 1 Question 1.
- Method 2: Use Max likelihood
  - Step 1: understand what is your $p(y|x)$, e.g. Gaussian, heavy tail distribution
  - Step 2: What is the log-likelihood of dataset $\{(x_i, y_i)\}_{i=1}^n$?
    - $\log \Pi_{i=1}^n p(y_i|x_i) = \sum_{i=1}^n \log p(y_i|x_i)$
  - Step 3: use $\log p(\cdot|x_i)$ as your loss function!

Sum! like Empirical Risk

ℓ_2 loss: means Gaussian Noise

$p(y|x) = N(f(x), \sigma I)$

How can I get the ℓ_2 loss using this methods?

Example: Logistic Regression

Consider a binary classification with $p(y_i = 1 \mid \mathbf{x}_i, \theta) = \sigma(\mathbf{x}_i^\top \theta) = \dfrac{1}{1 + e^{-\mathbf{x}_i^\top \theta}}$

Example: Gaussian with Learned Variance

Not Required

Example (*Gaussian with Learned Variance Leads to Sparsity*)

$$\ell(\mu, \sigma^2) = \sum_{i=1}^{n} \log P(y_i|\mu(x_i), \sigma(x_i)^2)$$

$$= \sum_{i=1}^{n} \left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma(x_i)^2) - \frac{(y_i - \mu(x_i))^2}{2\sigma(x_i)^2} \right)$$

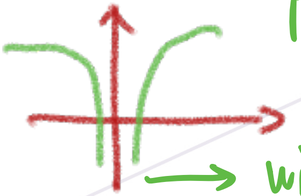
$$= -\frac{n}{2}\ln(2\pi(x_i)) - \underbrace{\frac{n}{2}\ln(\sigma(x_i)^2)}_{\text{sparse regularization}} - \underbrace{\sum_{i=1}^{n} \frac{(y_i - \mu(x_i))^2}{2\sigma(x_i)^2}}_{\text{weighted } \ell_2 \text{ loss}}$$

$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma}\right)$

ℓ_2 loss

weight by variance

Shape



$\ln(0) = -\infty$

→ will encourage $\sigma(x_i)$ to 0

Empirical Risk Minimization

I want an estimator to minimize the risk, but I can only get the empirial risk? What's the best thing I can do?

- Consider a parameterized family of prediction functions (often referred to as models) $f_\theta : \mathscr{X} \to \mathscr{Y}$, e.g. θ: parameter
  - Linear prediction $f(x) = \langle x, \theta \rangle$
  - Neural Network

^ means empirical

- Empirical Risk Minimization: $\hat{\theta} \in \hat{\mathscr{R}}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$.

Pro and Con of ERM

- Pro:
  - Flexible
  - Algorithms are available (e.g. SGD)
- Con:
  - can be relatively hard to optimize when the optimization formulation is not convex (e.g.,neural networks); (? in real application)
  - the dependence on parameters can be complex (e.g., neural networks);
  - need some capacity control to avoid overfitting (? in real application)

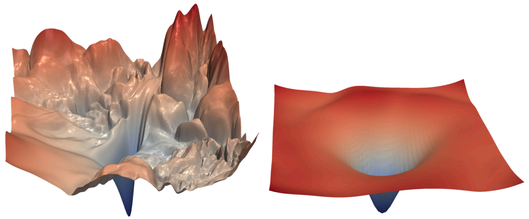


Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Our course is about overfitting!

The only theorem: Risk Decompoisition

$$\mathscr{R}(f_{\hat{\theta}}) - \mathscr{R}^* = \underbrace{\left\{ \mathscr{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) \right\}}_{\text{Estimation error}} + \underbrace{\left\{ \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) - \mathscr{R}^* \right\}}_{\text{Approximation error}}$$

The only theorem: Risk Decompoisition

$\theta^* = \arg\min_\theta R(f_\theta) \to$ best parameter.

$$\mathscr{R}(f_{\hat\theta}) - \mathscr{R}^* = \underbrace{\left\{ \mathscr{R}(f_{\hat\theta}) - \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) \right\}}_{\text{Estimation error}} + \underbrace{\left\{ \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) - \mathscr{R}^* \right\}}_{\text{Approximation error}}$$

Question:

$\Theta_1 \subseteq \Theta_2$

Approximation Error of Θ_1 is larger!

For an ERM Estimator: ||

$\theta^* = \arg\min_{\theta \in \Theta} R(f_\theta)$

$$\underbrace{\mathscr{R}(f_{\hat\theta}) - \hat{R}(f_{\hat\theta})}_{\text{Generalization error}} + \underbrace{\hat{R}(f_{\hat\theta}) - \hat{\mathscr{R}}(f_{\theta^*})}_{\text{Optimization error} \leq 0} + \underbrace{\hat{\mathscr{R}}(f_{\theta^*}) - \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'})}_{\text{Generalization error}}$$

The only theorem: Risk Decompoisition

$$\mathscr{R}(f_{\hat{\theta}}) - \mathscr{R}^* = \underbrace{\left\{ \mathscr{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) \right\}}_{\text{Estimation error}} + \underbrace{\left\{ \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) - \mathscr{R}^* \right\}}_{\text{Approximation error}}$$

For an ERM Estimator:

$$\underbrace{\mathscr{R}(f_{\hat{\theta}}) - \hat{R}(f_{\hat{\theta}})}_{\text{Generalization error}} + \underbrace{\hat{R}(f_{\hat{\theta}}) - \hat{\mathscr{R}}(f_{\theta^*})}_{\text{Optimization error}} + \underbrace{\hat{\mathscr{R}}(f_{\theta^*}) - \inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'})}_{\text{Generalization error}}$$

θ^* is the best

Optimization error ≤ 0

$$\leq 2 \sup_{\theta \in \Theta} |R(f_\theta) - \hat{R}(f_\theta)|$$

Uniform Bound!

Question.

$\Theta_1 \subseteq \Theta_2$

Generalization Bound of Θ_2 is larger.

No Free Lunch Theorem

Let $\mathcal{A}$ be any learning algorithm for the task of binary classification with respect to the 0/1-loss function over a domain $\mathcal{X}$. Let $m < \frac{|\mathcal{X}|}{2}$ be a number representing a training set size.

There exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:

- *there exists a function $f : \mathcal{X} \longrightarrow \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$;*
- *with probability at least $1/7$ over the choice of a sample $S \sim \mathcal{D}^m$ (of size m) we have that $L_{\mathcal{D}}(\mathcal{A}(S)) \geqslant 1/8$.*

Need Assumption over data.

https://www.cs.cornell.edu/courses/cs6783/2015fa/lec3.pdf

No Free Lunch Theorem

Let $\mathcal{A}$ be any learning algorithm for the task of binary classification with respect to the 0/1-loss function over a domain $\mathcal{X}$. Let $m < \frac{|\mathcal{X}|}{2}$ be a number representing a training set size.

There exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$ such that:

- *there exists a function $f : \mathcal{X} \longrightarrow \{0, 1\}$ with $L_{\mathcal{D}}(f) = 0$;*
- *with probability at least $1/7$ over the choice of a sample $S \sim \mathcal{D}^m$ (of size m) we have that $L_{\mathcal{D}}(\mathcal{A}(S)) \geqslant 1/8$.*

How to formulate $A(S)$ in math?

https://www.cs.cornell.edu/courses/cs6783/2015fa/lec3.pdf

No Free Lunch Theorem

$$\max_{1 \leqslant i \leqslant |T|} E_{S \sim \mathcal{D}^m}(L_{D_i}(A(S))) \geqslant \frac{1}{4}.$$

This means that for every $\mathcal{A}'$ that receives a training set of m examples from $\mathcal{X} \times \{0,1\}$ there exists $f : \mathcal{X} \longrightarrow \{0,1\}$ and a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that $L_{\mathcal{D}}(f) = 0$ and $E_{S \sim \mathcal{D}^m}(L_{\mathcal{D}}(\mathcal{A}'(S))) \geqslant \frac{1}{4}$.

No Free Lunch Theorem

$$\max_{1 \leqslant i \leqslant |T|} E_{S \sim \mathcal{D}^m}(L_{D_i}(A(S))) \geqslant \frac{1}{4}.$$

This means that for every $\mathcal{A}'$ that receives a training set of m examples from $\mathcal{X} \times \{0,1\}$ there exists $f : \mathcal{X} \longrightarrow \{0,1\}$ and a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that $L_{\mathcal{D}}(f) = 0$ and $E_{S \sim \mathcal{D}^m}(L_{\mathcal{D}}(\mathcal{A}'(S))) \geqslant \frac{1}{4}$.

No Free Lunch Theorem

Let $\mathcal{A}$ be any learning algorithm for the task of binary classification with respect to the 0/1-loss function over a domain $\mathcal{X}$. Let $m < \frac{|\mathcal{X}|}{2}$ be a number representing a training set size.

There exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:

- *there exists a function $f : \mathcal{X} \longrightarrow \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$;*
- *with probability at least $1/7$ over the choice of a sample $S \sim \mathcal{D}^m$ (of size m) we have that $L_{\mathcal{D}}(\mathcal{A}(S)) \geqslant 1/8$.*

Important to know what's the implicit assumption on target function

https://www.cs.cornell.edu/courses/cs6783/2015fa/lec3.pdf

Review

bayes Risk

$\theta^* = \arg\min_{\theta \in \Theta} R(f_\theta)$

$$R(f_{\hat\theta}) - R^* = \{R(f_{\hat\theta}) - R(f_{\theta^*})\} + \{R(f_{\theta^*}) - R^*\}$$

Estimation ⇓   Approximation

$$R(f_{\hat\theta}) - \hat R(f_{\hat\theta}) + \hat R(f_{\hat\theta}) - \hat R(f_{\theta^*}) + \hat R(f_{\theta^*}) - R(f_{\theta^*})$$

($\hat R(f_{\hat\theta}) - \hat R(f_{\theta^*}) \le 0$)

Difference between Empirical and population

uniform Bound $\le \sup_{\theta \in \Theta} |\hat R(f_\theta) - R(f_\theta)|$ generalization

	Θ_1 $\subseteq$	Θ_2
Approx	large	small
Gen	small	larger

"bias - Var" Trade-off

Difference between 401 and 402

Statistics Learning

- Difference 1: Parameter Convergence and Risk Convergence

$\hat{\theta} \to \theta^*$ $\quad R(f_{\hat{\theta}}) \to R(f_{\theta^*})$

Stats machine learning.

- Difference 2: Parametric and Non-parametric

① Assumption over data

ex. $y = \langle \theta, x \rangle + \epsilon$. $x \sim N(0,1)$

"minimum Ass" ex. $y = f(x)$, f is smooth, f is learnable.

② ① 1 data, $y = \langle \theta, x \rangle$; 10^5 data $y = \langle \theta, x \rangle$

② 1 data, 10^5 data: small NN, large NN

You use a parameterized family in Empirical risk minimization, why you call "non-parametric"?

Hardness of ERM

Error of ERM

IEMS 402 Focus

Approximation Error + Generalization Error + Optimization Error

Assume to be 0

$$\inf_{\theta' \in \Theta} \mathscr{R}(f_{\theta'}) - R^*$$

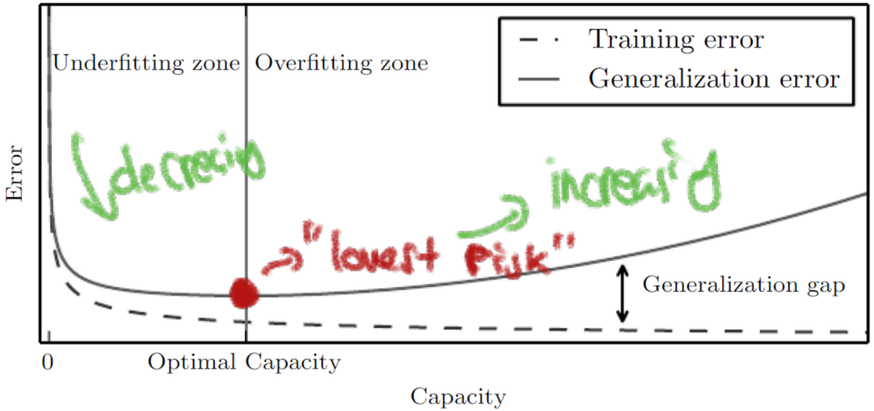
$$\sup_{\theta \in \Theta} |R(f_\theta) - \hat{R}(f_{\hat{\theta}})|$$

→ "Empirical Process"
⇒ Weak Convergence and Empirical Process

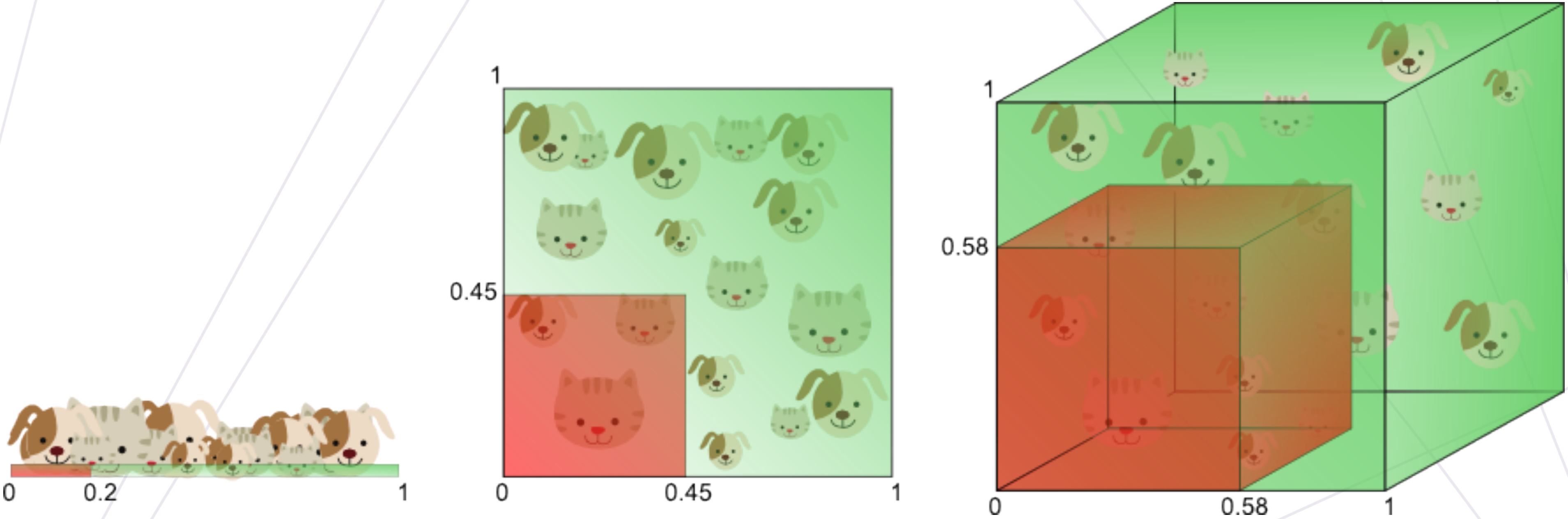
When we use more powerful parameterized family, e.g. Θ is larger:

- Approximation error is smaller!
- Generalization error is larger!

Bias-Variance Trade-off



Approximation: Curse of Dimensionality



Formulation: Approximate a smooth function

Fact. The number of parameters N required to achieve an approximation error of at most ϵ can be estimated by:

$$N \approx \left(\frac{1}{\epsilon}\right)^{d/s}$$

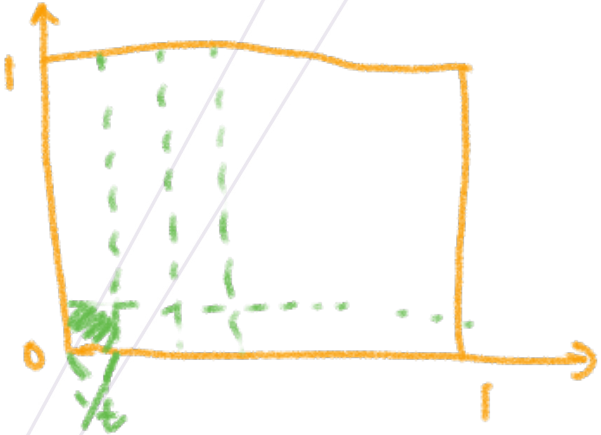
d Dimension

s smoothness

$N = d^s \cdot \left(\frac{1}{\epsilon}\right)^{d/s}$

s-th order Smoothness:

$f(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + f^{(s)}(x_0)(x - x_0)^s + O(\|x - x_0\|^{s+1})$



[box] $(0, 1/t)^d$ ① How many [box] I have t^d

② on each cube, replace the function by s-order Taylor Expansion

① How many parameter? $d^s \cdot t^d = N$

② Error? $\left(\frac{1}{t}\right)^s = \epsilon$

- Another Formulation see Homework 1 Question 3.

Formulation: Approximate a smooth function

Fact. The number of parameters N required to achieve an approximation error of at most ϵ can be estimated by:

$$N \approx \left(\frac{1}{\epsilon}\right)^{\frac{d}{s}}$$

d Dimension
s smoothness

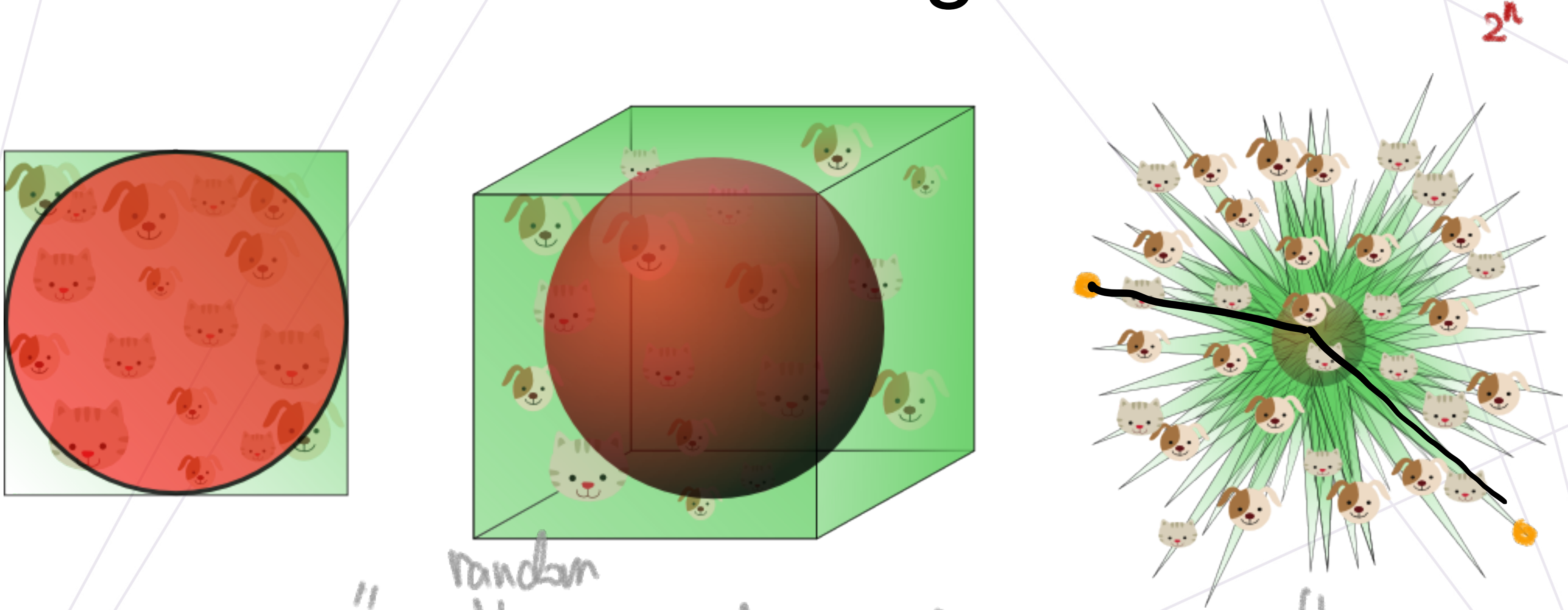
$N = d^s \left(\frac{1}{\epsilon}\right)^{\frac{d}{s}}$

① s is a constant. $N \approx (\frac{1}{\epsilon})^d$

② d/s is a constant. $s \propto d$, "d^s is a exp to d but it's a good poly respect to ϵ^{-1}"

- Another Formulation see Homework 1 Question 3.

How to think about High Dimension



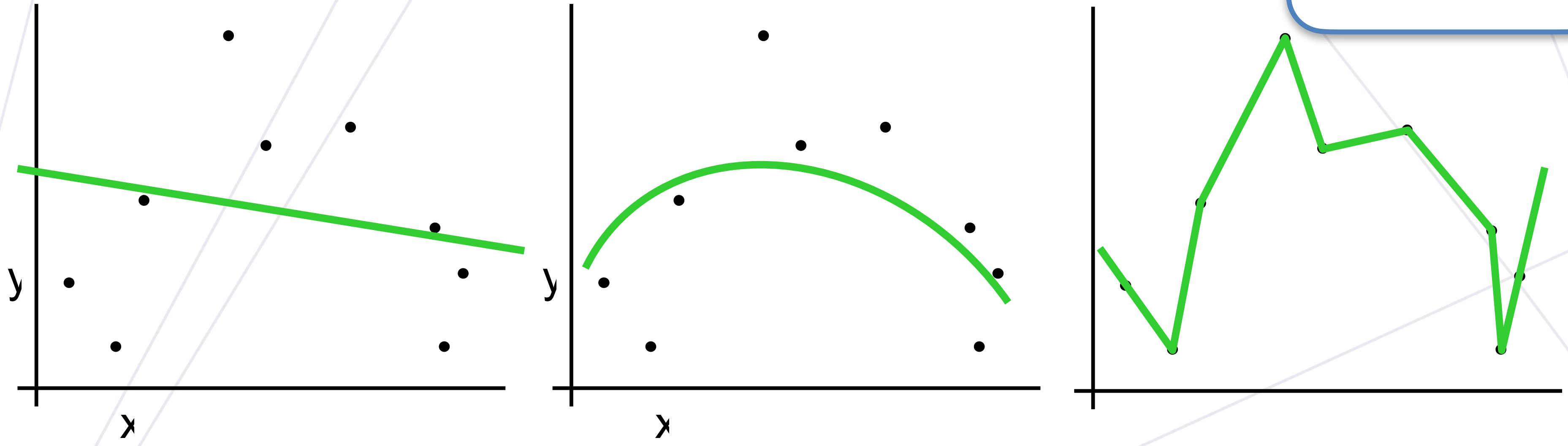
"Two random points' distance is a constant"

Generalization: Overfitting?

y = f(x) + noise

Can we learn f from this data?

Repeated Parrot
vs
understanding



Degree of Freedom

Suppose that we observe $y_i = r(x_i) + \epsilon_i (i = 1,\ldots,n)$, where the errors ϵ_i are uncorrelated with common variance $\sigma^2 > 0$

Now consider the fitted values $\hat{y}_i = \hat{r}(x_i)$ from a regression estimator $\hat{r}$.

Degree of freedom is defined as $\mathrm{df}(\hat{y}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(\hat{y}_i, y_i)$.

"How much I remember the label"

Degree of freedom

R $\hat{R}$

Fact. $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \hat{y}_i)^2\right] - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right] = \frac{2\sigma^2}{n}\,\mathrm{df}(\hat{y})\,.$

Generalization error

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} 2(y_i' - y_i)(\hat{y}_i - y_i) + (y_i' - y_i)^2\right]$$

$y_i' = r(x_i) + e_i'$

$y_i = r(x_i) + e_i$

$= 2(e_i - e_i')(\hat{y}_i - y_i)$ (independent)

$= 2e_i(\hat{y}_i - y_i) = 2\mathrm{cov}(y_i, \hat{y}_i) - 2\sigma^2$

$(y_i' - y_i)^2 \to 2\sigma^2$

Example of DOF 1

n labels for n data

① $\hat{y}_i = y_i$ $\quad df = \frac{1}{\sigma^2}\sum_{i=1}^{n} cov(\hat{y}_i, y_i) = n$

$= \sigma^2$

② $\hat{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i$ $\quad df = \frac{1}{\sigma^2}\sum_{i=1}^{n} cov(\hat{y}_i, y_i) = 1$

1 label for n data.

" DOF is the number of parameter we are using "

Example of DOF 2

Not Required

③ linear Regression $\mathbb{R}^n \ni \hat{y} = X\beta = X[(X^TX)^{-1}X^Ty]$

$\beta = (X^TX)^{-1}X^Ty$

$\frac{1}{\sigma^2} tr[cov(\underbrace{X(X^TX)^{-1}X^Ty}_{\mathbb{R}^n}, \underbrace{y}_{\mathbb{R}^n})]$

$= \frac{1}{\sigma^2} tr[X(X^TX)^{-1}X^T cov(y, y)]$

$= tr(X(X^TX)^{-1}X^T) = tr(X^TX(X^TX)^{-1}) = tr(I_p)$

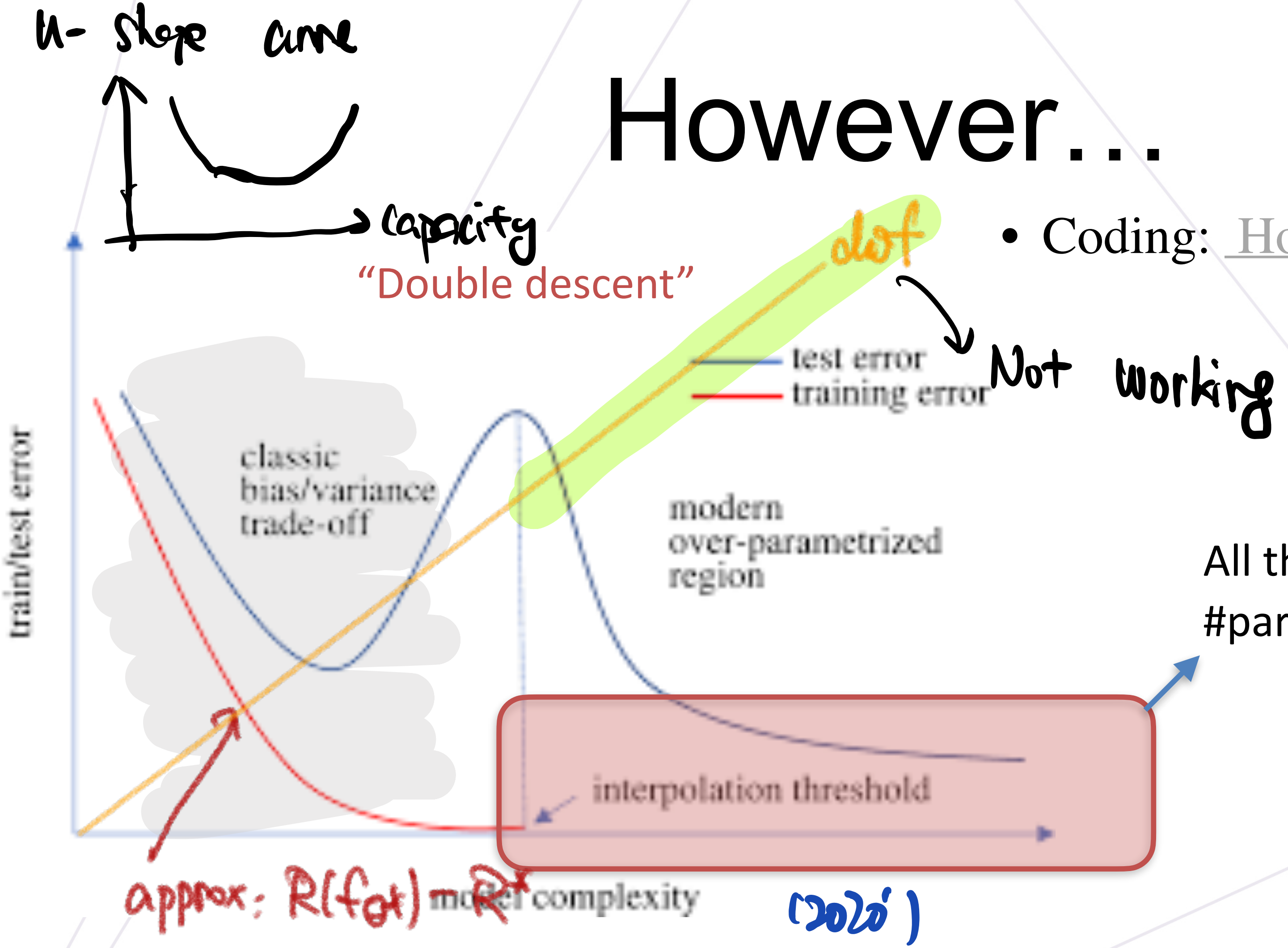
"number of parameter"

$p = tr(I_p)$

$tr(AB) = tr(BA)$

However…

- Coding: Homework 2 Question 3.



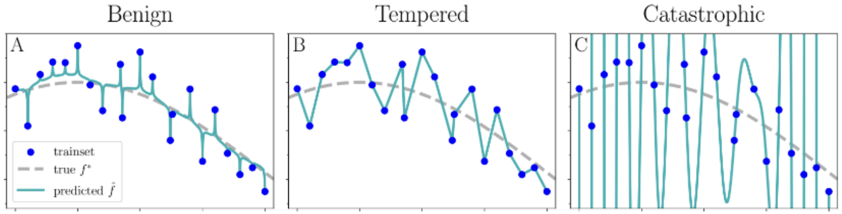
All the data can be remembered

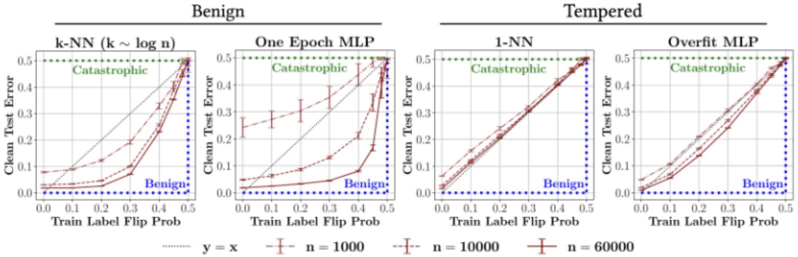
#parameter > #data

Taxonomy of (over)fitting

	Regression	Classification
Benign	$\lim_{n\to\infty} \mathcal{R}_n = R^*$	$\lim_{n\to\infty} \mathcal{R}_n = R^*$
Tempered	$\lim_{n\to\infty} \mathcal{R}_n \in (R^*, \infty)$	$\lim_{n\to\infty} \mathcal{R}_n \in (R^*, 1 - \frac{1}{K})$
Catastrophic	$\lim_{n\to\infty} \mathcal{R}_n = \infty$	$\lim_{n\to\infty} \mathcal{R}_n = 1 - \frac{1}{K}$

Table -1.1: Taxonomy of (over)fitting.



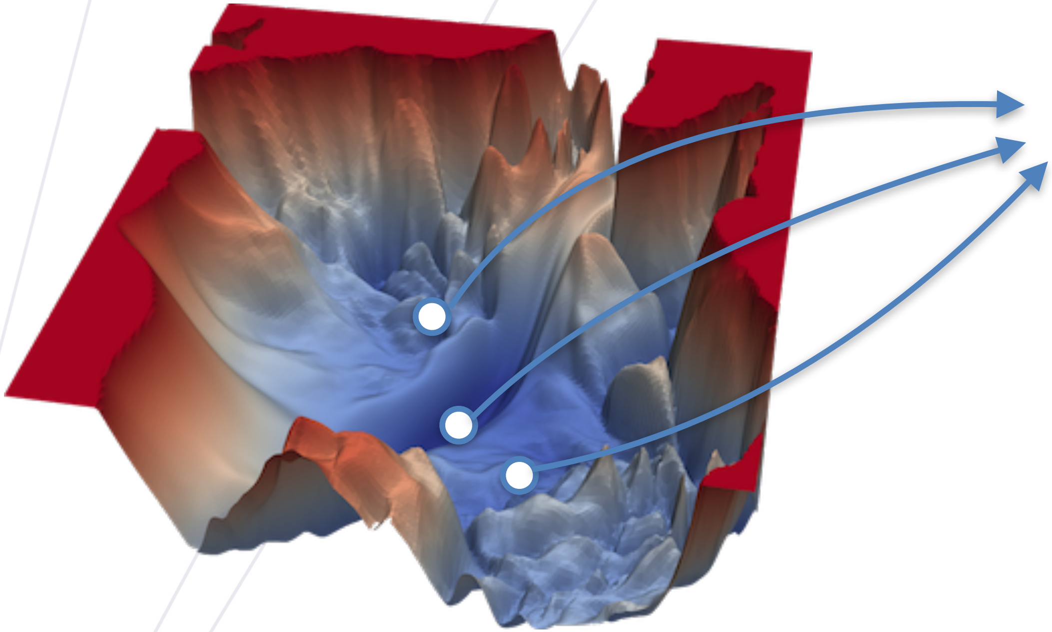


Mallinar, Neil, et al. "Benign, tempered, or catastrophic: A taxonomy of overfitting (2022)." arXiv preprint arXiv:2207.06569.

CIFAR100 : 6000
ResNext : #parameter 68.1M

Implicit bias

"Multiple Minima"

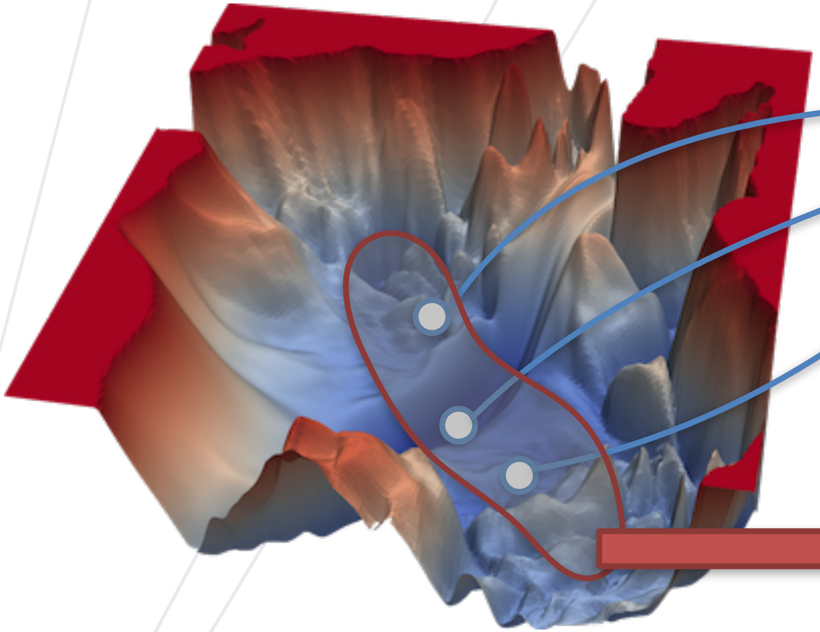


Many models can achieve low training loss

Loss landscape of VGG on CIFAR

Implicit bias

"Multiple Minima"



Loss landscape of VGG on CIFAR

Many models can achieve low training loss

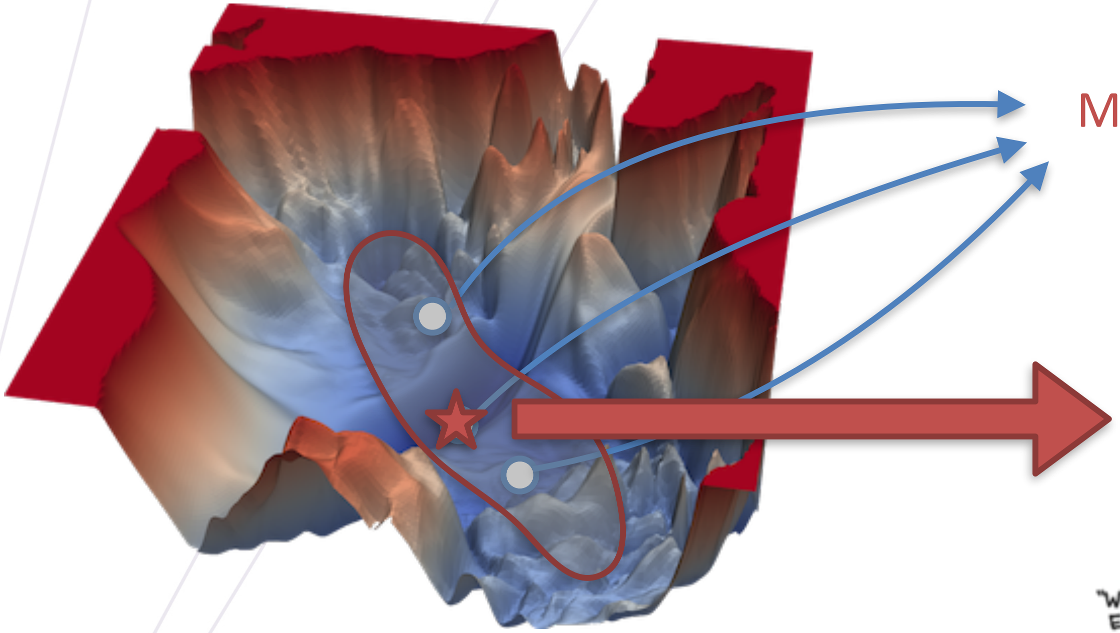
too pessimistic

Traditional bounds:

$$\sup_{\theta \in \Theta} |R(f_\theta) - \hat{R}(f_{\hat{\theta}})|$$

Implicit bias

"Multiple Minima"

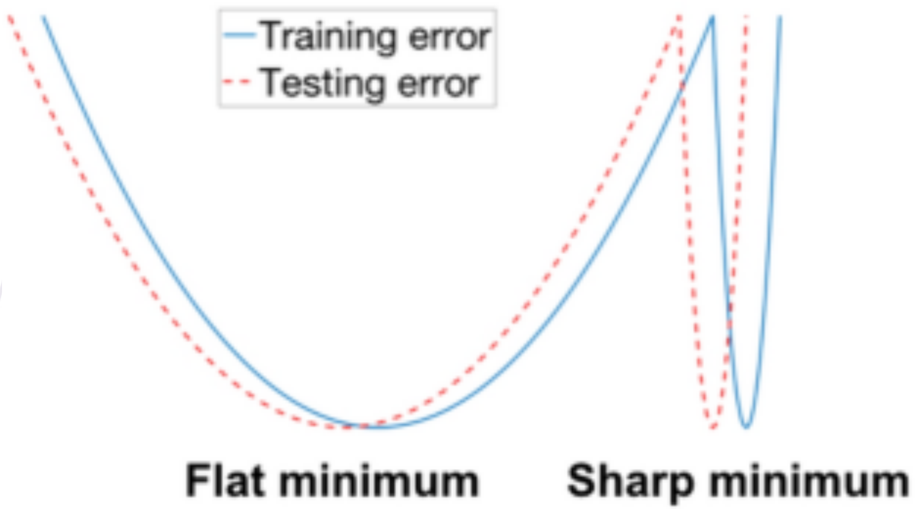


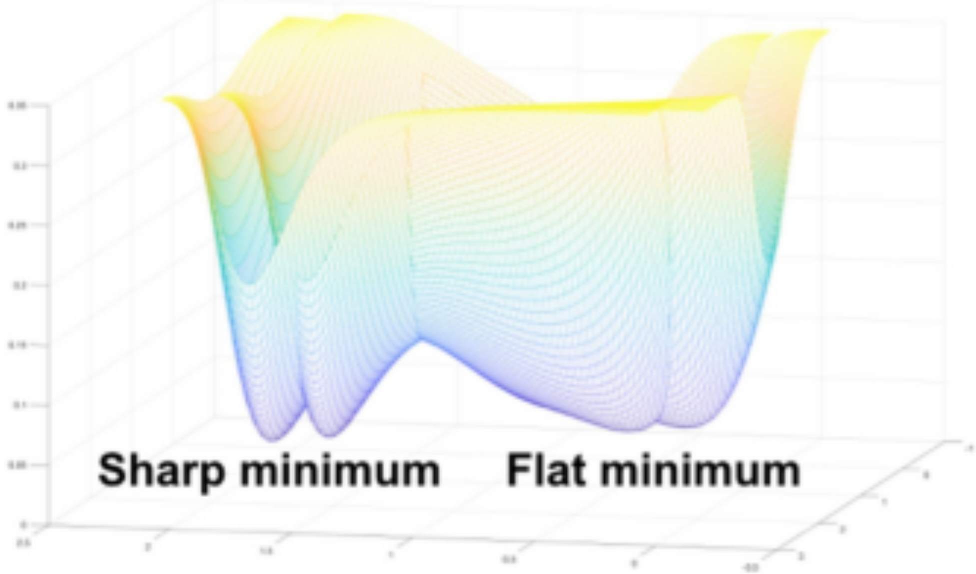
Many models can achieve low training loss

Loss landscape of VGG on CIFAR

What’s special about over-para

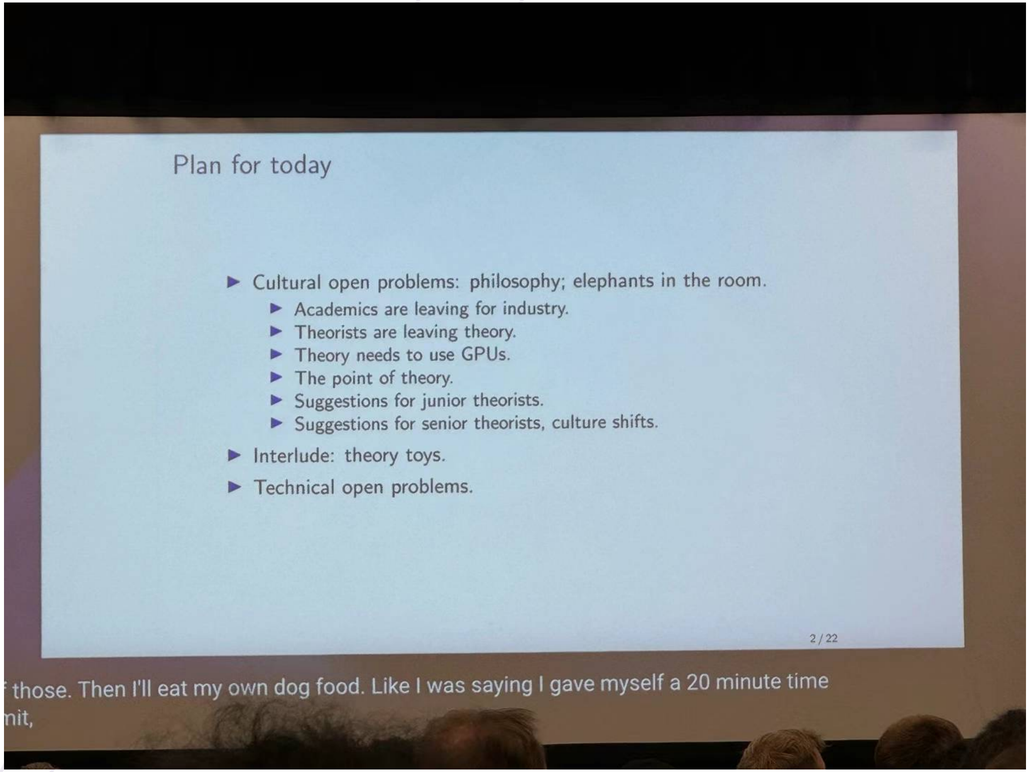
“Multiple Minima”





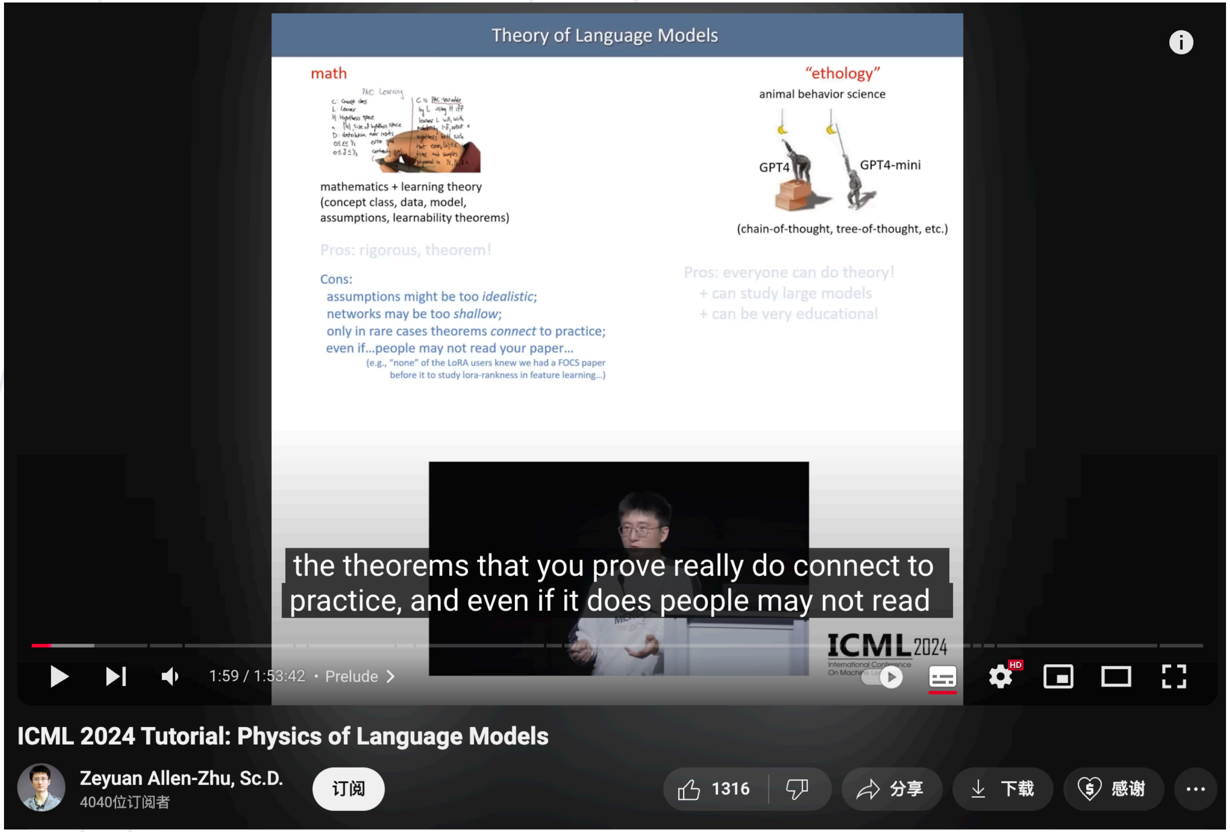
Last Note on Learning Theory

ML Theory workshop @Neurips24



https://cims.nyu.edu/~matus/neurips.2024.workshop/talk.pdf

Math-physics-ethology



Physics of language model
ICML 2024
https://shorturl.at/ZDwQE

Learning Theory Today