# Lecture 11: Rethinking Generalization: Modern Phenomena in Overparameterized Models

Readings: Bach (Ch. 12), ISL (Ch. 10), ESL (Ch. 11); code

Topics: overparameterization, implicit regularization, minimum-norm interpolation, gradient descent bias, double descent, benign overfitting, scaling laws, appendix: Wishart distribution

## 1 Motivation: Successes of Overparameterized Models in Deep Learning

### 1.1 The Classical View vs Modern Reality

**Classical Machine Learning Wisdom:**

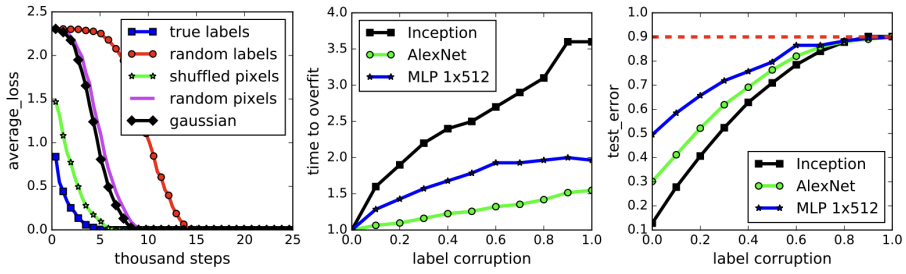
- A model is **overparameterized** if the number of learnable parameters $p > N$, in which case there are infinitely many interpolating solutions.
- Classical wisdom: generalization is governed by the **bias–variance tradeoff**, and overparameterized models should overfit severely.
- More parameters $\Rightarrow$ worse generalization.

**Modern Deep Learning Reality:**

- State-of-the-art deep learning models have far more parameters than training examples.
- They achieve near-zero training error (*perfect interpolation*).
- Yet they generalize extremely well on unseen data.

### 1.2 Understanding Deep Learning (Still) Requires Rethinking Generalization

**Zhang et al. (2017, 2021)**[1]: Deep neural nets can memorize random labels (noise) *and* still generalize on real data.




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

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

[1] `https://dl.acm.org/doi/10.1145/3446776`.

Why can overparameterized models generalize so well, despite classical theory predicting overfitting?

### 1.3 Why Overparameterization Can Help?

- **Implicit Regularization:** Optimization (e.g., gradient descent) naturally biases toward "simple" solutions.
- **Double Descent:** Test error can show multiple descents (e.g., as a function of model size, data size, or training time). How do we reconcile classical ML theory with the apparent success of overparameterized models?
- **Benign Overfitting:** Provides a theoretical explanation for double descent — even interpolating models can generalize due to implicit biases.
- **Scaling Laws:** Performance improves predictably with more parameters, data, and compute.
- **Infinite-Width Limit:** Gradient descent can exhibit global convergence, despite nonconvexity (via the Neural Tangent Kernel perspective).

Today we discuss the **successes** of overparameterized models, focusing on the lens of *implicit regularization* and *double descent*. In the next lecture, we will examine their **limitations and reliability issues**—both for overparameterized and other modern models.

## 2 Implicit Regularization

### 2.1 Implicit Regularization by Optimization

- Classical generalization relies on **explicit regularization**:

$$\hat{R}_N(\theta) + \lambda\|\theta\|^2 \quad \text{(ridge)}, \qquad \hat{R}_N(\theta) + \lambda\|\theta\|_1 \quad \text{(lasso)}.$$

- **Empirical observation:** In overparameterized settings, even without explicit penalties, GD biases toward "simple" solutions.
- This **implicit bias** can be viewed as "algorithmic regularization", which depends on:
  - initialization (e.g., $\theta^{(0)} = 0$),
  - learning rate $\eta$,
  - algorithm variant (GD, SGD, Adam, etc.),
  - any manipulation like dropout or batch normalization.
- The combination of initialization and the *architecture* of the predictor family further constrain model capacity implicitly.

Classical guarantees rely on *explicit* penalties (ridge, lasso), whereas GD provides an *implicit* bias even without such terms. How can we understand the implicit bias of GD?

### 2.2 Linear Models Trained with Gradient Descent

Given the training data $D_N = \{(x_i, y_i)\}_{i=1,\dots,N}$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, the linear model is

$$f_\theta(x) = \theta^T x = \hat{y}.$$

Here, $p = d$ is the number of learnable parameters.

- **Loss function (squared loss):**

$$L(\hat{y}, y) = (\hat{y} - y)^2.$$

- **Expected (population) risk**:

$$R(\theta) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\Big[L(x^\top\theta, y)\Big].$$

- **Empirical (training) risk:**

$$\hat{R}(\theta) = \hat{R}_N(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(x_i^\top\theta, y_i).$$

- **Excess risk:**

$$R(\theta) - R^*, \qquad \text{where } R^* = \min_\theta R(\theta).$$

### 2.3 Setup and Assumptions

Let us focus on ordinary least squares (OLS) regression[2]:

- **Design matrix:** $\mathbf{X} \in \mathbb{R}^{N\times d}$, with rows $x_i^\top$.
- **Response vector:** $\mathbf{y} \in \mathbb{R}^N$, with entries $y_i$.
- **Empirical risk (least squares):**

$$\hat{R}(\theta) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\theta\|_2^2. \tag{1}$$

- **Overparameterized regime:** $d > N$, so the system is underdetermined. There may be infinitely many solutions (interpolators) $\theta$ with $\mathbf{X}\theta = \mathbf{y}$.
- **Rank assumption:** $\operatorname{rank}(\mathbf{X}) = N$, so the Gram matrix $K = \mathbf{X}\mathbf{X}^\top$ is invertible.
- **Initialization & step size:** $\theta^{(0)} = 0$, and $0 < \eta < 1/\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of $\frac{1}{N}\mathbf{X}\mathbf{X}^\top$.

### 2.4 Gradient Descent Update and Main Theorem

**GD update:**

$$\theta^{(t)} = \theta^{(t-1)} - \eta\nabla\hat{R}_N(\theta^{(t-1)}) = \theta^{(t-1)} - \frac{\eta}{N}\mathbf{X}^\top(\mathbf{X}\theta^{(t-1)} - \mathbf{y}). \tag{2}$$

In Exercise 10.4, we have shown the following result.

**Theorem 2.1** (Implicit $\ell_2$-Regularization of GD)**.** *Under the earlier assumptions, with* $\theta^{(0)} = 0$ *and* $0 < \eta < 1/\lambda_{\max}$*,*

$$\theta^{(\infty)} = \lim_{t\to\infty}\theta^{(t)} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y}. \tag{3}$$

[2]We can also study the implicit bias of GD in classification and nonconvex settings but the analysis is more complicated (see Ch. 12.1 in Bach).

*Moreover,* $\theta^{(\infty)}$ *is the* minimum $\ell_2$-norm least-squares solution*, i.e.*

$$\theta^{(\infty)} = \arg\min_{\theta \in \mathbb{R}^d} \left\{ \|\theta\|_2 : \ \theta \in \arg\min_{\vartheta} \|\mathbf{X}\vartheta - \mathbf{y}\|_2 \right\}. \tag{4}$$

- Therefore, $\mathbf{X}\theta^{(t)} \to \mathbf{y}$ as $t \to \infty$. This is a linear convergence result, and the convergence rate depends on the condition number of the Gram matrix.
- No Gaussian assumption is needed for the convergence/limit identity.
- $\mathbf{X}^+ := \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}$ is also called the Moore–Penrose pseudoinverse. This is a right inverse, since $\mathbf{X}\mathbf{X}^+ = I_N$.

### 2.5 GD Trajectories Stay in the Span of the Data

- With $\theta^{(0)} = 0$, each GD update

  $$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{1}{N}\mathbf{X}^\top(\mathbf{X}\theta^{(t)} - \mathbf{y})$$

  adds a vector in $\text{span}(\mathbf{X}^\top)$.
- By induction,

  $$\theta^{(t)} \in \text{span}(\mathbf{X}^\top), \qquad \forall t \geq 0.$$

- Thus we can write

  $$\theta^{(t)} = \mathbf{X}^\top \alpha^{(t)}, \qquad \alpha^{(t)} \in \mathbb{R}^N.$$

- Then

  $$\mathbf{X}\theta^{(t)} = \mathbf{X}\mathbf{X}^\top \alpha^{(t)} = K\alpha^{(t)}, \qquad K = \mathbf{X}\mathbf{X}^\top.$$

- Since $\mathbf{X}\theta^{(t)} \to \mathbf{y}$ and $K$ is invertible, $\alpha^{(t)} \to K^{-1}\mathbf{y}$, and therefore

  $$\theta^{(\infty)} = \mathbf{X}^\top K^{-1}\mathbf{y}.$$

This is an **algorithmic version of the Representer Theorem**: although the parameter space is $\mathbb{R}^d$, GD solutions live in $\text{span}(\mathbf{X}^\top)$, i.e. a space of dimension at most $N$.

### 2.6 Dual Characterization: Minimum-Norm Interpolator

**Primal problem**:

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \qquad \text{s.t.} \qquad \mathbf{X}\theta = \mathbf{y}. \tag{5}$$

**Lagrangian:**

$$\mathcal{L}(\theta, \alpha) = \frac{1}{2}\|\theta\|_2^2 + \alpha^\top(\mathbf{y} - \mathbf{X}\theta), \qquad \alpha \in \mathbb{R}^N.$$

**Dual problem:**

$$\sup_{\alpha \in \mathbb{R}^N} g(\alpha), \qquad g(\alpha) = \inf_{\theta} \mathcal{L}(\theta, \alpha) = \alpha^\top \mathbf{y} - \frac{1}{2}\alpha^\top(\mathbf{X}\mathbf{X}^\top)\alpha.$$

**KKT conditions:**

$$\nabla_\theta \mathcal{L} = 0 \Rightarrow \theta = \mathbf{X}^\top \alpha, \qquad \nabla_\alpha \mathcal{L} = 0 \Rightarrow \mathbf{X}\mathbf{X}^\top \alpha = \mathbf{y}.$$

**Solution:**

$$\hat{\theta} = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y}. \tag{6}$$

This minimum-norm solution coincides with the GD limit $\theta^{(\infty)}$.

### 2.7 The Kernel View: Effective Complexity

The dual variables $\alpha \in \mathbb{R}^N$ live in **data space**, not parameter space. This is the kernel viewpoint: solutions are linear combinations of training points.

Recall from the dual/KKT formulation:

$$\hat{\theta} = \mathbf{X}^\top \alpha^\star, \qquad \alpha^\star = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y}.$$

**Prediction for a new input $x$:**

$$f(x) = x^\top \hat{\theta} = x^\top \mathbf{X}^\top \alpha^\star = \sum_{i=1}^{N} \alpha_i^\star \langle x, x_i \rangle.$$

Although $\hat{\theta} \in \mathbb{R}^d$ with $d \gg N$, the predictor depends only on the $N \times N$ Gram matrix

$$K = \mathbf{X}\mathbf{X}^\top.$$

Thus the effective model complexity is $O(N)$, not $O(d)$.

This is exactly the **representer theorem in action**: solutions live in $\text{span}(\mathbf{X}^\top)$, i.e. as linear combinations of training examples. Overparameterization does not increase effective complexity when optimization implicitly regularizes.

### 2.8 From Linear Models to Neural Networks

- In linear least-squares regression, GD from $\theta^{(0)} = 0$ converges to the *minimum $\ell_2$-norm interpolator*. Thus, GD introduces **implicit regularization** toward low $\ell_2$-norm solutions without explicit penalty.
- This connects directly to the representer theorem and to the kernel view of effective complexity.

**How about neural networks?**

- The parameter space is nonlinear and highly nonconvex.
- Implicit regularization of GD is less well understood in practice.
- But: in the *infinite-width limit*, the GD dynamics of neural networks reduce to kernel regression with the **Neural Tangent Kernel (NTK)**[3].

Thus the linear model analysis provides intuition: wide neural networks behave like kernel machines, with the NTK playing the role of the Gram matrix.

[3] See `https://arxiv.org/abs/1806.07572`.

### 2.9 Infinite-Width Limit and Neural Tangent Kernel

- In linear regression, GD solutions live in $\text{span}(\mathbf{X}^\top)$, and predictions depend only on the Gram matrix $K = \mathbf{X}\mathbf{X}^\top$.
- In wide neural networks, the same idea holds with a different kernel: the **Neural Tangent Kernel (NTK)**

$$K_{\text{NTK}}(x, x') = \nabla_\theta f_\theta(x)^\top \nabla_\theta f_\theta(x')\big|_{\theta=\theta_0}, \tag{7}$$

  where $\theta_0$ is the random initialization.

Wide neural nets thus behave like kernel machines where the kernel is determined by architecture and initialization. We will not analyze this in detail; see Ch. 12.4 in Bach.

How about **early stopping**? Can we understand its implicit regularization effect for simple models? See Exercise 11.1.

We've seen implicit regularization from **optimization dynamics** so far. There are also implicit/explicit regularization effects from **data perturbations**, such as Gaussian noise injection and Mixup; see Exercise 11.2.

### 2.10 Implicit Regularization: What We Know and Don't Know

**Well established for linear models:**

- GD from zero initialization converges to minimum $\ell_2$-norm solution.
- Early stopping provides spectral regularization.
- Kernel view explains effective complexity.

**Neural networks — empirical observations:**

- Networks seem to find "simpler" solutions among many possibilities.
- Different optimizers exhibit different implicit biases.
- Architecture matters enormously for what constitutes "simple".

**Theoretical understanding (limited):**

- Real neural networks are finite-width and perform *feature learning*. The NTK is therefore only an approximation — useful for intuition, but not a full explanation of practical deep learning.
- Multiple competing theories, no consensus.

⚠ Our theoretical understanding of implicit regularization in practical neural networks remains incomplete.

## 3 Double Descent

### 3.1 Two Complementary Perspectives

**Implicit regularization** is a general phenomenon: optimization biases solutions toward simpler predictors, even without explicit penalties. Overfitting is avoided by "algorithmic regularization".

**In the overparameterized regime, new behaviors emerge:**

- **Double Descent:** Test error does not just follow the classical U-curve. Around the interpolation threshold, it can explode and then *decrease again*.
- **Benign Overfitting:** Explains how interpolating models can still generalize — noise is often fit in "harmless" directions.
- **Infinite-Width Limit:** Neural networks trained by GD behave like kernel machines (NTK), where global convergence can be proven despite nonconvexity. Overparameterization can be a blessing for optimization.
- **Scaling Laws:** Performance often improves predictably with larger models, more data, and more compute.

We will study double descent in detail for some simple setting, and can only briefly discuss the other behaviors.

### 3.2 The Bias-Variance Decomposition

We know from Lecture 1 that the expected prediction error for a predictor $\hat{f}$ (with the true function $f^*$) at a test point $x$ can be decomposed into:

$$\mathbb{E}[(y - \hat{f}(x))^2] = \underbrace{(\mathbb{E}[\hat{f}(x)] - f^*(x))^2}_{\text{Bias}^2} + \underbrace{\text{Var}(\hat{f}(x))}_{\text{Variance}} + \sigma^2. \tag{8}$$

**Classical intuition:** fundamental tradeoff between bias and variance $\Rightarrow$ U-shaped test error.

**Classical wisdom:** Complex models have low bias but high variance. Simple models have high bias but low variance. The optimal model balances both.

**Traditional view:**

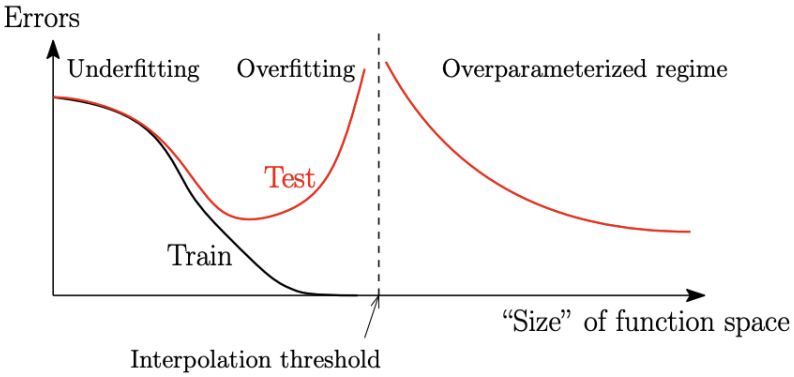
- Left: model too simple $\rightarrow$ underfitting $\rightarrow$ high bias
- Right: model too complex $\rightarrow$ overfitting $\rightarrow$ high variance
- Sweet spot: balanced complexity minimizes total error

### 3.3 Double Descent: Beyond the Classical U-Curve

Number of parameters $p$ ("size"/"capacity" of function space) vs. sample size $N$

**Three regimes:**

1. **Underparameterized** ($p < N$): classical bias-variance tradeoff
2. **Interpolation threshold** ($p \approx N$): test error explodes
3. **Overparameterized** ($p > N$): test error decreases again for $p \gg N$




See a short demo of empirical evidence in Ch. 12.2.2 in Bach.

## 3.4 Empirical Evidence (Belkin et al., 2019)

The double-descent risk curve manifests with not just neural networks, but also other models such as random forests[4].

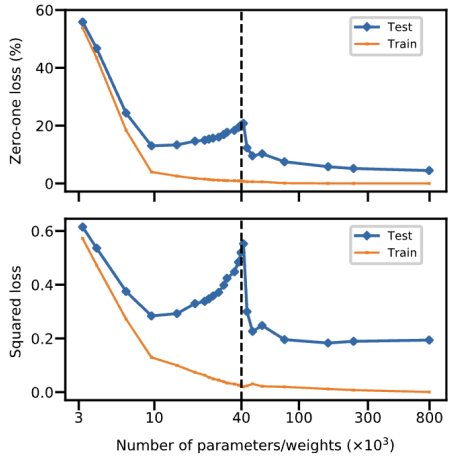



**Fig. 3.** Double-descent risk curve for a fully connected neural network on MNIST. Shown are training and test risks of a network with a single layer of $H$ hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d+1) \cdot H + (H+1) \cdot K$. The interpolation threshold (black dashed line) is observed at $n \cdot K$.

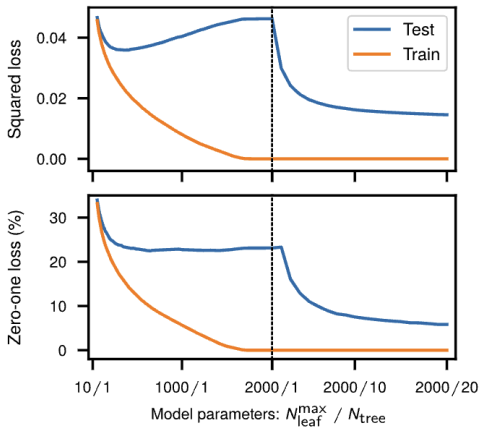



**Fig. 4.** Double-descent risk curve for random forests on MNIST. The double-descent risk curve is observed for random forests with increasing model complexity trained on a subset of MNIST ($n = 10^4$, 10 classes). Its complexity is controlled by the number of trees $N_{tree}$ and the maximum number of leaves allowed for each tree $N_{leaf}^{max}$.

## 3.5 Double Descent in Linear Models

- To analyze double descent rigorously, we consider simplified models where exact calculations are possible.
- Two setups that we can study:
  1. **Gaussian design regression:** inputs $x_i \sim \mathcal{N}(0, I_d)$ with varying dimension $d$.
  2. **Models with fewer symmetries:** capture a more realistic, full U-shaped curve (not covered here, see Ch. 12.2 in Bach).
- In the Gaussian setup:
  - We can show an **explosion of expected risk** at the interpolation threshold $d \approx N$.

[4] See `https://www.pnas.org/doi/abs/10.1073/pnas.1903070116`. For a comprehensive overview, see `https://arxiv.org/abs/2109.02355`.

– But we do not see a classical U-curve for $d < N$, since the prediction problem itself changes as $d$ grows.

💡 We now compute the exact expected risk of the **minimum-norm interpolator** (which is precisely the solution found by GD).

### 3.6 Gaussian Linear Model: Setup and Assumptions

**Random (Gaussian) design setting**[5]**:** both $x$ and $y$ are considered random, each pair $(x_i, y_i)$ assumed to be drawn i.i.d. from a probability distribution $\mathbb{P}$ on $\mathbb{R}^d \times \mathbb{R}$.

- Inputs: $x_i \sim \mathcal{N}(0, I_d)$ independently, $i = 1, \dots, N$.
- Responses:
$$y_i = x_i^\top \theta^* + \varepsilon_i, \qquad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$$
independent of the $x_i$.
- Design matrix: $\mathbf{X} \in \mathbb{R}^{N \times d}$ with rows $x_i^\top$.
- Kernel (Gram) matrix: $K = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{N \times N}$.
- Non-centered covariance matrix:
$$\hat{\Sigma} = \frac{1}{N}\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{d \times d}.$$
- Gradient descent from $\theta^{(0)} = 0$ converges to the minimum-norm interpolator:
$$\hat{\theta} = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{y}.$$
- Since $\Sigma = \mathbb{E}[xx^\top] = I_d$, the excess risk simplifies to
$$R(\hat{\theta}) - R^* = \|\hat{\theta} - \theta^*\|_2^2, \qquad R^* = \sigma^2.$$

### 3.7 Underparameterized Regime ($d < N$)

- When $d < N$, the estimator is the OLS estimator $\theta_{\text{OLS}}$, which is unbiased:
$$\mathbb{E}[\hat{\theta}] = \theta^*.$$
- Expected excess risk can be shown to be:
$$\mathbb{E}[R(\hat{\theta})] - R^* = \sigma^2 \, \mathbb{E}\Big[\operatorname{tr}\left((\mathbf{X}^\top \mathbf{X})^{-1}\right)\Big].$$
- Since $\mathbf{X}^\top \mathbf{X}$ is Wishart-distributed, for $N > d + 1$,
$$\mathbb{E}[R(\hat{\theta})] - R^* = \frac{\sigma^2 d}{N - d - 1} = \frac{\sigma^2 d}{N} \cdot \frac{1}{1 - \frac{d+1}{N}}. \tag{9}$$
- The risk $\to 0$ as $N \to \infty$ with fixed $d$.

[5]This is the so-called *well-specified* setting (Ch. 3.8 in Bach), where we assume that there exists a $\theta^*$ such that the data are actually generated from the conditional model.

### 3.8 Overparameterized Regime ($d > N$)

In this case, the Gram matrix $\mathbf{X}\mathbf{X}^\top$ is (almost surely) invertible. The GD minimum-norm interpolator is

$$\hat{\theta} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\theta^* + \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\varepsilon.$$

**Expected excess risk decomposition:**

$$\mathbb{E}[R(\hat{\theta})] - R^* = \mathbb{E}\Big[\|\hat{\theta} - \theta^*\|_2^2\Big] = \text{Bias}^2 + \text{Variance}.$$

- **Variance term:**

$$\sigma^2\,\mathbb{E}\Big[\text{tr}\big((\mathbf{X}\mathbf{X}^\top)^{-1}\big)\Big] = \frac{\sigma^2 N}{d-N-1}, \qquad d > N+1. \tag{10}$$

- **Bias$^2$ term:**

$$\mathbb{E}\Big[\|\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\theta^* - \theta^*\|_2^2\Big] = \frac{d-N}{d}\|\theta^*\|_2^2. \tag{11}$$

### 3.9 The Main Result: Expected Risk Across Regimes

Collecting all the pieces, we obtain the following result.

**Theorem 3.1** (Double Descent Risk Curve)**.** *For the Gaussian design setting with $x_i \sim \mathcal{N}(0, I_d)$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and the earlier assumptions, the **expected excess risk** is*

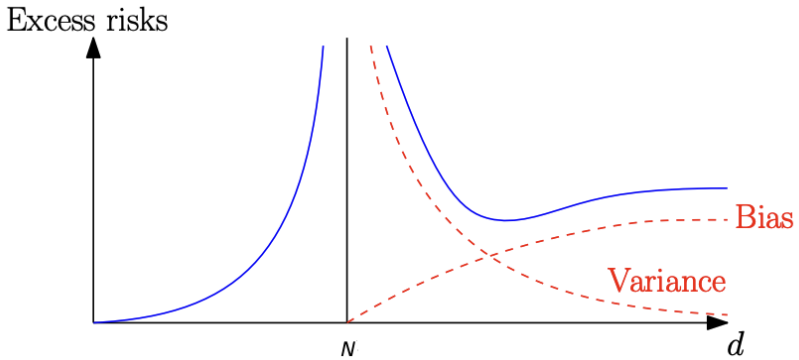
$$\mathbb{E}[R(\hat{\theta})] - R^* = \begin{cases} \dfrac{\sigma^2 d}{N-d-1}, & \text{if } d \le N-2, \\ \dfrac{\sigma^2 N}{d-N-1} + \dfrac{d-N}{d}\|\theta^*\|_2^2, & \text{if } d \ge N+2. \end{cases} \tag{12}$$

*Proof.* See blackboard. □

### 3.10 Remarks

From Equation (12), we observe the following:

- Blow-up at the interpolation threshold ($d \approx N$).
- For $d < N$: excess risk is purely from variance.
- For $d > N$: excess risk decomposes into **variance** + **bias$^2$**. Second descent occurs as $d \gg N$ (variance ↓, bias ↑).
- As $N \to \infty$ with fixed $d$, the risk approaches $\frac{\sigma^2 d}{N}$ (classical rate).



## 3.11 Empirical Evidence in Real-World Datasets

From `https://iclr-blogposts.github.io/2024/blog/double-descent-demystified/`:

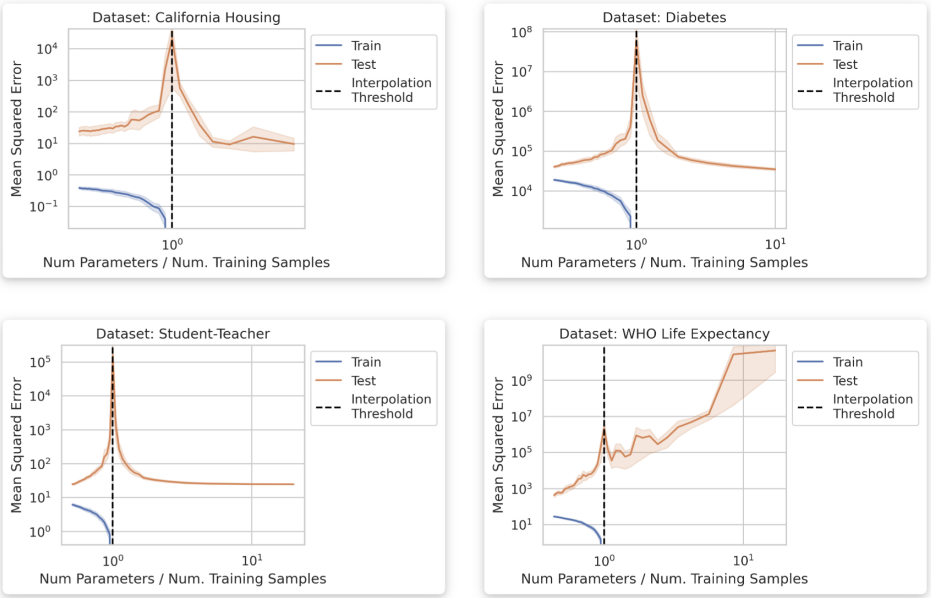



Figure 1. **Double descent in ordinary linear regression.** Three real datasets (California Housing, Diabetes, and WHO Life Expectancy) and one synthetic dataset (Student-Teacher) all exhibit double descent, with test loss spiking at the interpolation threshold. Blue is training error. Orange is test error.

## 3.12 Understanding the Interpolation Peak

**At $d \approx N$ (interpolation threshold):**

- The system $\mathbf{X}\theta = \mathbf{y}$ has (almost) a unique solution. At exactly $d = N$, generically a unique solution exists.
- No flexibility for the algorithm to select among multiple minimizers.
- Model is forced to fit noise as well as signal.
- Solution becomes extremely sensitive to data perturbations.

**Consequence:** The **risk diverges** as $d \to N$ from either side, producing the **peak** in the double descent curve. The mathematical formulas break down at $d \in \{N-1, N, N+1\}$ where required moments do not exist.

For $d > N$, the generalization risk has a U-shaped curve. Can we find its minimizer? See Exercise 11.5.

Could using ridge regression instead remove the peak near $d \approx N$, i.e. does regularization suppress the variance explosion? See Exercise 11.4 (b)–(c).

## 3.13 When Does Double Descent Occur?

Double descent emerges from the **interaction between the properties of the data** (distribution, covariance structure, noise level) and the **inductive biases of the learning algorithm** (e.g., GD $\to$ minimum-norm interpolator).

**Important Caveats:**

- Double descent is not universal — it depends on both data and algorithm.
- It arises specifically for the **minimum-norm interpolator** found by (stochastic) GD.
- If a different empirical risk minimizer is chosen, the phenomenon may not appear.
- If "model size" is measured differently (e.g., parameter norm instead of raw dimension/parameter count), the curve may look different.
- In fact, one can even **design the generalization curve**.

💡 Double descent is a phenomenon, not a law — its emergence depends on the interplay between data, architecture, and optimization.

### 3.14 Multiple Types and Contexts of Double Descent

**Nakkiran et al. (2019)[6]:** Double descent can appear along multiple axes[7]:

1. **Model-wise**: varying number of parameters
2. **Epoch-wise**: varying training time
3. **Sample-wise**: varying training set size

**Empirical evidence across contexts:**

- Linear regression with Gaussian design and noise
- Random feature models and kernel regression
- Deep neural networks (CNNs, transformers)
- Ensemble methods (e.g., random forests)

💡 Double descent is *not at odds* with the classical tradeoff: the spike near interpolation reflects variance blow-up at the boundary of model capacity. Beyond this regime, overparameterization and implicit bias allow the variance to shrink again.

### 3.15 Demonstration: Double Descent with Ridge Regularization

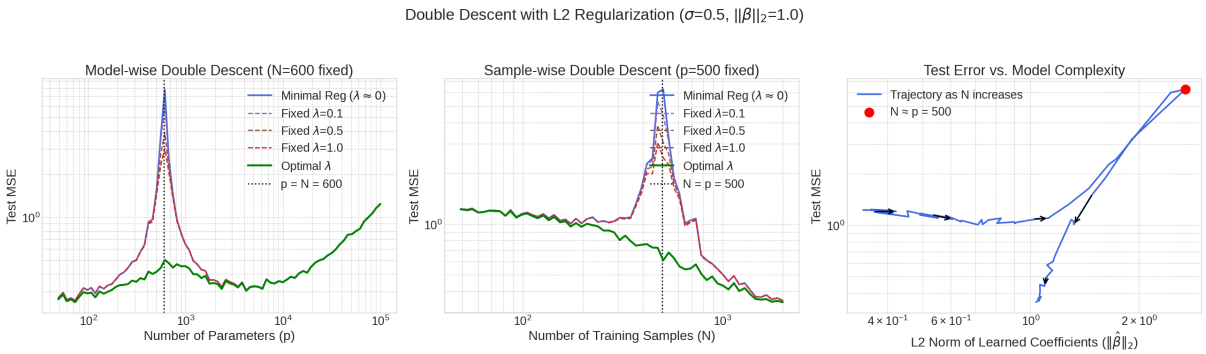
With stronger/optimally tuned $L_2$ (ridge) regularization, the sharp interpolation peak is *diminished or disappears*, showing how explicit control can smooth out double descent.

**Setting** (see the code): linear model with Gaussian features

$$y = X\beta^\star + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I),$$

where $\|\beta^\star\|_2 = 1$ is fixed.

[6] `https://arxiv.org/abs/1912.02292`.
[7] See `https://openai.com/index/deep-double-descent/` for a quick intro.




# 4 Benign Overfitting

## 4.1 From Double Descent to Benign Overfitting

**Key puzzle:** Why does test error decrease again when $d \gg N$, even though the model interpolates the noise?

- In high dimensions, the minimum-norm interpolator $\hat{\theta}$ aligns mostly with the signal subspace.
- Noise is fit in directions *orthogonal* (or nearly orthogonal) to $\theta^*$.
- These directions have little effect on prediction error.

**Benign Overfitting:** Overfitting is *benign* when the fitted noise lives in low-signal directions that do not harm generalization.

**Practical implication:** Overparameterization can sometimes *help* generalization, especially when the data has hidden structure.

## 4.2 Benign Overfitting: Why Overparameterization Helps

**Two contrasting covariance structures:**

- **Isotropic case** ($\Sigma = I$): as $d \to \infty$, excess risk $\to \|\theta^*\|_2^2$. Noise affects all directions equally $\to$ overfitting is harmful.
- **Anisotropic case** (decaying spectrum of $\Sigma$):
  - Signal concentrates in top eigendirections (large eigenvalues).
  - Noise is fit in weak eigendirections (small eigenvalues).
  - Weak directions contribute negligibly to prediction risk.

💡 Benign overfitting arises from *anisotropy*: noise gets absorbed in low-variance directions, leaving predictions accurate.

**Bartlett et al. (2020)**[8] provide a precise analysis along this line in the linear model setting.

[8] `https://arxiv.org/abs/1906.11300`. See also `https://arxiv.org/abs/2103.09177` for a survey.

# 5 Scaling Laws

## 5.1 Scaling Laws: Empirical Observation

**Surprising empirical fact:** In many domains, larger models consistently perform better, *even when already overparameterized.*

- **Language models:** GPT-1 (117M) → GPT-2 (1.5B) → GPT-3 (175B) → GPT-4 (∼1.7T).
- **Computer Vision:** Increasing depth and width improves accuracy.
- **Scientific ML:** Larger physics-informed NNs often yield better accuracy.

**Kaplan et al. (2020)[9]:** Study empirical scaling laws for language model performance on the cross-entropy loss, and show that performance follows approximate **power-law scaling**[10]:

$$\text{Error or Loss} \propto (\text{Model Size})^{-\alpha} \times (\text{Data Size})^{-\beta} \times (\text{Compute})^{-\gamma}.$$

## 5.2 Scaling Laws for Neural Language Models

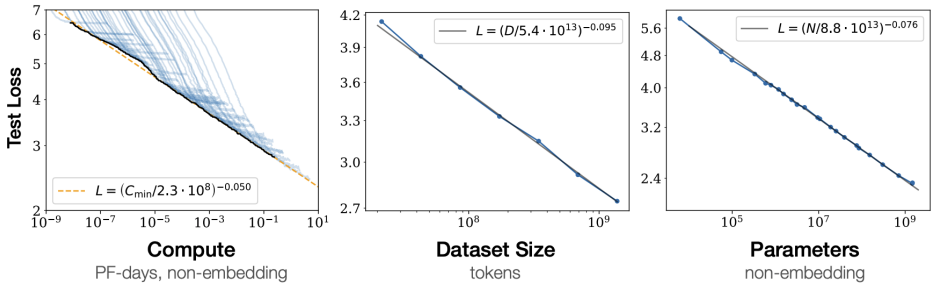



**Figure 1** Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

- Exponents $\alpha, \beta, \gamma > 0$ depend on task and architecture.
- Not universal across all ML domains, but holds remarkably well for language modeling.
- Suggests that we have not yet hit diminishing returns. Performance often improves predictably with more parameters, more data, and more compute.
- Open problems: when do the laws break down? How do they depend on data quality vs. quantity? How can we understand the empirical laws?

## 5.3 Active Research Directions

Many hot research directions in overparameterized ML and related topics:

- **Optimizer Dynamics and Implicit Bias**[11]: Which solutions GD-based methods select, and how optimizer dynamics affect this selection

[9] `https://arxiv.org/abs/2001.08361`. See also `https://arxiv.org/abs/2203.15556`.

[10] Interestingly, power-law behaviors appear throughout deep learning (see, e.g., `https://www.di.ens.fr/~simsekli/ht_ml_2023/hodgkinson.pdf` for an overview on the latest research), and even more broadly across the natural sciences (see `https://arxiv.org/abs/cond-mat/0412004`).

[11] See `https://dl.acm.org/doi/10.5555/3327757.3327921`, `https://arxiv.org/abs/1901.06053`.

- **Loss Geometry and Generalization**[12]: Why some minima generalize better than others
- **Mechanistic Interpretability**[13]: Understanding what neural networks actually learn and why
- **Lottery Ticket Hypothesis**[14]: Finding sparse subnetworks that train as well as full models
- **Grokking**[15]: Understanding sudden generalization after memorization

New discoveries regularly challenge our understanding of deep learning. *Theory is lagging behind experiments.* Bridging this gap could unlock principled improvements to existing methods and guide the development of new ones.

# 6 Key Takeaways

1. **Regularization Everywhere:** Regularization (explicit *or* implicit) is present in all regimes. In the overparameterized setting, it is *crucial* for explaining why interpolation can still generalize.
2. **Classical Theory Refined:** The bias–variance tradeoff alone is incomplete. Modern results extend it: test error can spike at the interpolation threshold and then decrease again (*double descent*). The curve depends on how model complexity is measured.
3. **Overparameterization Can Help:** Having more parameters can improve generalization, thanks to implicit biases of optimization.
4. **Optimization Matters:** Gradient descent does not pick an arbitrary interpolator — it converges to the minimum-norm solution. Algorithm choice shapes generalization.
5. **Practice Leads Theory:** Phenomena like benign overfitting and scaling laws were observed empirically before being partially explained theoretically[16].

### Key Papers and References


- Zhang et al. (2017): Understanding deep learning requires rethinking generalization
- Belkin et al. (2019): Reconciling modern ML practice and the bias–variance trade-off
- Nakkiran et al. (2019): Deep Double Descent
- Bartlett, Long, Lugosi, & Tsigler (2020): Benign overfitting in linear regression
- Hastie, Montanari, Rosset, & Tibshirani (2022): Surprises in high-dimensional ridgeless least squares interpolation
- Mei & Montanari (2019): The generalization error of random features regression
- D'Ascoli et al. (2020): Triple descent and the two kinds of overfitting
- Chen et al. (2020): Multiple descent: design your own generalization curve
- Curth et al. (2023): A U-turn on Double Descent: Rethinking Parameter Counting in Statistical Learning


[12] See `https://arxiv.org/abs/1609.04836` on sharp vs. flat minima.
[13] See `https://www.anthropic.com/research/tracing-thoughts-language-model` for a fun read.
[14] See `https://arxiv.org/abs/1803.03635`.
[15] See `https://arxiv.org/abs/2201.02177`.
[16] See an overview here: `https://arxiv.org/abs/2105.04026`.

# 7 Exercises

1. Consider linear regression with squared loss

$$L(\theta) = \tfrac{1}{2}\|\mathbf{X}\theta - \mathbf{y}\|_2^2, \qquad \mathbf{X} \in \mathbb{R}^{N\times d}, \quad \mathbf{y} \in \mathbb{R}^N.$$

   (a) Write down the gradient descent update for $\theta$ and state the stability condition on the step size $\eta$.

   (b) Analyze the dynamics of gradient descent by expressing the iterates in the eigenbasis of $A = \mathbf{X}^\top\mathbf{X}$. What is the effect of $\eta$ and the iteration number $t$ on each eigendirection?

   (c) Explain why early stopping acts as a form of *spectral regularization*, and compare this behavior to ridge regression.

2. Consider training with Gaussian input noise:

$$\tilde{x} = x + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I_d).$$

   For a loss function $\ell$ and predictor $f_\theta : \mathbb{R}^d \to \mathbb{R}$, the noise-regularized risk is

$$R_{\text{noise}}(\theta) = \mathbb{E}_{(x,y)}\, \mathbb{E}_\varepsilon\, \ell(f_\theta(x+\varepsilon), y).$$

   (a) Assume that the function $g(x) := \ell(f_\theta(x), y)$ is twice continuously differentiable in $x$. Using a second-order Taylor expansion in $\varepsilon$, show that

$$R_{\text{noise}}(\theta) \approx R(\theta) + \frac{\sigma^2}{2}\mathbb{E}_{(x,y)}\left[\operatorname{tr}(H_x \ell(f_\theta(x), y))\right],$$

   where $R(\theta) = \mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)]$ is the original risk and $H_x g(x)$ denotes the Hessian of $g(x)$ with respect to $x$.

   (b) Specialize to the squared-error loss

$$\ell(f, y) = \tfrac{1}{2}(f - y)^2$$

   and derive the correction term.

3. Consider the Vicinal Risk Minimization (VRM) objective, which trains on convex interpolations between points $x, x'$ drawn from the same class[a]:

$$\mathcal{L}_{\text{VRM}}(\theta) = \mathbb{E}_{(x,y)}\, \mathbb{E}_{\lambda\sim\text{Uniform}[0,1]}\, \mathbb{E}_{x'\sim p(\cdot|y)}\, \ell\big(f_\theta(x + \lambda(x' - x)), y\big).$$

   (a) Assume $\ell(f_\theta(x), y) = (f_\theta(x) - y)^2/2$. By applying a second-order Taylor expansion with respect to the perturbation $\delta = \lambda(x' - x)$, show that

$$\mathcal{L}_{\text{VRM}}(\theta) \approx \mathcal{L}_{\text{ERM}}(\theta) + \frac{1}{6}\mathbb{E}_{(x,y)}\big[\nabla_x f_\theta(x)^\top \Sigma_y \nabla_x f_\theta(x)\big],$$

   where $\mathcal{L}_{\text{ERM}}(\theta) = \mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)]$ and

$$\Sigma_y = \mathbb{E}_{x'|y}[(x' - x)(x' - x)^\top]$$

   is the class-conditional covariance matrix.

(b) Compare this penalty with the noise injection regularizer from the previous exercise. What is the key difference in the directions along which smoothness is enforced?

4. Consider the random-design linear model

$$y = x^\top \theta^* + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2), \quad x \sim P \text{ with covariance } \Sigma = \mathbb{E}[xx^\top],$$

where $\Sigma$ is positive definite. We observe $N$ i.i.d. samples $\{(x_i, y_i)\}_{i=1}^N$, and define the empirical covariance

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N x_i x_i^\top.$$

Let $\hat{\theta}_{\text{OLS}}$ denote the OLS estimator and $\hat{\theta}_\lambda$ the ridge estimator with $\lambda > 0$. The Bayes risk is $R^* = \sigma^2$.

(a) Prove that for OLS,

$$\mathbb{E}\big[R(\hat{\theta}_{\text{OLS}})\big] - R^* = \frac{\sigma^2}{N} \mathbb{E}\Big[\operatorname{tr}\big(\Sigma \hat{\Sigma}^{-1}\big)\Big].$$

(b) Show that for ridge regression,

$$\mathbb{E}\big[R(\hat{\theta}_\lambda)\big] - R^* = \lambda^2 \mathbb{E}\Big[\theta^{*\top} (\hat{\Sigma} + \lambda I)^{-1} \Sigma (\hat{\Sigma} + \lambda I)^{-1} \theta^*\Big] + \frac{\sigma^2}{N} \mathbb{E}\Big[\operatorname{tr}\big((\hat{\Sigma} + \lambda I)^{-2} \hat{\Sigma} \Sigma\big)\Big].$$

(c) Discuss whether a double-descent peak occurs as $d$ crosses $N$ for ridge with fixed $\lambda > 0$. Use the expression in (b) and consider the behavior of the eigenvalues of $\hat{\Sigma}$ when $d > N$. What changes when $\lambda \to 0$?

5. From the Double Descent Theorem, we have, for $d > N + 1$,

$$E_r(d) := \mathbb{E}[R(\hat{\theta})] - R^* = \underbrace{\Big(1 - \tfrac{N}{d}\Big) \|\theta^*\|^2}_{\text{bias term}} + \underbrace{\sigma^2 \cdot \tfrac{N}{d - N - 1}}_{\text{variance term}}.$$

Define the signal-to-noise ratio (SNR):

$$\text{SNR} = \frac{\|\theta^*\|^2}{\sigma^2}.$$

(a) Determine when $E_r(d)$ admits a critical point in $(N + 1, \infty)$. What condition on SNR is required?

(b) Assuming such a critical point exists, find the minimizer $d^*$ in terms of $N$ and SNR.

(c) Discuss limiting behaviors:

(i) As $d \downarrow N + 1$, which term dominates?

(ii) What is $\lim_{d \to \infty} E_r(d)$?

(iii) For which SNR does a finite minimizer $d^*$ exist? What happens to $d^*$ as $\text{SNR} \to 1^+$ and as $\text{SNR} \to \infty$?

[a] In standard mixup (Zhang et al., 2018); `https://arxiv.org/abs/1710.09412`, pairs $(x, y), (x', y')$ are drawn regardless of class, and both inputs and labels are interpolated: $\tilde{x} = \lambda x + (1 - \lambda) x'$, $\tilde{y} = \lambda y + (1 - \lambda) y'$. Here, we restrict to $x' \sim p(\cdot | y)$ so that labels remain unchanged for simplicity.

6. **[Experimental] Design Your Own Generalization Curve.** In classical linear regression, the *double descent* curve appears when test error decreases for $d < N$, spikes near $d = N$, and decreases again for $d > N$. Here we study the *feature-revealing* construction[a], where the distribution of newly added features can be chosen. This allows us to engineer *multiple descents*.

   Generate training data with $N = 50$ samples and $D = 100$ features, with $\theta^* = 0$, $\varepsilon_i \sim \mathcal{N}(0, 1)$, as follows:

   – For the first 58 features ($d = 1, \ldots, 58$):

   $$x_{i,d} \sim \mathcal{N}(0, 1).$$

   – For the next 6 features ($d = 59, \ldots, 64$): alternate between *benign* and *toxic* according to

   $$\Delta = \{\downarrow, \uparrow, \downarrow, \downarrow, \uparrow, \downarrow\}.$$

   where

   $\downarrow$ **Benign feature:**

   $$x_{i,d} \sim \mathcal{N}(0, \sigma^2).$$

   $\uparrow$ **Toxic feature:**

   $$x_{i,d} \sim \tfrac{1}{3}\mathcal{N}(-\mu, \sigma^2) + \tfrac{1}{3}\mathcal{N}(0, \sigma^2) + \tfrac{1}{3}\mathcal{N}(+\mu, \sigma^2).$$

   – For the remaining features ($d = 65, \ldots, 100$):

   $$x_{i,d} \sim \mathcal{N}(0, 1).$$

   Try the following for different values of $\mu$ and $\sigma$.

   (a) For each $d$, fit the minimum-norm interpolator

   $$\hat{\theta}_d = \mathbf{X}_d^+ \mathbf{y}$$

   where $\mathbf{X}_d$ is the design matrix restricted to the first $d$ features.

   (b) To evaluate the models, first generate a large, independent test set (e.g. $n_{\text{test}} = 2000$ samples). For each test sample, generate the full $D = 100$-dimensional feature vector and the corresponding label $y_i$ using the process described above. Then, for each estimator $\hat{\theta}_d$, calculate the excess test risk:

   $$L_d = \frac{1}{n_{\text{test}}} \sum_i (x_i^\top \hat{\theta}_d - y_i)^2 - 1.$$

   (c) Plot $L_d$ for $d \in [45, 70]$ and observe whether the test error exhibits descents and ascents that match the feature pattern

   $$\Delta = \{\downarrow, \uparrow, \downarrow, \downarrow, \uparrow, \downarrow\}.$$

   Discuss how this depends on the values of $\mu$ and $\sigma$ used.

[a] See `https://arxiv.org/abs/2008.01036`.

### Discussion: Double vs. Multiple Descent

- **Double descent:** With all features i.i.d. $\mathcal{N}(0,1)$, test error has the universal shape: decrease $\rightarrow$ spike at $d = N \rightarrow$ second descent.
- **Multiple descent**[17]**:** By controlling the distribution of revealed features, we can engineer additional peaks and valleys beyond $d = N$.
- **Out-of-distribution (OOD) connection:** Adding new features from a different distribution is a toy model of distribution shift.
  - *Benign features* (small-variance Gaussian) $\Rightarrow$ descent.
  - *Toxic features* (Gaussian mixture with shifted means) $\Rightarrow$ ascent.
- Generalization curves beyond interpolation are not universal; they depend on how future (possibly OOD) data distributions differ from the training setup.

This connects to our next lecture on *probabilistic approaches*, where we study how modeling uncertainty can help reason about generalization and the OOD case.

[17] https://arxiv.org/abs/2008.01036.

# A Appendix

## A.1 Auxiliary Results

**1D case:** If $z_1, \dots, z_N \sim \mathcal{N}(0,1)$ i.i.d., then

$$Q = \sum_{i=1}^{N} z_i^2 \sim \chi_N^2.$$

So the $\chi^2$ distribution describes the sampling variability of variance in 1D.

**Multi-D case:** If $x_i \sim \mathcal{N}(0, I_d)$ i.i.d., then

$$S = \sum_{i=1}^{N} x_i x_i^\top = \mathbf{X}^\top \mathbf{X}$$

is a random $d \times d$ positive semidefinite matrix.

**Wishart distribution:** The multivariate analogue of $\chi^2$, governing the distribution of covariance matrices.

## A.2 Definition: Wishart Distribution

See, e.g., `https://swnydick.github.io/assets/reports/Wishart_Distribution.pdf` for details.

**Definition A.1** (Wishart Distribution)**.** Let $x_1, \dots, x_N \sim \mathcal{N}(0, \Sigma)$ i.i.d. in $\mathbb{R}^d$, and define

$$S = \sum_{i=1}^{N} x_i x_i^\top \in \mathbb{R}^{d \times d}.$$

Then $S$ is said to follow a **Wishart distribution** with parameters $(\Sigma, N)$:

$$S \sim \mathcal{W}_d(\Sigma, N),$$

with density

$$p(S) = \frac{|S|^{\frac{N-d-1}{2}} \exp\big(-\frac{1}{2}\operatorname{tr}(\Sigma^{-1}S)\big)}{2^{Nd/2}|\Sigma|^{N/2}\Gamma_d(N/2)}, \qquad S \succeq 0,$$

where

$$\Gamma_d(a) = \pi^{\frac{d(d-1)}{4}} \prod_{j=1}^{d} \Gamma\left(a - \frac{j-1}{2}\right)$$

is the multivariate Gamma function, and $|S|$ is the determinant of $S$.

## A.3 Definition: Inverse-Wishart Distribution

**Definition A.2** (Inverse-Wishart Distribution)**.** A random matrix $\Sigma \in \mathbb{R}^{d \times d}$ follows an **inverse-Wishart distribution** with degrees of freedom $\nu$ and scale matrix $\Psi \succ 0$, denoted

$$\Sigma \sim \mathrm{IW}_d(\nu, \Psi),$$

if $\Sigma^{-1} \sim \mathcal{W}_d(\Psi^{-1}, \nu)$. The density is

$$p(\Sigma) = \frac{|\Psi|^{\nu/2}|\Sigma|^{-(\nu+d+1)/2} \exp\big(-\frac{1}{2}\operatorname{tr}(\Psi\Sigma^{-1})\big)}{2^{\nu d/2}\Gamma_d(\nu/2)}, \qquad \Sigma \succ 0.$$

**Key properties:**

- Requires $\nu > d-1$ for a proper distribution.
- Mean:
$$\mathbb{E}[\Sigma] = \frac{\Psi}{\nu - d - 1} \qquad (\nu > d+1).$$
- Mode:
$$\text{Mode}(\Sigma) = \frac{\Psi}{\nu + d + 1}.$$

## A.4 The Needed Wishart Distribution Facts

Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ with rows $x_i^\top \sim \mathcal{N}(0, I_d)$ i.i.d.

The covariance matrix

$$S = \mathbf{X}^\top\mathbf{X} \sim \mathcal{W}_d(I_d, N)$$

is Wishart distributed.

**Lemma A.3** (Key Properties)**.** *If $S \sim \mathcal{W}_d(I_d, N)$, then:*

*(i)* $\mathbb{E}[S] = NI_d$.

*(ii) If $N > d+1$, then*

$$\mathbb{E}[S^{-1}] = \frac{1}{N-d-1} I_d.$$

This form is used when analyzing the **underparameterized regime** ($d < N$).

*Connection to random matrix theory*[18]*:* In the large-$N, d$ limit with $d/N \to \gamma$, the eigenvalue distribution of $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$ converges to the **Marchenko–Pastur law**.

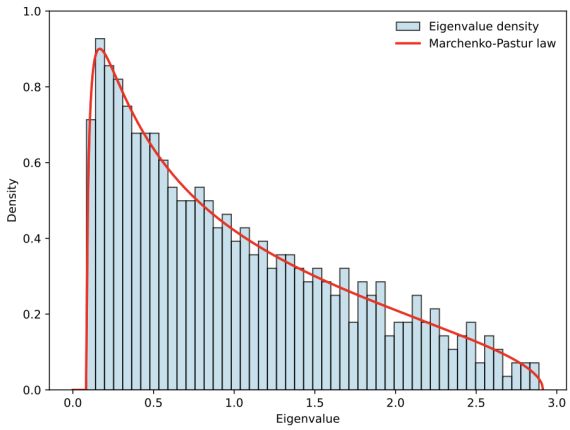



Figure 1: Histogram of eigenvalues of the empirical covariance matrix with $d = 500$ and $N = 1000$.

[18] See `https://zhenyu-liao.github.io/pdf/RMT4ML.pdf` for a nice introduction.

## A.5 The Needed Wishart Distribution Facts

Equivalently, the Gram matrix

$$K = \mathbf{X}\mathbf{X}^\top \sim \mathcal{W}_N(I_N, d)$$

follows a Wishart distribution.

**Lemma A.4** (Key Properties)**.** *If $K \sim \mathcal{W}_N(I_N, d)$, then:*

*(i)* $\mathbb{E}[K] = dI_N$.

*(ii) If $d > N + 1$, then*

$$\mathbb{E}[K^{-1}] = \frac{1}{d - N - 1} I_N.$$

This form is used when analyzing the **overparameterized regime** ($d > N$).

## A.6 Conjugacy and Statistical Connections

**Reminder from tutorial:** We studied conjugacy in Bayesian estimation (Normal–Normal for the mean). Here, a parallel arises in higher dimensions.

| **Aspect** | **1D Mean Estimation** | **Multi-D Covariance Estimation** |
|---|---|---|
| Parameter | Scalar $\theta$ | Covariance $\Sigma$ |
| Likelihood | $y \sim \mathcal{N}(\theta, \sigma^2)$ | $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \Sigma)$ |
| Conjugate Prior | Normal | Inverse-Wishart |
| Data Statistic | Sample mean | Sample covariance matrix |

- In both cases, the conjugate prior ensures closed-form posteriors.
- For covariance matrices, the **inverse-Wishart prior** is conjugate to the Gaussian likelihood.
- The inverse-Wishart prior $\mathrm{IW}(\nu_0, \Psi)$ requires degrees of freedom $\nu_0 > d - 1$ for a proper distribution.
- As $N \to \infty$, the posterior concentrates at the empirical statistic (**data dominates the prior**).

💡 Wishart/inverse-Wishart distributions used in our double descent analysis also appear naturally in Bayesian covariance learning.