LECTURE 6

DEEP REPRESENTATION LEARNING II

How Can a Model Generalize?

PRESS → OR SPACE.

© Marc Rußwurm
Licensed under **Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)**.
You may share and adapt these slides for teaching, research, and other non-commercial educational purposes with attribution.
Commercial training, paid workshops, consulting seminars, or incorporation into commercial course products require prior permission.

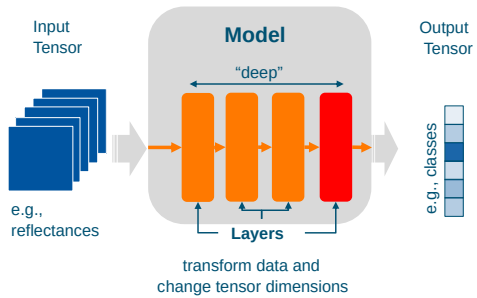
UNIVERSITÄT BONN

Learning Outcomes

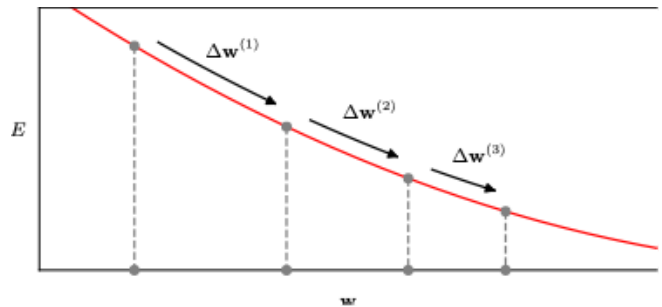
1. Explain **generalization** as the difference between optimizing a finite training sample and performing well on the underlying data distribution.
2. Explain how **model capacity, data, and regularization** determine underfitting and overfitting in the classical regime.
3. Explain why the **interpolation threshold, double descent, and grokking** challenge the classical bias–variance picture and motivate the modern overparameterized learning regime.

Recap Lecture 5 Deep Learning I

Deep Model Architectures



Parameter Optimization



Takeaway: We can now approximate any function with a neural network.

But is function approximation enough?

What is wrong here?

🧑‍🍳 Takeaway: The Pizza Baker Apprentice does not need to hold 📦 boxes to pick up a 🍕 pizza.



Approximation vs Generalization

Approximation

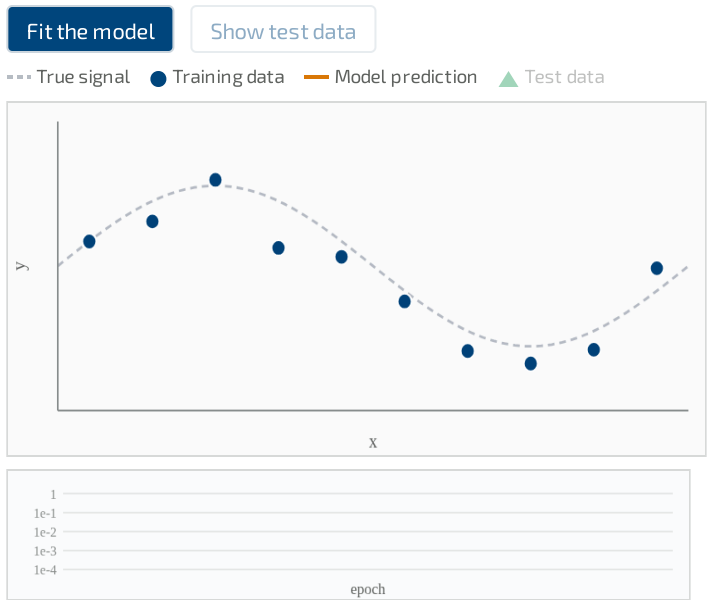
When training, we minimize the empirical risk

$$\widehat{\mathcal{R}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\big(f_\theta(\mathbf{x}_i), \mathbf{y}_i\big)$$

Generalization

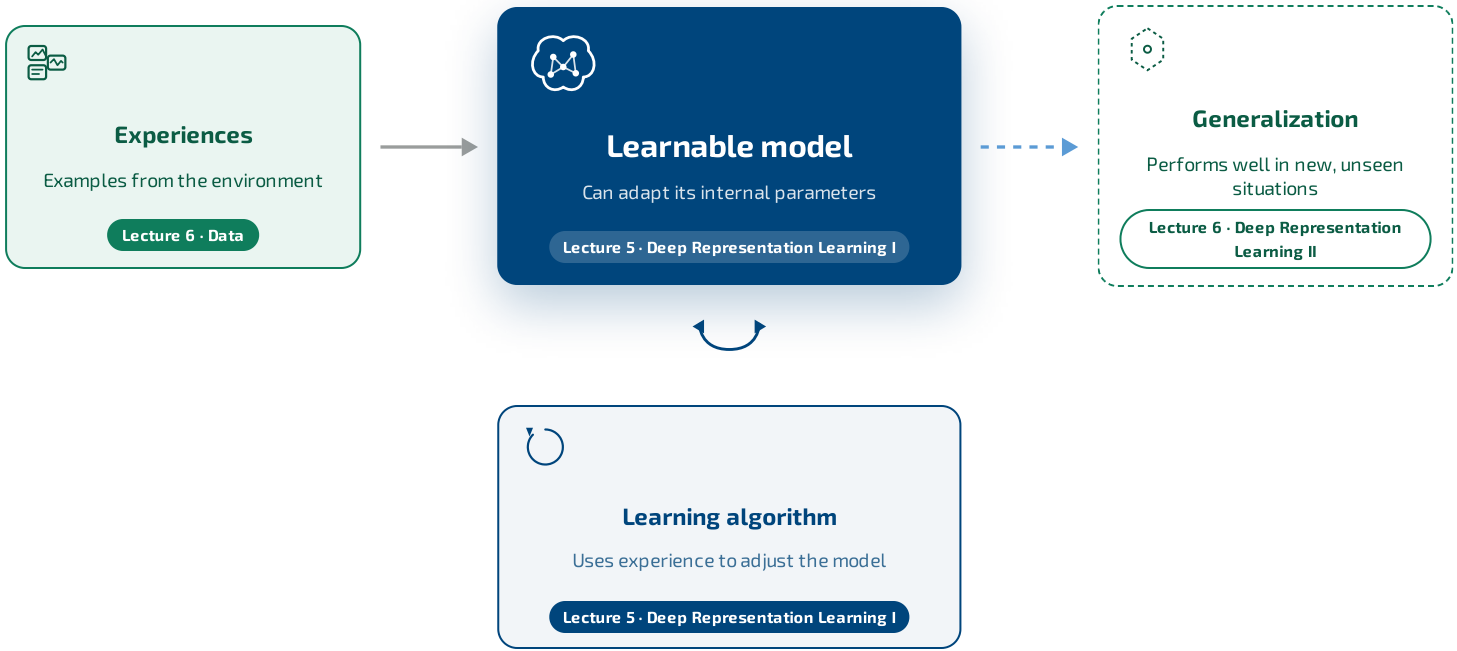
But in practice, we want the population (test) risk to be small

$$\mathcal{R}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{P}}\mathcal{L}\big(f_\theta(\mathbf{x}), \mathbf{y}\big)$$



A perfect training fit can still generalize poorly.

What is needed for a system to learn?



Data Samples and Distributions



Reality → Distribution → Dataset

Data we use in machine learning are not the world itself, but **samples drawn** (often imperfectly) from an **underlying distribution**

reality **distribution** **dataset samples**



design

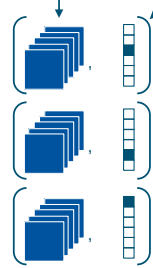
$P(X, Y)$

sampling

$\mathcal{D} = \{x_i, y_i\}_i^N$

e.g, S2 images

e.g., Land Cover



Geospatial Specialists design:

- Which input space X (e.g., S2 satellite)
- Which label space Y (e.g., 16 Land Covers)
- Which distribution P (e.g., location and time)

Reality → Distribution: Generative Perspective

Generative Perspective
(experimental data)

$$P(X, Y) = P(Y)P(X|Y)$$

1. **first** choosing desired **labels/areas of interest** $P(Y)$ e.g., according to a (stratified) strategy
2. **then** generating (i.e., acquiring, downloading) the (image) **data** $P(X|Y)$

e.g., Experimental Flight with UAV



Reality → Distribution: Discriminative Perspective

Discriminative Perspective
(non-experimental data)

$$P(X, Y) = P(X)P(Y|X)$$

1. **first** collecting remote sensing **images** $P(X)$
2. **then** annotating the data through manual **annotation** $P(Y|X)$

e.g., download and annotate of EO data



Data as Samples from a Distribution

Data Distribution → **Data Samples**

machine learning terminology: we "s ample" data

Remote sensing terminology: we " collect " data

supervised case:

$P(X, Y)$

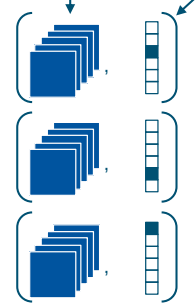
Random Variable X represents input data (e.g., S2 bands)

Random Variable Y represents output data e.g., Land Cover classes or Regression variables

$$\mathcal{D} = \{x_i, y_i\}_i^N$$

e.g, S2 images

e.g., Land Cover



Distribution → Dataset: I.I.D Assumption

Machine Learning Models assume most often that data is sampled **I** dentically and **I** ndependently **D** istributed **(I.I.D)** from the underlying distribution

development time

collected once:

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^{N} \sim P(X, Y)$$

2. random sampling after data collection

then partitioned randomly:

$$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \sim P(X, Y)$$

Distribution → Dataset: In-Distribution Generalization

development time

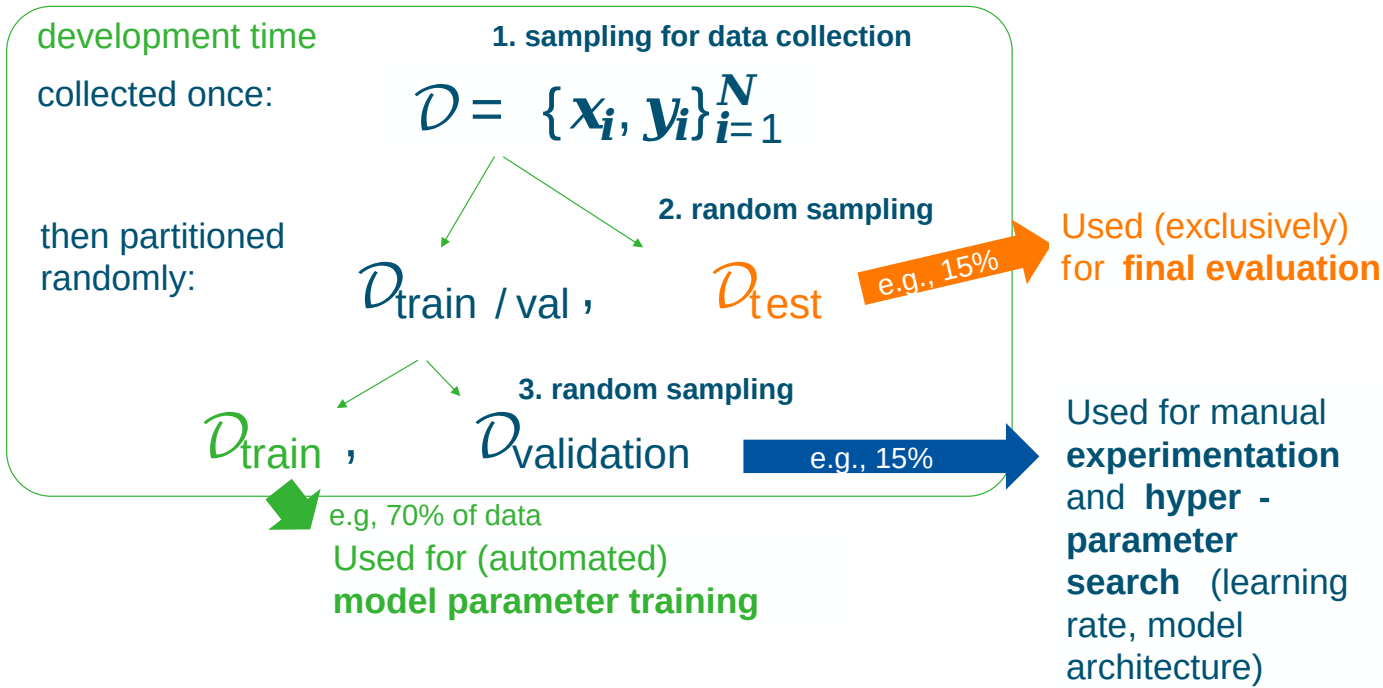
$$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \sim P(X, Y)$$

Test the model's generalization **in-distribution**

trained model

Random Sampling is sufficient for model comparisons

In Practice: Training, Validation, Test Data



Distribution → Dataset: Random Sampling

Collecting (i.e., sampling) data once ensures that data is sampled **I**dentically and **I**ndependently **D**istributed **(I.I.D)** but only measures **in-distribution generalization**

development time

collected once:

$$\mathcal{D} = \{X_i, y_i\}_{i=1}^{N} \sim p(X, Y)$$

random sampling

then partitioned randomly:

$$\mathcal{D}_{\text{train}} \leftrightarrow \mathcal{D}_{\text{test}} \sim p(X, Y)$$

in-distribution generalization

Real-World Models need to generalize Out-of-Distribution

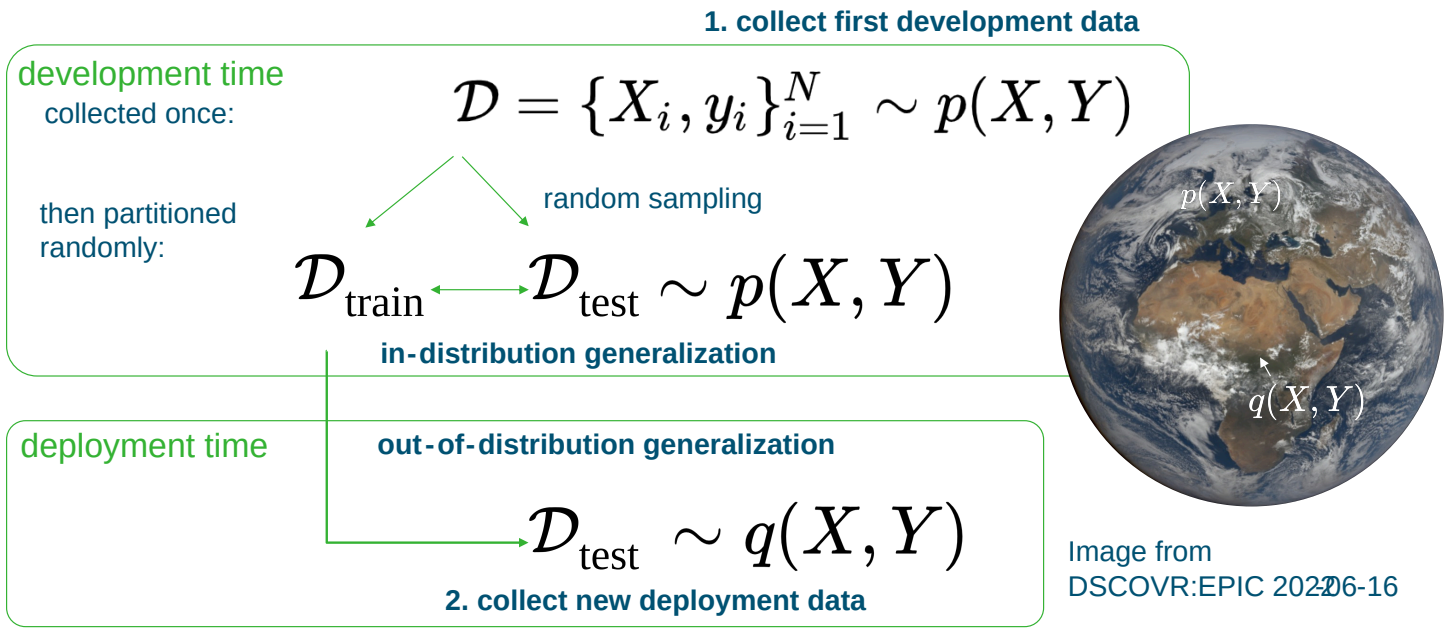


Image from DSCOVR:EPIC 2022-06-16

Chanes between Development and Deployment

Covariate Distribution Shifts $P(X) \neq Q(X)$

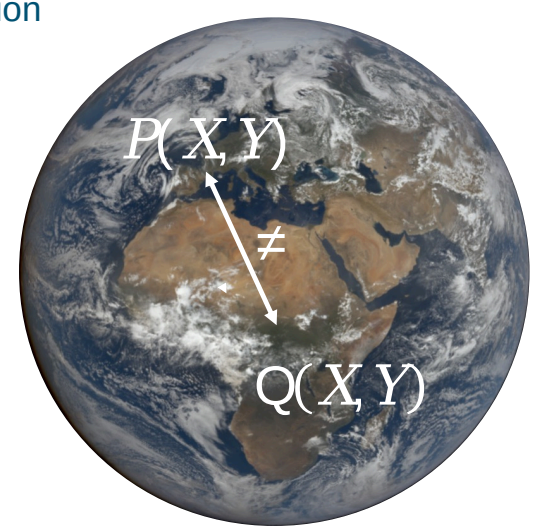
- **Different Location** : deploy model on different location
- **Different Time:** Summer/Winter
- **Different Sensors:** Sentinel - 2A vs 2B, Atmospheric correction algorithms

Concept Shifts $P(Y|X) \neq Q(Y|X)$

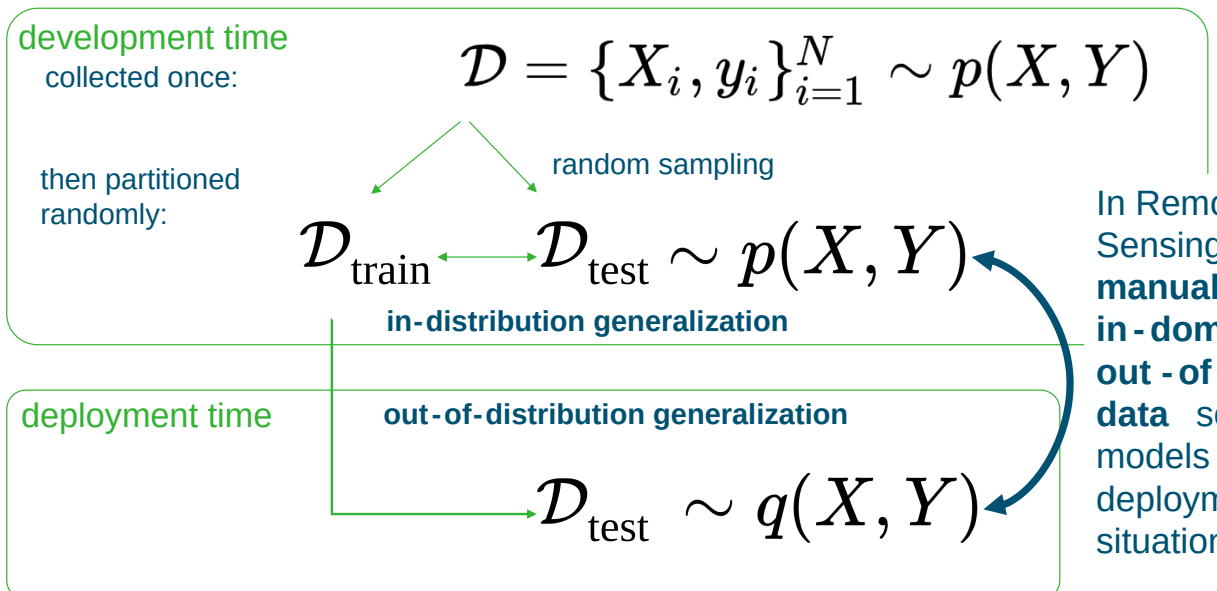
- **Different Intra - Class Variability** : Mixed Marine Debris, Different Forest Compositions
- **Different Data Annotators**

Label Shifts $P(Y) \neq Q(Y)$

- **Different Class Frequency:** One Category not present in test location.



Geospatial Models Need to Generalize Out-of-Distribution

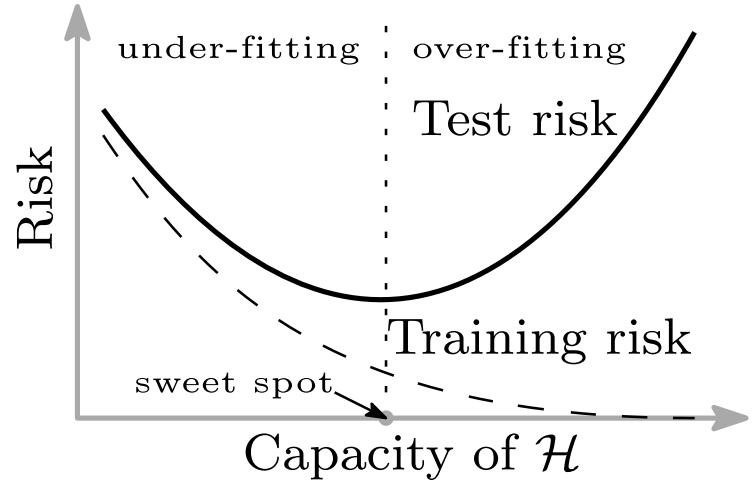


In Remote Sensing we often **manually align in-domain and out-of-domain data** so that models work in deployment situations

Takeaway: Data

- We model our world as distributions from which we sample (i.e., collect) datasets
- We use training data to train model weights, validation data for experimentation, and test data for final evaluation
- Random sampling of train, val, test data measures in-distribution accuracy and is sufficient for model comparisons
- Real-world deployments must consider distribution shifts $P(X \mid Y) \neq Q(X \mid Y)$ and assessing out-of-distribution accuracy on separately collected test data

Classical Generalization Theory



The Hypothesis Space $\mathcal{H}$

A **hypothesis** is a prediction function:

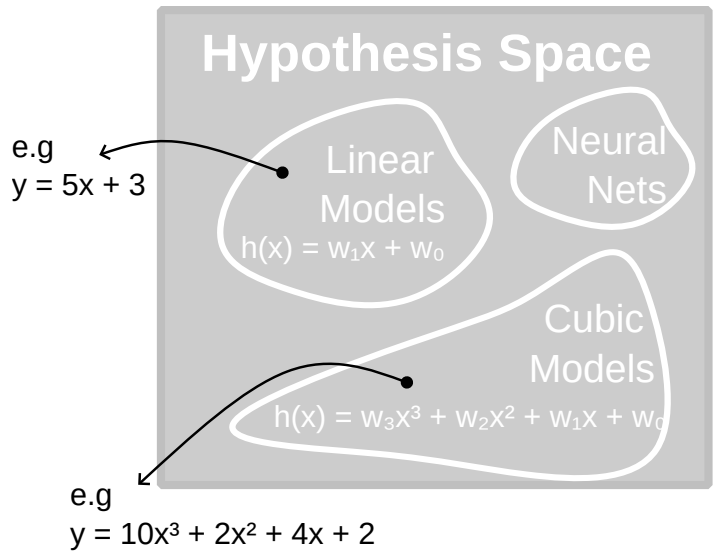
$$h : \mathcal{X} \to \mathcal{Y}$$

The **hypothesis space** is the set of functions the model can represent:

$$\mathcal{H} = \{h_\theta : \theta \in \Theta\}$$

- Θ is the **parameter space**
- $\theta \in \Theta$ is one parameter choice
- $h_\theta \in \mathcal{H}$ is the corresponding prediction function
- Training selects a hypothesis from $\mathcal{H}$

$$\boxed{\text{parameters } \theta \quad \longrightarrow \quad \text{hypothesis } h_\theta}$$



Example $\mathcal{H}_{\text{linear}}$

A **hypothesis** is a prediction function:

$$h : \mathcal{X} \to \mathcal{Y}$$

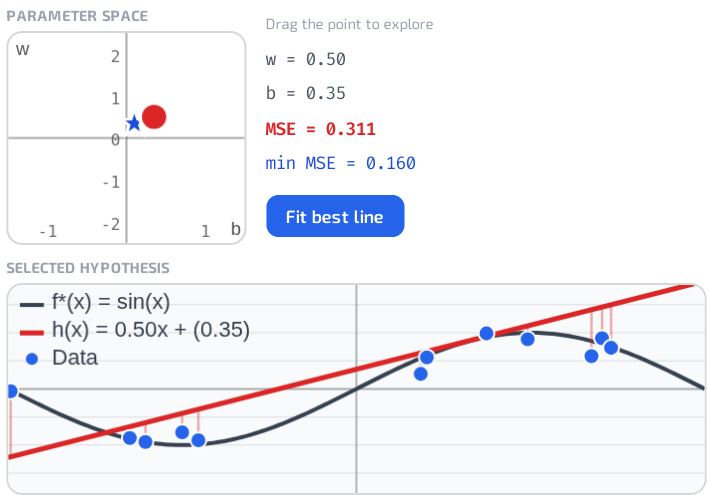
The **hypothesis space** is the set of functions the model can represent:

$$\mathcal{H} = \{h_\theta : \theta \in \Theta\}$$

- Θ is the **parameter space**
- $\theta \in \Theta$ is one parameter choice
- $h_\theta \in \mathcal{H}$ is the corresponding prediction function
- Training selects a hypothesis from $\mathcal{H}$

$$\boxed{\text{parameters } \theta \quad \longrightarrow \quad \text{hypothesis } h_\theta}$$

Example: $\mathcal{H}_{\text{linear}} = \{h_{w,b}(x) = wx + b : (w, b) \in \mathbb{R}^2\}$



Parameter Space and Hypothesis Space

Two parameter sets can map to the same Hypothesis

The **parameter space** contains the numerical values optimized during training:

$$\theta \in \Theta$$

The **hypothesis space** contains the prediction functions the model can represent:

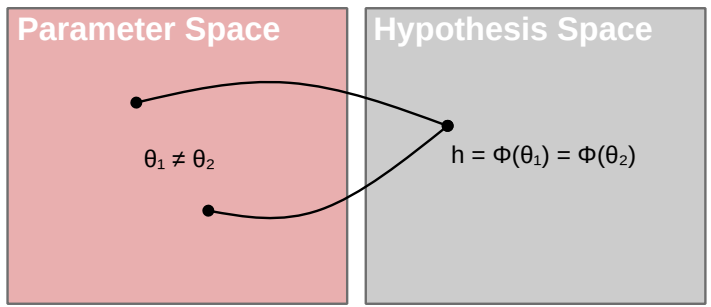
$$h_\theta \in \mathcal{H}$$

A parameterization maps parameter values to hypotheses:

$$\Phi : \Theta \to \mathcal{H}, \qquad \theta \mapsto h_\theta$$

This mapping does not have to be one-to-one:

$$\theta_1 \neq \theta_2 \qquad \text{but} \qquad \Phi(\theta_1) = \Phi(\theta_2)$$

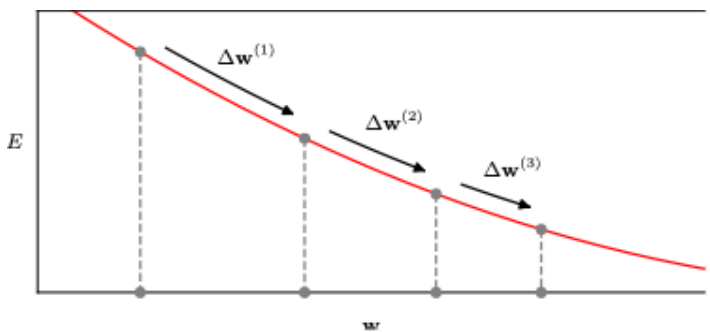


Training moves through parameter space, but predictions are determined by the corresponding hypothesis.

From Optimization to a Learned Hypothesis

Navigate parameter space

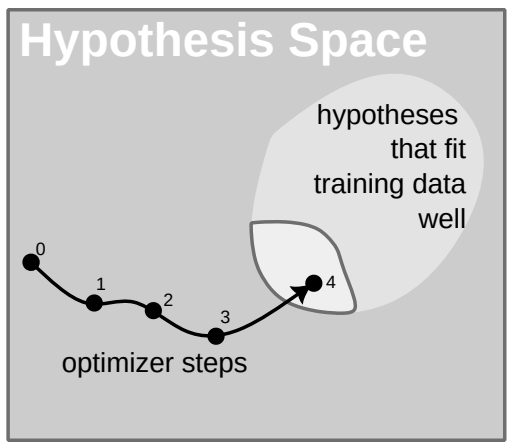
Gradient descent updates the parameters in a direction that aims to reduce the empirical risk:



Induce a path through $\mathcal{H}$

Every parameter value represents a prediction function:

$$h_{\theta^{(0)}} \to h_{\theta^{(1)}} \to \cdots \to h_{\theta^{(T)}}, \text{ so that } \hat{h}_D = h_{\theta^{(T)}} \in \mathcal{H}$$

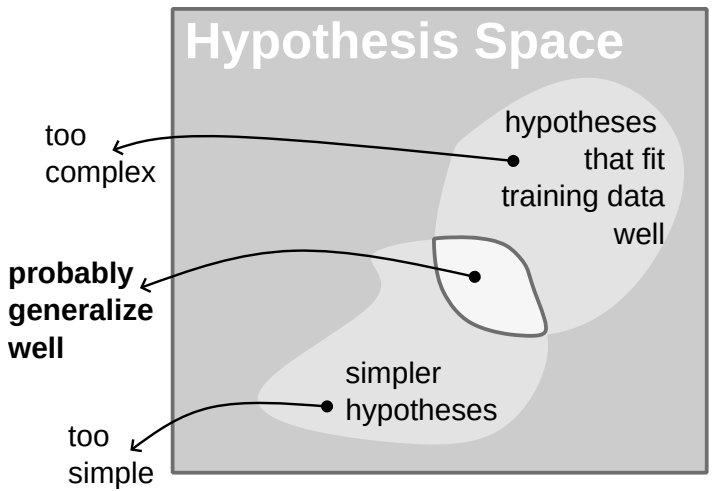


Occam's Razor Idea

Simpler functions should generalize better.

Willian of Ockham (c. 1287 – 9/10 April 1347)





Bias and Variance

$$\text{test risk} = \underbrace{\text{training risk}}_{\text{related to bias}} + \underbrace{(\text{test risk} - \text{training risk})}_{\text{related to variance}}$$

Bias

How systematically wrong is the model?

$$\text{Bias}(x) = \mathbb{E}_D[\hat{f}_D(x)] - f^*(x)$$

- Difference between the average learned model and the true function
- Often caused by a restrictive hypothesis space
- High bias leads to underfitting
- Both training and test error are typically high

Variance

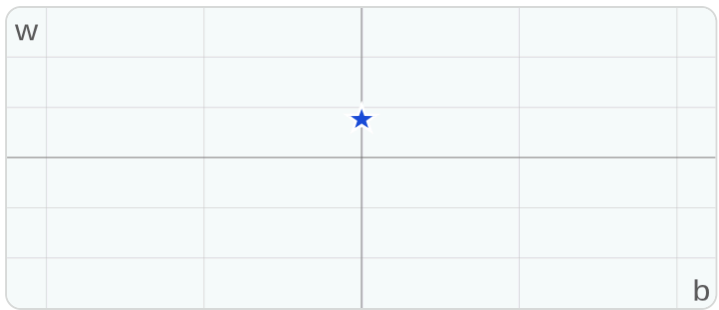
How much does the model depend on the dataset?

$$\text{Var}(x) = \mathbb{E}_D\left[\left(\hat{f}_D(x) - \mathbb{E}_D[\hat{f}_D(x)]\right)^2\right]$$

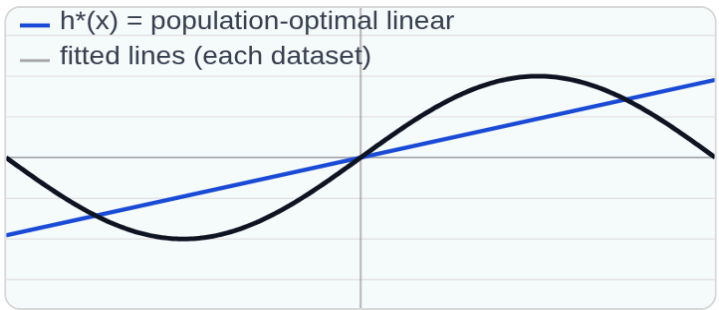
- Measures variation across different training datasets
- Often increases with model flexibility
- High variance leads to overfitting
- Test error can be much higher than training error

Example: Linear Regression

PARAMETER SPACE



X-Y PLANE SHOWING HYPOTHESES FUNCTIONS



Draw another dataset | Reset | N = 12 | M = 0

Task 1: Discuss the bias and variance of Linear Regression models.

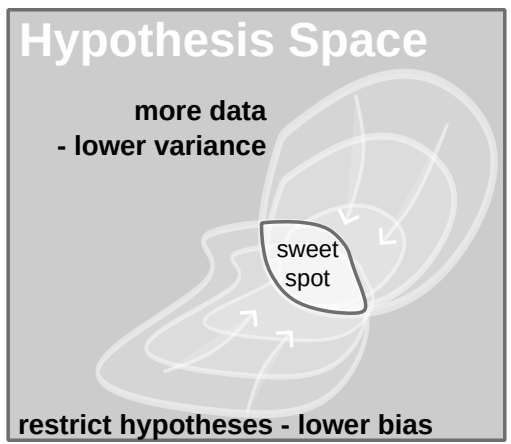
Task 2: Increase and Decrease the dataset size N, how does the variance change?

Controlling Bias and Variance

Strategy 1: Introduce Inductive Bias

Use domain knowledge to **restrict or prioritize** hypotheses:

- choose informative features;
- select an appropriate architecture;
- encode known invariances;
- regularize model complexity.



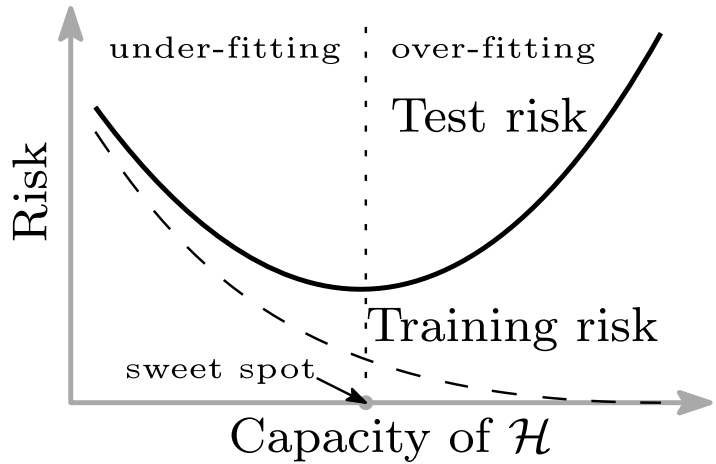
Strategy 2: Reduce Variance by more Data

Use additional information to rule out hypotheses that fit only by chance:

- collect more labelled data;
- apply valid data augmentation;
- exploit unlabelled data through pretraining.

Bias Variance Trade-Off

- **Under-fitting of simpler models** have high bias: they do not fit the training data well. Their predictions are fairly consistent across different training sets, so they have low variance.
- **Over-fitting of complex models** have lower bias: they fit the training data closely, sometimes perfectly. Their predictions depend more strongly on the training sample, so they have high variance and may perform poorly on new data.
- The classical **sweet spot** balances bias and variance to minimize test risk.



Vapnik-Chervonenkis (VC) Dimension

How to measure the Capacity of H

A set of n points is **shattered** by $\mathcal{H}$ if classifiers in $\mathcal{H}$ can realize all 2^n possible binary labelings. The **VC dimension** is the largest number of points that $\mathcal{H}$ can shatter:

$$\mathrm{VCdim}(\mathcal{H}) = \max\{n : \text{some } n \text{ points are shattered by } \mathcal{H}\}.$$

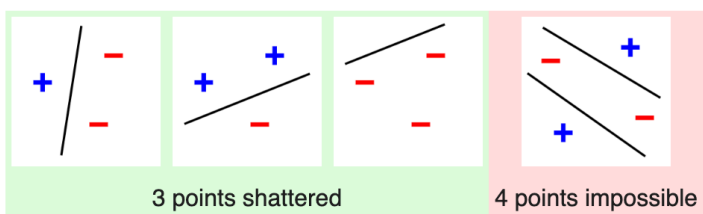
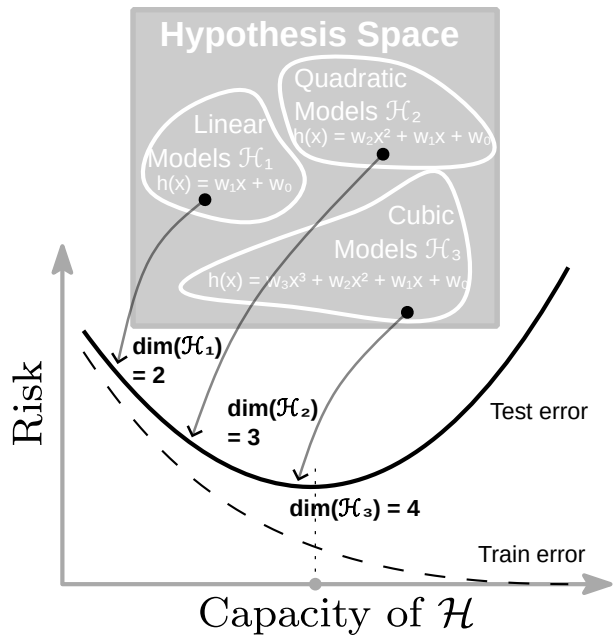


Image: Wikipedia, CC BY-SA 3.0



Worst-Case Generalization Gap

For a bounded loss, ignoring constants and logarithmic terms:

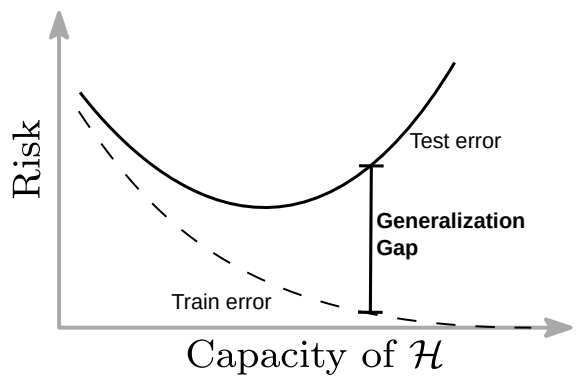
$$\left|R(h) - \widehat{R}_D(h)\right| \lesssim \sqrt{\frac{d}{n}}$$

with VC dimension d and sample size n

- A larger VC dimension gives a weaker guarantee for the same sample size.
- More training data reduces the worst-case gap approximately as $1/\sqrt{n}$.
- For a target gap ε, the required sample size scales approximately as

$$n \propto \frac{d}{\varepsilon^2}.$$

- The guarantee is useful only when $d \ll n$. When $d \approx n$, the bound becomes uninformative.



VC bounds are distribution-free and robust, but often pessimistic.

Example Polynomial Regression:

Increasing Polynomial Degree D: ↓ Bias ↑ Variance

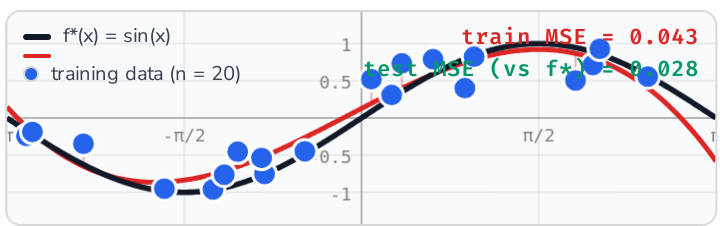
NUMBER OF SAMPLES **n = 20**

4 40

POLYNOMIAL DEGREE **d = 3**

1 20

Draw new dataset



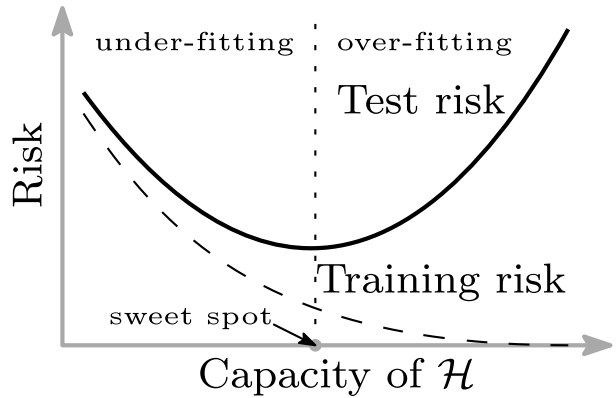
TRAIN / TEST ERROR VS. HYPOTHESIS-SPACE CAPACITY



degree d (hypothesis-space capacity) →

Takeaways Classical Generalization Theory

- Training minimizes **training risk**, while the goal is low **testing risk**.
- Simpler models are more stable but may underfit because of high bias.
- More flexible models reduce bias but may overfit because of high variance.
- When overfitting, reduce model complexity, increase regularization, or use more data.
- When underfitting, increase model capacity or reduce regularization.
- Use validation data to find the model complexity that best balances bias and variance.



Generalization beyond the Interpolation Threshold

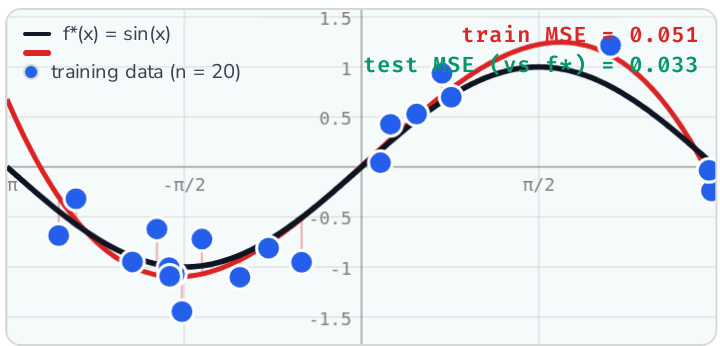
Photo by Ludvig Hedenborg, Pexels License

Beyond the Threshold: Double Descent in Polynomial Regression

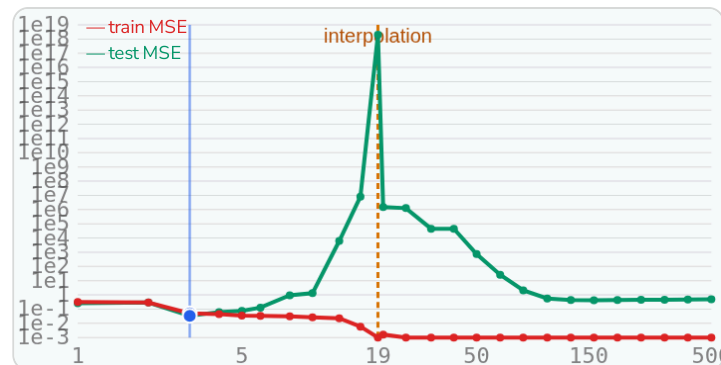
POLYNOMIAL DEGREE (LOG SCALE) **d = 3**

1 500

Draw new dataset



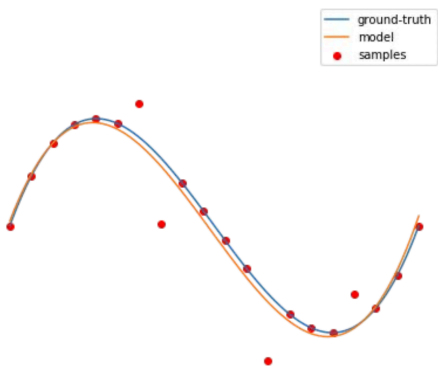
TRAIN / TEST ERROR VS. HYPOTHESIS-SPACE CAPACITY (LOG–LOG)



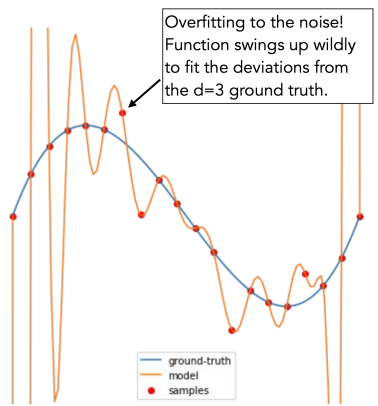
degree d (hypothesis-space capacity, log scale) →

Polynomial Regression in Interpolation Regime

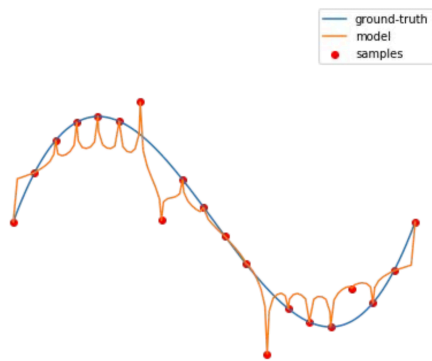
Underfitting — Degree 3



Overfitting — Degree 20



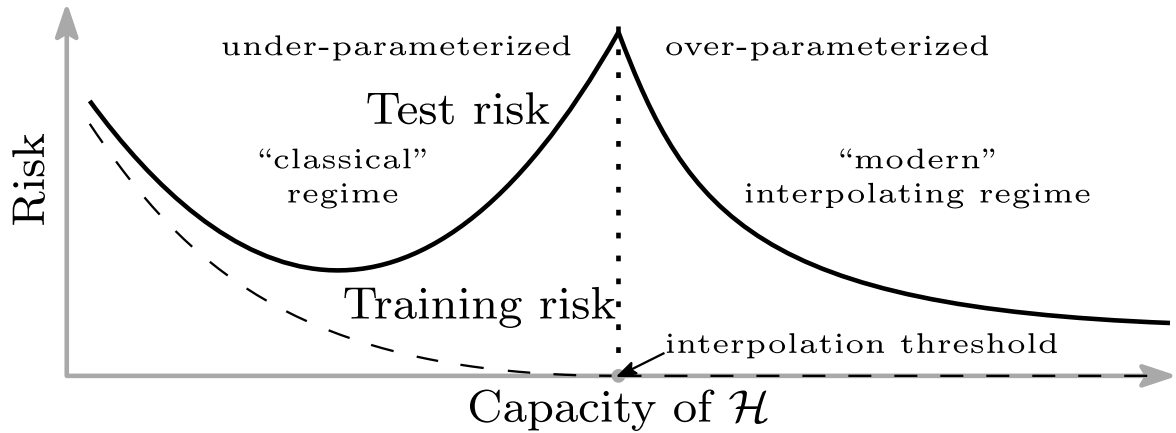
Interpolating — Degree 1000



Credit: Philip Isola, MIT 6.7960 Deep Learning (Fall 2024), Lecture 6: NN Generalization, MIT OpenCourseWare

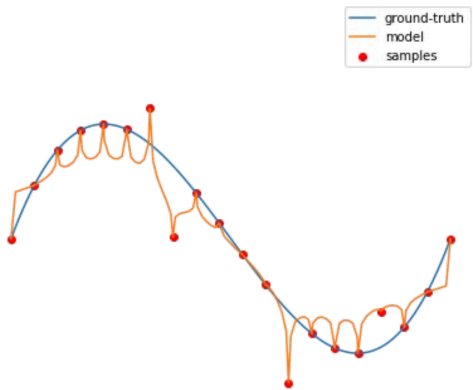
Double Descent Phenomenon

Overparamterized Models Improve Again after Overfitting



Credit: Belkin et al., 2019. Reconciling modern machine learning practice and the bias-variance trade-off

The Simple + Spiky Hypothesis



$$\text{learned model} = \underbrace{\text{“simple”}}_{\text{predictive component}} + \underbrace{\text{“spiky”}}_{\text{overfitting / memorization component}}$$

Belkin, Rakhlin & Tsybakov, 2018. Does data interpolation contradict statistical optimality?

Credit: Philip Isola, MIT 6.7960 Deep Learning (Fall 2024), Lecture 6: NN Generalization, MIT OpenCourseWare

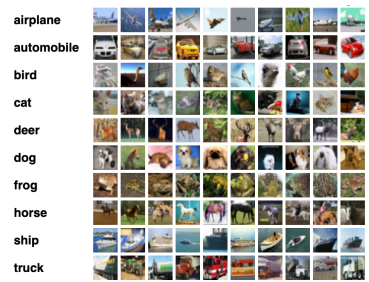
Neural Networks Interpolate and Generalize

The same hypothesis class can memorize noise and learn structure

The experiment

Inception on CIFAR-10

Training labels	Train accuracy	Test accuracy
True labels	100%	85.75%
Random labels	100%	9.78%



The VC bound becomes vacuous and does not explain Deep Networks' generalization

- **No contradiction:** The VC bound remains valid, but it is too pessimistic to distinguish the two learned solutions.

Zhang et al. (2017). *Understanding Deep Learning Requires Rethinking Generalization*. ICLR. arXiv:1611.03530

How to measure the Model Complexity now??

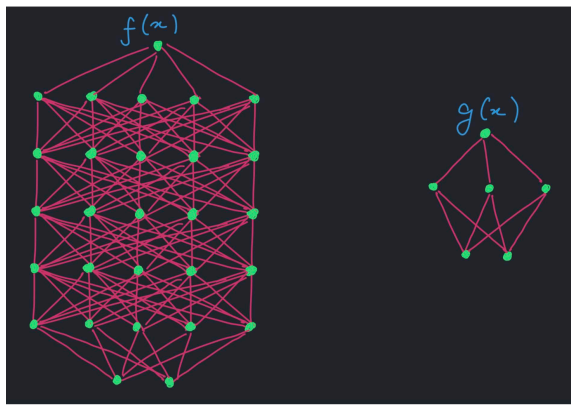
By the number of distinct functions the model can represent? (VC-dimension)

→ **No.** Not helpful — the generalization bound is too pessimistic. Following classical theory, neural networks should not generalize.

Slide from Philip Isola, MIT 6.7960 Deep Learning (Fall 2024), Lecture 6: NN Generalization, MIT OpenCourseWare

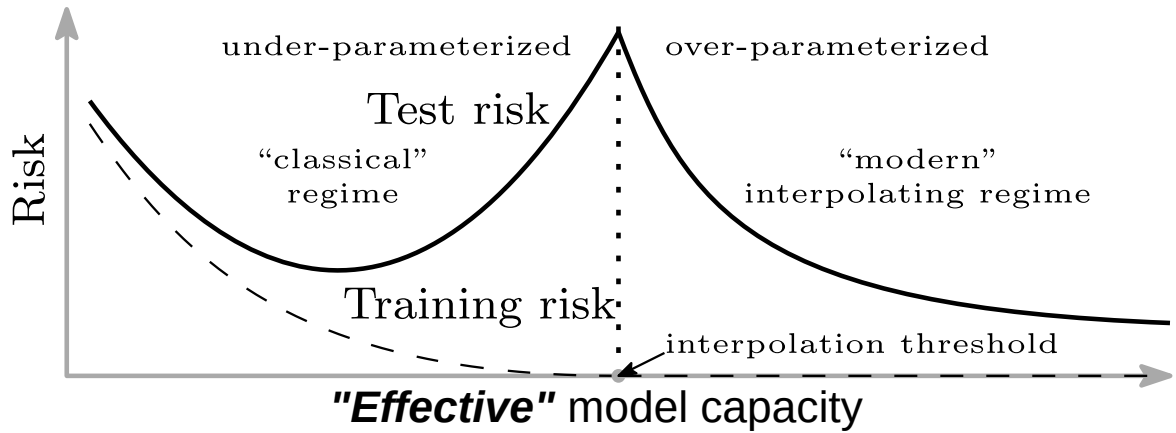
Why parameter count alone doesn't determine generalization for deep learning is still an open question.

By the number of parameters in a neural network?



Consider $h(x) = 10^{-100} f(x) + (1 - 10^{-100}) g(x)$. How many parameters does it have? Does it matter? **Not really.**

"Effective" Model Capacity



"Capacity" is intentionally abstract. Parameter count may indicate when interpolation becomes possible, but it does not determine the complexity of the learned solution or its test performance.

Recap So Far

WHAT WE OBSERVED

Deep networks generalize.
They make accurate predictions on inputs that were not part of the training data.

Deep networks also interpolate.
The same architectures can perfectly fit true labels and random labels.

Training fit alone cannot explain generalization.
Zero training error does not distinguish learning useful structure from memorizing the sample.

WHAT THIS IMPLIES

Generalization requires inductive bias.
Training must favor some interpolating solutions over many others that fit the data equally well.

Worst-case capacity is insufficient.
Parameter count and VC dimension describe the entire hypothesis space, but their bounds become vacuous for highly expressive neural networks.

The selected solution matters.
Architecture, optimization, regularization, and data properties all influence which solution training finds.

Next question: Which inductive biases make interpolating neural networks generalize?

Slide from Philip Isola, MIT 6.7960 Deep Learning (Fall 2024), Lecture 6: NN Generalization, MIT OpenCourseWare

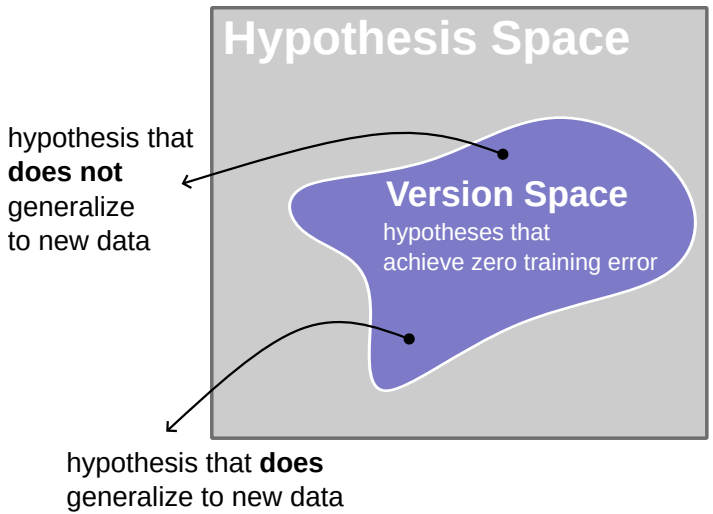
Version Space

The **version space** contains all hypotheses that fit the training data perfectly:

$$\mathcal{V}_{\mathcal{H}}(D) = \left\{ h \in \mathcal{H} : \widehat{R}_D(h) = 0 \right\}.$$

- In overparameterized models, many hypotheses belong to this set.
- Some generalize to unseen data, while others merely memorize.
- Training error cannot distinguish between them.

Deep models do generalize. Therefore, the training system must favor generalizing hypotheses within the version space.



Inductive Biases Select Interpolating Solutions

Sources of inductive bias

- **Architecture:** locality, parameter sharing, recurrence, and equivariance
- **Parameterization:** makes some functions easier to represent and reach
- **Initialization:** influences the path taken through parameter space
- **Optimization and loss:** SGD, batch size, learning rate, and loss function
- **Explicit regularization:** weight decay, dropout, and early stopping
- **Data and augmentation:** encode structure, invariances, and prior knowledge

What does “simpler” mean?

Not necessarily fewer parameters.

Depending on the setting, training may favor solutions with:

- low norm or large margin
- smoothness or invariance
- compressible representations

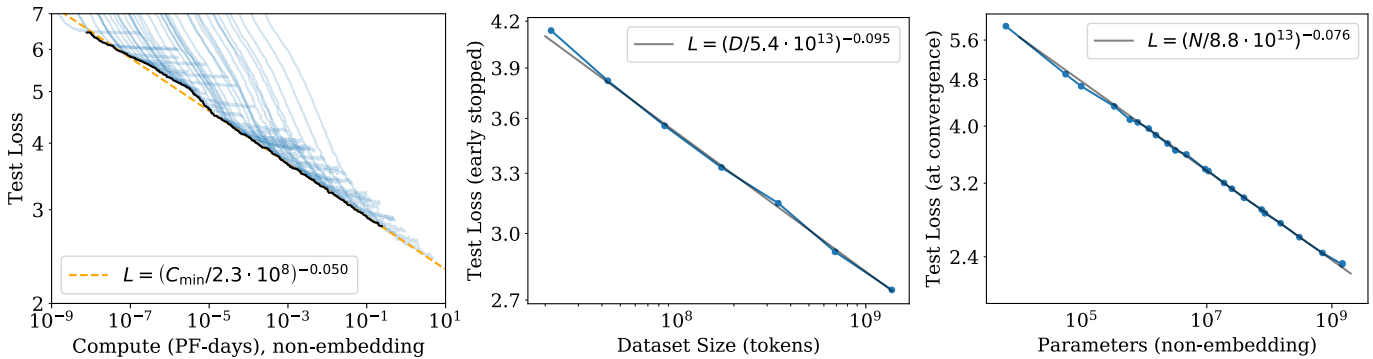
There is no single universal measure of the effective complexity of a deep network.

For concrete proposed theories, check Phillip Isola. *Lecture 6: Generalization Theory*. MIT 6.7960 Deep Learning, Fall 2024.

The Modern AI Acceleration

Scaling Laws for Transformers

Held-out loss follows power laws of compute, data, and model size



Compute

$$L(C_{\text{min}}) = \left(\frac{C_{\text{min}}}{2.3 \times 10^8}\right)^{-0.050}$$

Training data

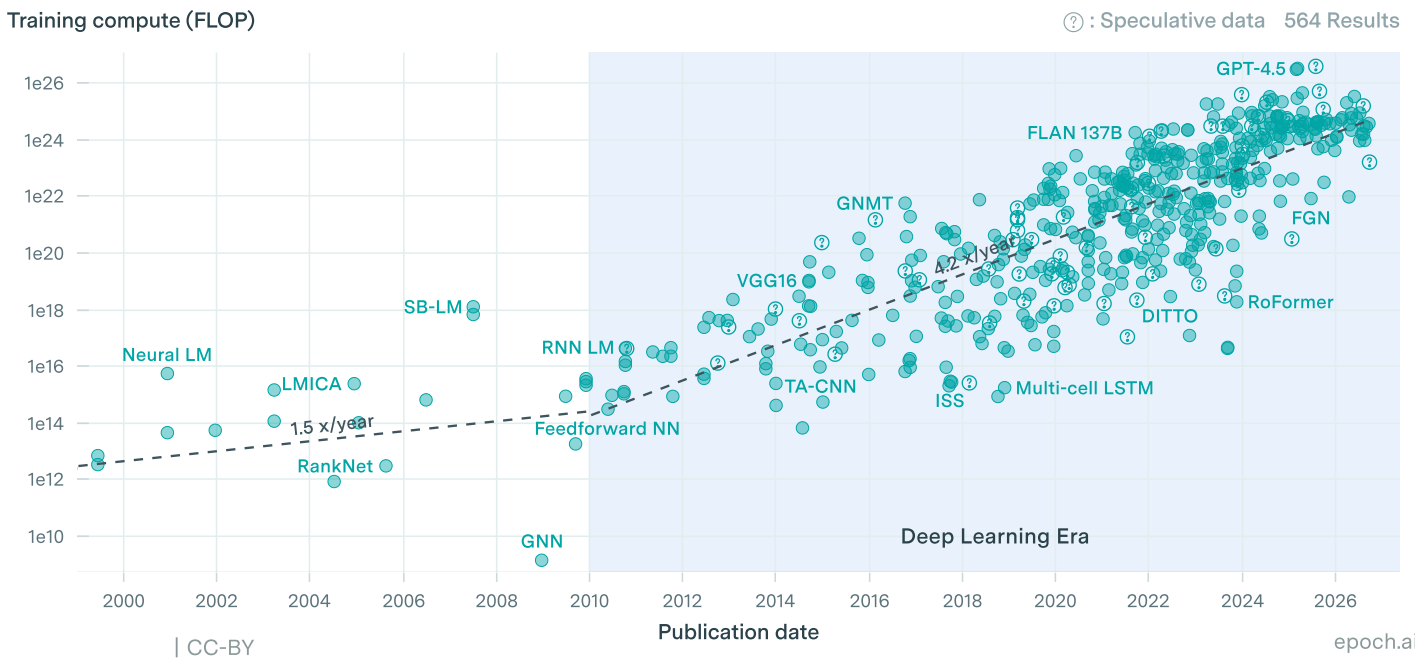
$$L(D) = \left(\frac{D}{5.4 \times 10^{13}}\right)^{-0.095}$$

Model size

$$L(N) = \left(\frac{N}{8.8 \times 10^{13}}\right)^{-0.076}$$

Kaplan et al. (2020). *Scaling Laws for Neural Language Models.*

Modern AI Acceleration



Epoch AI Database provided with CC-BY License

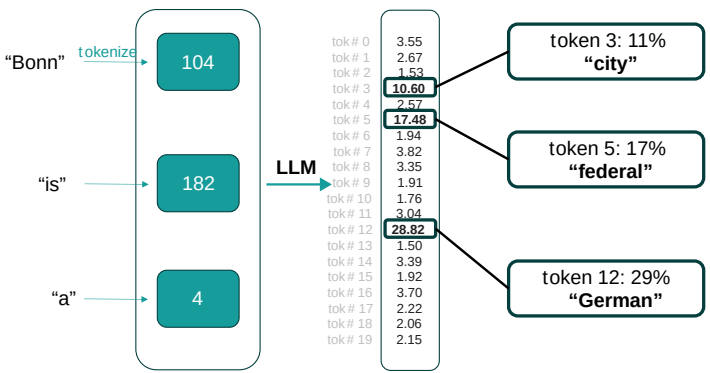
Larger Datasets and More Parameters

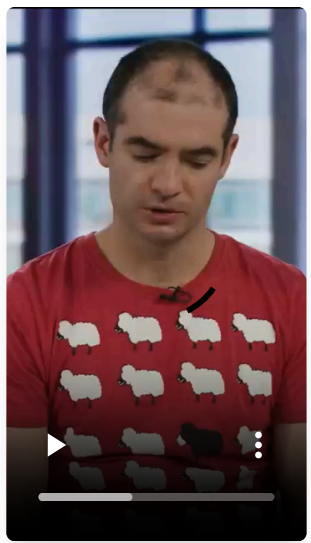


Epoch AI Database provided with CC-BY License

LLMs and Next Token Prediction

Recap - LLMs are Transformers trained with ordinary cross entropy to predict the next token.



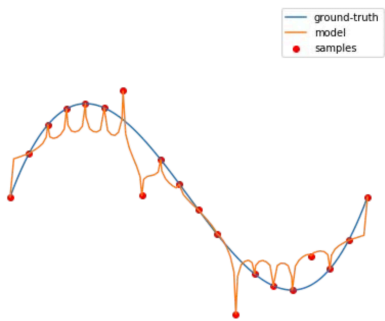


Copyright NVIDIA: NVIDIA Fireside Chat with Ilya Sutskever

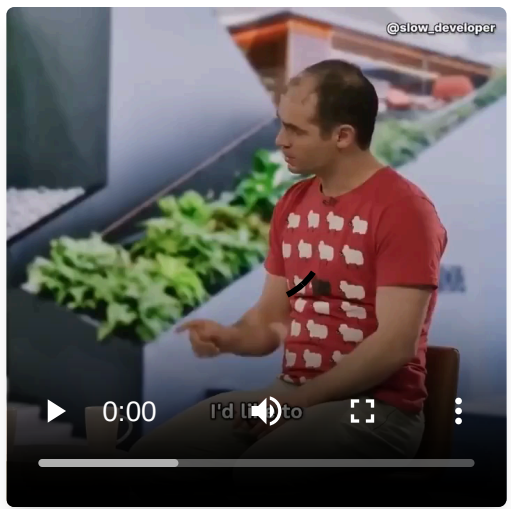
… and the murderer is ___!

Prediction requires more than memorization

Modern AI models **approximate** large corpora of training data while also **generalizing** by combining learned patterns into simple, plausible solutions for new inputs.



Outlook: Lecture 7 will focus on self-supervised training objectives for modern AI models.



Copyright NVIDIA: NVIDIA Fireside Chat with Ilya Sutskever