DEMYSTIFYING DEEP LEARNING THROUGH HIGH-DIMENSIONAL STATISTICS

JEFFREY PENNINGTON

GOOGLE BRAIN

CU BOULDER

2-19-21

OUTLINE

1. Background
2. Problem Setup and Motivation
3. Part I: What Causes Double Descent?
4. Part II: Beyond Double Descent

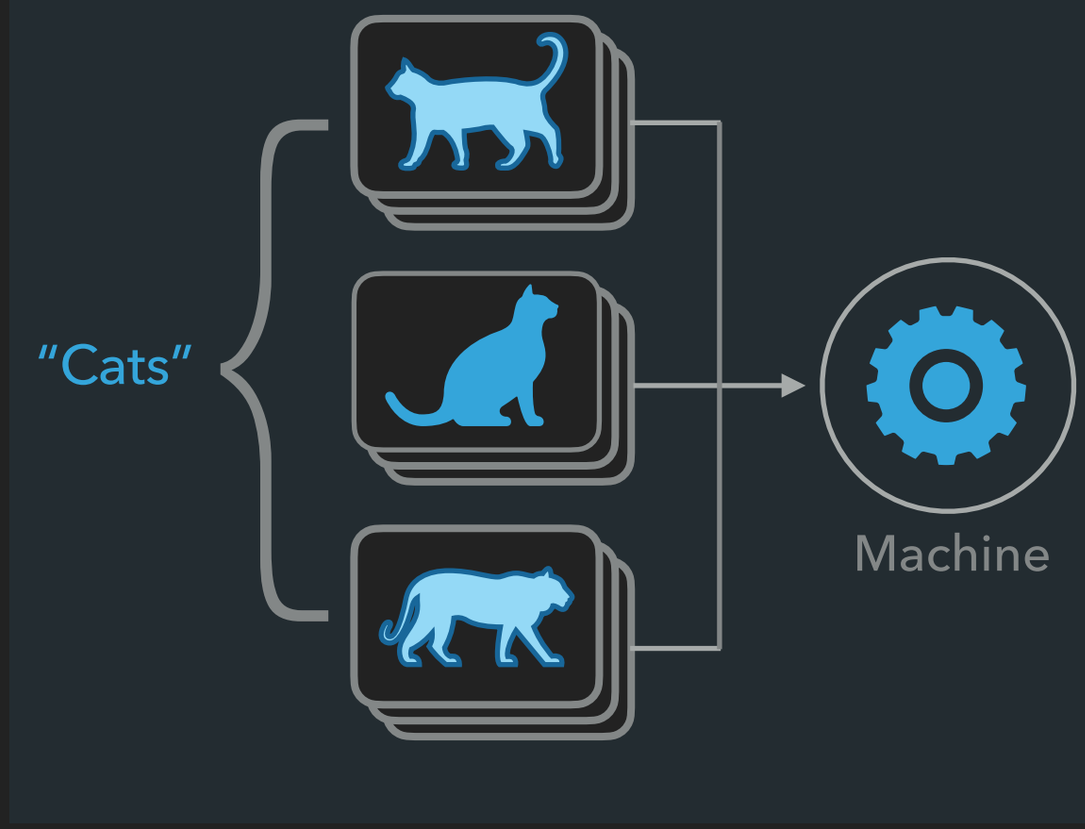
OUTLINE

1. Background
2. Problem Setup and Motivation
3. Part I: What Causes Double Descent?
4. Part II: Beyond Double Descent

SUPERVISED LEARNING

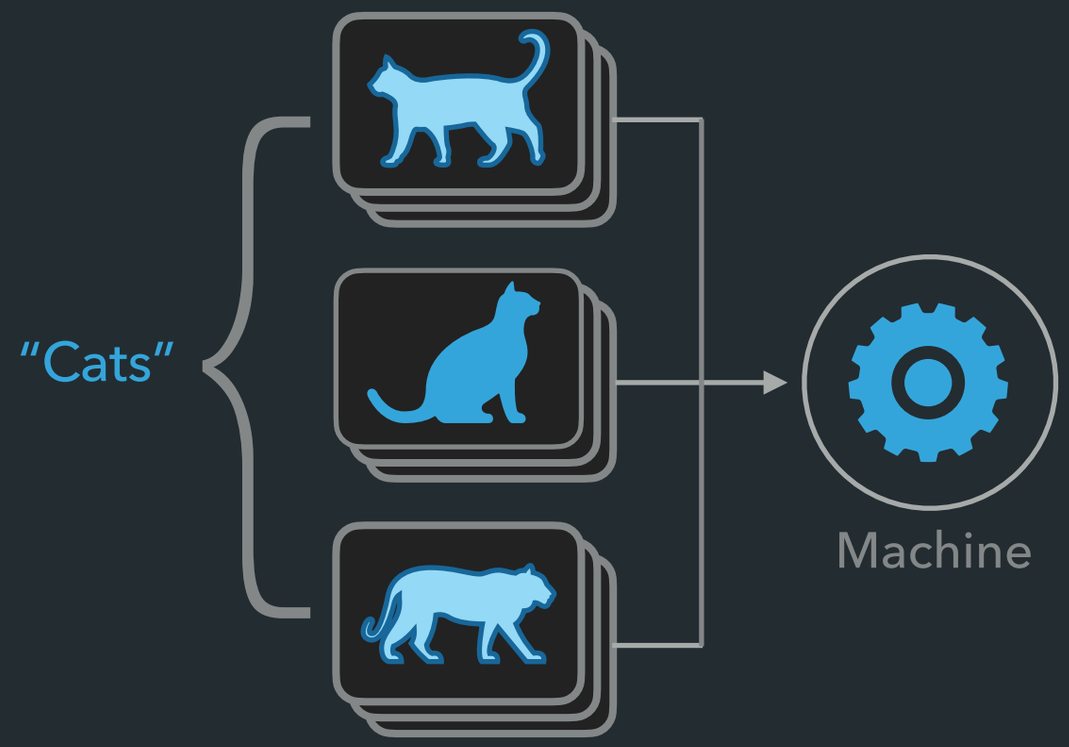
SUPERVISED LEARNING

1. Provide machine learning algorithm with labeled datapoints

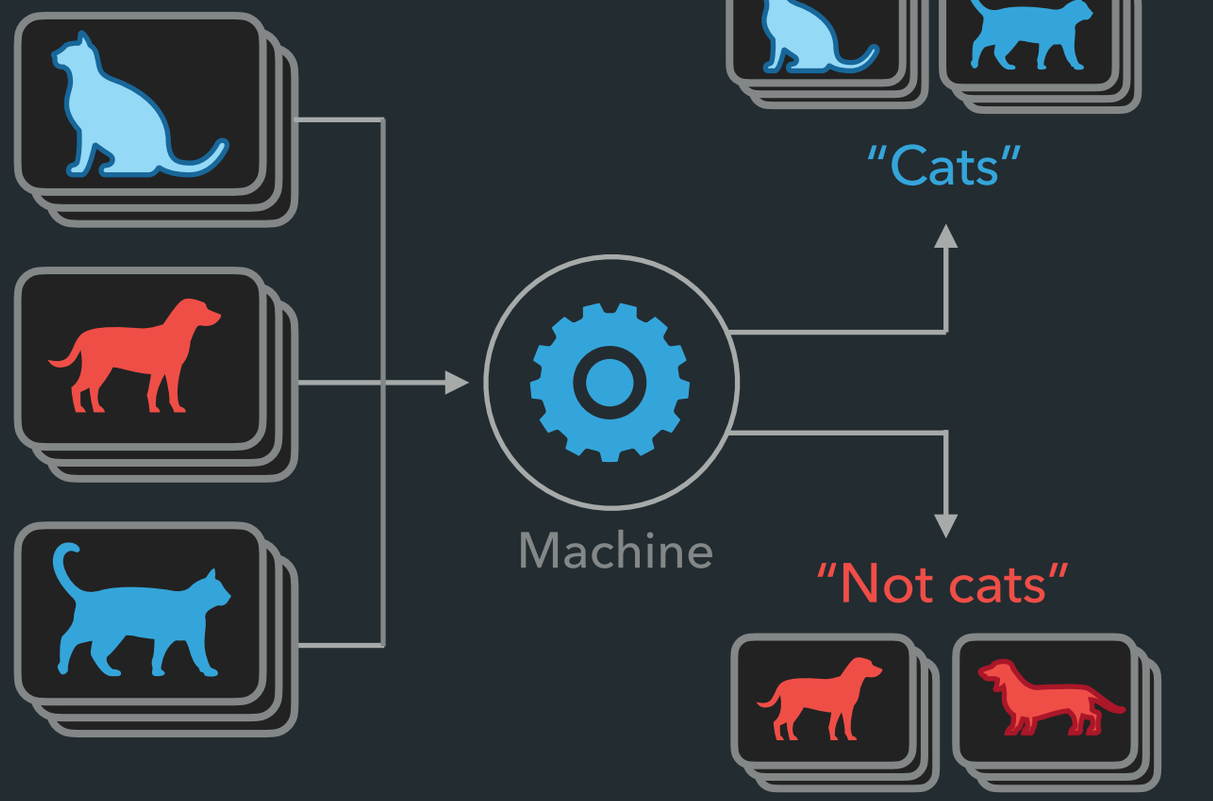


SUPERVISED LEARNING

1. Provide machine learning algorithm with labeled datapoints

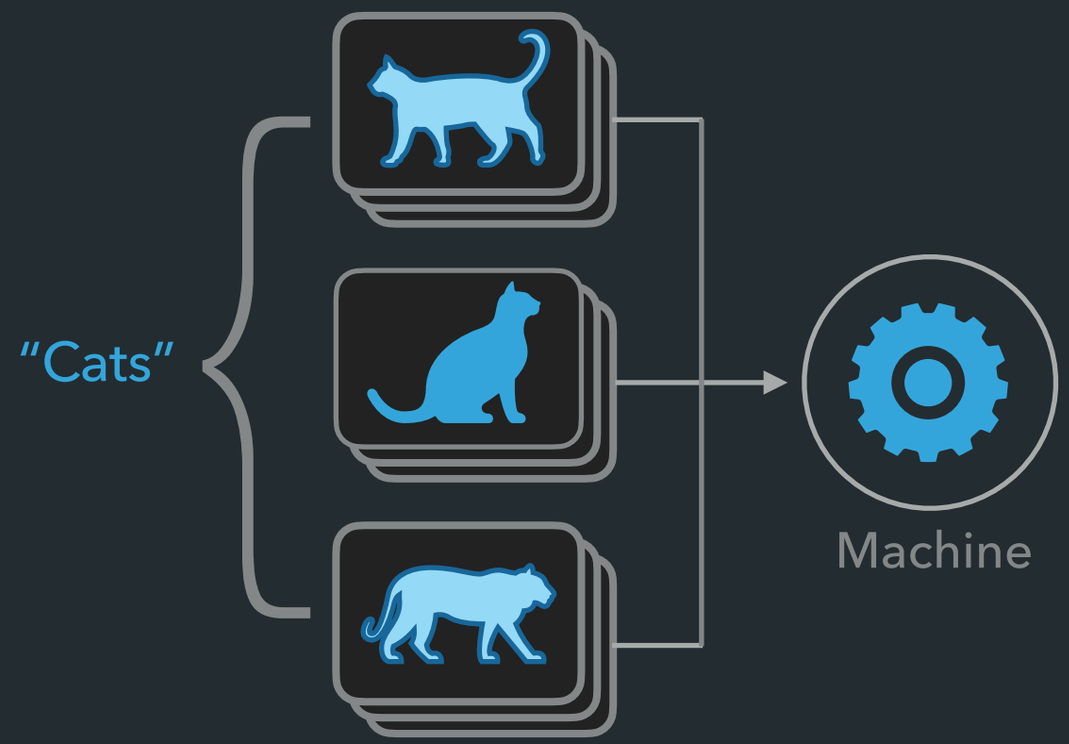


3. Provide new, unlabeled datapoints for the machine to automatically label

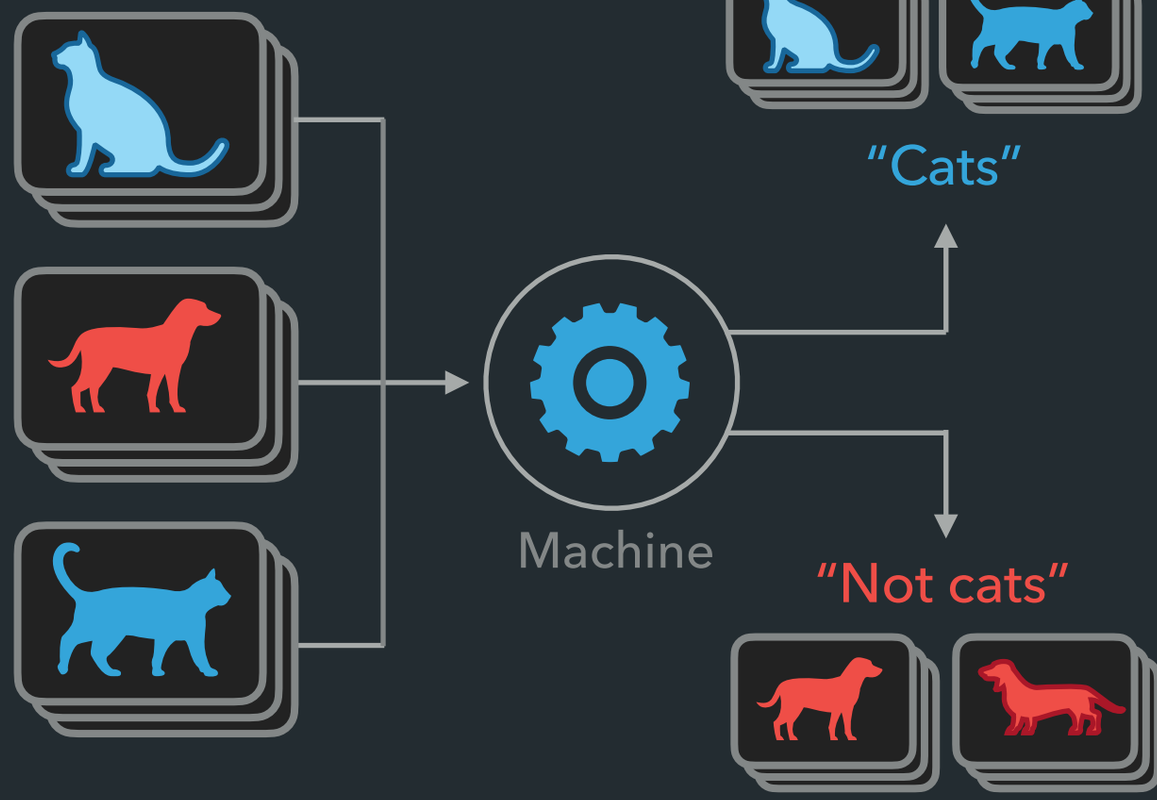


SUPERVISED LEARNING

1. Provide machine learning algorithm with labeled datapoints

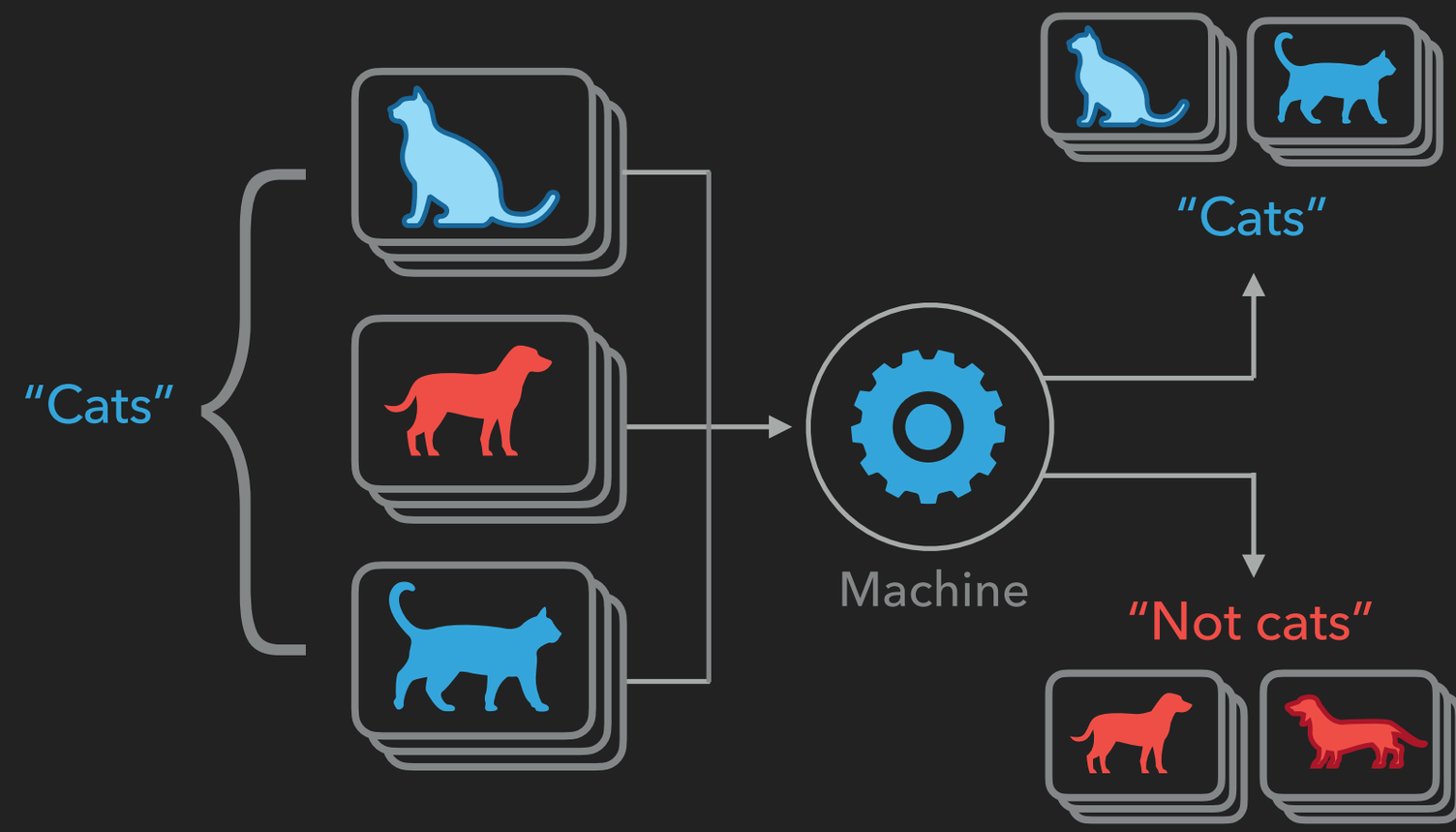


3. Provide new, unlabeled datapoints for the machine to automatically label

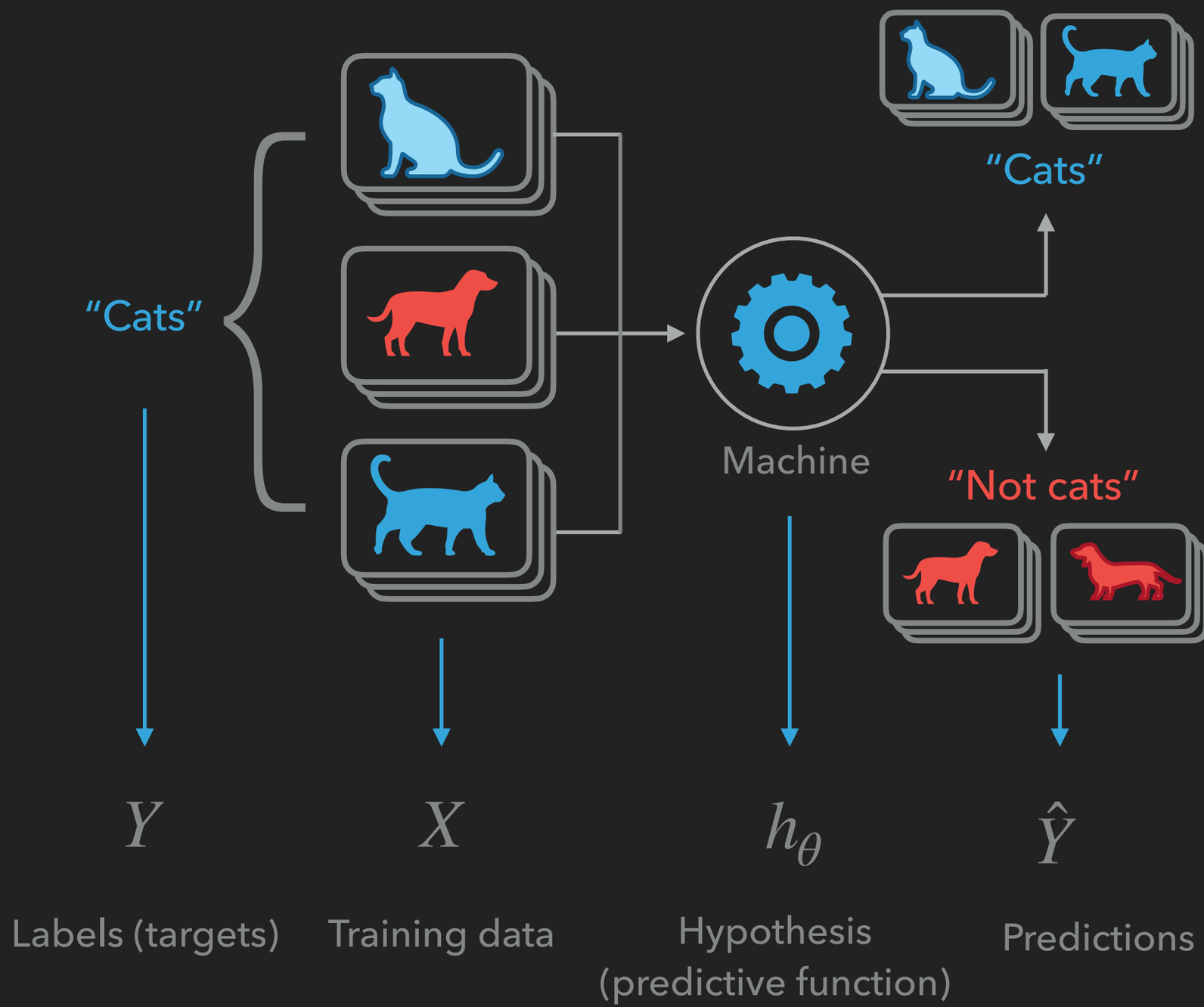


WHAT ABOUT STEP 2? HOW TO TRAIN THE MACHINE?

FORMALIZING THE PROBLEM



FORMALIZING THE PROBLEM



STEP 2: LEARNING A PREDICTIVE FUNCTION

Strategy: Introduce a loss function $L(\hat{Y}, Y)$ whose minimizer gives a good predictive function

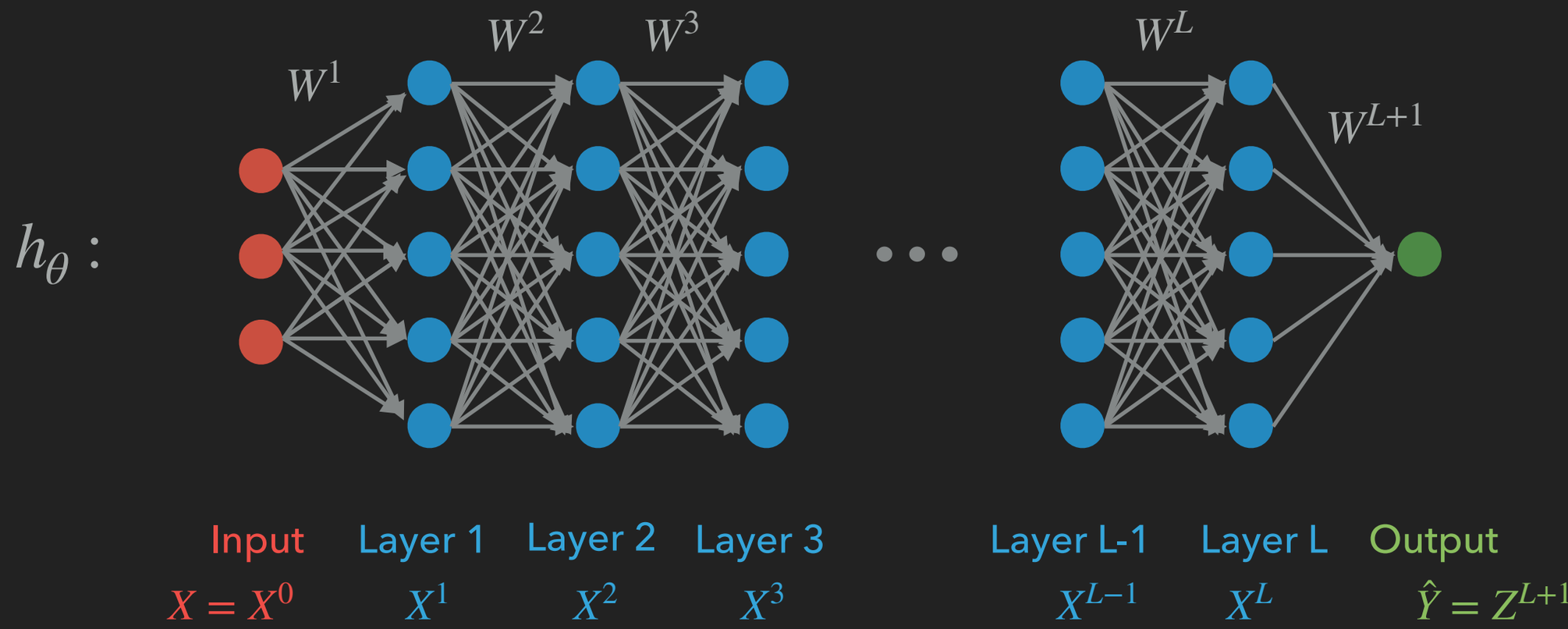
Ex: Least squares regression

$$
\begin{aligned}
L &= \sum_{i\mu} (Y_{i\mu} - \hat{Y}_{i\mu})^2 \\
&= \|Y - \hat{Y}\|_F^2 \\
&= \|Y - h_\theta(X)\|_F^2
\end{aligned}
$$

Goal is to find $\text{argmin}\{L \,|\, h_\theta \in \mathscr{H}\}$ for a well-specified hypothesis class $\mathscr{H}$.

DEEP NEURAL NETWORKS

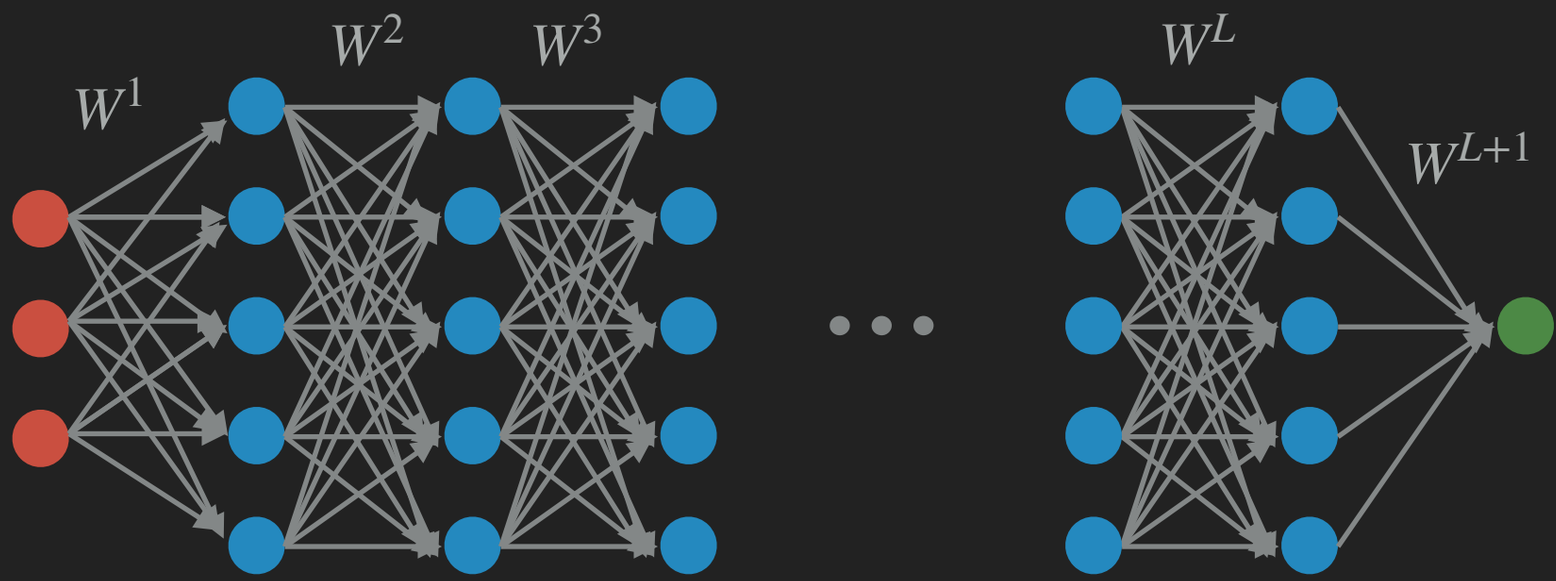
In practice, one common and effective hypothesis class is defined by deep neural networks



DEEP NEURAL NETWORKS

In practice, one common and effective hypothesis class is defined by deep neural networks

h_θ :



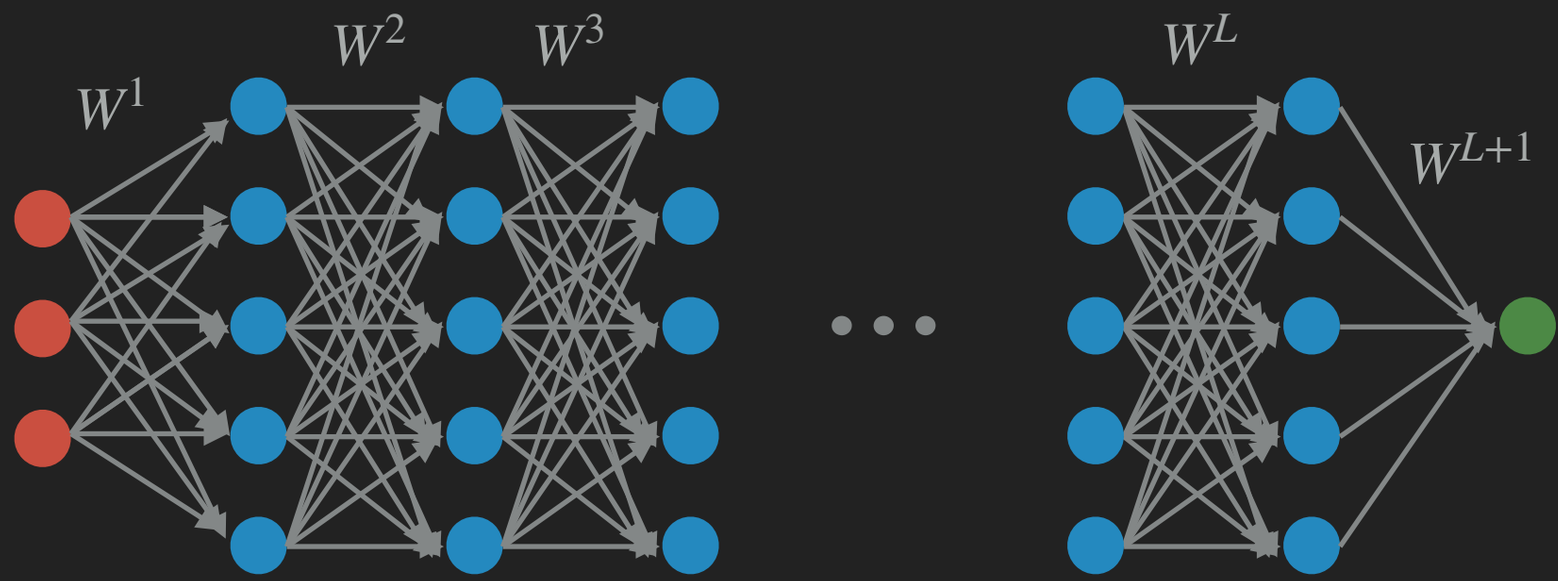
Input $X = X^0$ | Layer 1 X^1 | Layer 2 X^2 | Layer 3 X^3 | Layer L-1 X^{L-1} | Layer L X^L | Output $\hat{Y} = Z^{L+1}$

$$X^l_{i\mu} = f(Z^l_{i\mu}), \quad Z^{l+1} = W^{l+1}X^l + B^{l+1}$$

DEEP NEURAL NETWORKS

In practice, one common and effective hypothesis class is defined by deep neural networks

h_θ :



Input	Layer 1	Layer 2	Layer 3	Layer L-1	Layer L	Output
$X = X^0$	X^1	X^2	X^3	X^{L-1}	X^L	$\hat{Y} = Z^{L+1}$

$$X^l_{i\mu} = f(Z^l_{i\mu}), \quad Z^{l+1} = W^{l+1}X^l + B^{l+1}$$

(Post) activation neurons

Activation function

(Pre) activation neurons

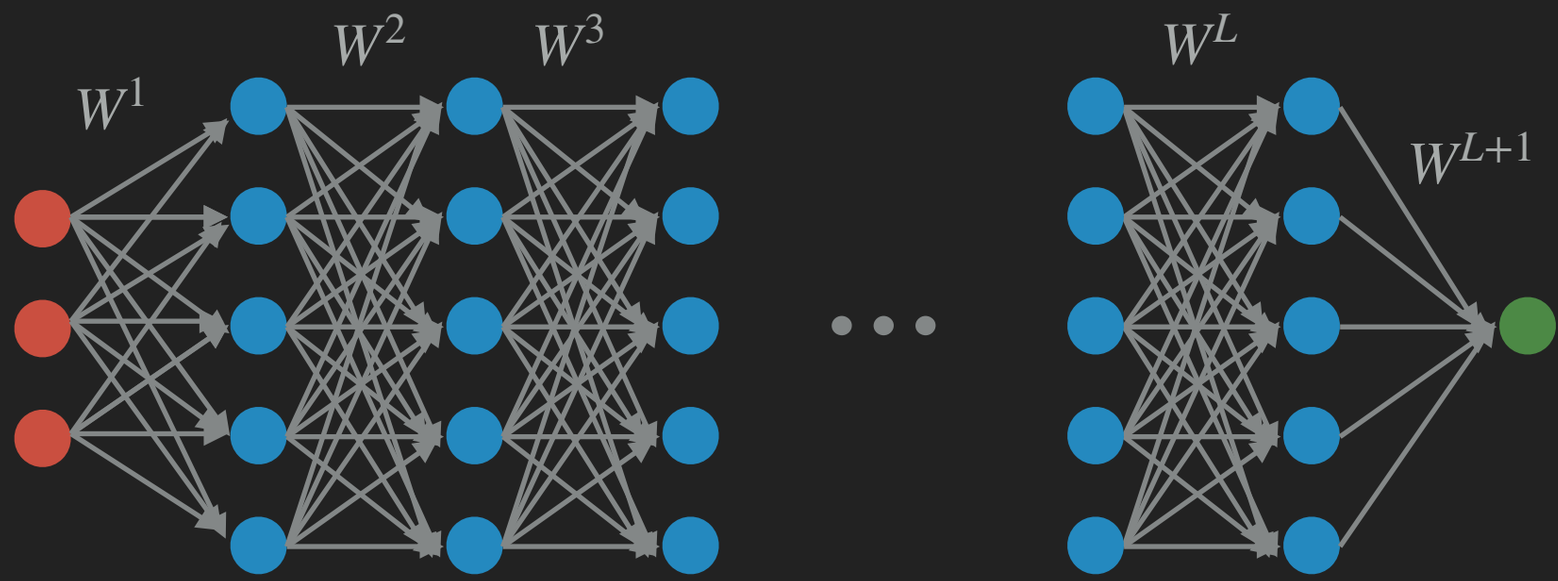
Weights

Biases

DEEP NEURAL NETWORKS

In practice, one common and effective hypothesis class is defined by deep neural networks

h_θ :



Input $X = X^0$ | Layer 1 X^1 | Layer 2 X^2 | Layer 3 X^3 | Layer L-1 X^{L-1} | Layer L X^L | Output $\hat{Y} = Z^{L+1}$

$$X^l_{i\mu} = f(Z^l_{i\mu}), \quad Z^{l+1} = W^{l+1}X^l + B^{l+1} \qquad \theta = \{W^l, B^l\}_{l=1\ldots L+1}$$

(Post) activation neurons

Activation function

(Pre) activation neurons

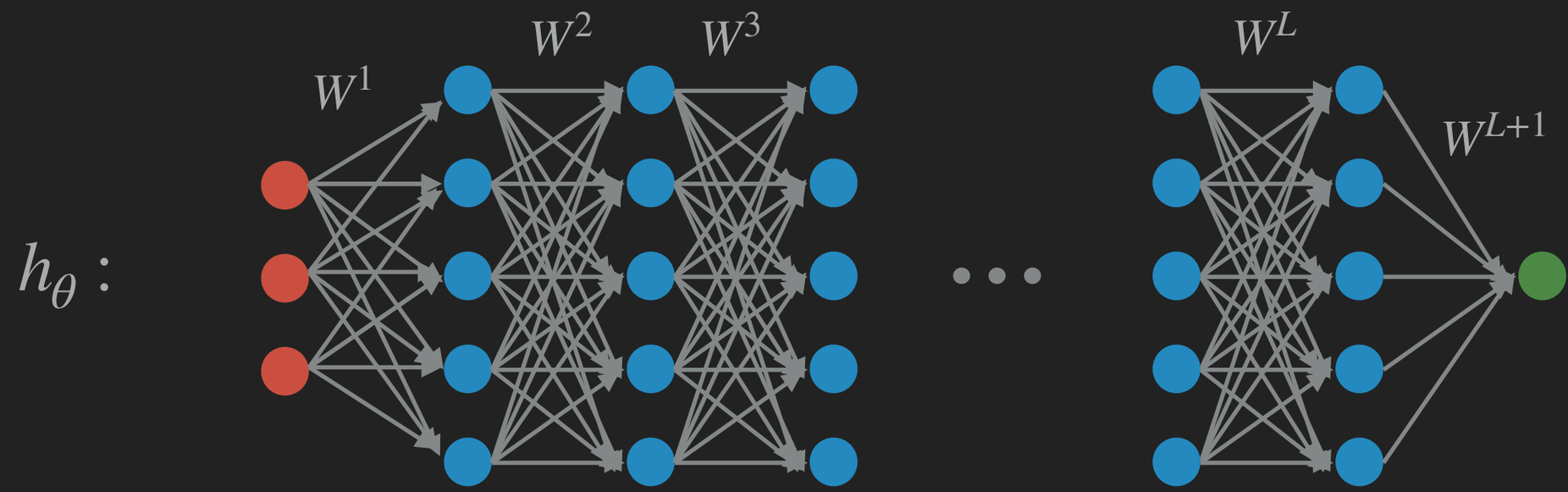
Weights

Biases

Parameters

DEEP NEURAL NETWORKS

In practice, one common and effective hypothesis class is defined by deep neural networks



m - # samples

$$X^l_{i\mu} = f(Z^l_{i\mu}), \quad Z^{l+1} = W^{l+1}X^l + B^{l+1} \qquad \theta = \{W^l, B^l\}_{l=1\ldots L+1}$$

(Post) activation neurons

Activation function

(Pre) activation neurons

Weights

Biases

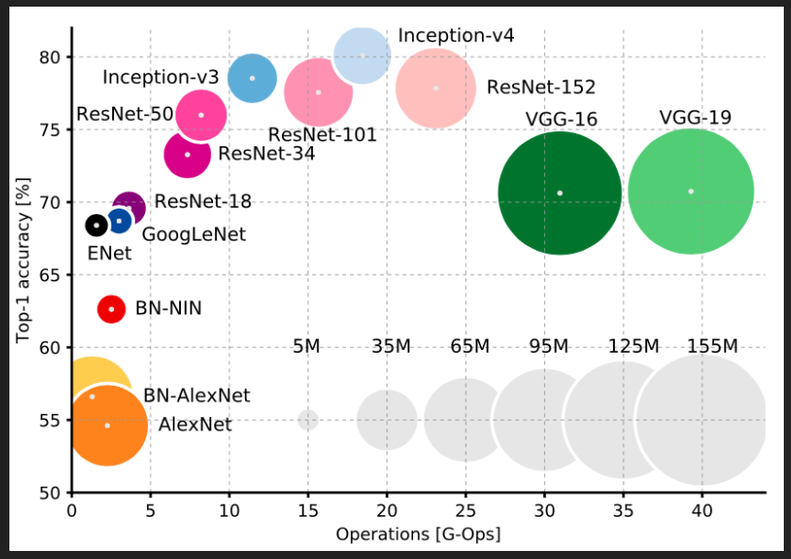
Parameters ($p = |\theta|$)

WHY DOES DEEP LEARNING WORK?

Deep neural networks define a very flexible and expressive class of functions.

State-of-the-art models have millions or billions of parameters

- Meena: 2.6 billion
- Turing NLG: 17 billion
- GPT-3: 175 billion



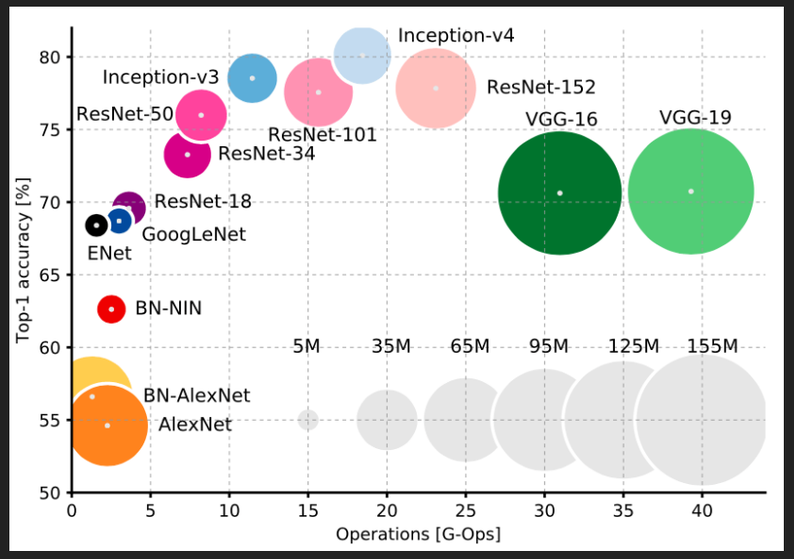
WHY DOES DEEP LEARNING WORK?

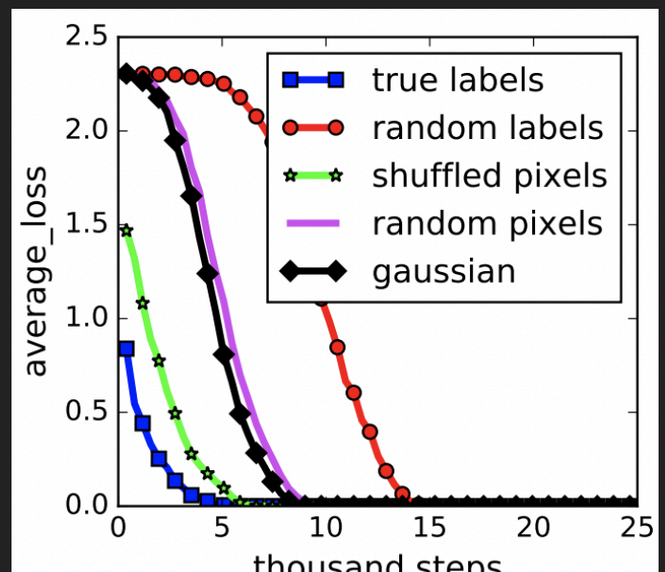
Deep neural networks define a very flexible and expressive class of functions.

State-of-the-art models have millions or billions of parameters

- Meena: 2.6 billion
- Turing NLG: 17 billion
- GPT-3: 175 billion

Models that perform well on real data can easily memorize noise (Zhang et al., 2017)



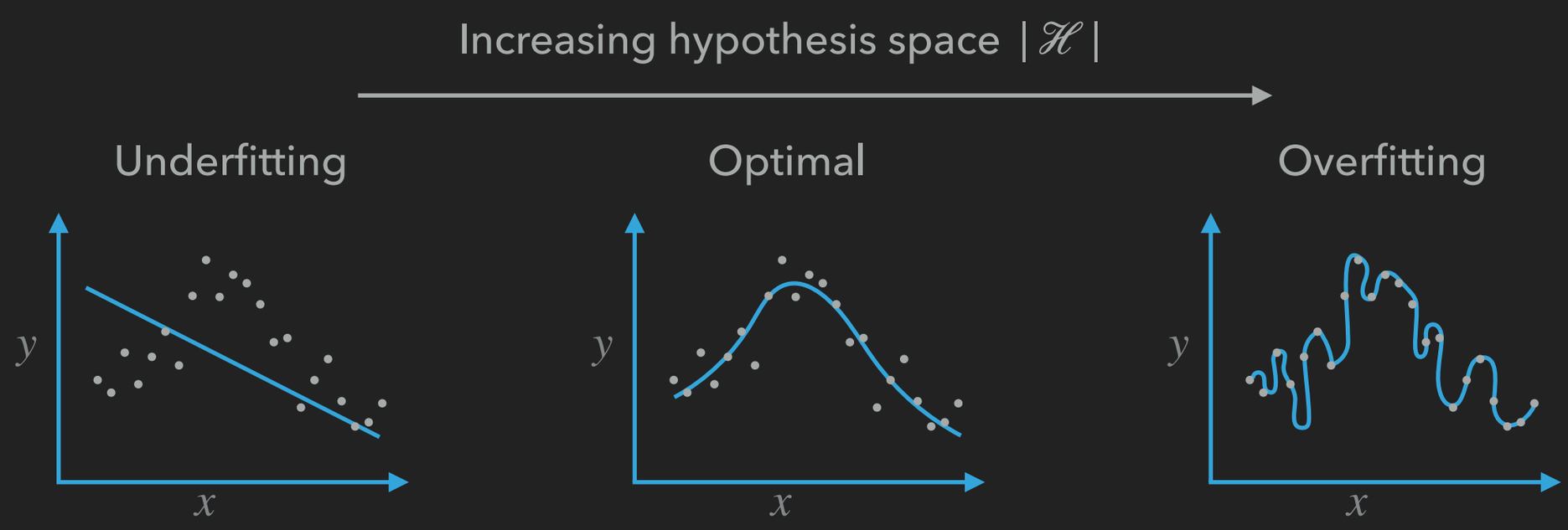


WHY DOES DEEP LEARNING WORK?

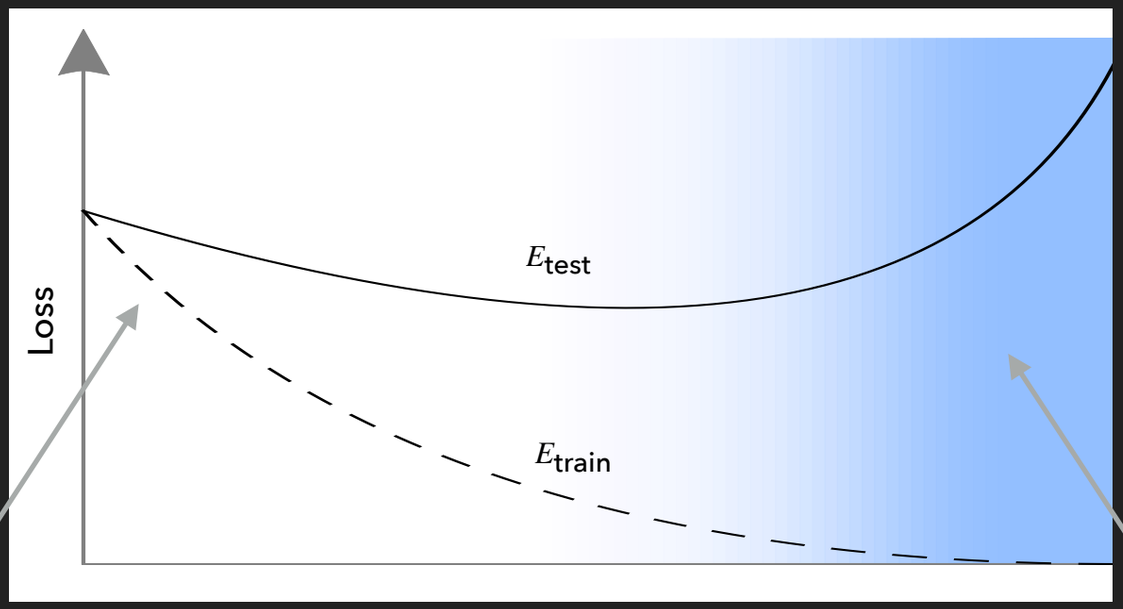
Deep neural networks define a very flexible and expressive class of functions.

➡ Standard wisdom suggests they should overfit, i.e.

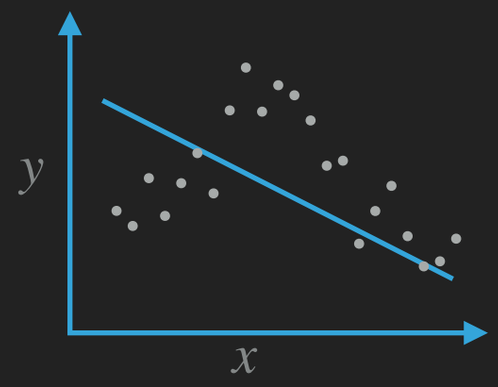
$$E_{\text{train}} \equiv \frac{1}{m}\sum_{\mu=1}^{m} L\big(h_\theta(X_\mu), Y_\mu\big) \to 0 \quad \text{but} \quad E_{\text{test}} \equiv \mathbb{E}_{(x,y)} L\big(h_\theta(x), y\big) \gg 0$$



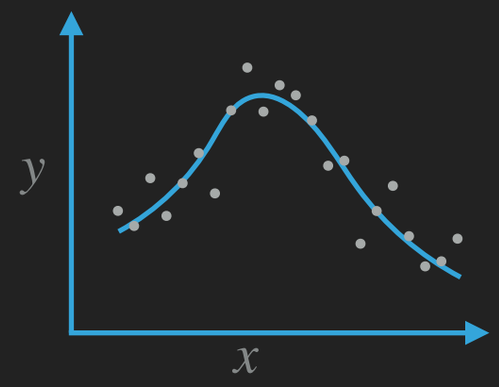
WHY DOES DEEP LEARNING WORK?



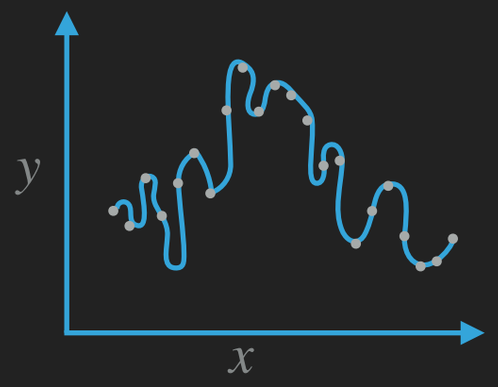
Underfitting



Optimal



Overfitting



BIAS-VARIANCE DECOMPOSITION

Condition on a test point x, then decompose the per-sample loss

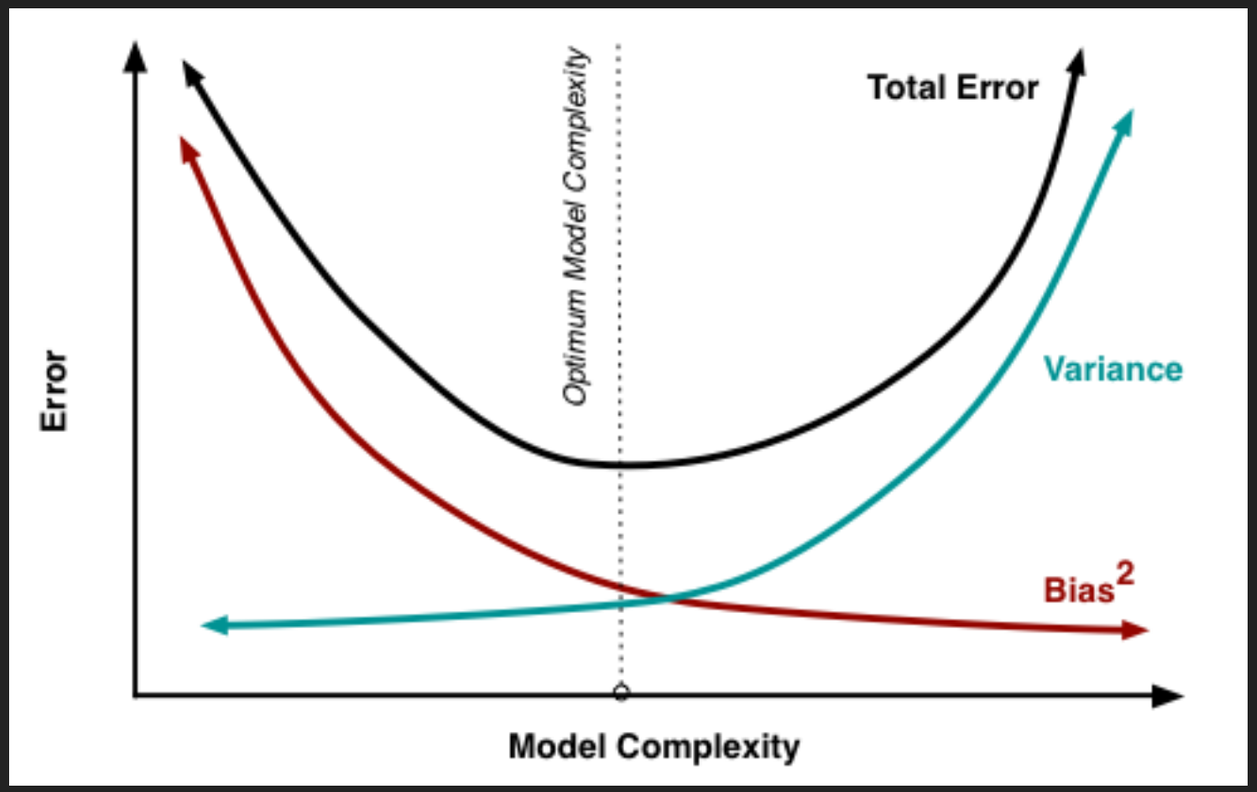
$$\mathbb{E}[(\hat{y}(x) - y(x))^2] = (\mathbb{E}\hat{y}(x) - \mathbb{E}y(x))^2 + \mathbb{V}[\hat{y}(x)] + \mathbb{V}[y(x)]$$

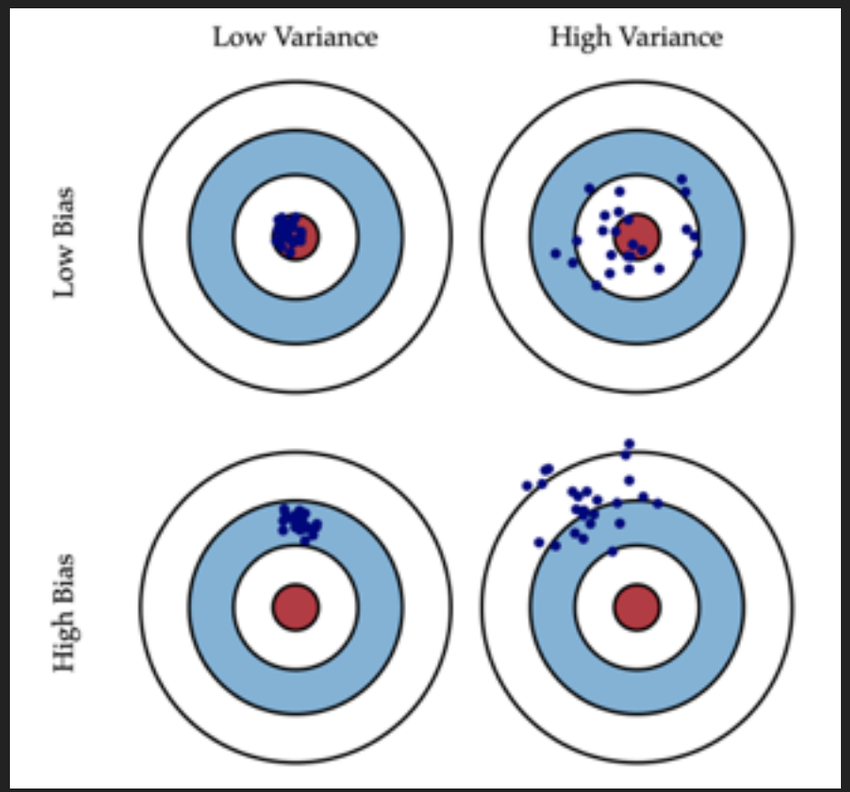
Averaging over x gives:

Bias

Variance

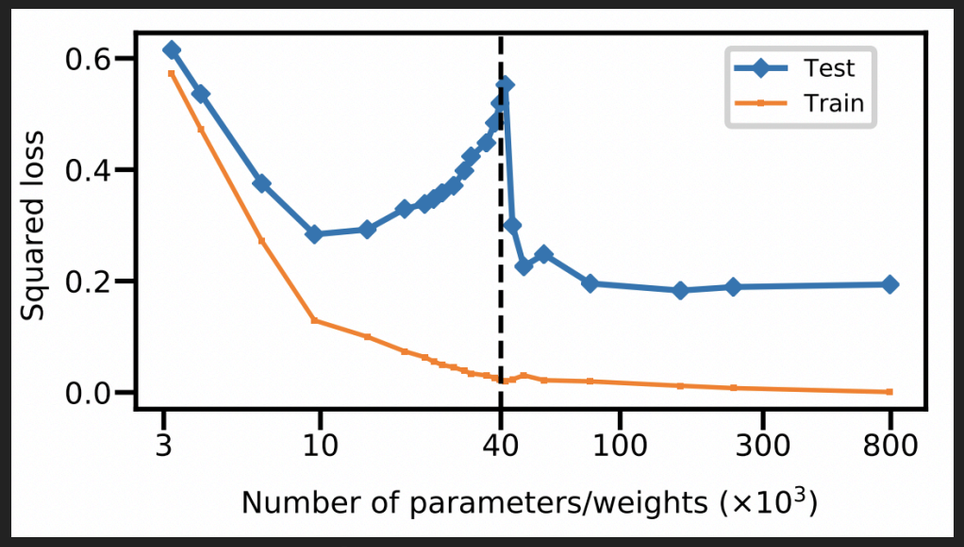
Noise



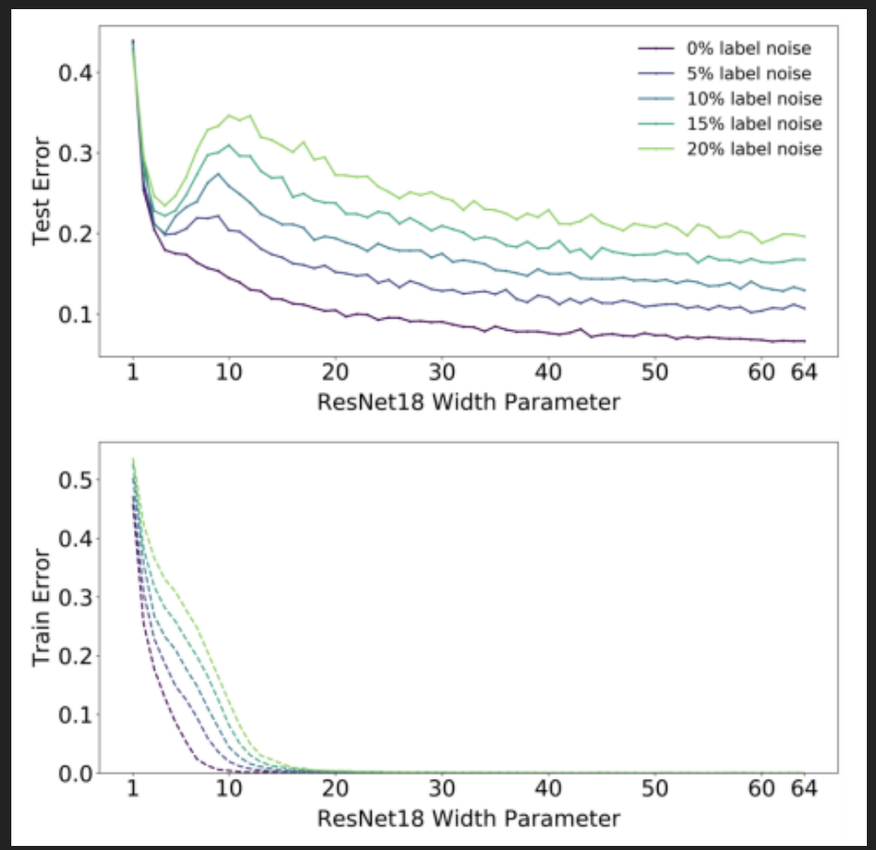


DOUBLE DESCENT

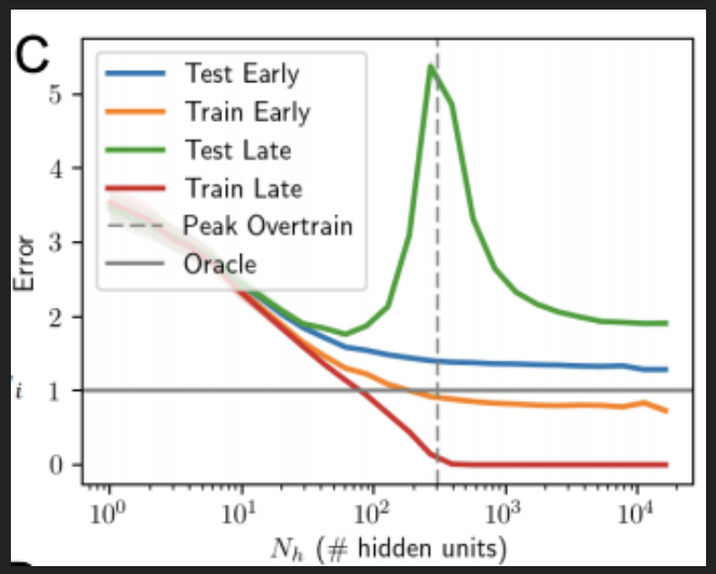
However, large NNs do not seem to obey the classical theory:



Belkin *et al.*, 2018



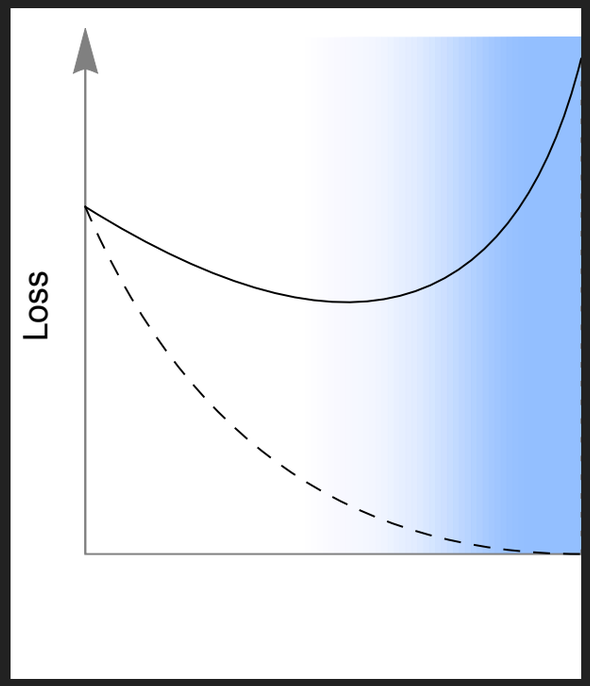
Nakkiran *et al.*, 2019



Advani and Saxe, 2017

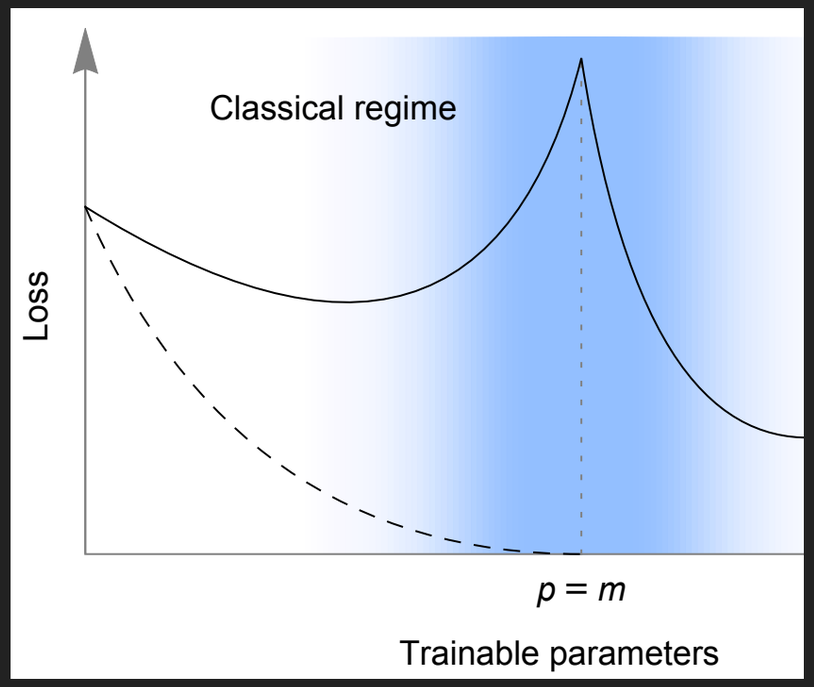
DOUBLE DESCENT

However, large NNs do not seem to obey the classical theory:



DOUBLE DESCENT

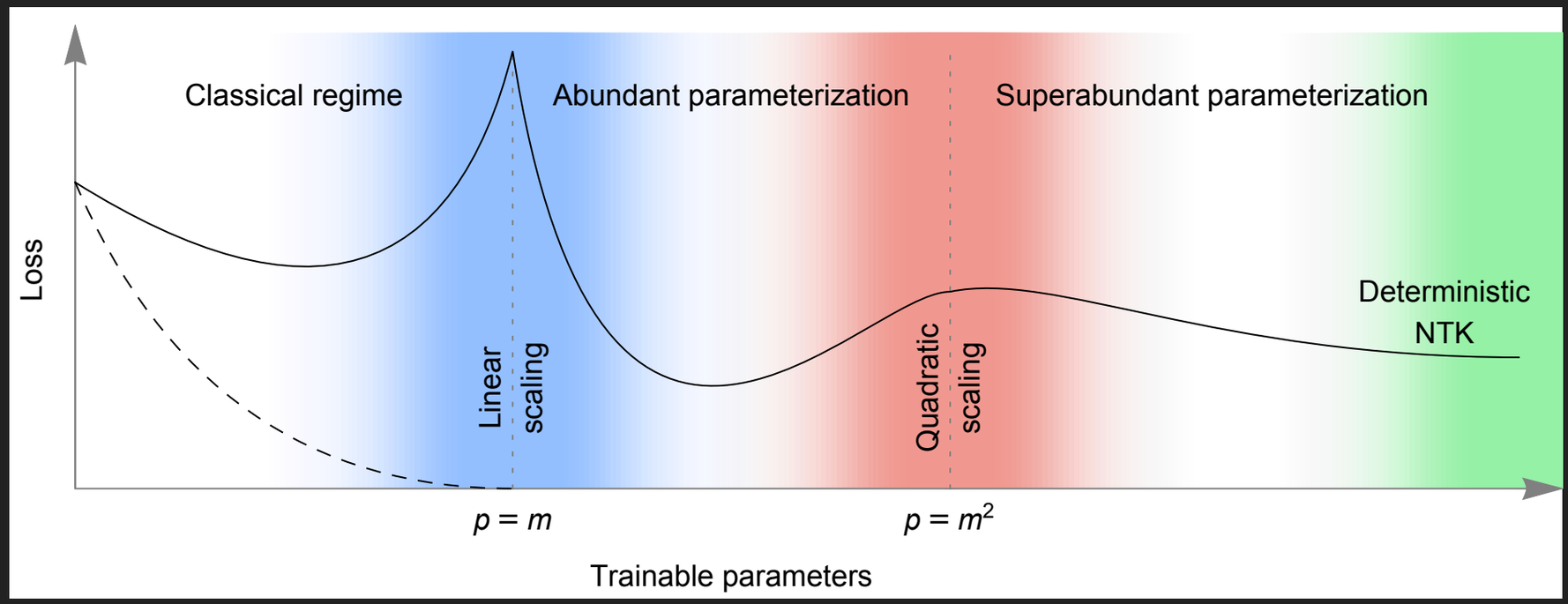
The emerging paradigm of *double descent* offers some insight:



Part I: What is causing the peak? What about the bias/variance tradeoff?

TRIPLE DESCENT

In some scenarios, NNs can exhibit non-monotonic behavior deep in the overparameterized regime



Part II: Triple descent and multi-scale theory of generalization

OUTLINE

1. Background
2. Problem Setup and Motivation
3. Part I: What Causes Double Descent?
4. Part II: Beyond Double Descent

DEEP LEARNING MODELS ARE HIGH-DIMENSIONAL

Deep learning models employ large numbers of parameters.
At least two practically-relevant high-dimensional regimes:

1. Quadratic overparameterization ($n_l \sim m$, i.e. $p \sim m^2$)
2. Linear overparameterization ($p \sim m$)

Examples:

	Width n_l	# Samples m	# Parameters p
FC/ CIFAR-10	10^3	10^4	10^6
ResNet/ ImageNet	10^3	10^7	10^8

HIGH-DIMENSIONAL SCALING LIMITS

We will focus on the following high-dimensional asymptotics of one hidden-layer networks:

1. Dataset size $m \to \infty$
2. Input dimensionality $n_0 \to \infty$
3. Hidden-layer size $n_1 \to \infty$

with the ratios $\phi = n_0/m$ and $\psi = n_0/n_1$ held constant

HIGH-DIMENSIONAL SCALING LIMITS

We will focus on the following high-dimensional asymptotics of one hidden-layer networks:

1. Dataset size $m \to \infty$
2. Input dimensionality $n_0 \to \infty$
3. Hidden-layer size $n_1 \to \infty$

with the ratios $\phi = n_0/m$ and $\psi = n_0/n_1$ held constant

I) For random features F, $p = n_1$, i.e. linear overparameterization

HIGH-DIMENSIONAL SCALING LIMITS

We will focus on the following high-dimensional asymptotics of one hidden-layer networks:

1. Dataset size $m \to \infty$
2. Input dimensionality $n_0 \to \infty$
3. Hidden-layer size $n_1 \to \infty$

with the ratios $\phi = n_0/m$ and $\psi = n_0/n_1$ held constant

I) For random features F, $p = n_1$, i.e. linear overparameterization

II) For NTK features J, $p = n_1(n_0 + 1)$, i.e. quadratic overparameterization

RANDOM FEATURE KERNEL RIDGE REGRESSION

Basic linear regression cannot capture double descent

- Data dimensionality tied to number of parameters ($p = n_0$)

RANDOM FEATURE KERNEL RIDGE REGRESSION

Basic linear regression cannot capture double descent

- Data dimensionality tied to number of parameters ($p = n_0$)

Random feature kernel regression can capture double descent

RANDOM FEATURE KERNEL RIDGE REGRESSION

Basic linear regression cannot capture double descent

- Data dimensionality tied to number of parameters ($p = n_0$)

Random feature kernel regression can capture double descent

- Linear regression on random features matrix $F = f(W_1 X)$

RANDOM FEATURE KERNEL RIDGE REGRESSION

Basic linear regression cannot capture double descent

- Data dimensionality tied to number of parameters ($p = n_0$)

Random feature kernel regression can capture double descent

- Linear regression on random features matrix $F = f(W_1 X)$
  - $F = X_1$, i.e. first post-activation matrix of NN at init
  - $W_1 \in \mathbb{R}^{n_1 \times n_0}$, $[W_1]_{ij} \sim \mathcal{N}(0, 1/n_0)$

RANDOM FEATURE KERNEL RIDGE REGRESSION

Basic linear regression cannot capture double descent

- Data dimensionality tied to number of parameters ($p = n_0$)

Random feature kernel regression can capture double descent

- Linear regression on random features matrix $F = f(W_1 X)$
  - $F = X_1$, i.e. first post-activation matrix of NN at init
  - $W_1 \in \mathbb{R}^{n_1 \times n_0}$, $[W_1]_{ij} \sim \mathcal{N}(0, 1/n_0)$
  - Equivalent to training only the top layer of single-layer NN

RANDOM FEATURE KERNEL RIDGE REGRESSION

Basic linear regression cannot capture double descent

- Data dimensionality tied to number of parameters ($p = n_0$)

Random feature kernel regression can capture double descent

- Linear regression on random features matrix $F = f(W_1 X)$
  - $F = X_1$, i.e. first post-activation matrix of NN at init
  - $W_1 \in \mathbb{R}^{n_1 \times n_0}$, $[W_1]_{ij} \sim \mathcal{N}(0, 1/n_0)$
  - Equivalent to training only the top layer of single-layer NN
  - For simple data, test error can computed asymptotically:
    - $X_{ij} \sim \mathcal{N}(0,1)$, $Y = \beta^\top X + \varepsilon$
    - $n_0, n_1, m \to \infty$ with $\phi \equiv n_0/m$, $\psi = n_0/n_1$ held constant

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$L = \|WF - Y\|_F^2 + \gamma \|W\|_F^2, \qquad Y = \beta^T X + \epsilon \qquad F = f(W_1 X)$$

$$W^* = YQF^T, \qquad Q = (K + \gamma I)^{-1}, \qquad K = \frac{1}{n_1} F^T F$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$L = \|WF - Y\|_F^2 + \gamma\|W\|_F^2, \qquad Y = \beta^T X + \epsilon \qquad F = f(W_1 X)$$

$$W^* = YQF^T, \qquad Q = (K + \gamma I)^{-1}, \qquad K = \frac{1}{n_1} F^T F$$

Consider an unseen test point $\tilde{x}$, with random features $\tilde{f} = f(W_1\tilde{x})$ and targets $\tilde{y} = \beta^T\tilde{x}$.

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$L = \|WF - Y\|_F^2 + \gamma\|W\|_F^2, \qquad Y = \beta^T X + \epsilon \qquad F = f(W_1 X)$$

$$W^* = YQF^T, \qquad Q = (K + \gamma I)^{-1}, \qquad K = \frac{1}{n_1} F^T F$$

Consider an unseen test point $\tilde{x}$, with random features $\tilde{f} = f(W_1\tilde{x})$ and targets $\tilde{y} = \beta^T\tilde{x}$.

$$E_{test} = \mathbb{E}_{\tilde{x}} \|W^*\tilde{f} - \tilde{y}\|_F^2$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$L = \|WF - Y\|_F^2 + \gamma\|W\|_F^2, \qquad Y = \beta^T X + \epsilon \qquad F = f(W_1 X)$$

$$W^* = YQF^T, \qquad Q = (K + \gamma I)^{-1}, \qquad K = \frac{1}{n_1} F^T F$$

Consider an unseen test point $\tilde{x}$, with random features $\tilde{f} = f(W_1\tilde{x})$ and targets $\tilde{y} = \beta^T\tilde{x}$.

$$\begin{aligned} E_{test} &= \mathbb{E}_{\tilde{x}} \|W^*\tilde{f} - \tilde{y}\|_F^2 \\ &= \mathbb{E}_{\tilde{x}}\, tr[(YQF^T\tilde{f} - \tilde{y})^T(YQF^T\tilde{f} - \tilde{y})] \\ &= \mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T FQY^T YQF^T\tilde{f}] - 2\mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T FQY^T\tilde{y}] + \mathbb{E}_{\tilde{x}}\, tr[\tilde{y}^T\tilde{y}] \end{aligned}$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$L = \|WF - Y\|_F^2 + \gamma\|W\|_F^2, \qquad Y = \beta^T X + \epsilon \qquad F = f(W_1 X)$$

$$W^* = YQF^T, \qquad Q = (K + \gamma I)^{-1}, \qquad K = \frac{1}{n_1} F^T F$$

Consider an unseen test point $\tilde{x}$, with random features $\tilde{f} = f(W_1\tilde{x})$ and targets $\tilde{y} = \beta^T\tilde{x}$.

$$\begin{aligned} E_{test} &= \mathbb{E}_{\tilde{x}} \|W^*\tilde{f} - \tilde{y}\|_F^2 \\ &= \mathbb{E}_{\tilde{x}}\, tr[(YQF^T\tilde{f} - \tilde{y})^T(YQF^T\tilde{f} - \tilde{y})] \\ &= \mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T FQY^T YQF^T\tilde{f}] - 2\mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T FQY^T\tilde{y}] + \mathbb{E}_{\tilde{x}}\, tr[\tilde{y}^T\tilde{y}] \end{aligned}$$

How to simplify these terms?

- Complex dependence on W_1, X, F

GAUSSIAN EQUIVALENTS

Asymptotic universality – can replace $F = f(WX)$ by another matrix such that all relevant first and second moments are preserved

$$F \simeq F^{lin} \equiv \sqrt{\zeta} WX + \sqrt{\eta - \zeta}\, A$$

$$\eta = \int dz\, \frac{e^{-z^2/2}}{\sqrt{2\pi}} f(\sigma_w \sigma_x z)^2 \qquad \zeta = \left[\sigma_w \sigma_x \int dz\, \frac{e^{-z^2/2}}{\sqrt{2\pi}} f'(\sigma_w \sigma_x z) \right]^2 \qquad A_{ij} \sim \mathcal{N}(0,1)$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$E_{test} = \mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T F Q Y^T Y Q F^T \tilde{f}] - 2\mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T F Q Y^T \tilde{y}] + \mathbb{E}_{\tilde{x}}\, tr[\tilde{y}^T \tilde{y}]$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$E_{test} = \mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T F Q Y^T Y Q F^T \tilde{f}] - 2\mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T F Q Y^T \tilde{y}] + \mathbb{E}_{\tilde{x}}\, tr[\tilde{y}^T \tilde{y}]$$

To simplify, apply the linearizations:

$$F \to F^{lin} \equiv \sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A \qquad \tilde{f} \to \tilde{f}^{lin} \equiv \sqrt{\zeta} W_1 \tilde{x} + \sqrt{\eta - \zeta}\, \tilde{a}$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

$$E_{test} = \mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T F Q Y^T Y Q F^T \tilde{f}] - 2\mathbb{E}_{\tilde{x}}\, tr[\tilde{f}^T F Q Y^T \tilde{y}] + \mathbb{E}_{\tilde{x}}\, tr[\tilde{y}^T \tilde{y}]$$

To simplify, apply the linearizations:

$$F \to F^{lin} \equiv \sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A \qquad \tilde{f} \to \tilde{f}^{lin} \equiv \sqrt{\zeta} W_1 \tilde{x} + \sqrt{\eta - \zeta}\, \tilde{a}$$

The expectations over $\beta, \varepsilon, \tilde{a}, \tilde{x}$ are trivial because

$$Q \to ((F^{lin})^T F^{lin} + \gamma I)^{-1} = \left((\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)^T (\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)\right)^{-1}$$

depends only on W_1, X, A.

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

After applying linearization and performing the trivial expectations, the result can be written as

$$E_{test} = \sum_i tr[R_i Q S_i Q] + \sum_i tr[T_i Q]$$

where R_i, S_i, T_i are low-order polynomials in W_1, X, A, and

$$Q = \left((\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)^T (\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A) \right)^{-1}$$

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

After applying linearization and performing the trivial expectations, the result can be written as

$$E_{test} = \sum_i tr[R_i Q S_i Q] + \sum_i tr[T_i Q]$$

where R_i, S_i, T_i are low-order polynomials in W_1, X, A, and

$$Q = \left((\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)^T (\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)\right)^{-1}$$

Q: How to evaluate the trace of a *rational function* of random matrices?

KERNEL RIDGE REGRESSION IN HIGH DIMENSIONS

After applying linearization and performing the trivial expectations, the result can be written as

$$E_{test} = \sum_i tr[R_i Q S_i Q] + \sum_i tr[T_i Q]$$

where R_i, S_i, T_i are low-order polynomials in W_1, X, A, and

$$Q = \left((\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)^T (\sqrt{\zeta} W_1 X + \sqrt{\eta - \zeta}\, A)\right)^{-1}$$

Q: How to evaluate the trace of a *rational function* of random matrices?

A: Linear pencil + operator-valued free probability

GENERALIZATION ERROR

$$K = \frac{1}{n_1} F^\top F + \gamma I$$

Lemma 1. *Let* $\eta = \mathbb{E}[\sigma(g)^2]$ *and* $\zeta = (\mathbb{E}[g\sigma(g)])^2$ *for* $g \sim \mathcal{N}(0,1)$. *Then, in the high-dimensional asymptotics defined above, the traces* $\tau_1(\gamma) = \frac{1}{m}\mathbb{E}\,\mathrm{tr}(K^{-1})$ *and* $\tau_2(\gamma) = \frac{1}{m}\mathbb{E}\,\mathrm{tr}(\frac{1}{n_0} X^\top X K^{-1})$ *are given by the unique solutions to the coupled polynomial equations,*

$$\zeta\tau_1\tau_2\left(1-\gamma\tau_1\right) = \phi/\psi\left(\zeta\tau_1\tau_2 + \phi(\tau_2-\tau_1)\right) = \left(\tau_1-\tau_2\right)\phi\left((\eta-\zeta)\tau_1 + \zeta\tau_2\right)\,, \tag{21}$$

such that $\tau_1, \tau_2 \in \mathbb{C}^+$ *for* $\gamma \in \mathbb{C}^+$.

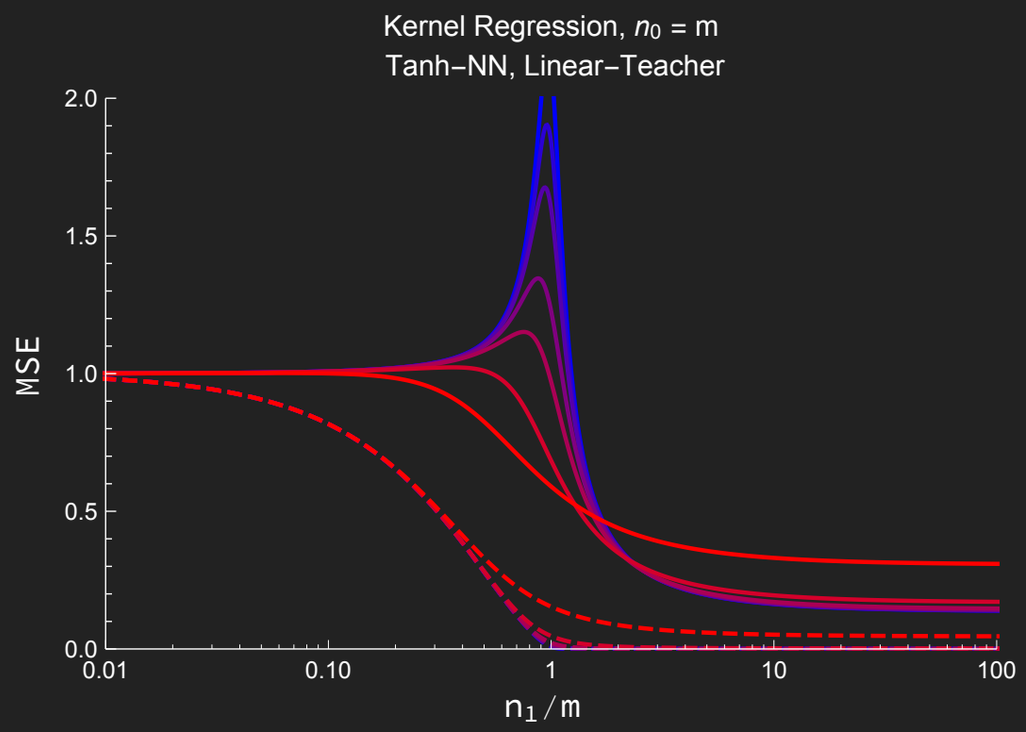
Theorem 1. *Let* $\gamma = Re(z)$ *and let* τ_1 *and* τ_2 *be defined as in Lemma 1 with* $Im(z) \to 0^+$. *Then the asymptotic training error* $E_{train} = \frac{1}{m}\mathbb{E}\|Y - \hat{y}(X)\|_F^2$ *is given by,*

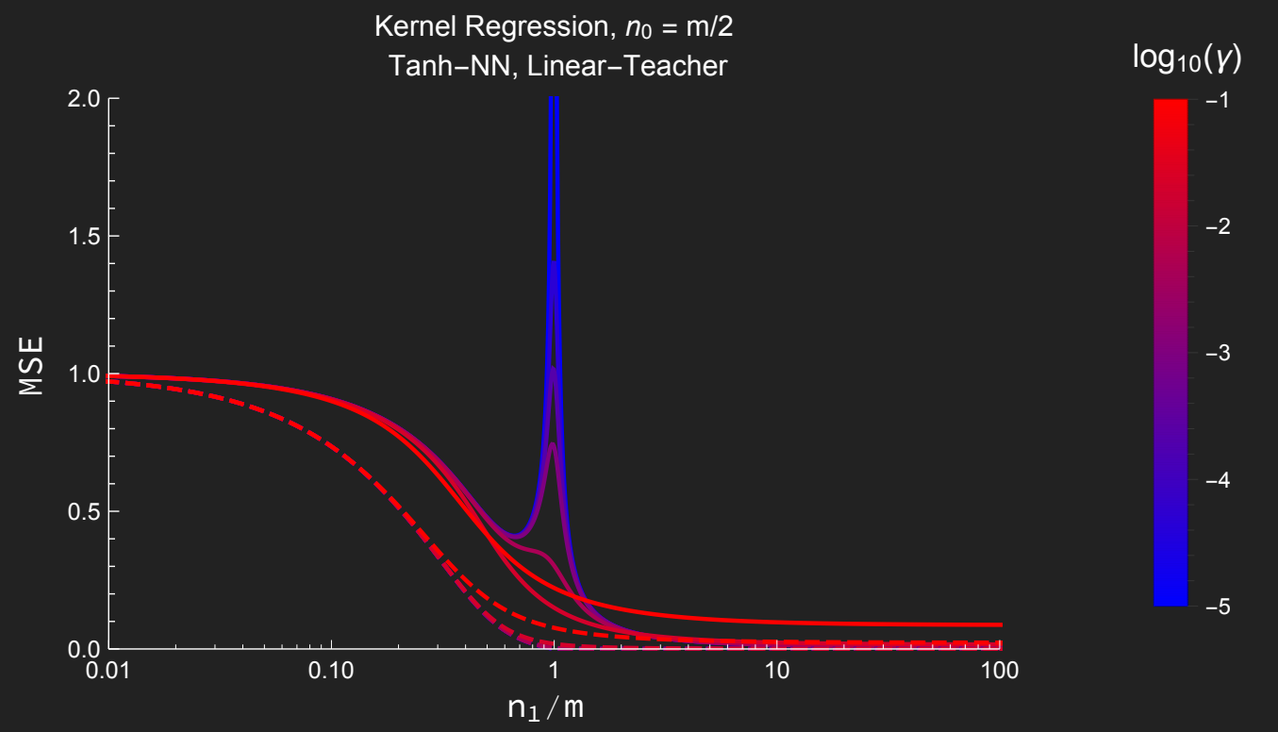
$$E_{train} = -\gamma^2(\sigma_\varepsilon^2\tau_1' + \tau_2')\,, \tag{22}$$

and the asymptotic test error $E_{test} = \mathbb{E}(y - \hat{y}(\mathbf{x}))^2$ *is given by*

$$E_{test} = (\gamma\tau_1)^{-2}E_{train} - \sigma_\varepsilon^2\,. \tag{23}$$

GENERALIZATION ERROR





OUTLINE

1. Background
2. Problem Setup and Motivation
3. Part I: What Causes Double Descent?
4. Part II: Beyond Double Descent

BIAS–VARIANCE DECOMPOSITION

Recall that the per-sample loss can be decomposed as

$$\mathbb{E}[(\hat{y}(x) - y(x))^2] = (\mathbb{E}\hat{y}(x) - \mathbb{E}y(x))^2 + \mathbb{V}[\hat{y}(x)] + \mathbb{V}[y(x)]$$

If the additive label noise ε on the training points is the only randomness, then we can use the classical decomposition:

$$E_{test} = \mathbb{E}_x\mathbb{E}_\varepsilon\left[\hat{y}(x) - y(x)\right]^2 = \underbrace{\mathbb{E}_x\mathbb{V}_\varepsilon\left[\hat{y}(x)\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_x(\mathbb{E}_\varepsilon[\hat{y}(x)] - y(x)^2)}_{\text{Bias}}$$

BIAS–VARIANCE DECOMPOSITION

Recall that the per-sample loss can be decomposed as

$$\mathbb{E}[(\hat{y}(x) - y(x))^2] = (\mathbb{E}\hat{y}(x) - \mathbb{E}y(x))^2 + \mathbb{V}[\hat{y}(x)] + \mathbb{V}[y(x)]$$

If the additive label noise ε on the training points is the only randomness, then we can use the classical decomposition:

$$E_{test} = \mathbb{E}_x\mathbb{E}_\varepsilon\left[\hat{y}(x) - y(x)\right]^2 = \underbrace{\mathbb{E}_x\mathbb{V}_\varepsilon\left[\hat{y}(x)\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_x(\mathbb{E}_\varepsilon[\hat{y}(x)] - y(x)^2)}_{\text{Bias}}$$

But what about any other sources of randomness ($\Theta = \{W_1, X\}$)?

MULTIVARIATE APPROACH: SYMMETRIC DECOMPOSITION

There is a unique way to symmetrically decompose the variance:

Proposition 1. *Let $X_1, \ldots, X_K$, and Y be random variables and $\mathcal{X} := \{X_1, \ldots, X_K\}$. We define a* variance decomposition *of Y to be a multiset $\{V_1, \ldots, V_N\}$ of nonnegative real numbers such that $\mathbb{V}[Y] = \sum_i V_i$. Then there exists a unique variance decomposition $\mathcal{V} := \{V_s : s \subseteq \mathcal{X}\}$ such that $\mathcal{V}$ is invariant under permutations of $\mathcal{X}$, and such that for all $S \subseteq \mathcal{X}$ the marginal variances satisfy the subset-sum relation,*

$$\mathbb{VE}[Y|X_j \text{ for } j \in S] = \sum_{s \subseteq S} V_s \,. \tag{5}$$

Example 1. *Consider the case of two random variables, the parameters P and the data D. Then $\mathcal{X} = \{P, D\}$ and the decomposition satisfying Proposition 1 is given by,*

$$V_P := \mathbb{E}_{\mathbf{x}}\mathbb{VE}[\hat{y}|P] \tag{6}$$

$$V_D := \mathbb{E}_{\mathbf{x}}\mathbb{VE}[\hat{y}|D] \tag{7}$$

$$V_{PD} := \mathbb{E}_{\mathbf{x}}\mathbb{VE}[\hat{y}|P, D] - \mathbb{E}_{\mathbf{x}}\mathbb{VE}[\hat{y}|P] - \mathbb{E}_{\mathbf{x}}\mathbb{VE}[\hat{y}|D] \,. \tag{8}$$

We can interpret V_{PD} as the variance explained by the parameters and data together beyond what they explain individually.

MULTIVARIATE APPROACH: SYMMETRIC DECOMPOSITION

Example 2. *Further decomposing D into randomness from sampling the inputs X and label noise $\boldsymbol{\epsilon}$, we can write $\mathcal{X} = \{P, X, \boldsymbol{\epsilon}\}$ and the decomposition satisfying Proposition 1 is given by,*

$$V_X := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X], \tag{9}$$

$$V_{\boldsymbol{\epsilon}} := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|\boldsymbol{\epsilon}], \tag{10}$$

$$V_P := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P], \tag{11}$$

$$V_{X\boldsymbol{\epsilon}} := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X, \boldsymbol{\epsilon}] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|\boldsymbol{\epsilon}], \tag{12}$$

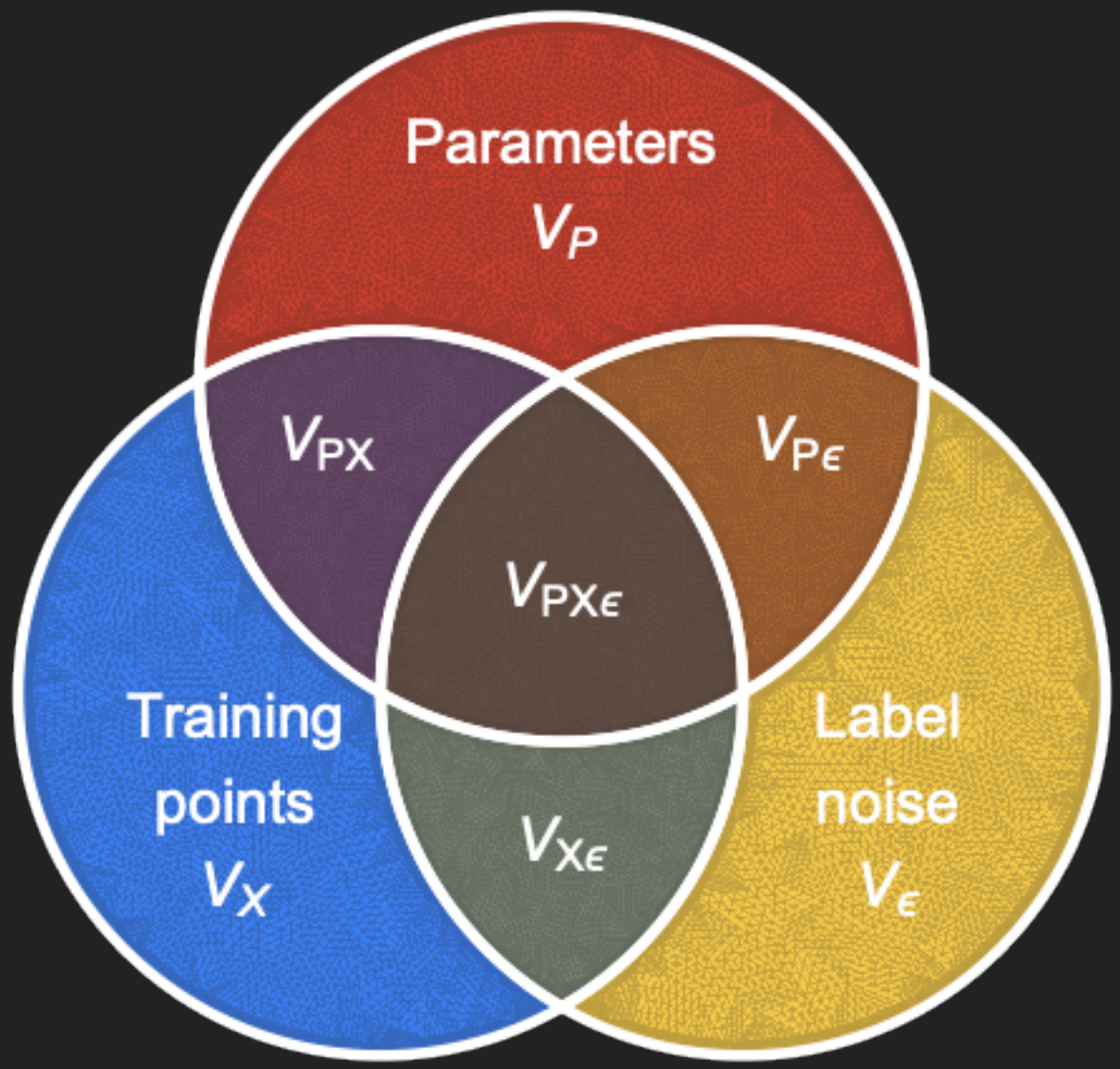
$$V_{PX} := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P, X] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P], \tag{13}$$

$$V_{P\boldsymbol{\epsilon}} := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X, \boldsymbol{\epsilon}] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|\boldsymbol{\epsilon}] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P], \tag{14}$$

$$\begin{aligned} V_{PX\boldsymbol{\epsilon}} := \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P, X, \boldsymbol{\epsilon}] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X, \boldsymbol{\epsilon}] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P, X] - \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X, \boldsymbol{\epsilon}] \\ + \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|X] + \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|\boldsymbol{\epsilon}] + \mathbb{E}_{\mathbf{x}}\mathbb{V}\mathbb{E}[\hat{y}|P]. \end{aligned} \tag{15}$$

MULTIVARIATE APPROACH: SYMMETRIC DECOMPOSITION

Remark 1. *Because* $V_s \geq 0$ *and* $V = \mathbb{V}[\hat{y}] = \sum_s V_s$, *the subset-sum relation (5) yields an interpretation of* V *as the union of disjoint areas, forming a Venn diagram.*



MULTIVARIATE APPROACH: SYMMETRIC DECOMPOSITION

Lemma 1. *Let* $\eta = \mathbb{E}[\sigma(g)^2]$ *and* $\zeta = (\mathbb{E}[g\sigma(g)])^2$ *for* $g \sim \mathcal{N}(0,1)$. *Then, in the high-dimensional asymptotics defined above, the traces* $\tau_1(\gamma) = \frac{1}{m}\operatorname{tr}(K^{-1})$ *and* $\tau_2(\gamma) = \frac{1}{m}\operatorname{tr}(\frac{1}{n_0}X^\top X K^{-1})$ *are given by the unique solutions to the coupled polynomial equations,*

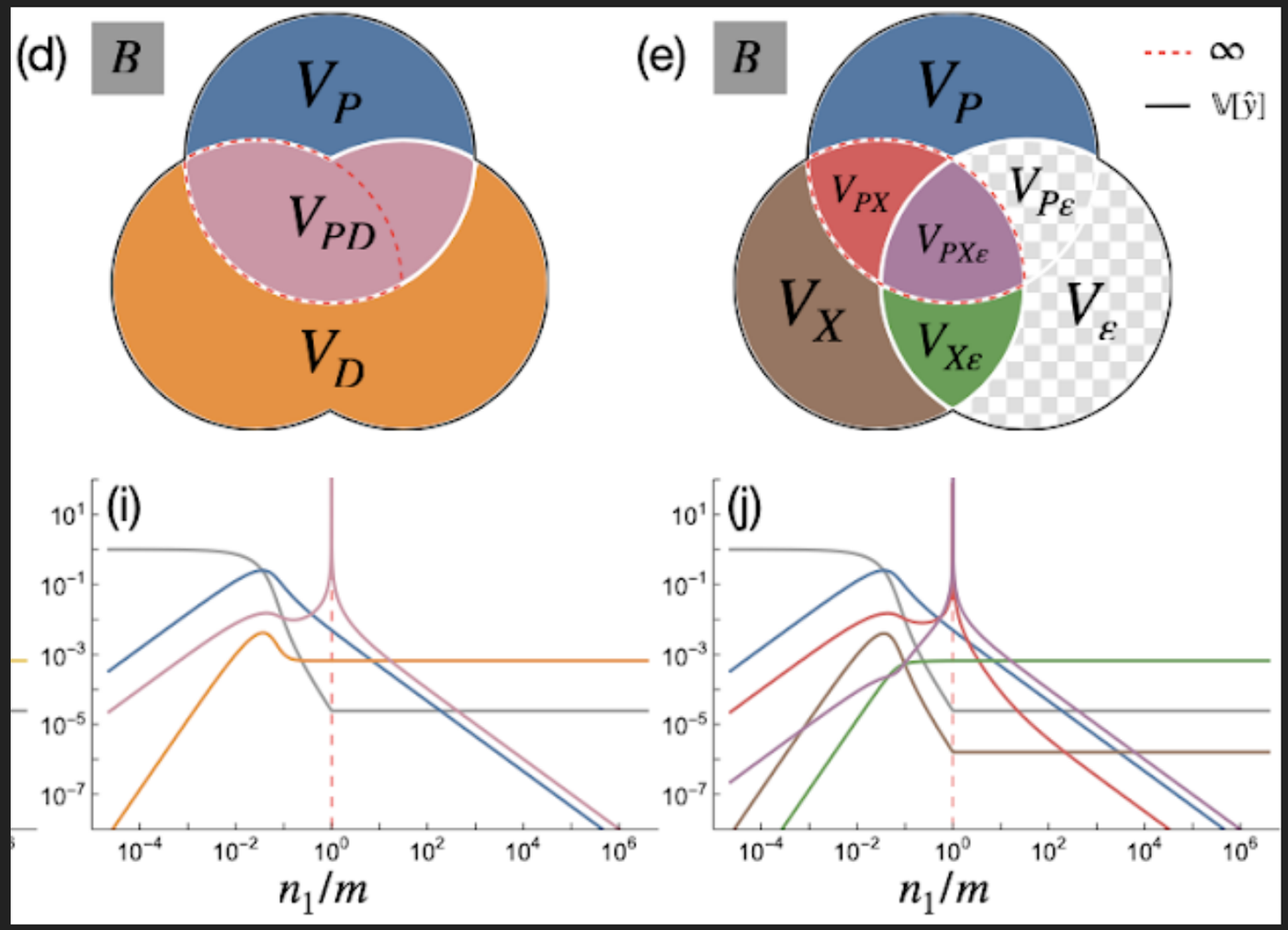
$$\zeta\tau_1\tau_2\left(1-\gamma\tau_1\right) = \phi/\psi\left(\zeta\tau_1\tau_2 + \phi(\tau_2-\tau_1)\right) = (\tau_1-\tau_2)\,\phi\left((\eta-\zeta)\tau_1 + \zeta\tau_2\right), \tag{21}$$

such that $\tau_1, \tau_2 \in \mathbb{C}^+$ *for* $\gamma \in \mathbb{C}^+$.

Theorem 1. *Let* τ_1 *and* τ_2 *be defined as in Lemma 1. Then the asymptotic bias and variance terms of eqns.* (9)-(15) *are given by,*

$$\begin{aligned}
B &= \tau_2^2/\tau_1^2 & V_{PX} &= -\tau_2'/\tau_1^2 - B - V_P - V_X \\
V_P &= \tau_2'/\tau_1' - B & V_{P\boldsymbol{\varepsilon}} &= 0 \\
V_X &= \phi B(\tau_1-\tau_2)^2/(\tau_1^2 - \phi(\tau_1-\tau_2)^2) & V_{X\boldsymbol{\varepsilon}} &= \sigma_{\boldsymbol{\varepsilon}}^2 V_X/B \\
V_{\boldsymbol{\varepsilon}} &= 0 & V_{PX\boldsymbol{\varepsilon}} &= \sigma_{\boldsymbol{\varepsilon}}^2(-\tau_1'/\tau_1^2 - 1) - V_{X\boldsymbol{\varepsilon}}\,.
\end{aligned} \tag{22}$$

A UNIFYING PERSPECTIVE ON BIAS–VARIANCE DECOMPOSITIONS



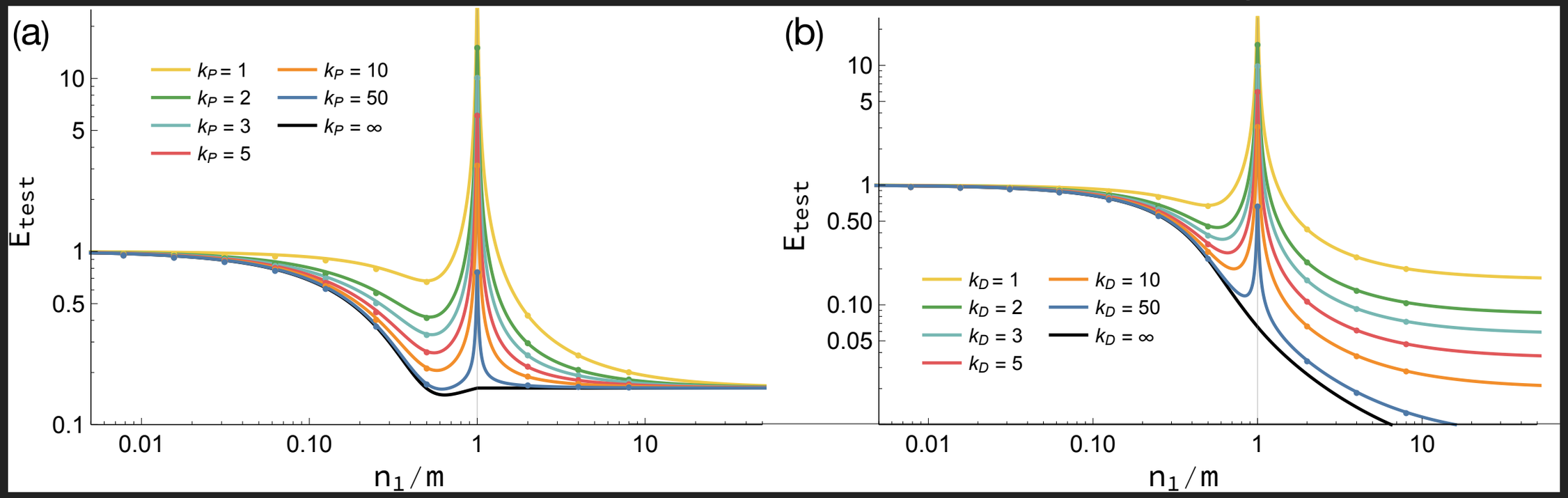
The symmetric decomposition reveals that the divergence comes from $V_{PD} = V_{PX} + V_{PX\varepsilon}$

- This the variance that is explained by the *interaction of the* parameters and the data
- Predicts that ensembling over parameters or samples will eliminate divergence

THE ORIGIN OF DOUBLE DESCENT

Parameter-ensemble of k_P different models with same data

Data-ensemble of k_D different models with same parameters



OUTLINE

1. Background
2. Problem Setup and Motivation
3. Part I: What Causes Double Descent?
4. Part II: Beyond Double Descent

FROM KERNELS TO NEURAL NETWORKS AND BACK AGAIN

Now consider a single-layer neural network in which all the parameters are trained, $N(x; \theta = \{W_1, W_2\}) = W_2 f(W_1 x)$

FROM KERNELS TO NEURAL NETWORKS AND BACK AGAIN

Now consider a single-layer neural network in which all the parameters are trained, $N(x; \theta = \{W_1, W_2\}) = W_2 f(W_1 x)$

As the width grows, parameters move less during the course of gradient descent, i.e. $\theta(t) \approx \theta(0)$

FROM KERNELS TO NEURAL NETWORKS AND BACK AGAIN

Now consider a single-layer neural network in which all the parameters are trained, $N(x; \theta = \{W_1, W_2\}) = W_2 f(W_1 x)$

As the width grows, parameters move less during the course of gradient descent, i.e. $\theta(t) \approx \theta(0)$

This motivates a linear approximation

$$N(x; \theta(t)) \approx N(x; \theta(0)) + \left.\frac{\partial N}{\partial \theta}\right|_{\theta=\theta(0)} (\theta(t) - \theta(0)) + \mathcal{O}(\theta(t) - \theta(0))^2$$

FROM KERNELS TO NEURAL NETWORKS AND BACK AGAIN

Now consider a single-layer neural network in which all the parameters are trained, $N(x; \theta = \{W_1, W_2\}) = W_2 f(W_1 x)$

As the width grows, parameters move less during the course of gradient descent, i.e. $\theta(t) \approx \theta(0)$

This motivates a linear approximation

$$N(x; \theta(t)) \approx N_0 + J_0(\theta(t) - \theta(0))$$

FROM KERNELS TO NEURAL NETWORKS AND BACK AGAIN

Now consider a single-layer neural network in which all the parameters are trained, $N(x; \theta = \{W_1, W_2\}) = W_2 f(W_1 x)$

As the width grows, parameters move less during the course of gradient descent, i.e. $\theta(t) \approx \theta(0)$

This motivates a linear approximation

$$N(x; \theta(t)) \approx N_0 + J_0(\theta(t) - \theta(0))$$

The dynamics are determined by the Neural Tangent Kernel Θ

$$\Theta = J_0^T J_0$$

NEURAL TANGENT KERNEL

The offset N_0 contributes unnecessary variance. Can set $N_0 = 0$ by subtracting two copies of the model with same initialization

$$N^{VR}(x; \{\theta_1, \theta_2\}) = \frac{1}{\sqrt{2}}\big(N(x;\theta_1) - N(x;\theta_2)\big) \qquad \theta_1(0) = \theta_2(0)$$

$$N_0^{VR} = 0, \quad \Theta^{VR} = \Theta$$

NEURAL TANGENT KERNEL

The NTK is a random feature model with features $J_0 \in \mathbb{R}^{p \times m}$.

NEURAL TANGENT KERNEL

The NTK is a random feature model with features $J_0 \in \mathbb{R}^{p \times m}$.

- Quadratically overparameterized ($p = n_0(n_1 + 1) \sim m^2$)

NEURAL TANGENT KERNEL

The NTK is a random feature model with features $J_0 \in \mathbb{R}^{p \times m}$.

- Quadratically overparameterized ($p = n_0(n_1 + 1) \sim m^2$)

Kernel decomposes into per-layer kernels $\Theta = J_0^T J_0 = \Theta_1 + \Theta_2$

NEURAL TANGENT KERNEL

The NTK is a random feature model with features $J_0 \in \mathbb{R}^{p \times m}$.

- Quadratically overparameterized ($p = n_0(n_1 + 1) \sim m^2$)

Kernel decomposes into per-layer kernels $\Theta = J_0^T J_0 = \Theta_1 + \Theta_2$

The second-layer kernel is what we studied earlier

- $\Theta_2 = K = F^T F$

NEURAL TANGENT KERNEL

The NTK is a random feature model with features $J_0 \in \mathbb{R}^{p \times m}$.

- Quadratically overparameterized ($p = n_0(n_1 + 1) \sim m^2$)

Kernel decomposes into per-layer kernels $\Theta = J_0^T J_0 = \Theta_1 + \Theta_2$

The second-layer kernel is what we studied earlier

- $\Theta_2 = K = F^T F$

The first layer kernel can be analyzed similarly

- $\Theta_1 = (F')^T D_{W_2} F' \odot X^T X, \qquad F' = f'(W_1 X)$

NEURAL TANGENT KERNEL

The NTK is a random feature model with features $J_0 \in \mathbb{R}^{p \times m}$.

- Quadratically overparameterized ($p = n_0(n_1 + 1) \sim m^2$)

Kernel decomposes into per-layer kernels $\Theta = J_0^T J_0 = \Theta_1 + \Theta_2$

The second-layer kernel is what we studied earlier

- $\Theta_2 = K = F^T F$

The first layer kernel can be analyzed similarly

- $\Theta_1 = (F')^T D_{W_2} F' \odot X^T X, \qquad F' = f'(W_1 X)$
- Only the mean of $(F')^T D_{W_2} F'$ contributes at leading order
- $\Theta_1 \simeq c_1 I + c_2 X^T X$

QUADRATIC OVERPARAMETERIZATION

Proposition 1. *As $n_0, n_1, m \to \infty$ with $\phi = n_0/m$ and $\psi = n_0/n_1$ fixed, the traces $\tau_1(z) := \frac{1}{m}\mathbb{E}\,\mathrm{tr}(K(z)^{-1})$ and $\tau_2(z) := \frac{1}{m}\mathbb{E}\,\mathrm{tr}(\frac{1}{n_0}X^\top X K(z)^{-1})$ are given by the unique solutions to the coupled polynomial equations,*

$$\begin{aligned}
&\phi\left(\zeta\tau_2\tau_1 + \phi(\tau_2-\tau_1)\right) + \zeta\tau_1\tau_2\psi\left(z\tau_1 - 1\right) \\
&\quad = -\zeta\tau_1\tau_2\sigma_{W_2}^2\left(\zeta\left(\tau_2-\tau_1\right)\psi + \tau_1\psi\eta' + \phi\right) \\
&\zeta\tau_1^2\tau_2\left(\eta'-\eta\right)\sigma_{W_2}^2 + \zeta\tau_1\tau_2\left(z\tau_1 - 1\right) \\
&\quad = \left(\tau_2-\tau_1\right)\phi\left(\zeta\left(\tau_2-\tau_1\right) + \eta\tau_1\right),
\end{aligned} \tag{25}$$

such that $\tau_1, \tau_2 \in \mathbb{C}^+$ for $z \in \mathbb{C}^+$.

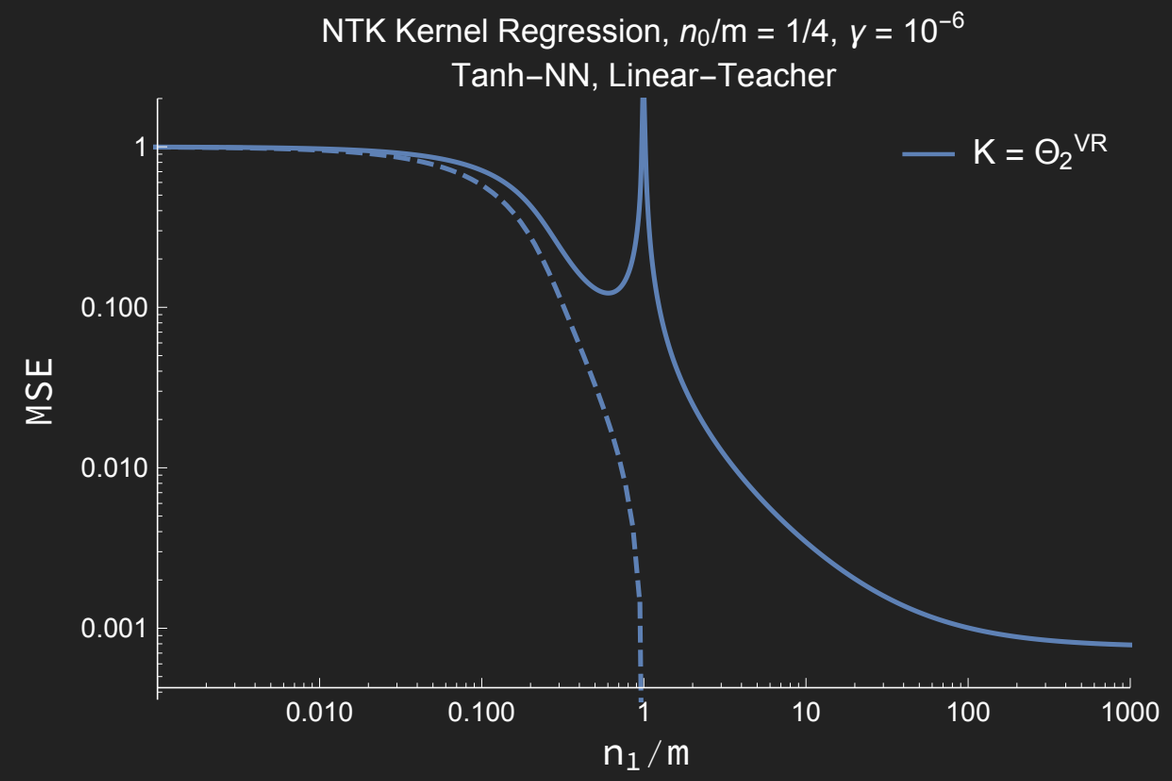
Theorem 1. *Let $\gamma = Re(z)$ and let τ_1 and τ_2 be defined as in Proposition 1 with $Im(z) \to 0^+$. Then the asymptotic training error $E_{train} = \frac{1}{m}\mathbb{E}\|Y - \hat{y}(X)\|_F^2$ is given by,*

$$\begin{aligned}
E_{train} = &-\gamma^2(\sigma_\varepsilon^2\tau_1' + \tau_2') + \nu\sigma_{W_2}^2\gamma^2(\tau_1 + \gamma\tau_1') \\
&+ \nu\sigma_{W_2}^4\gamma^2\left((\eta'-\zeta)\tau_1' + \zeta\tau_2'\right),
\end{aligned} \tag{26}$$

and the asymptotic test error $E_{test} = \mathbb{E}(y - \hat{y}(\mathbf{x}))^2$ is given by

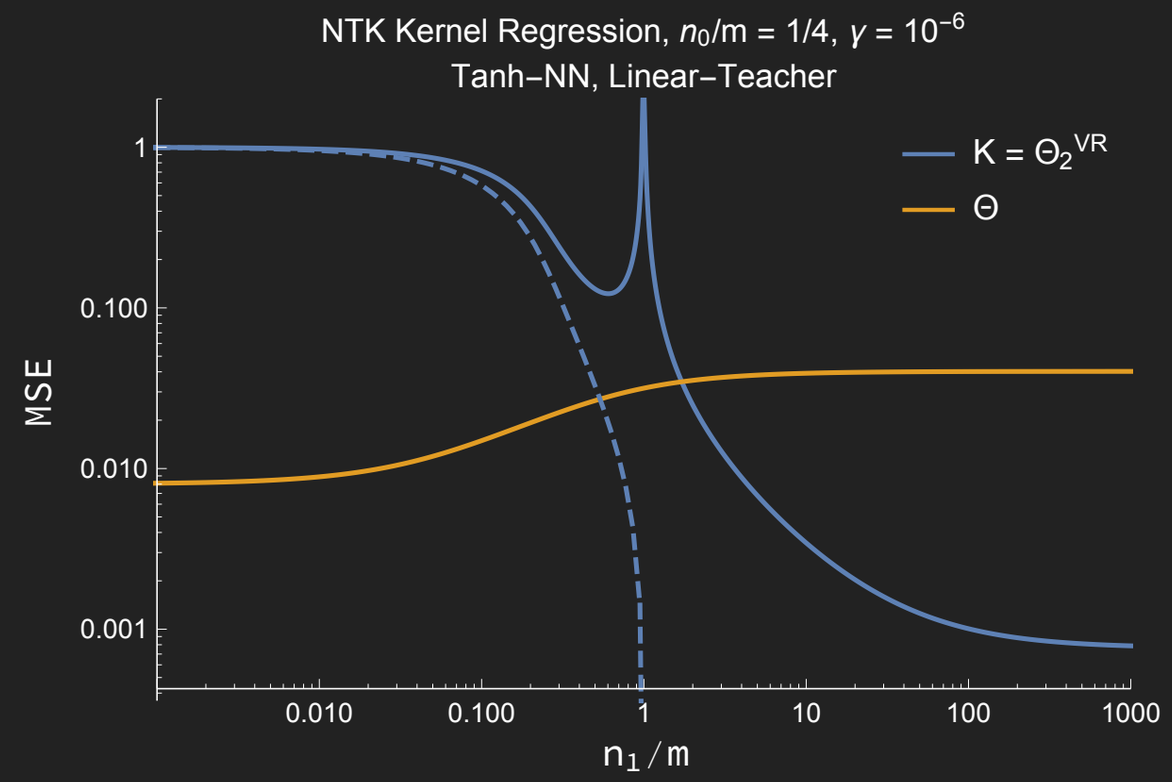
$$E_{test} = (\gamma\tau_1)^{-2}E_{train} - \sigma_\varepsilon^2\,. \tag{27}$$

QUADRATIC OVERPARAMETERIZATION



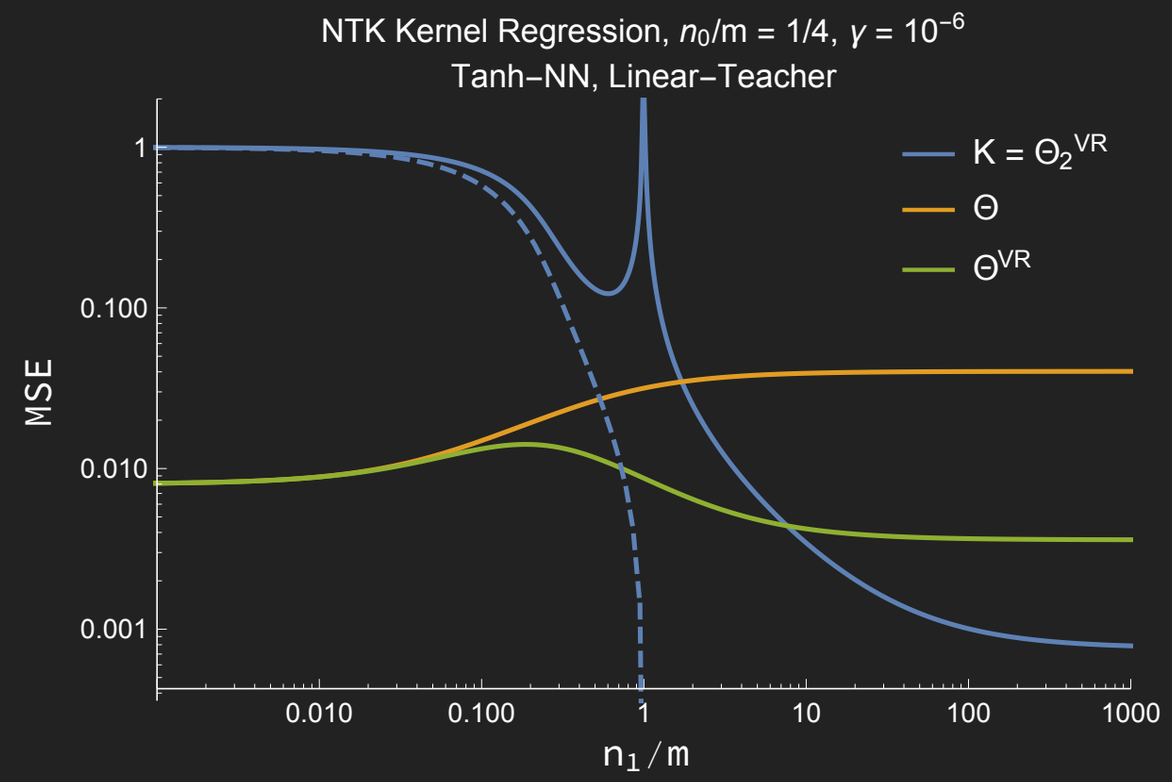
QUADRATIC OVERPARAMETERIZATION

The network can be too overparametrized

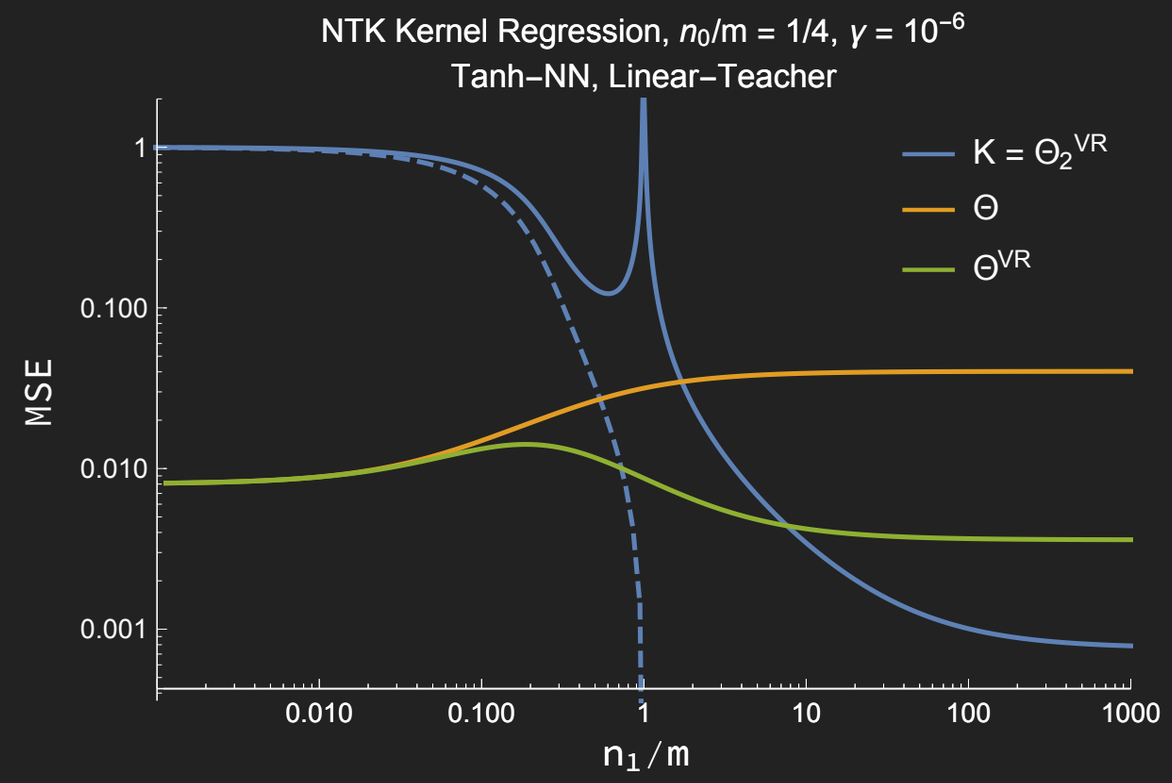


QUADRATIC OVERPARAMETERIZATION

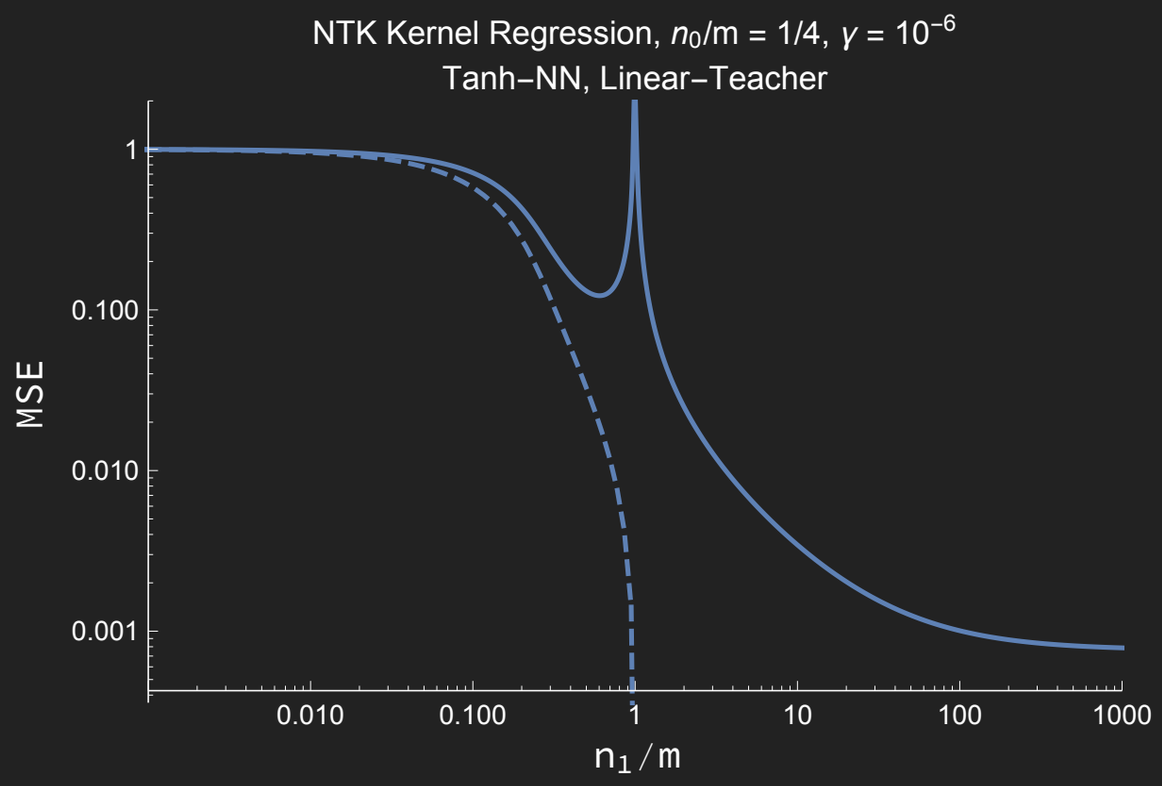
Reducing the variance helps, but a peak emerges

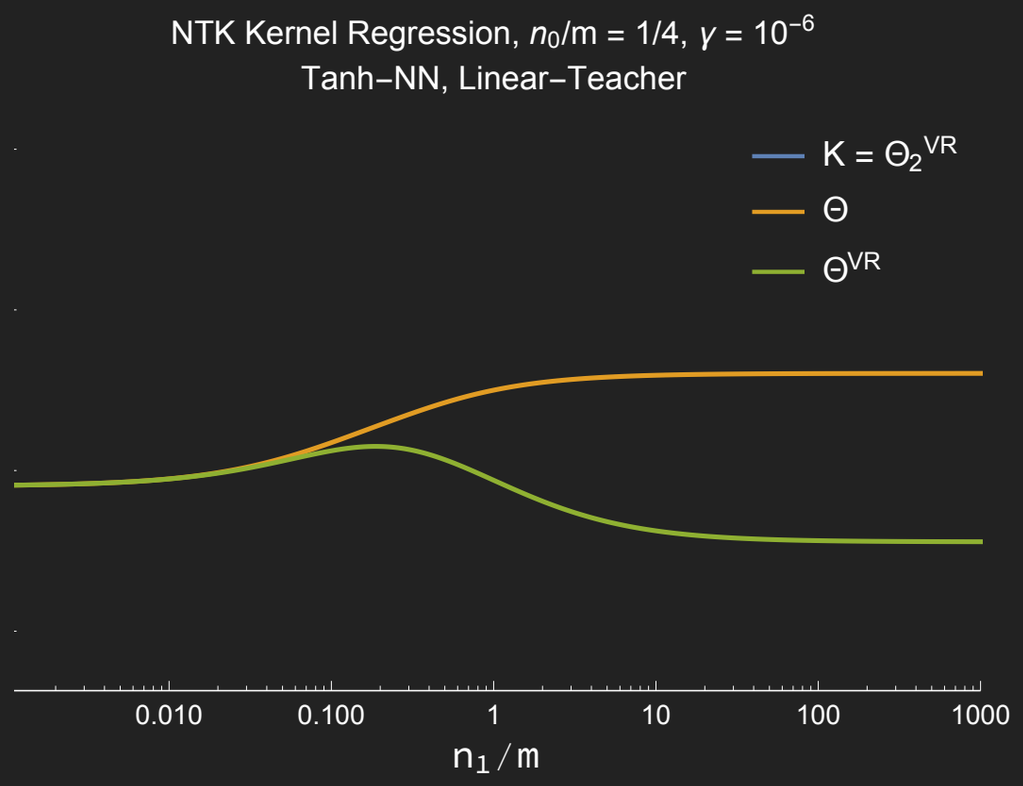


TWO OVERPARAMETERIZATION SCALES



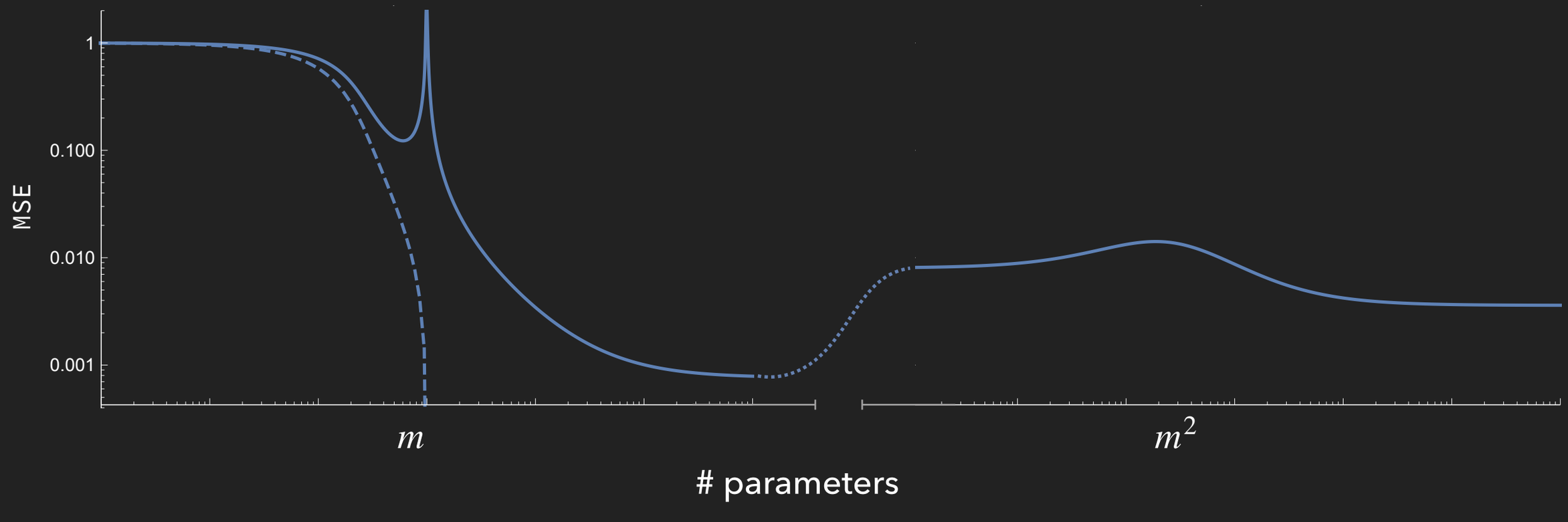
TWO OVERPARAMETERIZATION SCALES





TWO OVERPARAMETERIZATION SCALES

NTK Kernel Regression, $n_0/m = 1/4$, $\gamma = 10^{-6}$
Tanh–NN, Linear–Teacher

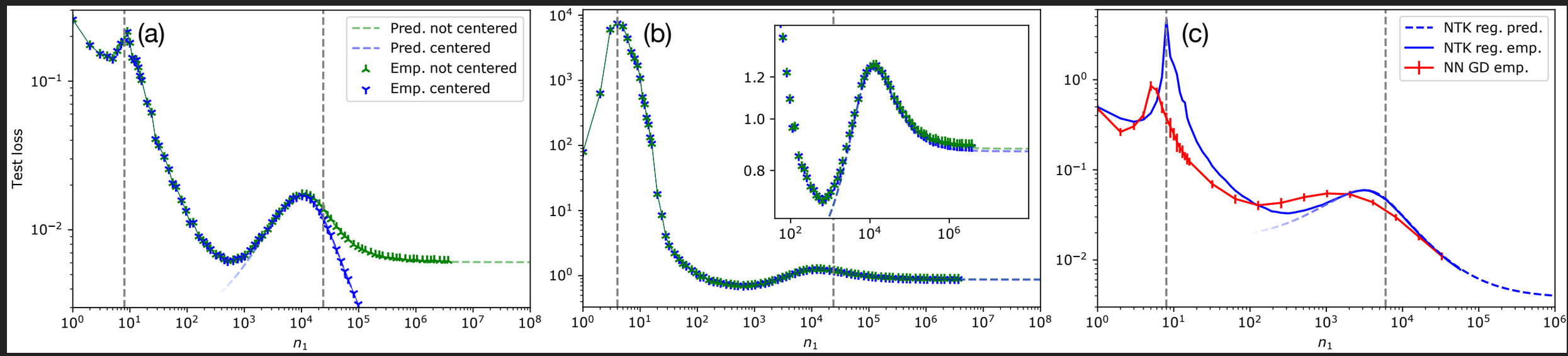


TWO OVERPARAMETERIZATION SCALES

Triple descent in finite-sized kernel regression models

Global minimum need not be at infinity

Qualitative picture holds for gradient descent as well



TWO OVERPARAMETERIZATION SCALES