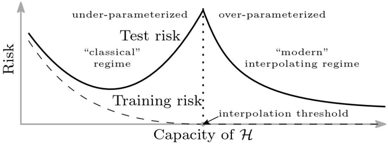


Yikun Zhang
Department of Statistics,
University of Washington
∗ Part of the slides were made when I was an
Advanced Algorithmic Engineer at Trip.com

Shall We Always Avoid Overfitting?

A generalized framework of the classical bias-variance trade-off in modern deep learning regime

November 4, 2021

Table of Contents

What is Overfitting?

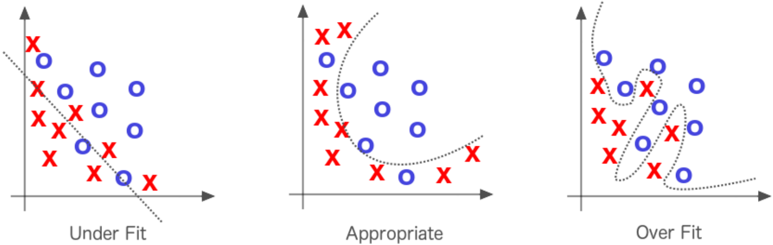


Figure 1: A binary classification problem with underfitting, just-right, and overfitting decision boundaries (or classifiers).

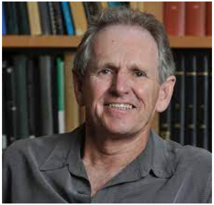
Survey: Is Overfitting Good or Bad?

How many of you think that overfitting is a *bad* phenomenon in machine learning practices and should be avoided?

[1] Cited from *The Element of Statistical Learning, Second Edition*, Hastie et al. (2009).

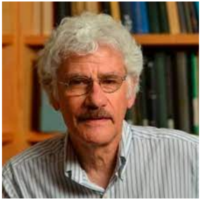
Survey: Is Overfitting Good or Bad?

How many of you think that overfitting is a *bad* phenomenon in machine learning practices and should be avoided?



(a) Trevor Hastie

(b) Robert Tibshirani



(c) Jerome H. Friedman

- "...Fitting the training data too well can lead to overfitting, which degrades the risk on future prediction." [1]

[1] Cited from *The Element of Statistical Learning, Second Edition*, Hastie et al. (2009).

Survey: Is Overfitting Good or Bad?

- "...With so many candidate models, **overfitting is a real danger. Is the monkey who typed Hamlet actually a good writer?"** [2]

Figure 3: A typing monkey (Image source: iStock).

[2] Cited from *Model Selection and Model Averaging*, Claeskens et al. (2008).

A Central Tenet in Machine Learning: Bias-Variance Trade-off

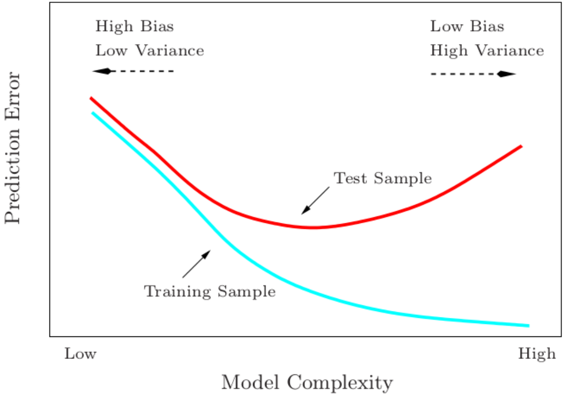


Figure 4: Training and test errors with respect to model complexity.

Practical Remedies for Overfitting

Take the neural network as an example:

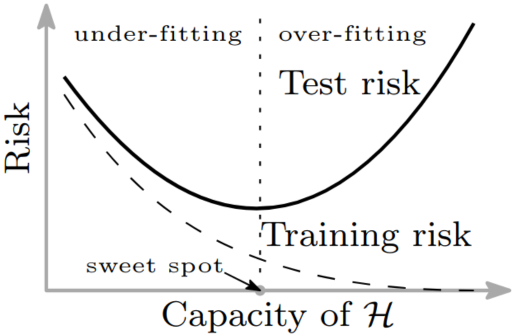
- Pick a simpler neural network architecture:
  1. Reducing the number of layers or neurons
  2. Dropout, weight decay, ...
- Regularization:
  1. Add penalized terms to the loss function

$$h_n = \arg\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(h(x_i), y_i\right)$$

$$\Longrightarrow \quad h_n = \arg\min_{h\in\mathcal{H}} \left[\frac{1}{n}\sum_{i=1}^{n} \ell\left(h(x_i), y_i\right) + \lambda||h||_{\mathcal{H}}\right].$$

  2. Early stopping,
  3. Stochastic gradient descent (implicit regularization), ...

Pursue a “Sweet Spot” Model



Training Risk: $\frac{1}{n}\sum_{i=1}^{n} \ell\left(h(x_i), y_i\right).$

Test Risk: $\mathbb{E}_{(x,y)\sim P}\left[\ell(h(x), y)\right].$

Assume that the data samples are sampled randomly from a probability distribution P.

So Far So Good...

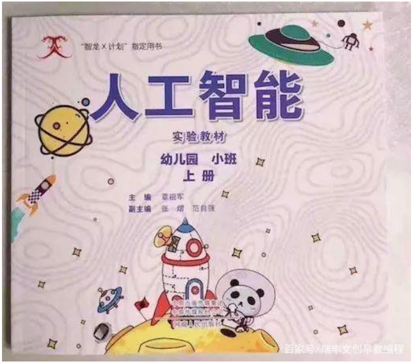
To be an artificial intelligence (AI) engineer, we only need to

1. Understand the bias-variance trade-off principle.
2. Know how to do computer programming.
3. Avoid overfitting with those standard techniques.

So Far So Good...

To be an artificial intelligence (AI) engineer, we only need to

1. Understand the bias-variance trade-off principle.
2. Know how to do computer programming.
3. Avoid overfitting with those standard techniques.



The picture is an AI textbook designed for children at the kindergarten level in China. There are some news saying that several kindergartens begin teaching computer programming to their 3-5 years-old kids.

Concerns

If a kindergarten kid is capable of tackling AI tasks, why do we need years of subsequent education and training?
A big waste of time?

Concerns

If a kindergarten kid is capable of tackling AI tasks, why do we need years of subsequent education and training?
A big waste of time?

There is something weird happening in real-world applications...

Success of Deep Learning

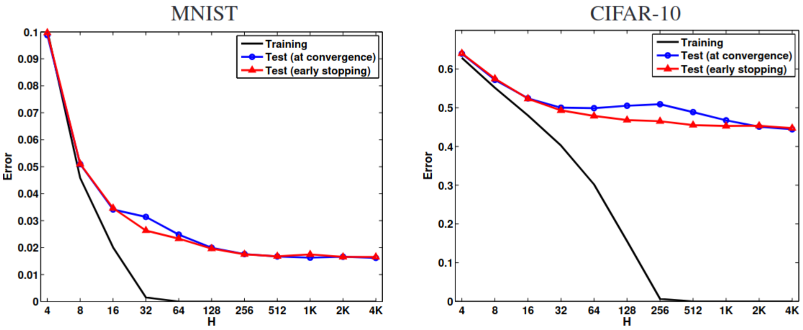
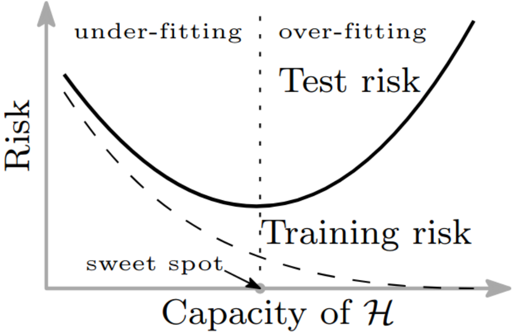


Figure 5: The training and test errors based on different stopping criteria when two-layer Neural Networks (NNs) with different number of hidden units H are trained on MNIST and CIFAR-10 data (Neyshabur et al., 2014).

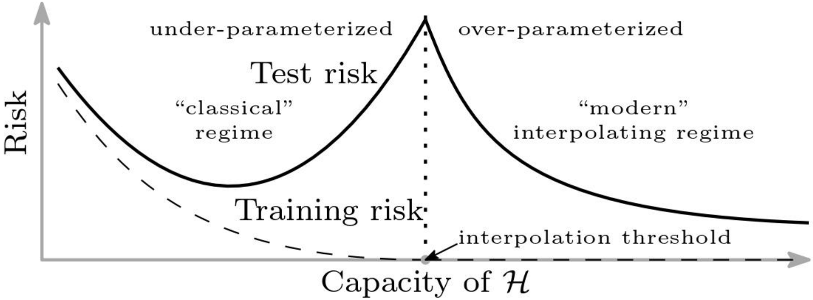
- Notes: The number of parameters/weights is $H(d+K)$ for each two-layer NN, where d is the number of features and K is the size of the output layer.

Deep Learning Paradox

The preceding evidence in modern deep learning regime indicates a contradiction to the classical bias-variance trade-off.



“Double-Descent“ Curve



An extension of the classical bias-variance trade-off.

Illustration of Interpolation

- Data: $Y_i = \sin(X_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, ..., 20$.
- Model (*cubic spline*): $h(X) = \sum_{k=0}^{3} a_k X^k + \sum_{j=1}^{N-3} b_j (X - \xi_j)_+^3$

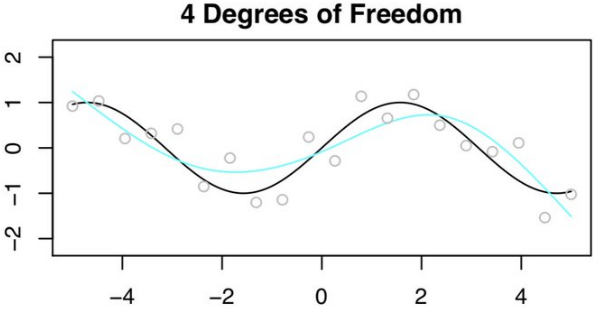


Figure 6: Fitting the true sine function (black) with cubic spline (cyan) (Image source: *Prof. Daniela Witten*'s Twitter).

Illustration of Interpolation

- Data: $Y_i = \sin(X_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, ..., 20$.
- Model (*cubic spline*): $h(X) = \sum_{k=0}^{3} a_k X^k + \sum_{j=1}^{N-3} b_j (X - \xi_j)_+^3$

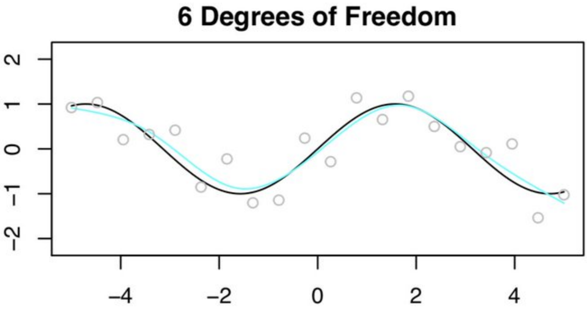


Figure 6: Fitting the true sine function (black) with cubic spline (cyan) (Image source: *Prof. Daniela Witten*'s Twitter).

Illustration of Interpolation

- Data: $Y_i = \sin(X_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, ..., 20$.
- Model (*cubic spline*): $h(X) = \sum_{k=0}^{3} a_k X^k + \sum_{j=1}^{N-3} b_j (X - \xi_j)_+^3$

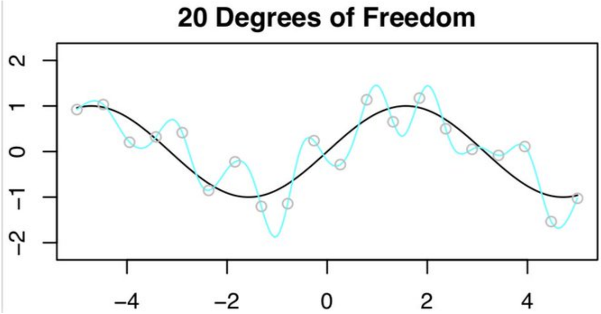


Figure 6: Fitting the true sine function (black) with cubic spline (cyan) (Image source: *Prof. Daniela Witten*'s Twitter).

Illustration of Interpolation

- Data: $Y_i = \sin(X_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, ..., 20$.
- Model (*cubic spline*): $h(X) = \sum_{k=0}^{3} a_k X^k + \sum_{j=1}^{N-3} b_j (X - \xi_j)_+^3$

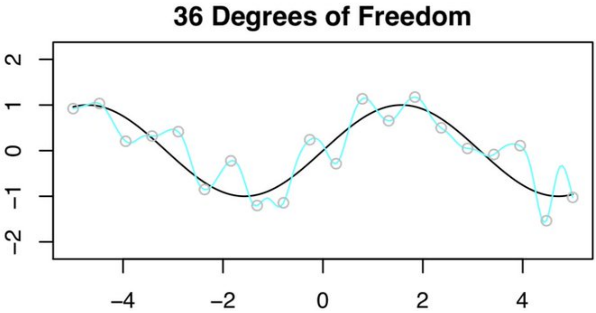
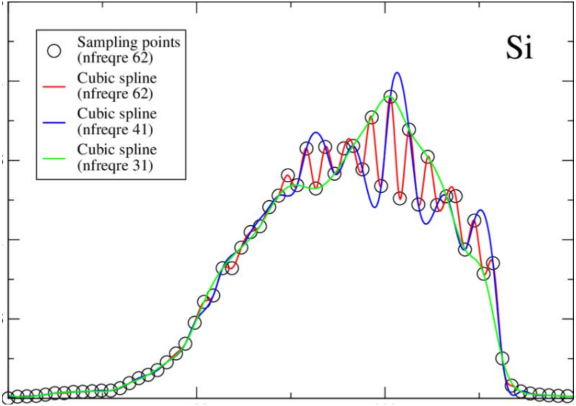


Figure 6: Fitting the true sine function (black) with cubic spline (cyan) (Image source: *Prof. Daniela Witten*'s Twitter).

Over-Parameterization Regime

Problem: Among all the interpolating models in a function class, which one should we choose?



Over-parameterization Regime

Solution: Choose the *smoothest* one, i.e,

$$\arg\min_{h \in \mathcal{H}'} ||h||_{\mathcal{H}},$$

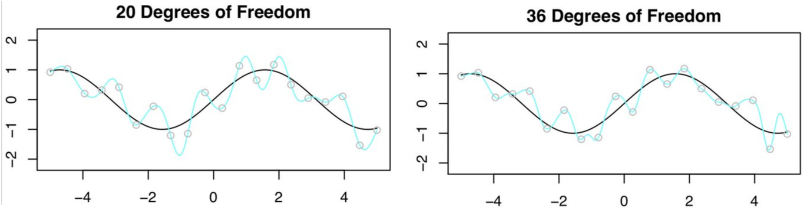
where $\mathcal{H}' \subset \mathcal{H}$ denotes the collection of all interpolating models. This principle is known as **Occam's razor** (Blumer et al., 1987).

Over-parameterization Regime

Solution: Choose the *smoothest* one, i.e,

$$\arg\min_{h \in \mathcal{H}'} ||h||_{\mathcal{H}},$$

where $\mathcal{H}' \subset \mathcal{H}$ denotes the collection of all interpolating models. This principle is known as **Occam's razor** (Blumer et al., 1987).



Experiment (I): Random Fourier Features (RFF)

The model family:

$$\mathcal{H}_N = \left\{ h : \mathbb{R}^d \to \mathbb{C} : h(x) = \sum_{i=1}^{N} a_k \phi(x; v_k) \text{ with } \phi(x; v) := e^{\sqrt{-1}\langle v_k, x\rangle} \right\},$$

where $v_1, ..., v_N \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, I_d)$ and $\mathcal{H}_N$ is a class of real-valued functions with $2N$ parameters.

- Notes: As $N \to \infty$, $\mathcal{H}_N$ approximates the reproducing kernel Hilbert space (RKHS) using the Gaussian kernel.

Experiment (I): Random Fourier Features (RFF)

Optimization details on RFF:

1. Find $h_{n,N} \in \mathcal{H}_N$ via Empirical Risk Minimization (ERM):

$$h_{n,N} = \arg\min_{h \in \mathcal{H}_N} \frac{1}{n} \sum_{i=1}^{n} (h(x_i) - y_i)^2 .$$

2. When the minimizer is not unique (as is always the case when $N > n$), choose the one with minimal

$$||h_{n,N}||_2 := \sqrt{\sum_{i=1}^{n} |a_i|^2}.$$

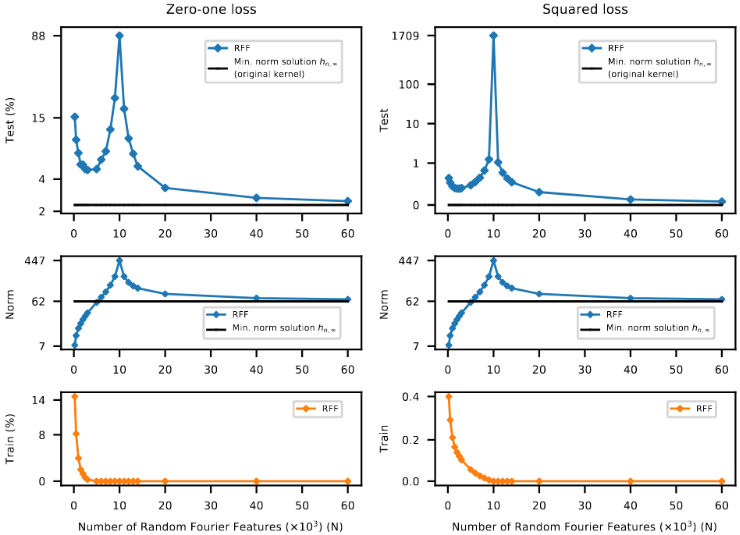


Figure 8: RFF on MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$.

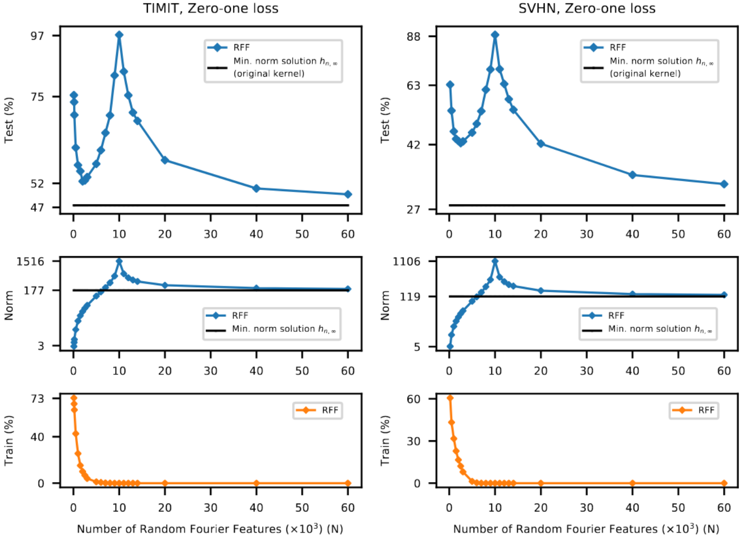


Figure 9: RFF on TIMIT ($n = 1.1 \times 10^6$, 48 classes) and SVHN ($n = 7.3 \times 10^4$, 10 classes).

Experiment (II): Fully Connected Two-Layer Neural Networks

Setup:

- Given a $(n \times d)$ training set with K classes, a fully connected neural network with a single layer of H hidden units has the number of parameters $H(d+K)$.
- It can be trained via stochastic gradient descent (SGD)

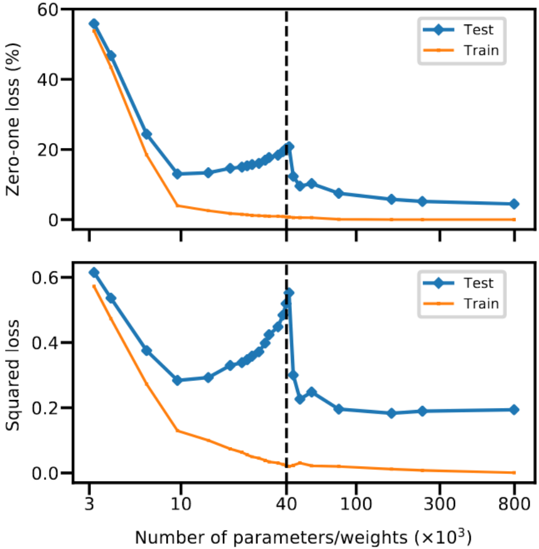


Figure 10: A fully connected neural network on a subset of MNIST ($n = 4 \times 10^3$, $d = 784$, $K = 10$ classes). The interpolation threshold is achieved at $N = 4 \times 10^4$.

Experiment (III): Random Forests

- In order to interpolate a $(n \times d)$ training set, a tree with n leaves (or fewer) will be learned.
- Beyond the interpolation threshold, the number of such trees will be increased.

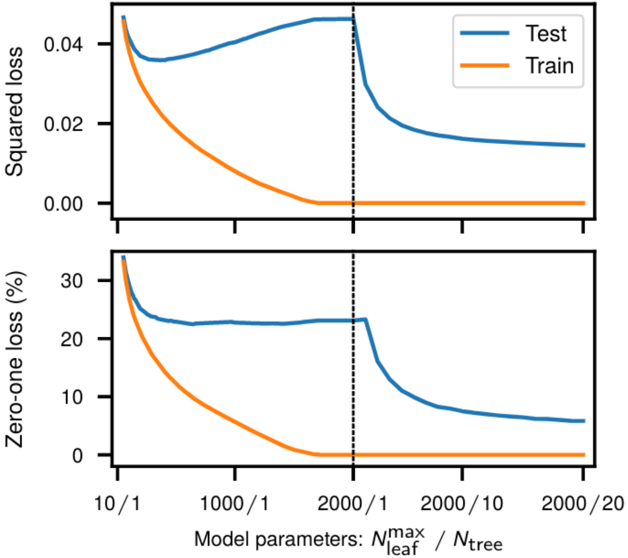


Figure 11: Random Forests on a subset of MNIST ($n = 10^4$, 10 classes).

“Double Descent” in Linear Regression

Consider the model:

$$Y_i = \boldsymbol{\beta}^T X_i + \epsilon_i, \quad (X_i, \epsilon_i) \sim P_X \times P_\epsilon,$$

where $\mathbb{E}(X_i) = \mathbf{0}$, Cov$(X_i) = \Sigma$, $\mathbb{E}(\epsilon_i) = 0$, and Var$(\epsilon_i) = \sigma^2$.

“Double Descent” in Linear Regression

Consider the model:

$$Y_i = \boldsymbol{\beta}^T X_i + \epsilon_i, \quad (X_i, \epsilon_i) \sim P_X \times P_\epsilon,$$

where $\mathbb{E}(X_i) = \mathbf{0}$, $\text{Cov}(X_i) = \Sigma$, $\mathbb{E}(\epsilon_i) = 0$, and $\text{Var}(\epsilon_i) = \sigma^2$.

$\Longrightarrow$

Least square regression estimator:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\boldsymbol{Y} - \boldsymbol{\beta}^T \boldsymbol{X}\|_2 = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^+ \boldsymbol{X}\boldsymbol{Y},$$

where $\boldsymbol{X} = (X_1, ..., X_n)^T \in \mathbb{R}^{n \times p}$, $\boldsymbol{Y} = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$, and $(\boldsymbol{X}^T \boldsymbol{X})^+$ is the pseudoinverse of $\boldsymbol{X}^T \boldsymbol{X}$.

“Double Descent” in Linear Regression

- $\gamma := \frac{p}{n}$ is the overparametrization ratio (as $n, p \to \infty$).
- SNR $= \frac{\|\boldsymbol{\beta}\|_2^2}{\sigma^2}$ is the signal-to-noise ratio.

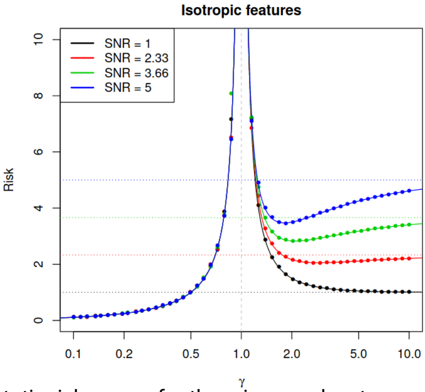


Figure 12: Asymptotic risk curves for the min-norm least square estimator (Hastie et al., 2019).

“Multiple Descent” in Linear Regression

Minimum ℓ_1-norm interpolation (Li and Wei, 2021):

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} ||\boldsymbol{\beta}||_1 \quad \text{subject to } Y_i = \boldsymbol{\beta}^T X_i, i = 1, ..., n.$$

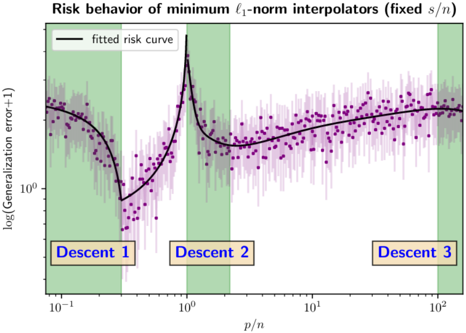


Figure 13: Triple descent in sparse linear regression. Here, s is the ratio of sparsity in the true signal.

Why are linear models are informative? (Neural Tangent Kernel Theory)

When the number of parameter p is very large, we approximate the model $\boldsymbol{z} \mapsto f(\boldsymbol{z}; \boldsymbol{\theta})$ by

$$\boldsymbol{z} \mapsto \nabla_{\boldsymbol{\theta}} f(\boldsymbol{z}; \boldsymbol{\theta}_0)^T \boldsymbol{\beta},$$

where we suppose that $f(\boldsymbol{z}; \boldsymbol{\theta}_0) \approx 0$ and let $\boldsymbol{\theta} = \boldsymbol{\theta}_0 + \boldsymbol{\beta}$.

This argument can be made rigorous via **Neural Tangent Kernel** theory (Jacot et al., 2018), especially when $p > n$. See, for instance, Allen-Zhu et al. (2019):

- Allen-Zhu, Zeyuan, Yuanzhi Li, and Zhao Song. *"A convergence theory for deep learning via over-parameterization."* International Conference on Machine Learning. PMLR, 2019.

Conclusion Thoughts

- Shall we always avoid overfitting?

Conclusion Thoughts

- Shall we always avoid overfitting? The answer is “NOT Necessarily”!

Conclusion Thoughts

- Shall we always avoid overfitting? The answer is “NOT Necessarily”!
- The classical bias-variance trade-off is still useful when
    1. The training set is of large scale.
    2. An interpolating class cannot be fitted.

A Final Reflection

Is the interpolation or overparametrization theory correct?

A Final Reflection

Is the interpolation or overparametrization theory correct?

- Balestriero, Randall, Jerome Pesenti, and Yann LeCun. *”Learning in High Dimension Always Amounts to Extrapolation*.” arXiv preprint arXiv:2110.09485 (2021).

The interpolation point of view does not seem to be right!

A Final Reflection

Is the interpolation or overparametrization theory correct?

- Balestriero, Randall, Jerome Pesenti, and Yann LeCun. *"Learning in High Dimension Always Amounts to Extrapolation*." arXiv preprint arXiv:2110.09485 (2021).

The interpolation point of view does not seem to be right!

However, in my opinion, an overparametrized model is still effective in practice! Why?

A Final Reflection

Is the interpolation or overparametrization theory correct?

- Balestriero, Randall, Jerome Pesenti, and Yann LeCun. ”*Learning in High Dimension Always Amounts to Extrapolation*.” arXiv preprint arXiv:2110.09485 (2021).

The interpolation point of view does not seem to be right!

However, in my opinion, an overparametrized model is still effective in practice! Why?

- Fefferman, Charles, Sanjoy Mitter, and Hariharan Narayanan. ”*Testing the manifold hypothesis*.” Journal of the American Mathematical Society 29.4 (2016): 983-1049.

Thank You

Comments or Questions?

yikunzhang@trip.com

References I

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

G. Claeskens, N. L. Hjort, et al. Model selection and model averaging. *Cambridge Books*, 2008.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, second edition, 2009. URL `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Y. Li and Y. Wei. Minimum ℓ_1-norm interpolators: Precise asymptotics and multiple descent. *arXiv preprint arXiv:2110.09502*, 2021.

B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.