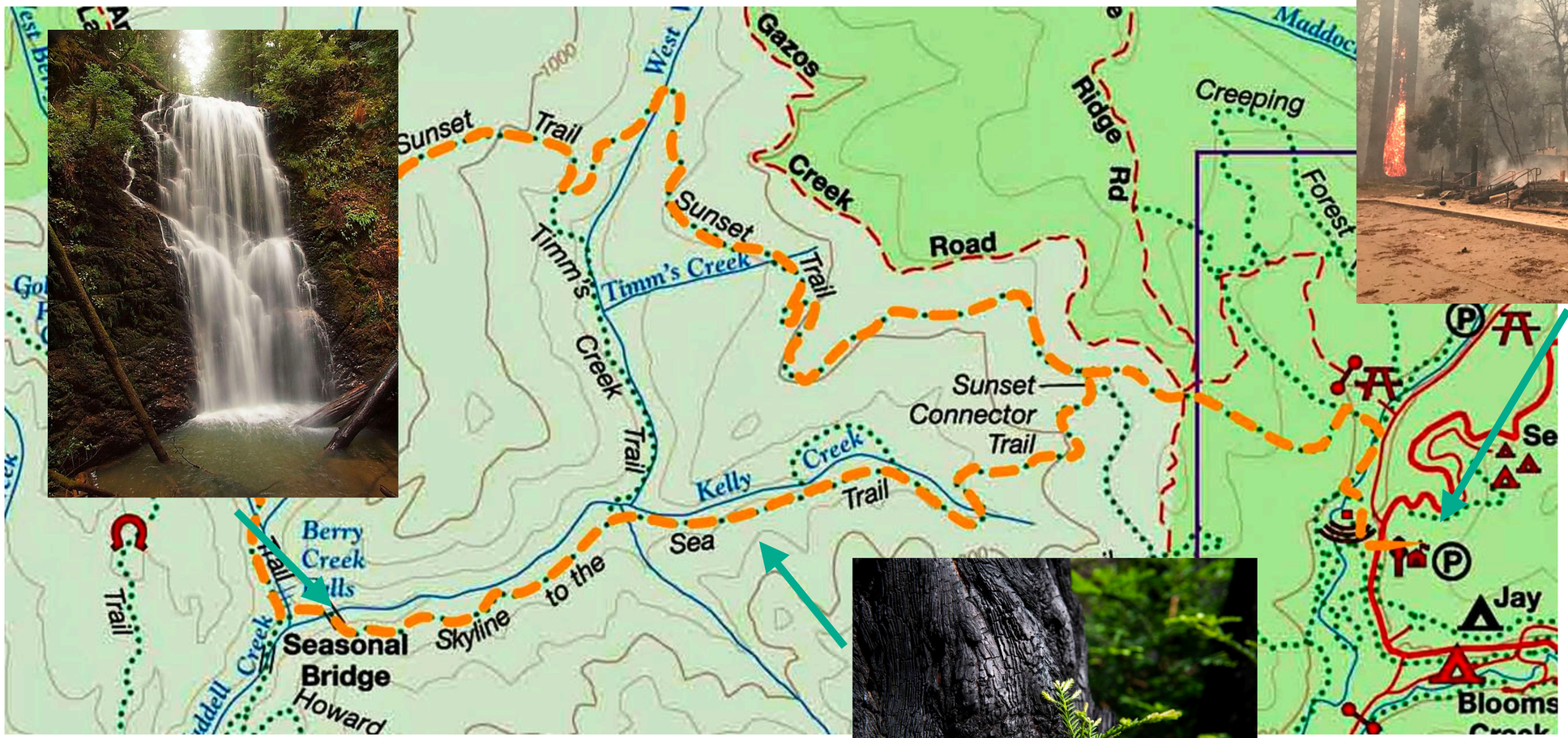
Why do over-parameterized neural networks work? An overview of benign overfitting and inductive bias

Clayton Sanford

November 22, 2022

This talk

Neural Network Theory State Park



Central issue of deep learning theory

- **Goal of ML theory:**
  - Rigorous mathematical understanding capabilities and limitations of ML algorithms, which translate to practical recommendations for practitioners.
- **Problem:**
  - ML theory is too pessimistic for deep learning; lack of theoretical explanations for neural networks' practical success.
  - Two conflicting narratives for what makes ML models succeed: classical ML theory vs modern deep learning practice.

Supervised learning setting

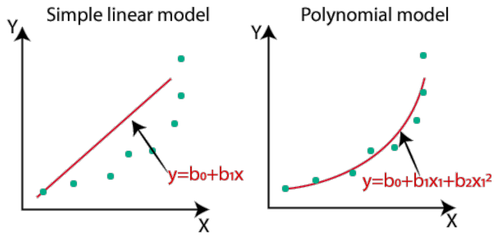
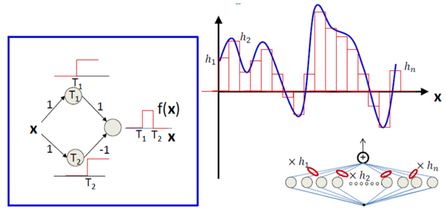
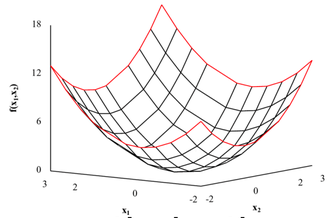
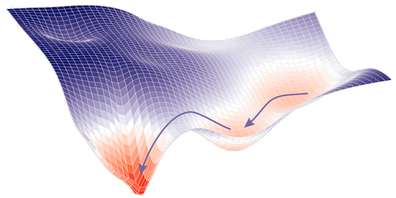
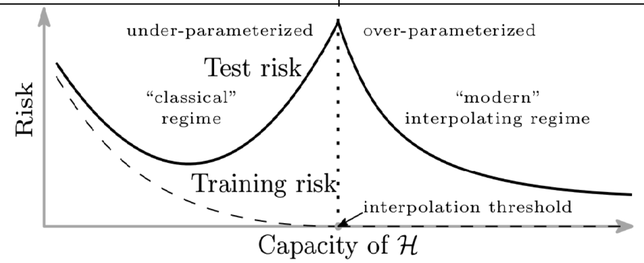
- Given samples $(x_1, y_1), \ldots, (x_n, y_n) \sim \mathscr{D}$.
- Want to learn $h : \mathbb{R}^d \to \mathscr{Y}$ such $R(h) = \mathbb{E}[\ell(h(x), y)]$ is small for new $(x, y) \sim \mathscr{D}$.
  - $\mathscr{Y} = \{\pm 1\}$ for classification, $\mathscr{Y} = \mathbb{R}$ for regression.
- How? Find $h \in \mathscr{H}$ minimizing training error: $\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$.

$$\underbrace{R(h)}_{\text{population error}} = \underbrace{\hat{R}(h)}_{\text{training error}} + \underbrace{R(h) - \hat{R}(h)}_{\text{generalization error}}$$

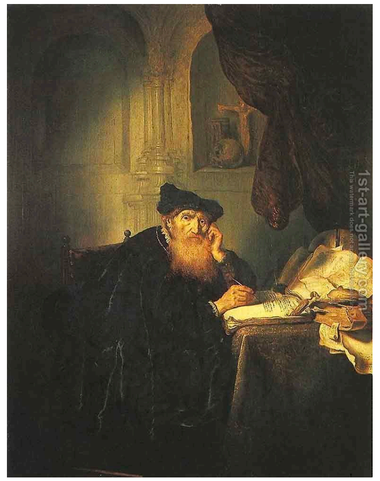
Supervised learning setting

- Given samples $(x_1, y_1), \ldots, (x_n, y_n) \sim \mathscr{D}$.
- Want to learn $h : \mathbb{R}^d \to \mathscr{Y}$ such $R(h) = \mathbb{E}[\ell(h(x), y)]$ is small for new $(x, y) \sim \mathscr{D}$.
  - $\mathscr{Y} = \{\pm 1\}$ for classification, $\mathscr{Y} = \mathbb{R}$ for regression.
- How? Find $h \in \mathscr{H}$ minimizing training error: $\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$.

$$\underbrace{R(h)}_{\text{population error}} = \underbrace{\min_{\tilde{h} \in h} \hat{R}(\tilde{h})}_{\text{approximation error}} + \underbrace{\hat{R}(h) - \min_{\tilde{h} \in h} \hat{R}(\tilde{h})}_{\text{optimization error}} + \underbrace{R(h) - \hat{R}(h)}_{\text{generalization error}}$$

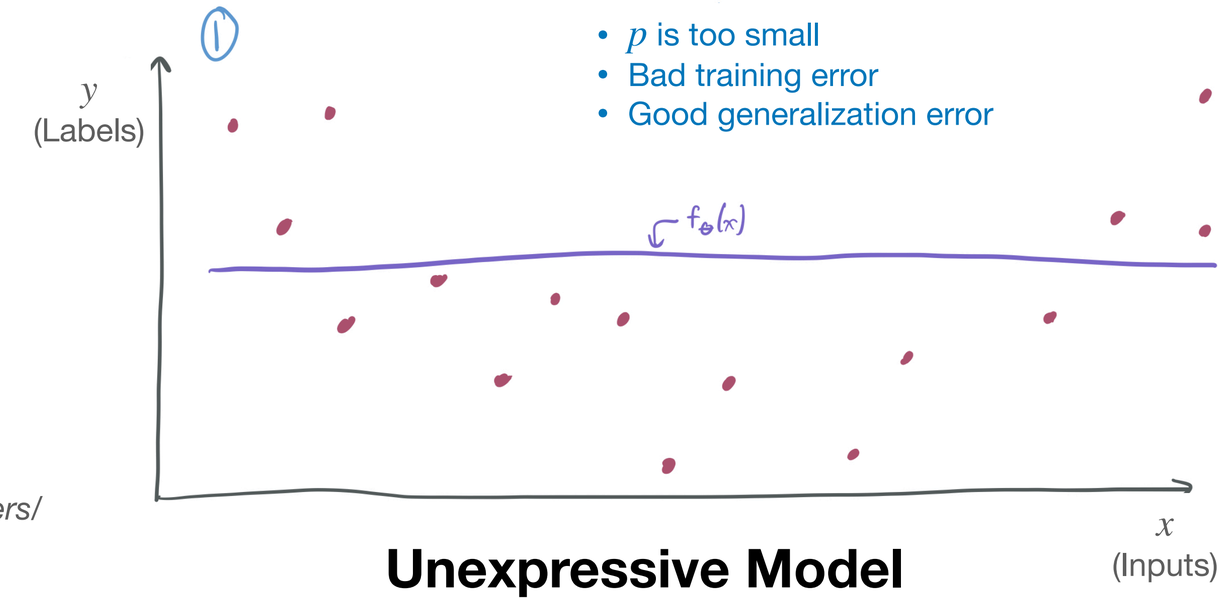
	Classical ML	Deep Learning
Approximation	 Limited capacity models	 Universal approximation
Optimization	 Convex optimization and/or poly-time algorithms	 Non-convex optimization No guarantees
Generalization	 Capacity-based	"Benign overfitting"

Narrative #1: Classical Theory



- ML textbook trade-offs: greater model complexity requires more samples
- Delicate balance between overfitting (high generalization error $R(h) - \hat{R}(h)$) and over-simplification (high training error $\hat{R}(h)$)

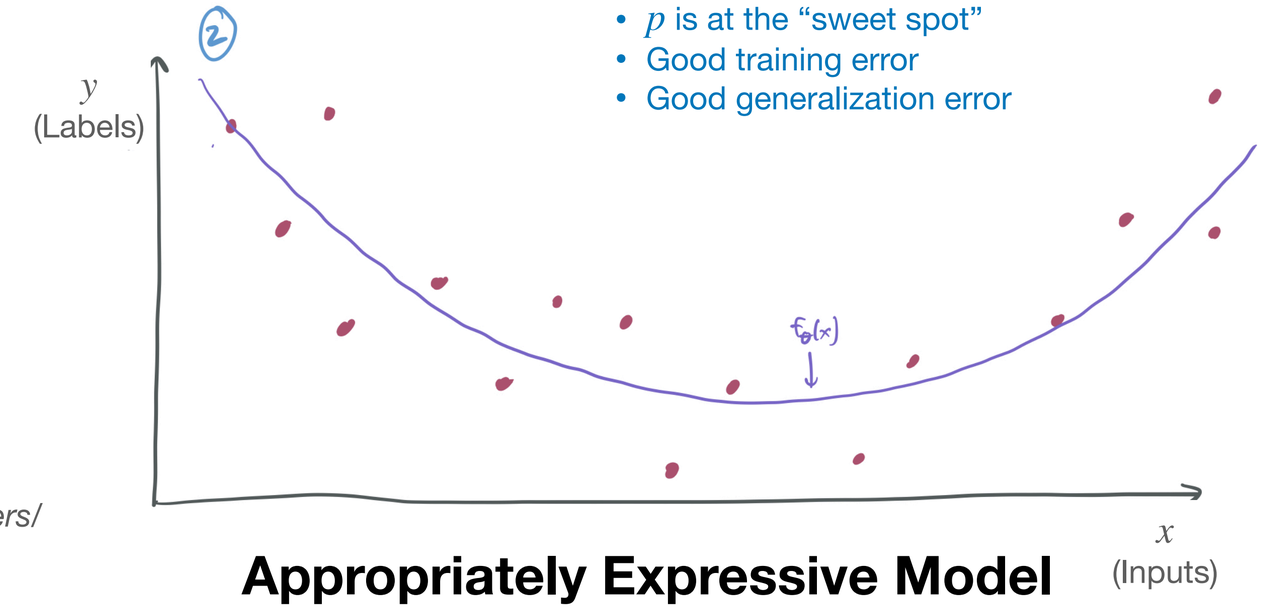
Narrative #1: Classical Theory



Unexpressive Model

- ML textbook trade-offs: greater model complexity requires more samples
- Delicate balance between overfitting (high generalization error $R(h) - \hat{R}(h)$) and over-simplification (high training error $\hat{R}(h)$)

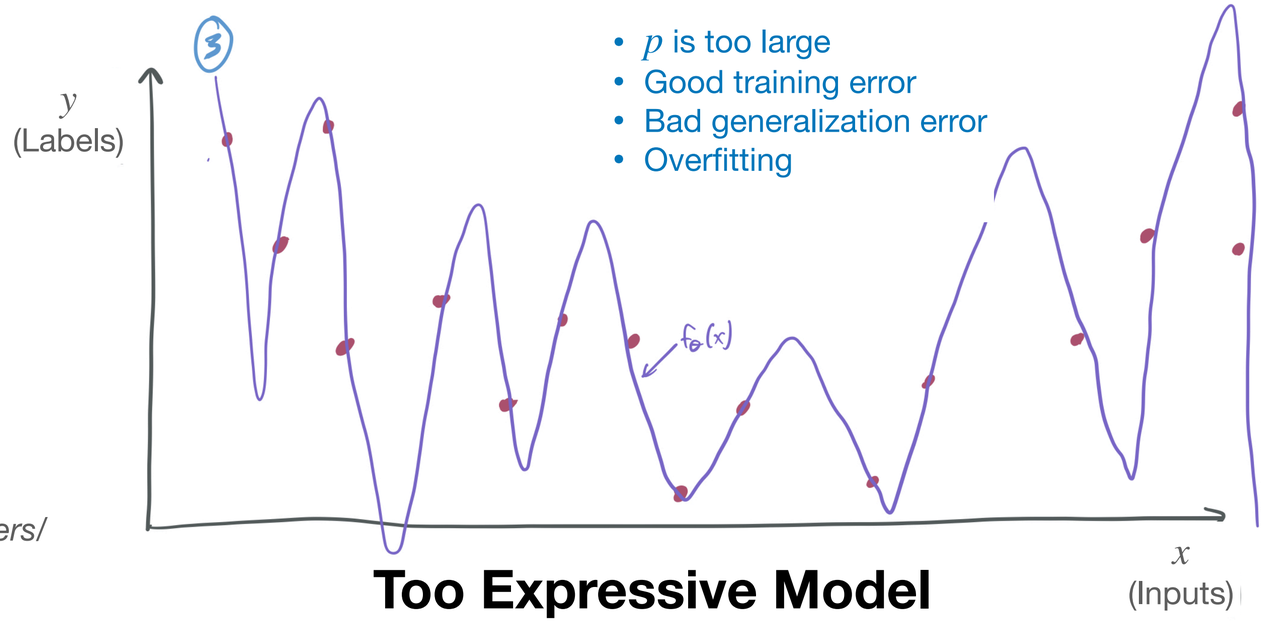
Narrative #1: Classical Theory



Appropriately Expressive Model

- ML textbook trade-offs: greater model complexity requires more samples
- Delicate balance between overfitting (high generalization error $R(h) - \hat{R}(h)$) and over-simplification (high training error $\hat{R}(h)$)

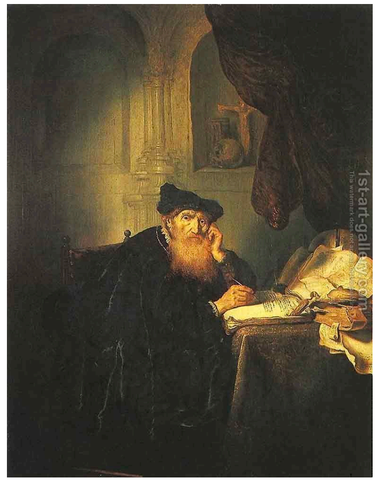
Narrative #1: Classical Theory



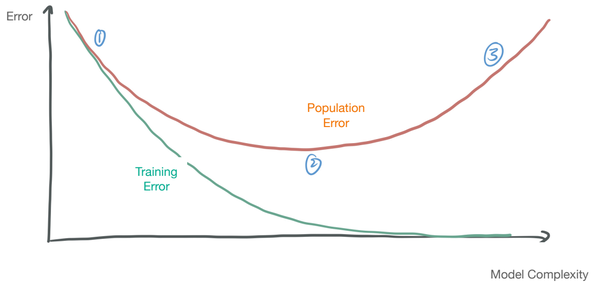
Too Expressive Model

- ML textbook trade-offs: greater model complexity requires more samples
- Delicate balance between overfitting (high generalization error $R(h) - \hat{R}(h)$) and over-simplification (high training error $\hat{R}(h)$)

Narrative #1: Classical Theory



- Made rigorous with measurements of model complexity, like VC-dimension**,** Rademacher complexity, fat-shattering dimension.
- **VC-dimension** measures ability of hypotheses in $\mathscr{H}$ to correctly classify different labelings y_i.
- Generalization bound based on VC-dimension:
  - $R(h) - \hat{R}(h) = O(\sqrt{VC(\mathscr{H})/n})$ for all $h \in \mathscr{H}$.
- Example: Linear classification
  - $VC(\mathscr{H}) = d + 1.$
  - $R(h) - \hat{R}(h) = O(\sqrt{d/n})$ for all $h \in \mathscr{H}$.



Narrative #2: Deep Learning Practice

- **Past decade**: empirical dominance of deep learning over other ML models
- **How to train a neural network:**
  - Initialize a very large model (# params > # samples)
  - Train with gradient descent until convergence to very small training error
  - Necessary tips & tricks: dropout, Adam, batch size, regularization, specialized architectures, choice of loss function, etc.

Clash between narratives

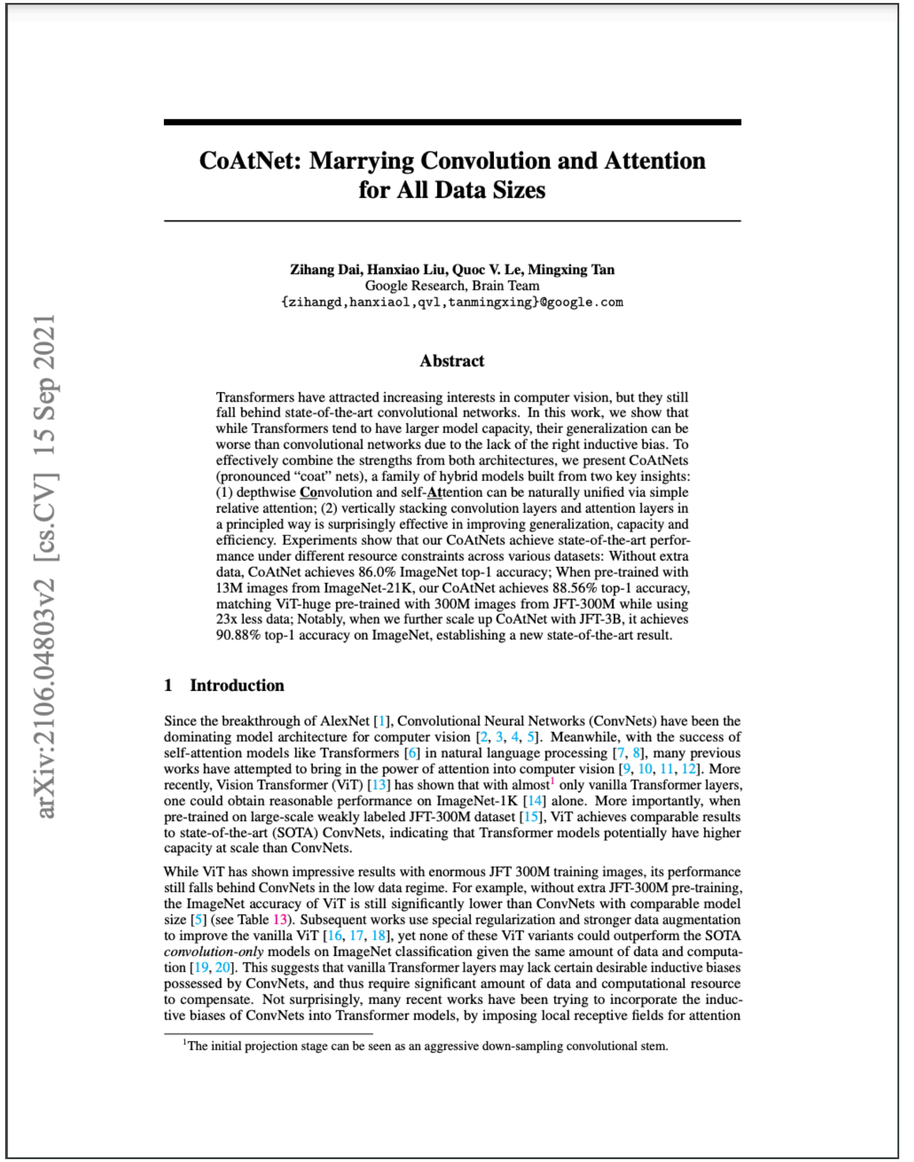
Unprincipled alchemy!

Irrelevant theory!

Clash between narratives

An example

- **CoAtNet:** current (as of May 2022) holder of SOTA for ImageNet image classification (ignoring ensemble models)
- Achieves **86.09%** accuracy on 1000-class classification by a NN with **168M** parameters and **13M** training samples.
- VC dimension of NNs with fixed depth and w parameters is $\Theta(w \log w)$ [Bartlett, et al '98].
- Generalization bound is vacuous: $R(h) - \hat{R}(h) = \tilde{O}(\sqrt{w/n})$.



CoAtNet: Marrying Convolution and Attention for All Data Sizes

Zihang Dai, Hanxiao Liu, Quoc V. Le, Mingxing Tan
Google Research, Brain Team
{zihangd,hanxiaol,qvl,tanmingxing}@google.com

Abstract

Transformers have attracted increasing interests in computer vision, but they still fall behind state-of-the-art convolutional networks. In this work, we show that while Transformers tend to have larger model capacity, their generalization can be worse than convolutional networks due to the lack of the right inductive bias. To effectively combine the strengths from both architectures, we present CoAtNets (pronounced "coat" nets), a family of hybrid models built from two key insights: (1) depthwise **Co**nvolution and self-**At**tention can be naturally unified via simple relative attention; (2) vertically stacking convolution layers and attention layers in a principled way is surprisingly effective in improving generalization, capacity and efficiency. Experiments show that our CoAtNets achieve state-of-the-art performance under different resource constraints across various datasets: Without extra data, CoAtNet achieves 86.0% ImageNet top-1 accuracy; When pre-trained with 13M images from ImageNet-21K, our CoAtNet achieves 88.56% top-1 accuracy, matching ViT-huge pre-trained with 300M images from JFT-300M while using 23x less data; Notably, when we further scale up CoAtNet with JFT-3B, it achieves 90.88% top-1 accuracy on ImageNet, establishing a new state-of-the-art result.

1 Introduction

Since the breakthrough of AlexNet [1], Convolutional Neural Networks (ConvNets) have been the dominating model architecture for computer vision [2, 3, 4, 5]. Meanwhile, with the success of self-attention models like Transformers [6] in natural language processing [7, 8], many previous works have attempted to bring in the power of attention into computer vision [9, 10, 11, 12]. More recently, Vision Transformer (ViT) [13] has shown that with almost[1] only vanilla Transformer layers, one could obtain reasonable performance on ImageNet-1K [14] alone. More importantly, when pre-trained on large-scale weakly labeled JFT-300M dataset [15], ViT achieves comparable results to state-of-the-art (SOTA) ConvNets, indicating that Transformer models potentially have higher capacity at scale than ConvNets.

While ViT has shown impressive results with enormous JFT 300M training images, its performance still falls behind ConvNets in the low data regime. For example, without extra JFT-300M pre-training, the ImageNet accuracy of ViT is still significantly lower than ConvNets with comparable model size [5] (see Table 13). Subsequent works use special regularization and stronger data augmentation to improve the vanilla ViT [16, 17, 18], yet none of these ViT variants could outperform the SOTA *convolution-only* models on ImageNet classification given the same amount of data and computation [19, 20]. This suggests that vanilla Transformer layers may lack certain desirable inductive biases possessed by ConvNets, and thus require significant amount of data and computational resource to compensate. Not surprisingly, many recent works have been trying to incorporate the inductive biases of ConvNets into Transformer models, by imposing local receptive fields for attention

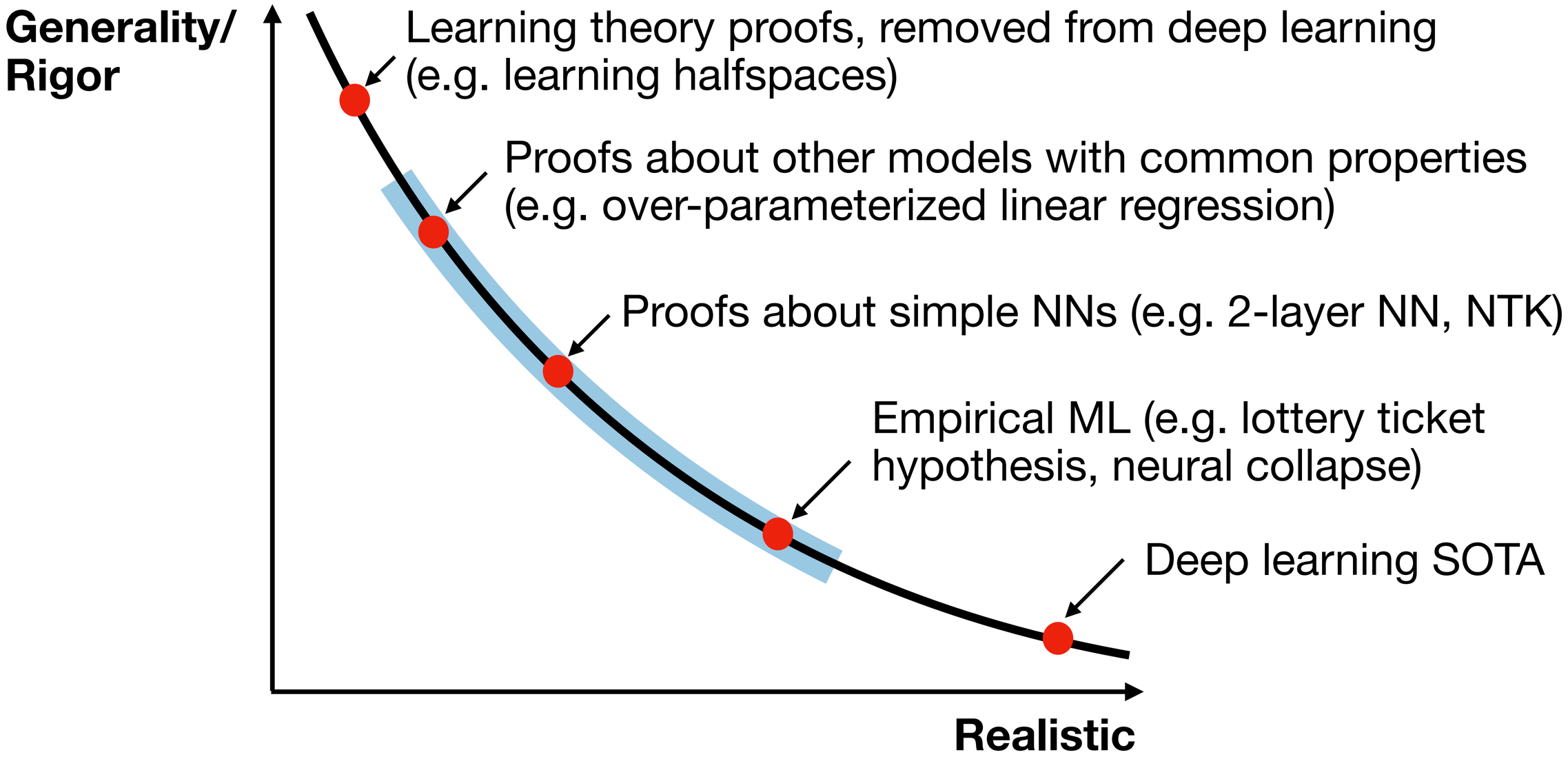
[1]The initial projection stage can be seen as an aggressive down-sampling convolutional stem.

Can they be reconciled?

- **Goal:** a theory of **benign overfitting** that mathematically describes why some overfitting models generalize.
- **Challenge:** It's mathematically very difficult to say things about neural networks

$$
\begin{aligned}
&\sum_{k_0,k_1,k_2} \mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right) W^{(\ell+1)}_{i_0k_0} W^{(\ell+1)}_{i_1k_1} W^{(\ell+1)}_{i_2k_2} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{\mathrm{d}H}^{(\ell)}_{k_0k_1k_2;\delta_0\delta_1\delta_2}\right] \\
=&\frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \sum_{\delta\in\mathcal{D}} \delta_{i_0i_1} \left\langle\!\!\left\langle \frac{\partial \mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma'^{(\ell)}_{k;\delta_0} \sigma'^{(\ell)}_{k;\delta_1} \sigma'^{(\ell)}_{m;\delta_2} \sigma^{(\ell)}_{m;\delta} \widehat{\mathrm{d}H}^{(\ell)}_{kkm;\delta_0\delta_1\delta_2}\right] \right\} \\
&+ \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \sum_{\delta\in\mathcal{D}} \delta_{i_0i_2} \left\langle\!\!\left\langle \frac{\partial \mathcal{O}}{\partial z^{(\ell+1)}_{i_1;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma'^{(\ell)}_{k;\delta_0} \sigma'^{(\ell)}_{m;\delta_1} \sigma'^{(\ell)}_{k;\delta_2} \sigma^{(\ell)}_{m;\delta} \widehat{\mathrm{d}H}^{(\ell)}_{kmk;\delta_0\delta_1\delta_2}\right] \right\} \\
&+ \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \sum_{\delta\in\mathcal{D}} \delta_{i_1i_2} \left\langle\!\!\left\langle \frac{\partial \mathcal{O}}{\partial z^{(\ell+1)}_{i_0;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma'^{(\ell)}_{m;\delta_0} \sigma'^{(\ell)}_{k;\delta_1} \sigma'^{(\ell)}_{k;\delta_2} \sigma^{(\ell)}_{m;\delta} \widehat{\mathrm{d}H}^{(\ell)}_{mkk;\delta_0\delta_1\delta_2}\right] \right\} \\
&+ O\left(\frac{1}{n^2}\right) . \qquad (11.40)
\end{aligned}
$$

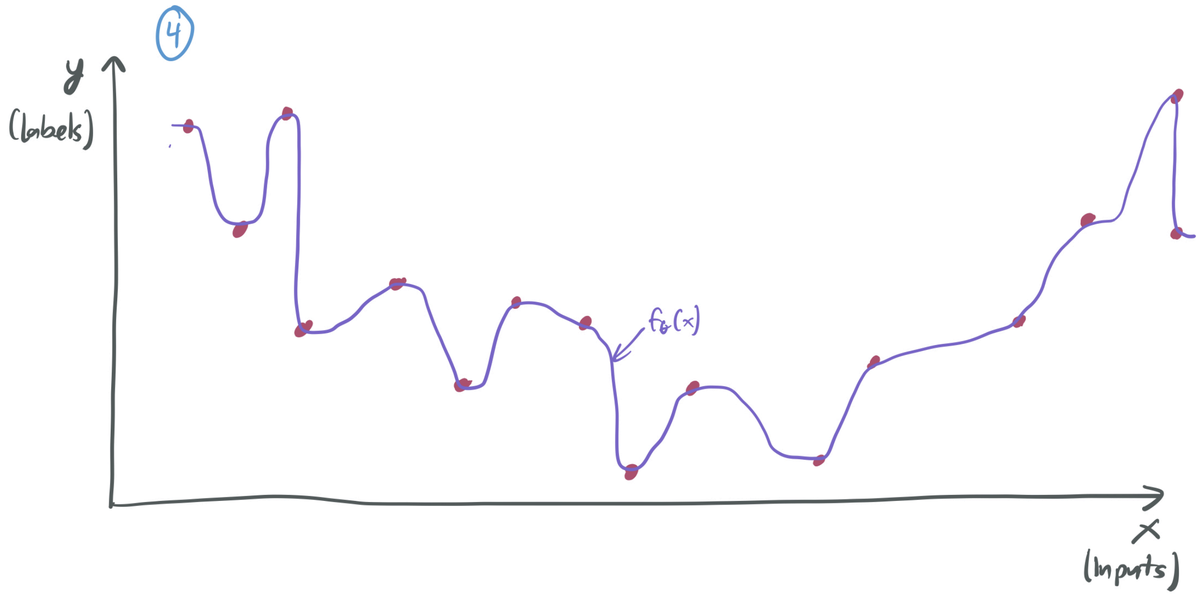
Can they be reconciled? (ctd)



Benign overfitting and double-descent

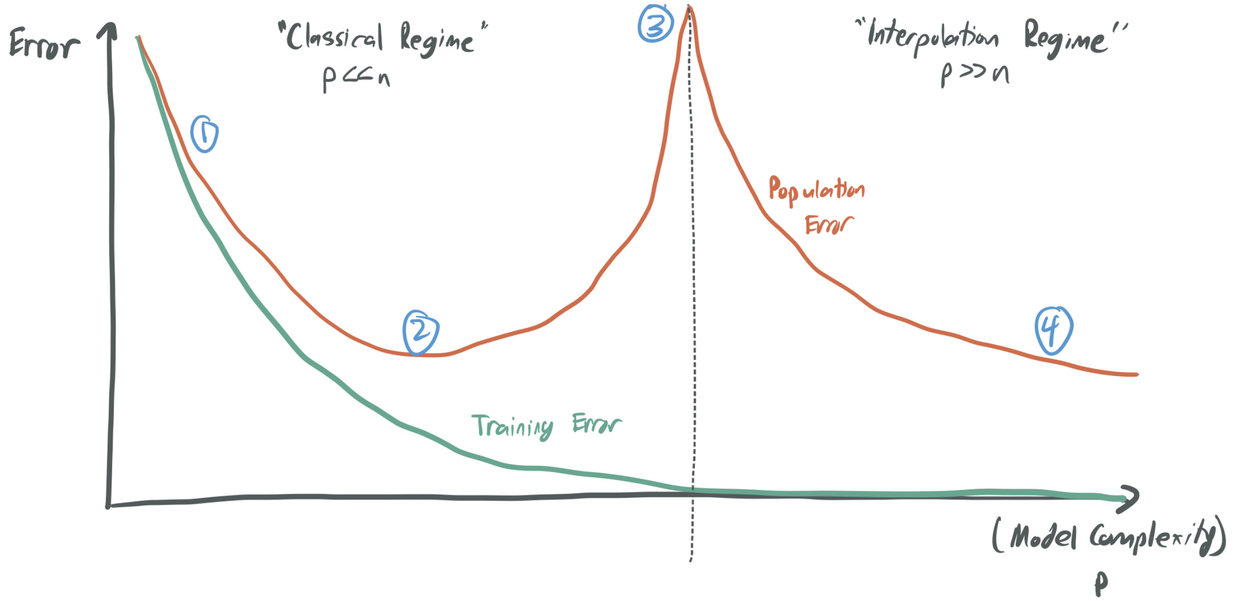
Benign Overfitting

Model generalizes despite over-parameterization and very small training error



Double Descent

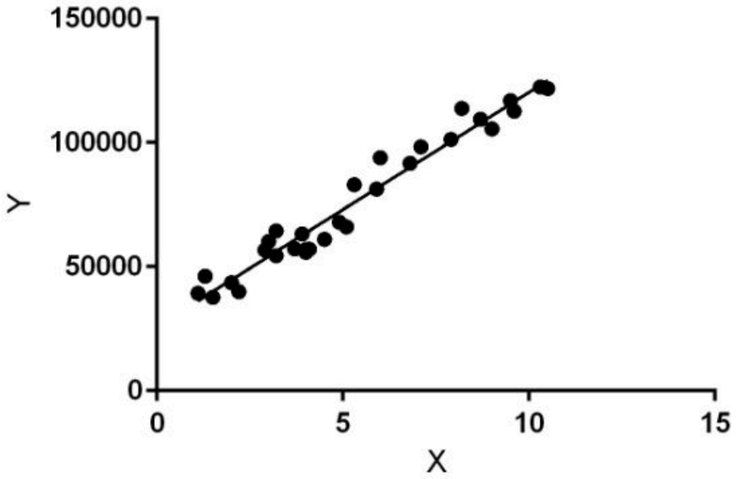
Increasing model complexity beyond initial point of overfitting causes second descent of generalization error

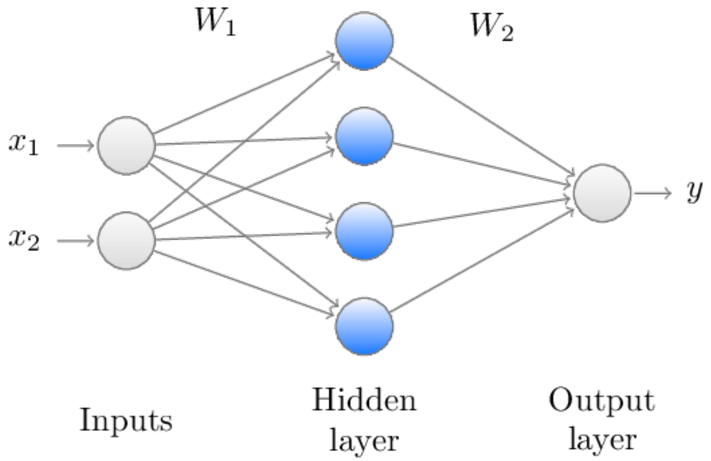


Two Vignettes

1. **Benign overfitting in linear models**:
   simplicity via minimum-norm interpolation

2. **Benign overfitting in 2-layer neural nets**:
   simplicity via adaptivity to low dimensions

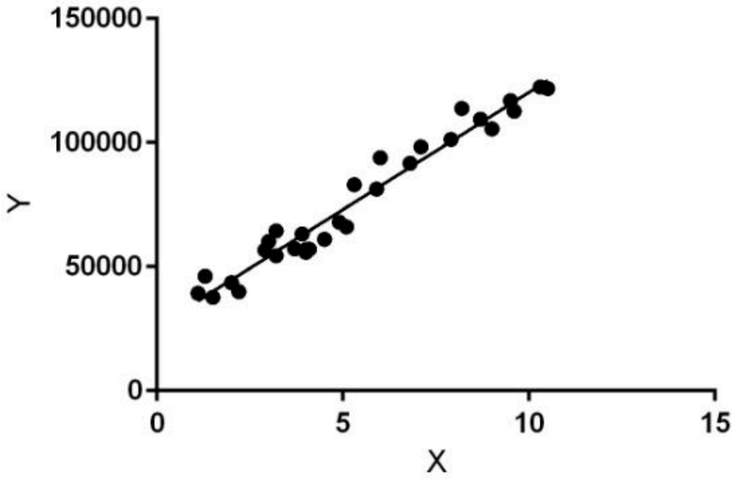


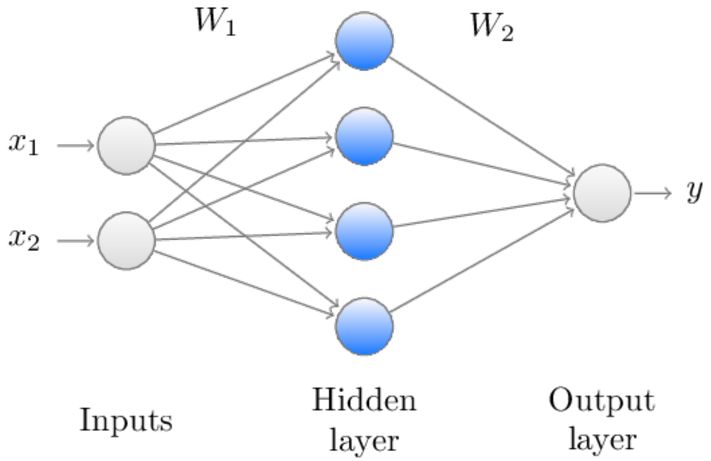


Two Vignettes

1. **Benign overfitting in linear models**:
   simplicity via minimum-norm interpolation

2. **Benign overfitting in 2-layer neural nets**:
   simplicity via adaptivity to low dimensions





Where can you find benign overfitting?

- **Least-squares regression** [BHX19, **BLLT19**, HMRT19, Mitra19, MVSS19]
- Ridge regression [TB20]
- Kernel regression [RZ19, LRZ20]
- **Support vector machines** [**MNSBHS20**, CL20, **ASH20**]
- Random feature models [MM19]
- Boosting [BFLS98]

Linear regression

- Sample $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$. $(X, y) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$.
- Learn $x \mapsto \hat{\theta}^T x$.
- **Ordinary least-squares (OLS)** (classical, $n \gg d$):
  - $\hat{\theta} \in \mathbb{R}^d$ minimizes $\sum_{i=1}^{n} (\hat{\theta}^T x_i - y_i)^2$, or $\hat{\theta} = X^\dagger y = (X^T X)^{-1} X^T y$.
- **Minimum-norm interpolation** (interpolation, $d \gg n$):
  - $\hat{\theta} \in \mathbb{R}^d$ minimizes $\|\hat{\theta}\|$ such that $\hat{\theta}^T x_i = y_i$, or $\hat{\theta} = X^\dagger y = X(XX^T)^{-1} Y$.
- Classical generalization bound: $R(h) - \hat{R}(h) \leq O(\sqrt{d/n})$ [Audibert and Catoni, ’10]

Benign overfitting: feature importance

[Bartlett, Long, Lugosi, Tsigler '19]

- Analysis of when benign overfitting occurs for over-parameterized OLS ($d \gg n$).
- Special case: **bi-level ensemble** for $k \leq n$:
  - Subgaussian $x_i \in \mathbb{R}^d$ with independent components and diagonal covariance Λ with $\Lambda_{1,1}, \ldots, \Lambda_{k,k} = 1$ and $\Lambda_{k+1,k+1}, \ldots, \Lambda_{d,d} = \lambda < n/d$.
  - Optimal weight $\theta^* \in \mathbb{S}^{d-1}$, and subgaussian noise σ.
- **Theorem**: With probability 0.99:

$$R(h) = O\left(\frac{k + \lambda n}{n} + \sigma^2\left(\frac{k}{n} + \frac{n}{d}\right)\right)$$

- Proof by **bias-variance** decomposition and eigenvalue concentration bounds.

Benign overfitting: feature importance

[Bartlett, Long, Lugosi, Tsigler '19]

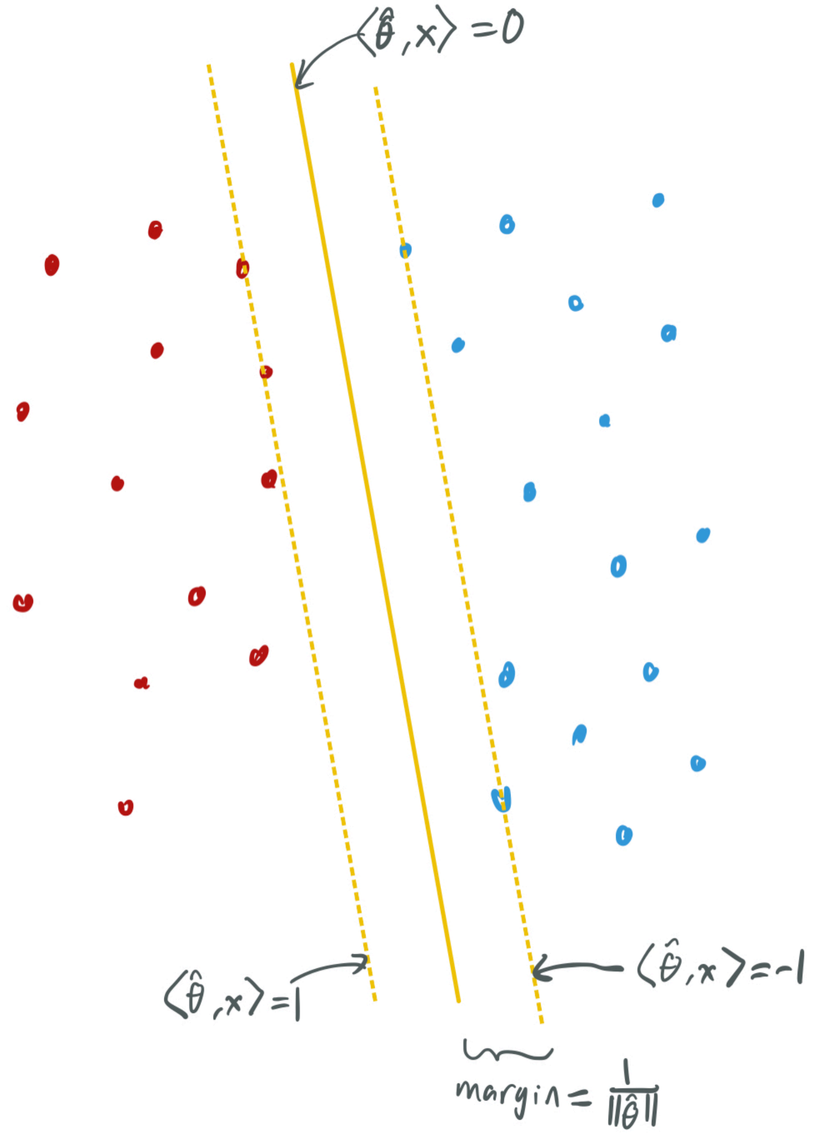
- $\Lambda_{1,1}, \ldots, \Lambda_{k,k} = 1$ and $\Lambda_{k+1,k+1}, \ldots, \Lambda_{d,d} = \lambda < n/d$.
- **Theorem**: With probability 0.99:

$$R(h) = O\left(\frac{k + \lambda n}{n} + \sigma^2\left(\frac{k}{n} + \frac{n}{d}\right)\right)$$

- Benign overfitting occurs when…
  - $k \ll n$ (small number of high-variance/"significant" features)
  - $\lambda < n/d$ (all other "noisy" features are relatively low-variance)
  - $n \ll d$ (problem has *much* more parameters than necessary to fit)

Hard SVM or maximum-margin classification

- Linearly separable
  $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1, 1\}$.
- Learn $x \mapsto \text{sign}(\hat{\theta}^T x)$.
- $\hat{\theta} \in \mathbb{R}^d$ minimizes $\|\hat{\theta}\|_2$ such that
  $y_i \hat{\theta}^T x_i \geq 1$.
- x_i is a **support vector** if $\hat{\theta}^T x_i = y_i$.
- Classical generalization bounds rely on bounding number of support vectors.



SVM benign overfitting by connection to OLS [MNSBHS20]

Exhibits benign overfitting for SVM linear classification of Gaussian data with bi-level variances:

1. For certain distributions, over-parameterized OLS regression for $y_i \in \mathbb{R}$ has benign overfitting. [BLLT19]

2. Then, "OLS classification" with $y_i \in \{\pm 1\}$ and prediction rule $x \mapsto \operatorname{sign}(x^T \theta_{OLS})$ also has benign overfitting.

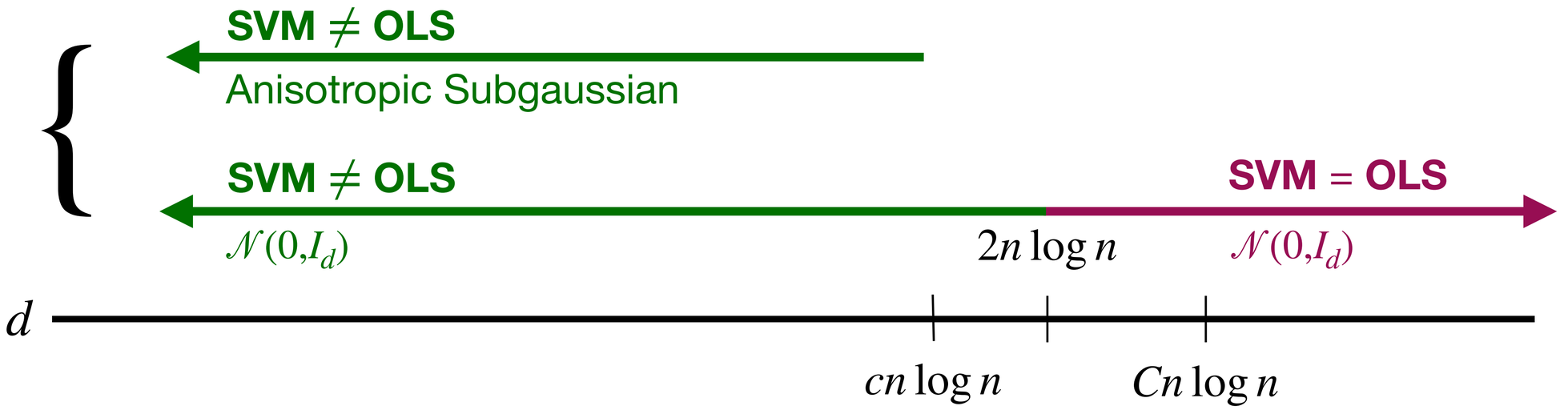
3. Given dimension $d = \Omega(n^{3/2} \log n)$, same weights returned by OLS classification and SVM: $\theta_{OLS} = \theta_{SVM}$.

Tightening [MNSBHS20]…

Question: For what $d = d(n)$ do we have **SVM** = **OLS** with high probability?

- **[MNSBHS20]**

- **[ASH21]**



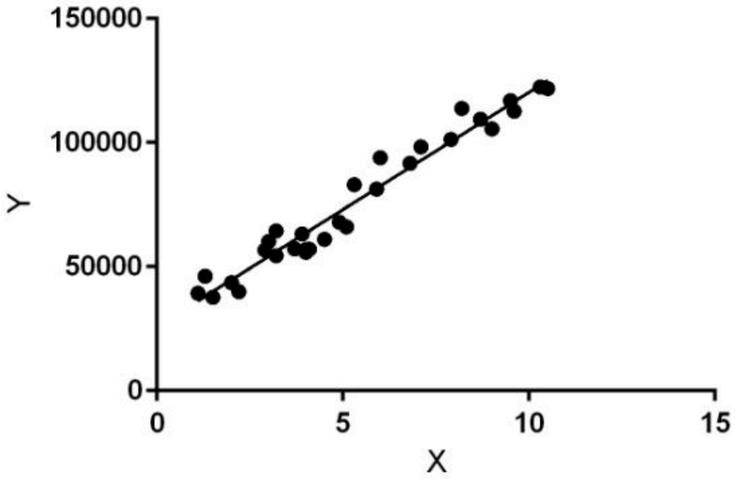
Zooming out

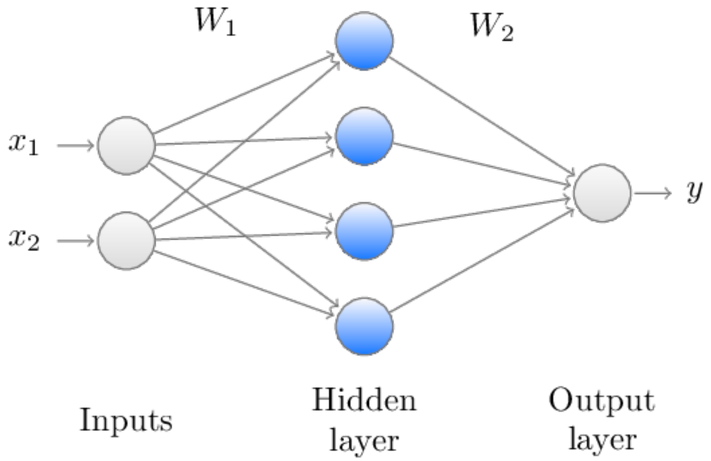
Or: It’s 2022, who cares about SVMs or OLS??

- Benign overfitting isn’t only for neural nets!
- Gradient descent sometimes biased in favor of max-margin classifier
  - Classification tasks with logistic loss function converge to the max margin solution [Soudry et al, ’17], [Ji & Telgarsky ’19]
  - Wide two-layer neural nets with logistic loss function also converge to max margin solutions [Chizat et al, ’20]

Two Vignettes

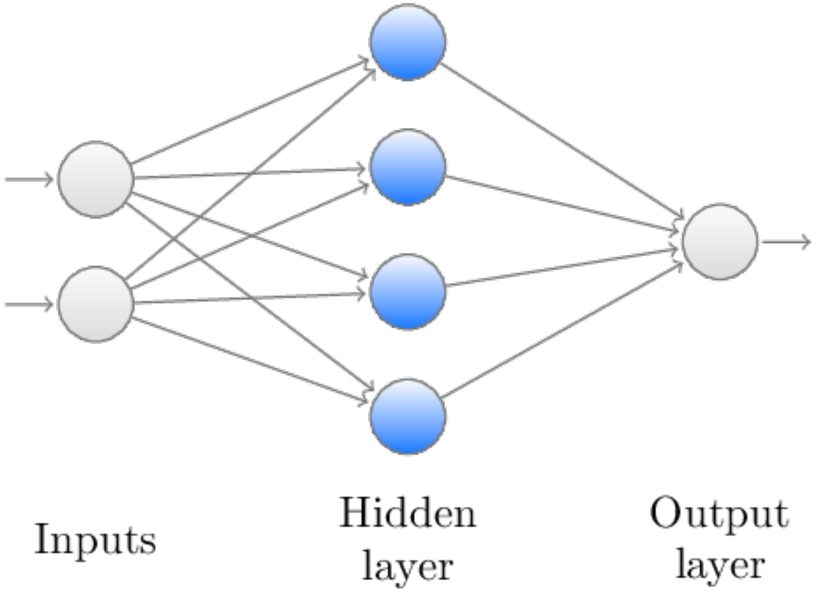
1. **Benign overfitting in linear models**:
   simplicity via minimum-norm interpolation
2. **Benign overfitting in 2-layer neural nets**:
   simplicity via adaptivity to low dimensions





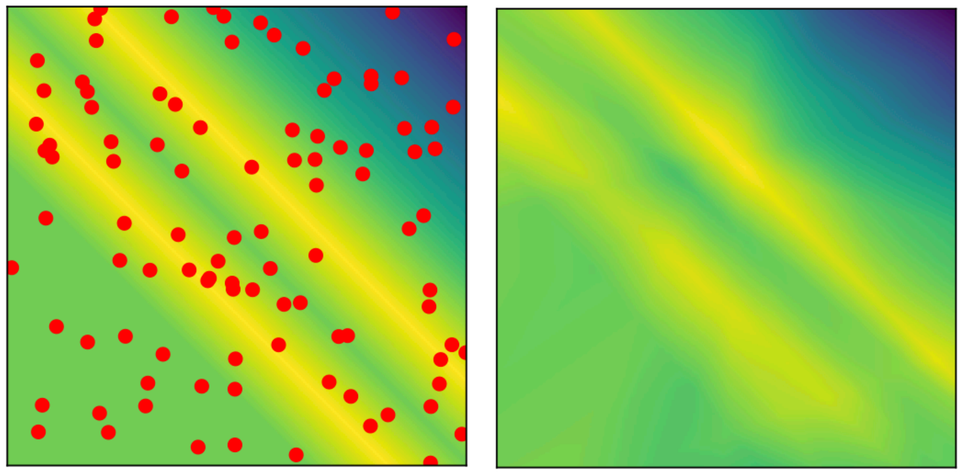
Two-layer neural networks

- $f(x;\theta) = \sum_{j=1}^{m} a_j \phi(w_j^T x + b_j)$
  - Parameters $\theta = (a_j, b_j, w_j)_{j\in[m]} \in (\mathbb{R} \times \mathbb{R} \times \mathbb{R}^d)^m$
  - Rectified Linear Unit (ReLU): $\phi(t) = \max(0,t)$
- Train with gradient descent: $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla \hat{R}(\theta^{(t)})$
  - $\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i;\theta), y_i)$, for n training samples $(x_i, y_i)_{i\in[n]}$
  - Non-convex!



Learning with Wide Neural Nets

- Without data assumptions, doomed to require $\exp(d)$ samples.
- This is known as **curse of dimensionality**.
- Wide neural nets can **beat** the curse of dimensionality for regression. [Bach ’17]
- Adaptivity to smoothness and **low dimensional structure** (data lies on a low dimensional manifold).
- How do NNs achieve this?



[Parhi et al. ’22]

Two training regimes

$$f(x;\theta) = \sum_{j=1}^{m} a_j \phi(w_j^T x + b_j)$$

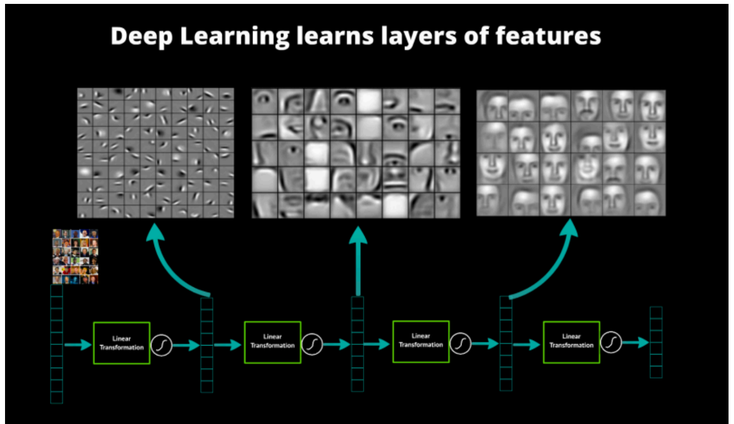
Kernel learning

(a.k.a. random feature model or neural tangent kernel)

- Bottom-layer weights (w_j, b_j) either fixed or remain close to initialization
- Top-layer weights trained with gradient descent
- Convex optimization problem with known convergence rate 🙂

Feature learning

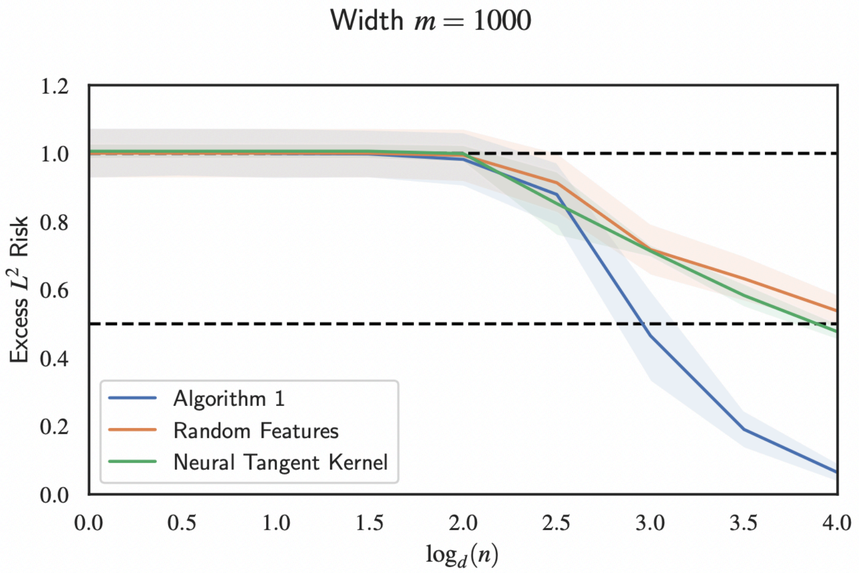
- All weights are trained simultaneously
- Bottom-layer weights move a significant distance from initialization to represent “features” that can be combined
- Non-convex with poorly understood optimization landscape 😬



Theoretical Benefits of Feature Learning

A toy problem [Damian, Lee, Soltanolkotobi ’22]

- Target function $y_i = f^*(x_i)$ for degree-q polynomial $f^*(x) = p(u_1^T x, \ldots, u_r^T x)$
- Non-adaptivity of kernel learning:
  - Sample complexity $n = \Theta(d^q)$
- Adaptivity of feature learning:
  - Sample complexity $n = O(d^2 r + d r^q)$
  - Special learning algorithm:
    train bottom-layer for one step, train top-layer after



Inductive biases + connections to benign overfitting

- **Key idea**: gradient descent with proper initializations is biased in favor of intrinsically simple neural networks
- Recipe for understanding feature learning
    1. Identify inductive biases that influence learning algorithm
    2. Relate that inductive bias to generalization/adaptivity

Variational norm

Weight norm: $\|f(\,\cdot\,;\theta)\|_{\mathscr{R}} = \sum_{j=1}^{m} |\,a_j\,|\,,\ f(x;\theta) = \sum_{j=1}^{m} a_j \phi(w_j^T x + b_j),\ \|w_j\| = 1,$

1. *Identify inductive biases that influence learning algorithm*
   - For large m, gradient descent converges to $\arg\min_{\theta} \hat{R}(\theta) + \lambda \|f(\,\cdot\,;\theta)\|_{\mathscr{R}}$
     [Bach & Chizat, ’21]
2. *Relate that inductive bias to generalization/adaptivity*
   - If target f^* has $\|f^*\|_{\mathscr{R}} \leq B$ for all d, then sample complexity $\tilde{O}(n^{-\frac{d+3}{2d+3}})$
     [Parhi & Nowak, ’22]

Variational norm (ctd.)

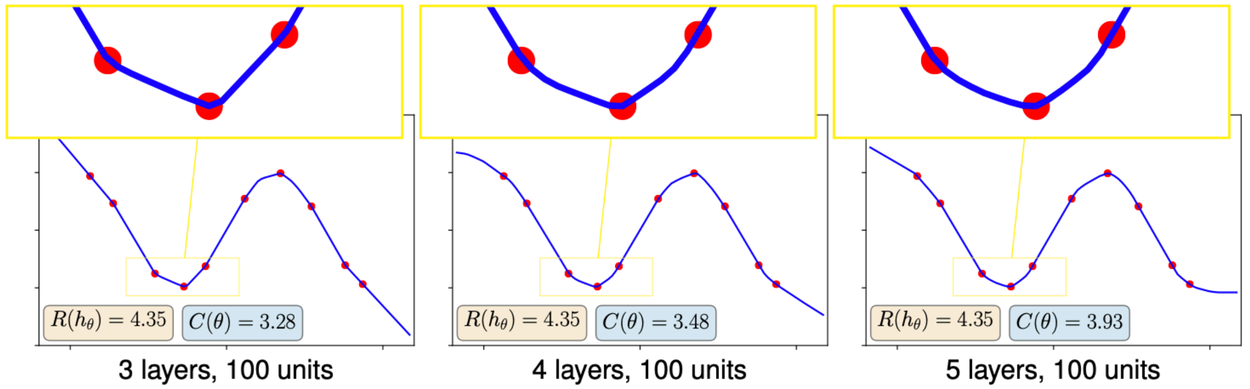
Weight norm: $\|f(\,\cdot\,;\theta)\|_{\mathscr{R}} = \sum_{j=1}^{m} |\,a_j\,|\,,\ f(x;\theta) = \sum_{j=1}^{m} a_j \phi(w_j^T x + b_j),\ \|w_j\| = 1,$

Question: How does $f(\,\cdot\,;\theta)$ for $\theta = \arg\min_{\theta} \hat{R}(\theta) + \lambda \|f(\,\cdot\,;\theta)\|_{\mathscr{R}}$ behave?

A. Piecewise-linear interpolation for $d = 1$ case:

Linear spline interpolation is one solution
[Savarese et al ’19]

All solutions have linear splines when “local convexity” changes
[Hanin ’21]



[Savarese ’19]

Variational norm (ctd.)

Weight norm: $\|f(\,\cdot\,;\theta)\|_{\mathscr{R}} = \sum_{j=1}^{m} |a_j|\,,\ f(x;\theta) = \sum_{j=1}^{m} a_j \phi(w_j^T x + b_j),\ \|w_j\| = 1,$

Question: How does $f(\,\cdot\,;\theta)$ for $\theta = \arg\min_{\theta} \hat{R}(\theta) + \lambda \|f(\,\cdot\,;\theta)\|_{\mathscr{R}}$ behave?

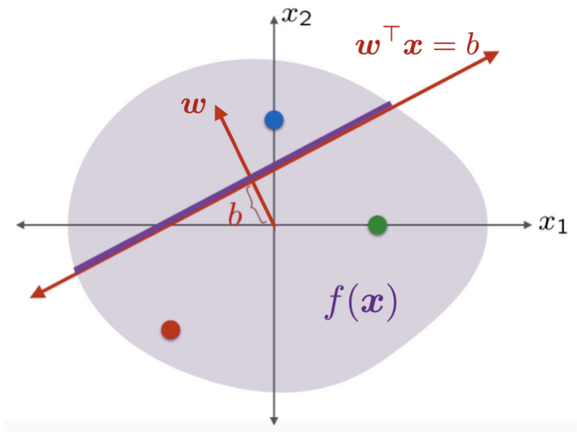
B. Radon transform minimization for general d case:

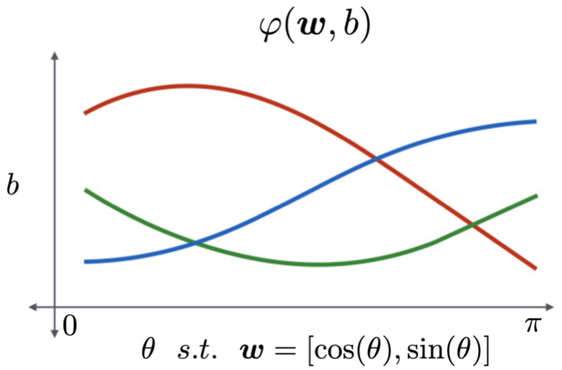
Exact characterization of norm in terms of Radon transform of Laplacian:

$$\|f\|_{\mathscr{R}} = \|\mathscr{R}(\Delta^{\frac{d+1}{2}} f)\|_{\mathbb{L}^1(\mathbb{S}^{d-1}\times[-c_0,c_0])}$$

[Ongie et al, ’19]

… but doesn’t tell us about interpolation





Variational norm (ctd.)

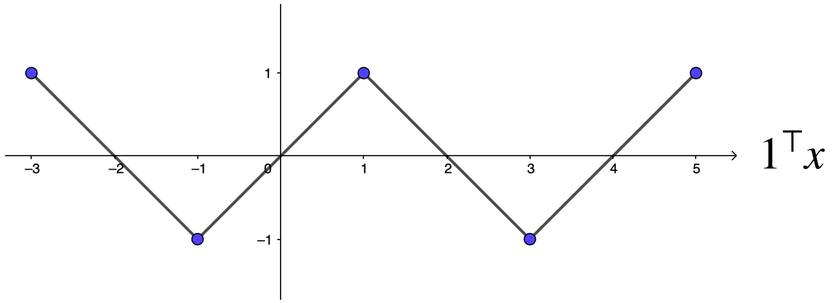
Weight norm: $\|f(\,\cdot\,;\theta)\|_{\mathscr{R}} = \sum_{j=1}^{m} |\,a_j\,|\,,\ f(x;\theta) = \sum_{j=1}^{m} a_j \phi(w_j^T x + b_j),\ \|w_j\| = 1,$

Question: How does $f(\,\cdot\,;\theta)$ for $\theta = \arg\min_{\theta} \hat{R}(\theta) + \lambda \|f(\,\cdot\,;\theta)\|_{\mathscr{R}}$ behave?

C. Inductive bias can favor averaging-based solutions over projections:

Even if data are intrinsically 1-dimensional, $f(\,\cdot\,;\theta)$ may not be.

For parity dataset $(x, x_1 x_2 \ldots x_d)$, for $x \in \{\pm 1\}^d$, $f(\,\cdot\,;\theta)$ averages together partial solutions rather than learning 1-d projection.
[Sanford, Ardeshir, Hsu '22]



Recap

- Classical ML theory fails to explain benign overfitting in neural networks
- Benign overfitting can be analyzed more cleanly in simpler linear models and connected to certain kinds of “simplicity” enabled by a plethora of parameters
- While more difficult to study in neural networks, can consider simplicities that gradient descent is biased towards (e.g. variational norm) and connect to generalization (e.g. dimension adaptivity)
- These notions of simplicity are all intrinsically-linked in various ways (low intrinsic dimensionality, Radon transform, weight norm minimization, spline interpolation, averaging)… future questions involve connecting them.

Thank you