Announcements

- Email me if you have not yet been added to our PSC allocation
- Lab 2 assigned today, due in 3 weeks
- Add a discussion question on Canvas by Sept. 22
    - Auditors can submit questions via email

Discussion Questions - Thomas Huang (Waymo)

On Sept. 29, Thomas Huang (Research Scientist, Waymo AI Foundations) will give a guest lecture about foundation models at Waymo.

As part of this guest lecture, we will have a structured Q&A / discussion.

- To guide this discussion, please write a question, and post it under this discussion (i.e., as a reply).
- Your question should be unique! Please read other questions which have already been posted before posting your question.
- If your question is clearly off-limits (i.e., related to proprietary technical details), I will respond to your post and request that you submit a new question.

Reply

L7: Applied Machine Learning

18-848 RadarML

Tianshu Huang

Crash Course: Applied Machine Learning

- Introduction to Machine Learning
  - Empirical Risk Minimization
  - Deep Learning & Transformers
- **Training & Evaluation**
  - The ML Life Cycle
  - **Variance of Evaluation**
- Learning at Scale
  - Foundation Models
  - Scaling Laws

Review: **Variance, Stddev, Stderr**

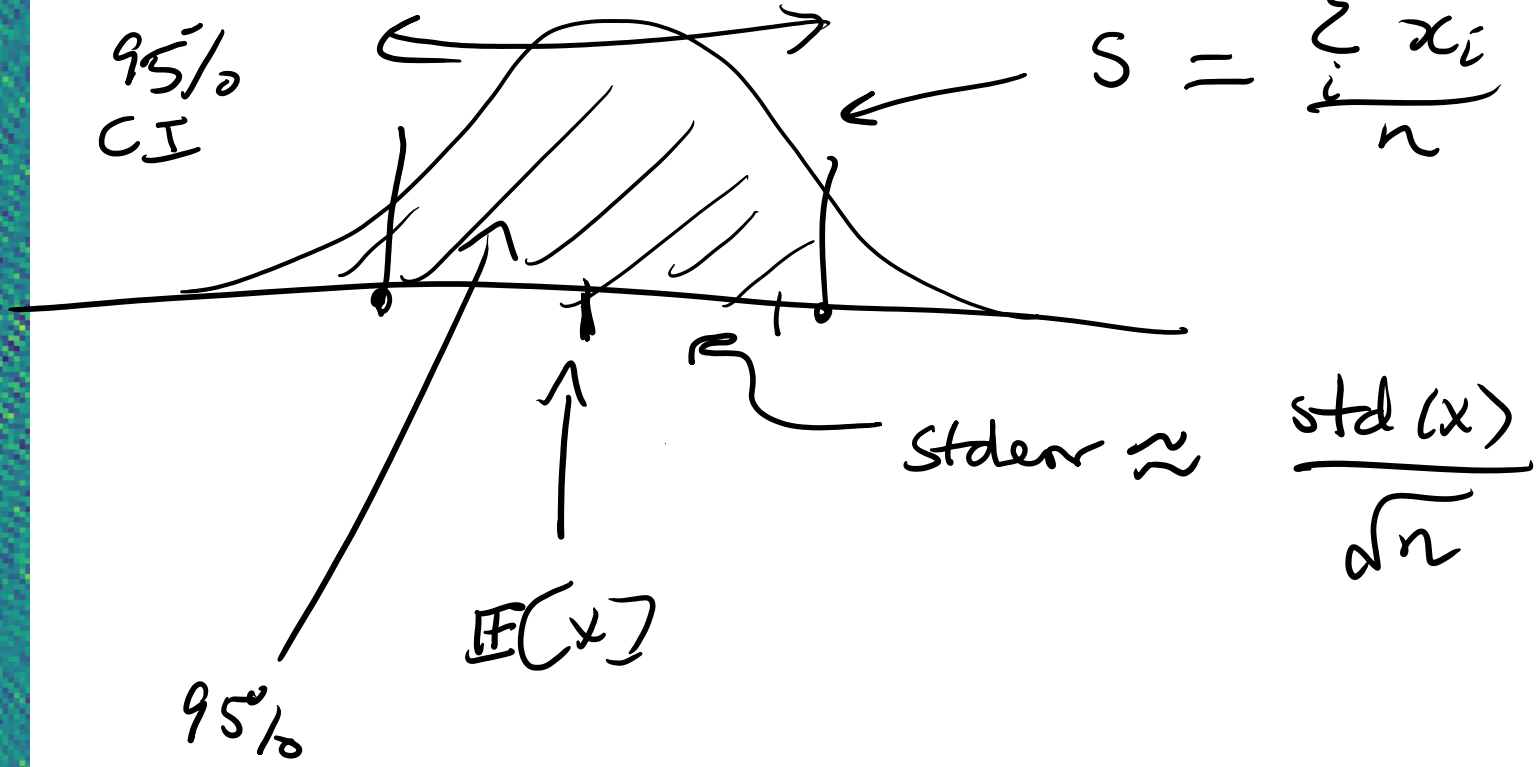
$$Var(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

$$= \mathbb{E}[(X - \mathbb{E}[X])^2] \leftarrow \text{stable}$$

$$std(X) = \sqrt{var(X)}$$

$$stderr(X) = \frac{std(X)}{\sqrt{N}}$$

Review: **Confidence Intervals**



How much test data do we Need?

Evaluation Scales with Standard Error

Definition 2.2 (Evaluation noise). *An empirical mean over n_{test} i.i.d. test samples has standard error*

$$SE = \frac{\sigma}{\sqrt{n_{test}}}.$$

Uncertain in single estimate

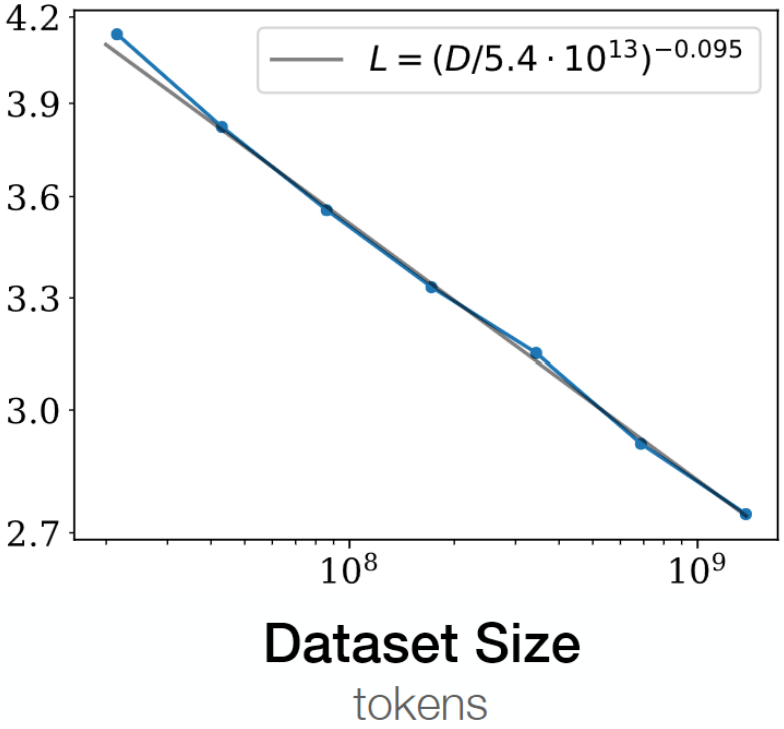
Assumption 2.2 (Regret of a noisy decision). *Choosing among candidates by their measured scores, when each score is off by $O(\sigma/\sqrt{n_{test}})$, costs us quality of the same order. Writing the shortfall relative to a perfect oracle decision:*

$$U_{test}(n_{test}) = -\frac{b}{\sqrt{n_{test}}}, \qquad b > 0.$$

regret of choosing wrong model

$$\propto \frac{1}{\sqrt{n}}$$

Data Scaling is Approximately Logarithmic



$L \propto \log(N)$

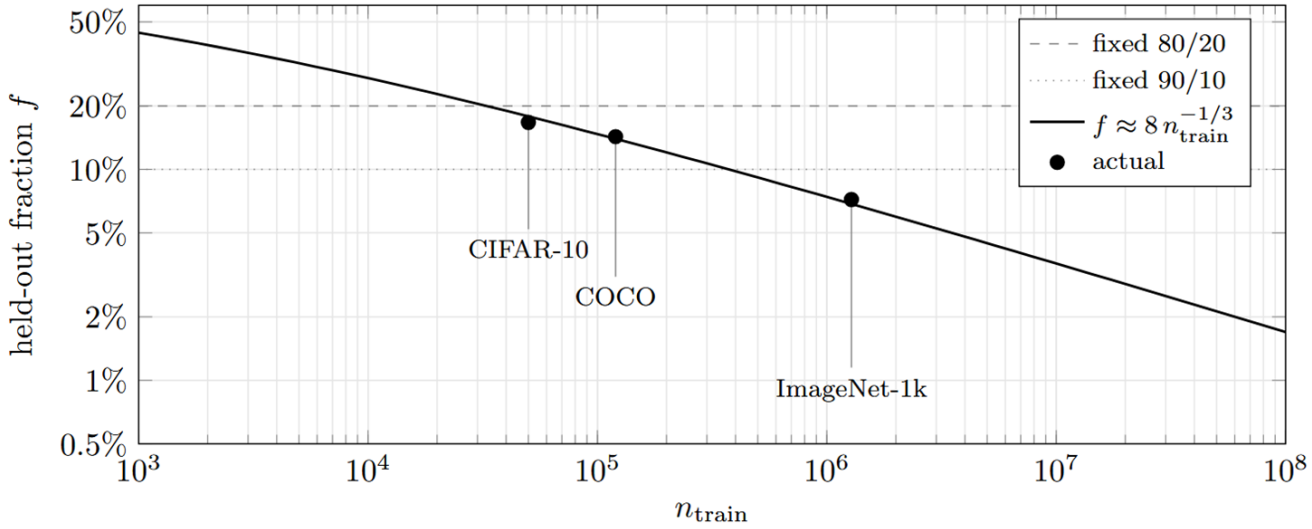
How much val/test data do we need?

If we assume that model quality scales with $\log(n_{\text{train}})$ and evaluation accuracy with $n_{\text{test}}^{-1/2}$, then:

$$\underbrace{\frac{\partial U_{\text{train}}}{\partial n_{\text{train}}} \propto \frac{1}{n_{\text{train}}}}_{\text{log data scaling}} = \underbrace{\frac{\partial U_{\text{test}}}{\partial n_{\text{test}}} \propto n_{\text{test}}^{-3/2}}_{\sqrt{n}\text{ estimation error}} \implies \boxed{n_{\text{test}} \propto n_{\text{train}}^{2/3}}$$

more data
⇒ more test #
less test %

How much val/test data do we need?



What if we are measuring rare events?

$$X = \begin{cases} 0 & \text{w.p. } 1-p \\ 1 & \text{w.p. } p \end{cases}$$

$p \approx 0$

$$\text{Var}(X) = p(1-p)$$

$$\text{CV}(X) = \frac{\sqrt{p(1-p)}}{p} \approx \frac{\sqrt{1-p}}{\sqrt{p}} \approx \frac{1}{\sqrt{p}}$$

$$\text{CV stderr} = \frac{1}{\sqrt{np}}$$

$p \downarrow$ $n \uparrow$ prop.

Review: **Coefficient of Variation**

$N \longrightarrow SE = \pm 1\%$

$E[x] = 1\%$

$$\frac{SE(x)}{E[x]} =$$ relative SE or coefficient of variation of SE

How do we compare two methods?

(↑ better)

$Y_1 \sim N(\mu, \sigma^2)$

$Y_2 \sim N(\mu + \delta, \sigma^2)$

$Y_2 = Y_1 + \delta$

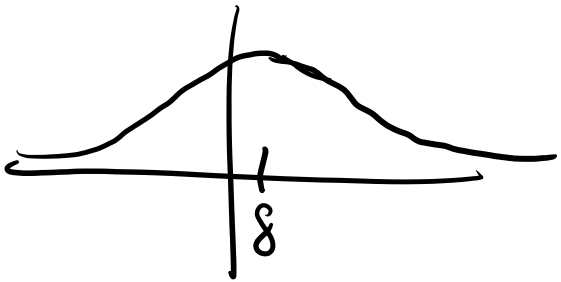
$Y_2 - Y_1 = \delta$

	Y_1	Y_2
x_1	→	→
x_2	→	→
x_3	→	→
⋮		

$S_1 \sim N(\mu, \frac{\sigma^2}{n})$

$S_2 \sim N(\mu + \delta, \frac{\sigma^2}{n})$

$S_2 - S_1 \sim N(\delta, 2\frac{\sigma^2}{n})$



Aside: what test?

You may have heard of:

- t-test – small samples
- permutation test
- chi-square test
- …

A Z-test is almost always good enough in deep learning

Similar performance = Rare events!

$Y_2 = Y_1 + \Delta$

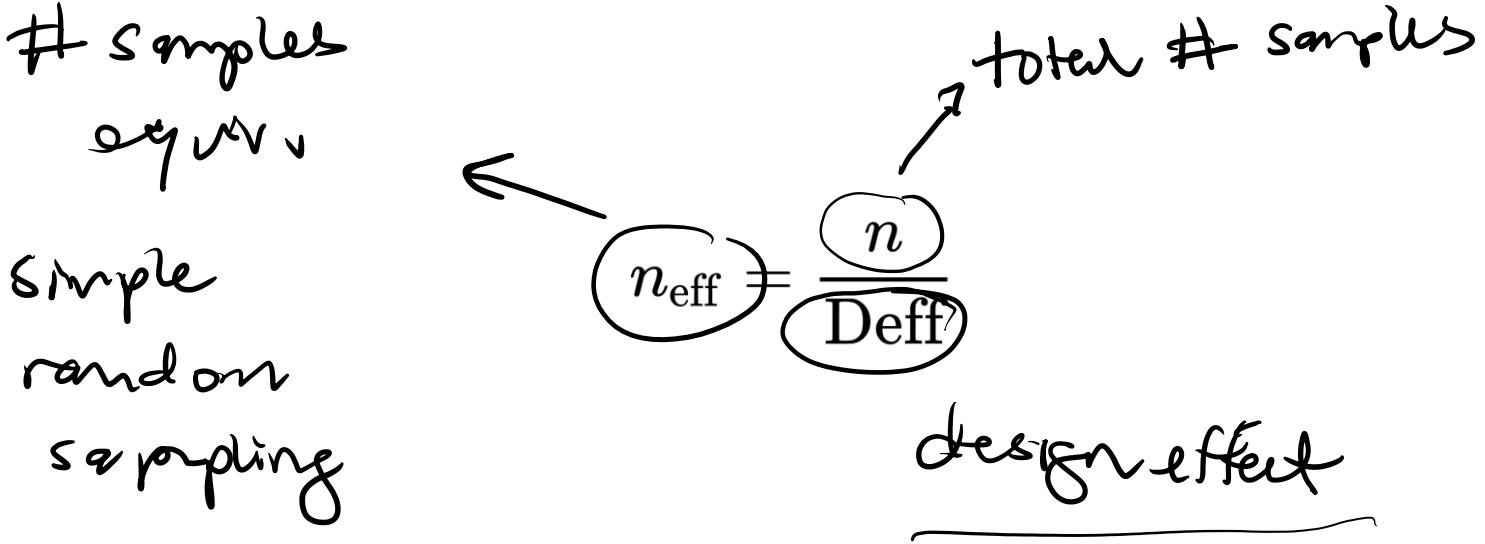
$\Delta \sim \text{Bernoulli}(p)$

$\Rightarrow \begin{cases} \text{w.p. } 1-p, & Y_2 = Y_1 \\ \text{w.p. } p, & Y_2 = Y_1 + 1 \end{cases}$

(D) $\Rightarrow Y_2 - Y_1 \simeq \Delta$

more similar $\Rightarrow$ prop. more data

Effective Sample Size (ESS)



Correlation ⇒ ESS < N ≈ "real samples"

- duplicate data
- similar scenes, repeated setting
- videos / time series

ESS > N?

(positive) correlation $\Rightarrow$ ESS < N

- "antithetical sampling"
- data curation

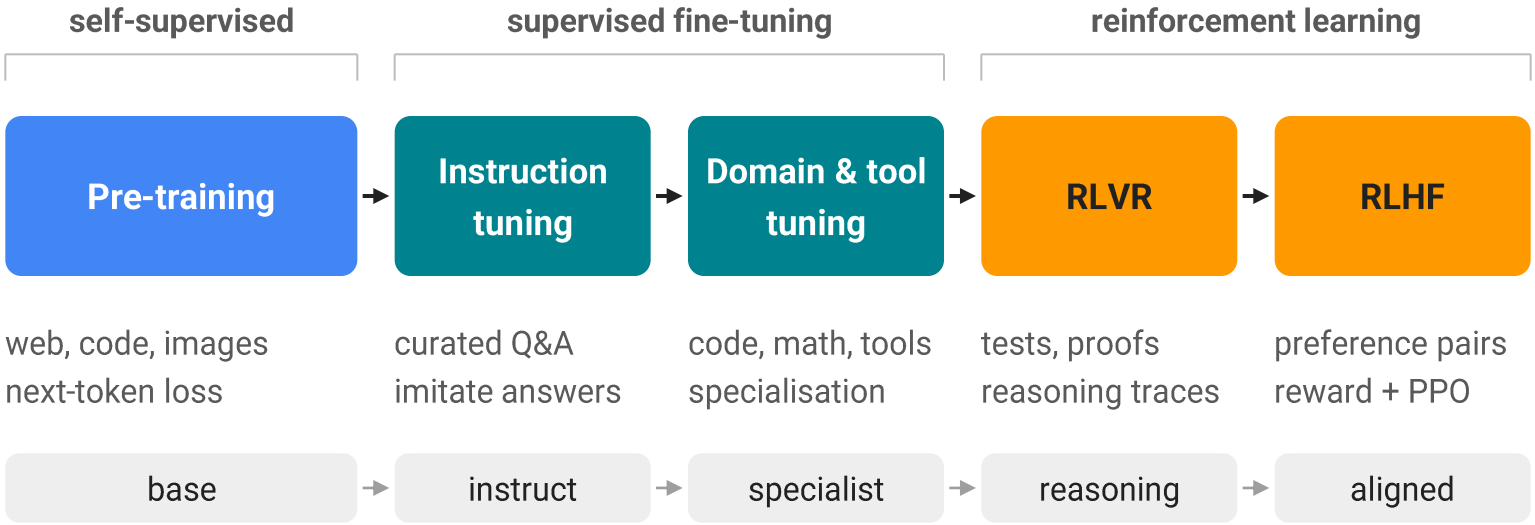
More Training vs Evaluation Data

↑ Eval Split	↑ Train Split
Noisier metrics	Consistent metrics
More complicated method	Simpler method
Correlated data (ESS < N)	Uncorrelated or curated data
Rare events / similar methods	Dense evaluation metrics
Weaker scaling $\sim \frac{1}{\sqrt{n}}$	Stronger scaling $\log$
If you are uncertain	

Crash Course: Applied Machine Learning

- Introduction to Machine Learning
  - Empirical Risk Minimization
  - Deep Learning & Transformers
- Training & Evaluation
  - The ML Life Cycle
  - Variance of Evaluation
- **Learning at Scale**
  - **Foundation Models**
  - Scaling Laws

A Modern Foundation Model Pipeline



Density of Supervision

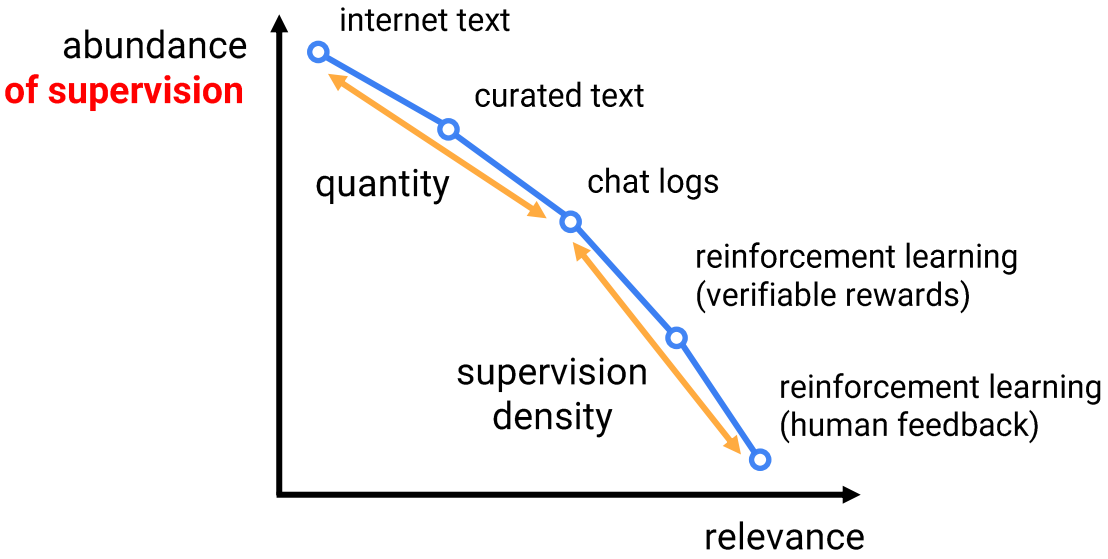
Yann Lecun, 2016

- "Pure" Reinforcement Learning (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- Supervised Learning (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- Unsupervised/Predictive Learning (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

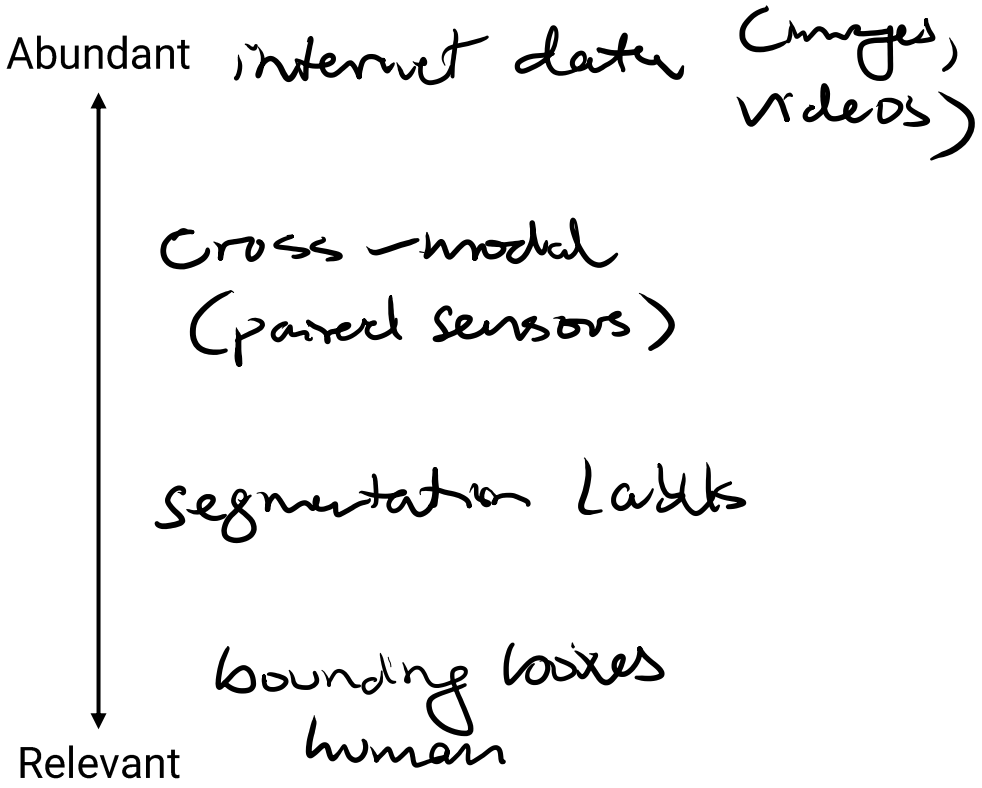
- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Pareto Frontier of Dataset Types

Relevance vs abundance and density of supervision



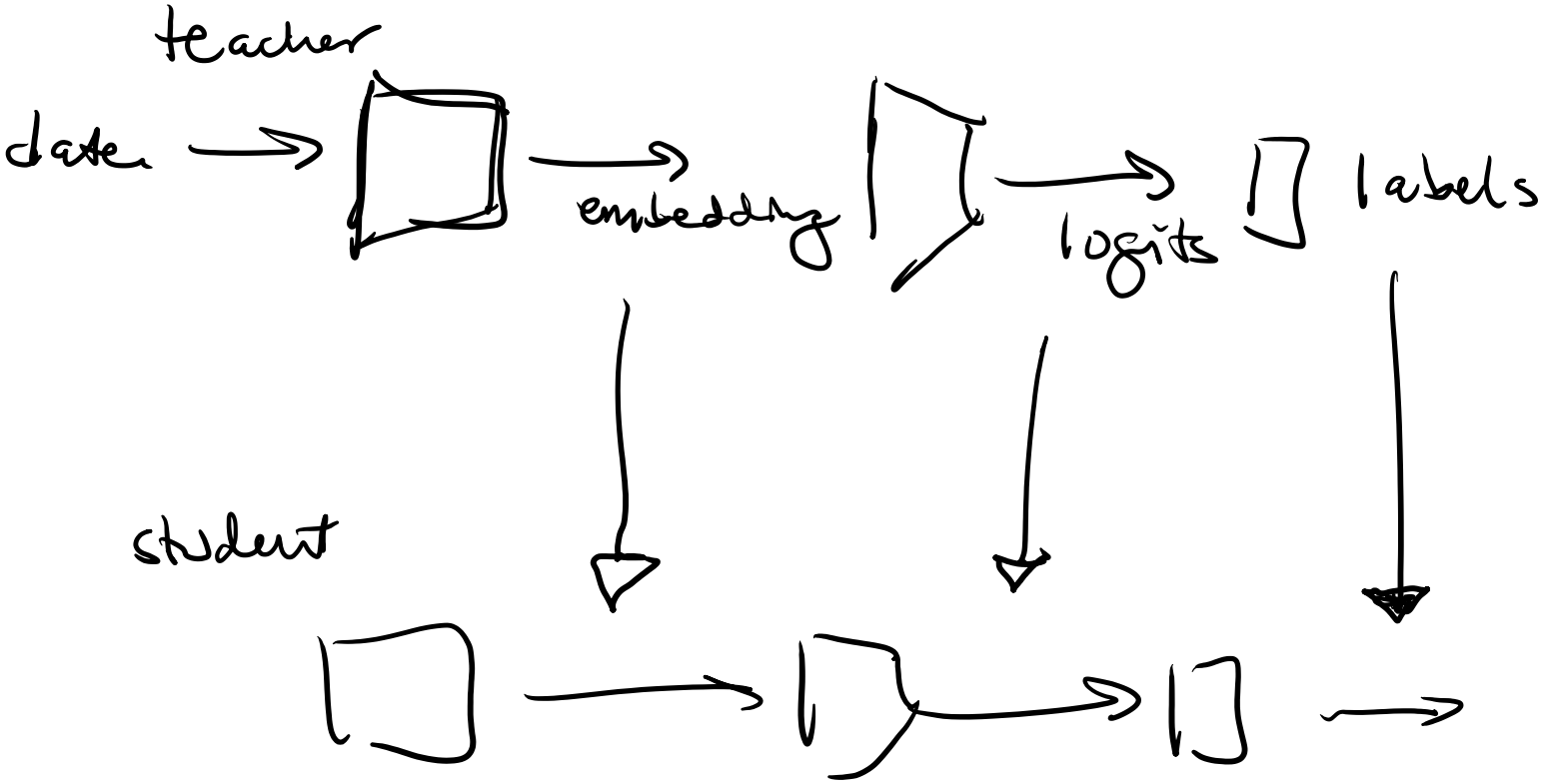
Perception Foundation Models



What to do with a foundation model?

- deploy directly
- distillation
- as a benchmark, upper bound

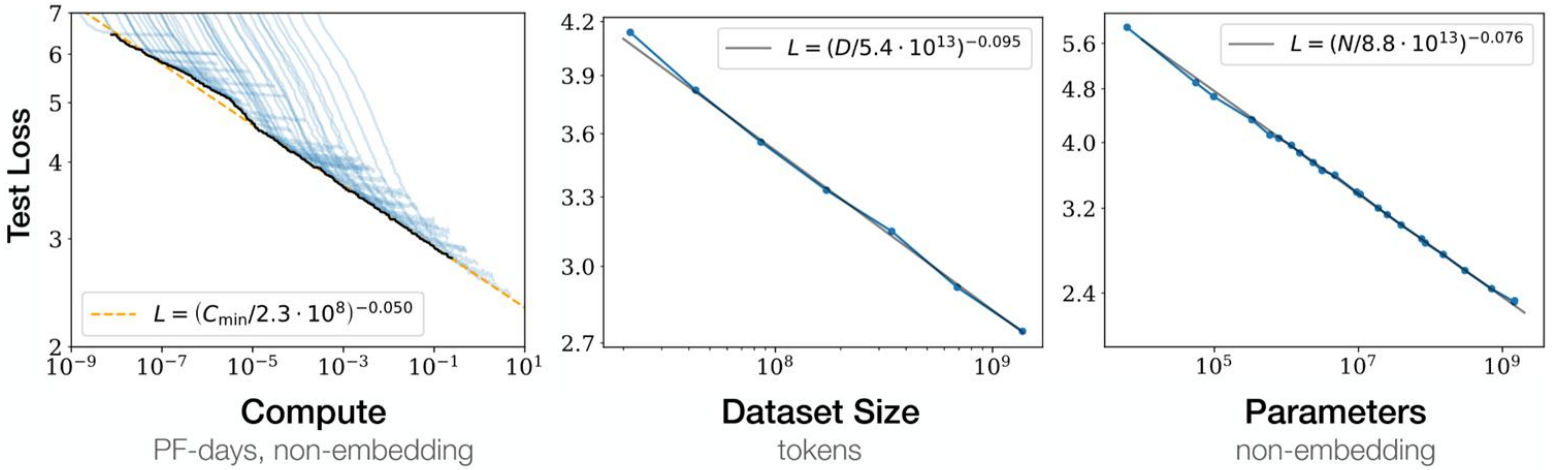
Distillation is a more dense form of supervision



Crash Course: Applied Machine Learning

- Introduction to Machine Learning
  - Empirical Risk Minimization
  - Deep Learning & Transformers
- Training & Evaluation
  - The ML Life Cycle
  - Variance of Evaluation
- **Learning at Scale**
  - Foundation Models
  - **Scaling Laws**

What is a scaling law?

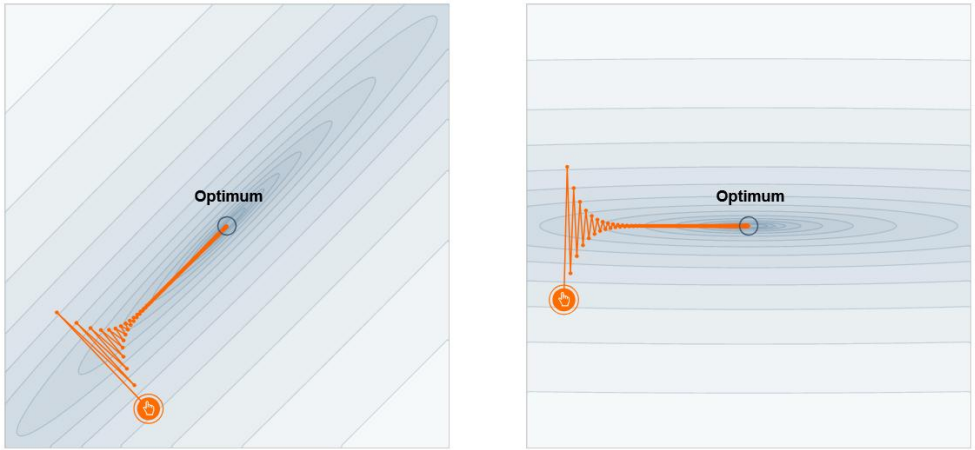


“Scaling Laws for Neural Language Models”, https://arxiv.org/abs/2001.08361

Model Scaling

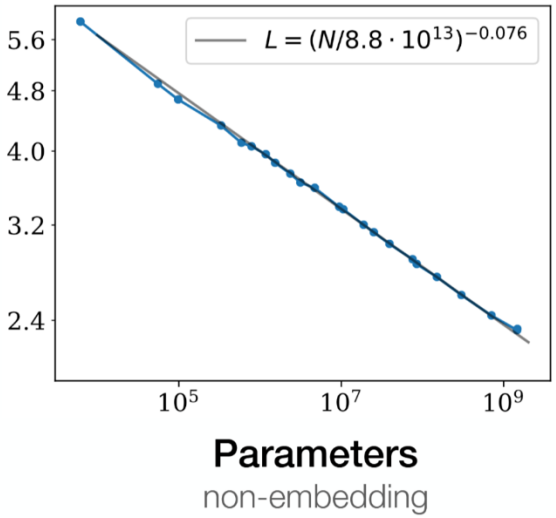
Gradient descent is intrinsically regularizing

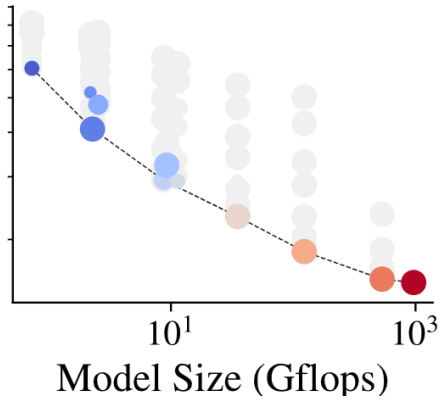
Here's the trick. There is a very natural space to view gradient descent where all the dimensions act independently — the eigenvectors of A.



https://distill.pub/2017/momentum/

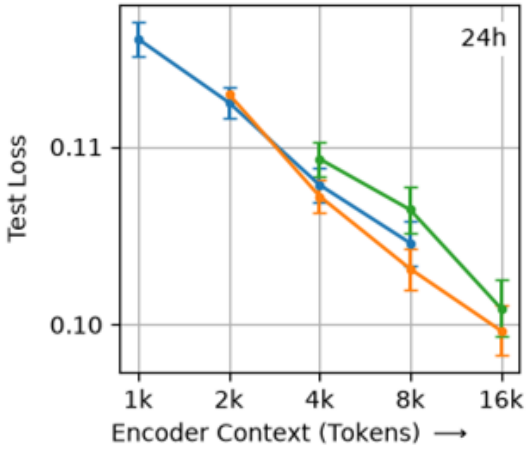
Model Scaling

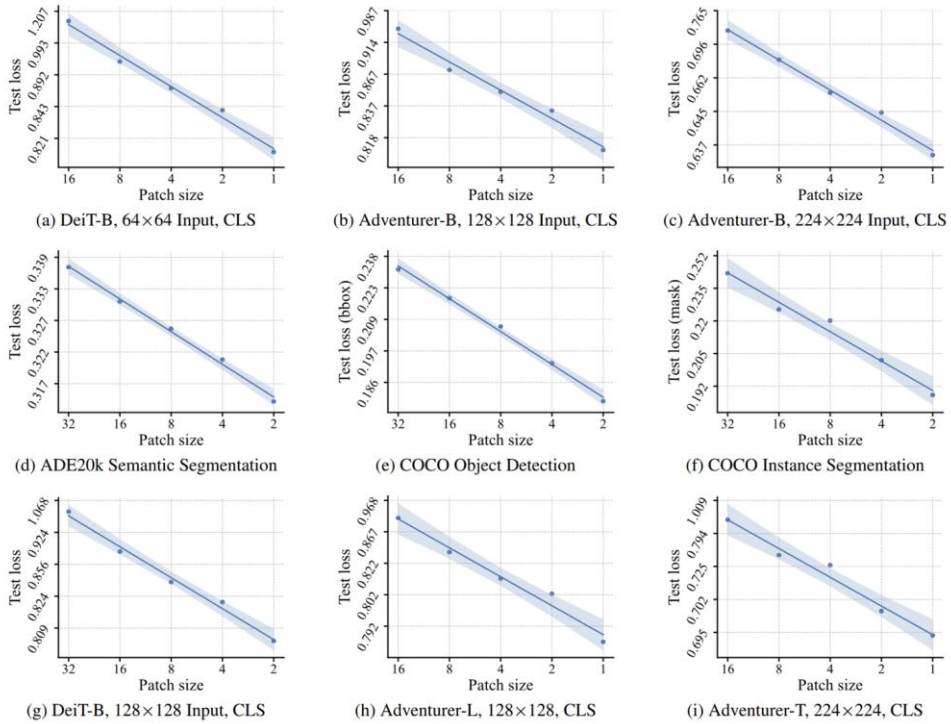




Left: Scaling Laws for Neural Language Models https://arxiv.org/abs/2001.08361
Right: Scaling Vision Transformers https://arxiv.org/abs/2106.04560

Context Scaling





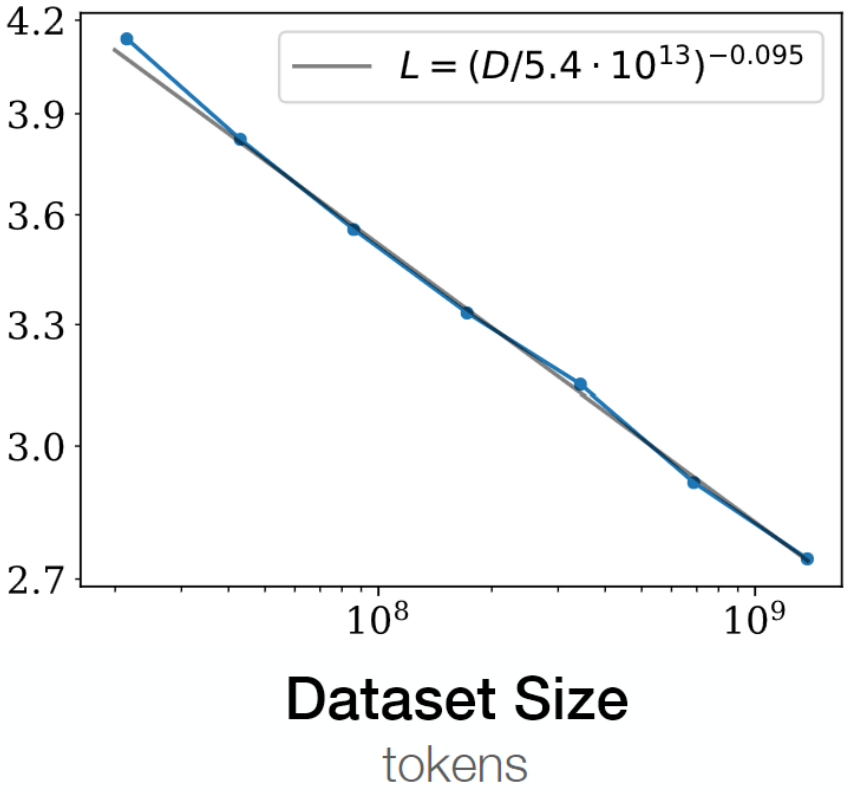
Left: Unpublished radar paper
Right: Scaling Laws in Patchification https://arxiv.org/abs/2502.03738

Data Scaling

$$\log(L) = -\alpha \cdot \log(N) + b$$

$$L = \exp(\alpha \log(N) + b)$$

$$\approx e^{b} \cdot N^{-\alpha}$$



Linear Regression Excess EPE Power Law

$$y = X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I_N).$$

$$\hat{\beta} = \arg\min_\beta \|y - X\beta\|^2 = (X^T X)^{-1} X^T y.$$

$\mathbb{E}[(y - \hat{y})^2]$

expected pred. error

$$\mathbb{E}[EPE] = \sigma^2 \left(1 + \frac{d}{N}\right)$$

irreducible reducible

Linear Regression Excess EPE Power Law

$$\mathbb{E}[EPE] = \sigma^2 \left(1 + \frac{d}{N}\right)$$

excess EPE $\sim \sigma^2 \cdot \frac{d}{N}$

L

$L \propto \frac{d}{N}$

$\log L \propto \log(d) - \log(N)$

NTK (Neural Tangent Kernel) Theory

“Every Model Learned by Gradient Descent is Approximately a Kernel Machine”
https://arxiv.org/abs/2012.00152

Linear	fixed **d**
Kernel	**d = N**
NN Implicit Kernel	**d ∝ N**

$\sigma^2 \cdot \frac{d}{N} \qquad d = N^{1-\alpha}$

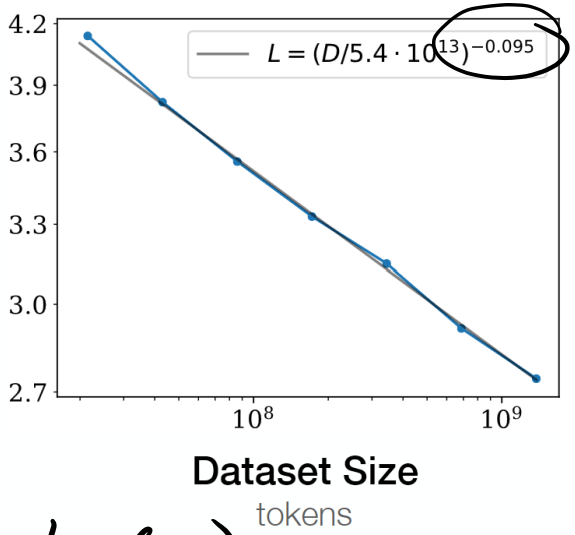
$\sigma^2 \cdot N^{-\alpha} \qquad \propto \qquad -\alpha \log(N)$

Why is data scaling approximately logarithmic?

2. For large models trained with a limited dataset with early stopping:

$$L(D) = (D_c/D)^{\alpha_D}\ ;\ \alpha_D \sim 0.095,\quad D_c \sim 5.4 \times 10^{13}\ \text{(tokens)}$$

$$\beta N^{-\alpha} = \beta e^{-\alpha \log N} \approx \beta \left(1 - \alpha \log N + \underbrace{O(\alpha^2)}_{\approx 0}\right) = \beta\,(1 - \alpha \log N) \propto \alpha \log(cN)$$



Key Takeaways

- Foundation models let us exploit large datasets which do not exactly match the downstream task
- Scaling among many axes (but especially data) is a power law which is approximately logarithmic

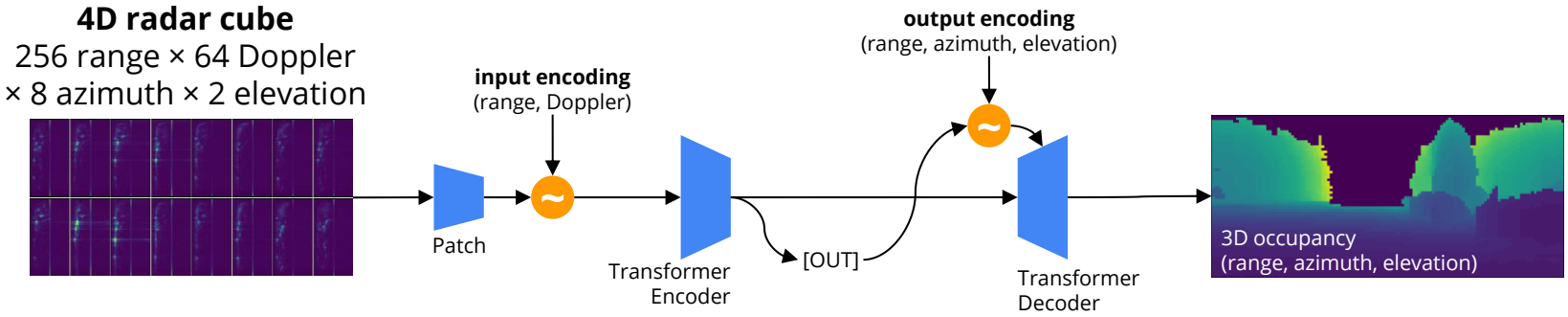
Lab 2 Posted

- Submit via canvas; due 10/06
  - One week revision period after comments are posted
- Training takes time. **Start early!**
  - Lab 2 infra has a substantial learning curve
  - PSC can have long queues
  - **No deadline extensions due to failure to plan ahead!**
- Lab 2 checkpoint due 09/22: log in to PSC
  - Or show that you have your own compute set up

In Lab 2, you will:

- Work with (relatively) complex infrastructure which may be new to you
- Propose and implement a change to the GRT model
- Run your own data scaling study
- Estimate the uncertainty of your results

Generalizable Radar Transformer (GRT) at a Glance



Programming Constructs & Tooling

Some things which may be new to you:

- Types (+ protocols, generics, Jaxtyping, …)
- Dependency Injection
- uv, hydra, slurm
- Linters, type checking, unit tests, pre-commit, CI (github actions)

Next Time

- What hardware components go into a radar?
- How do quirks in the hardware affect the radar signals?
- How do we choose the right hardware & components?