高等机器学习

深层神经网络

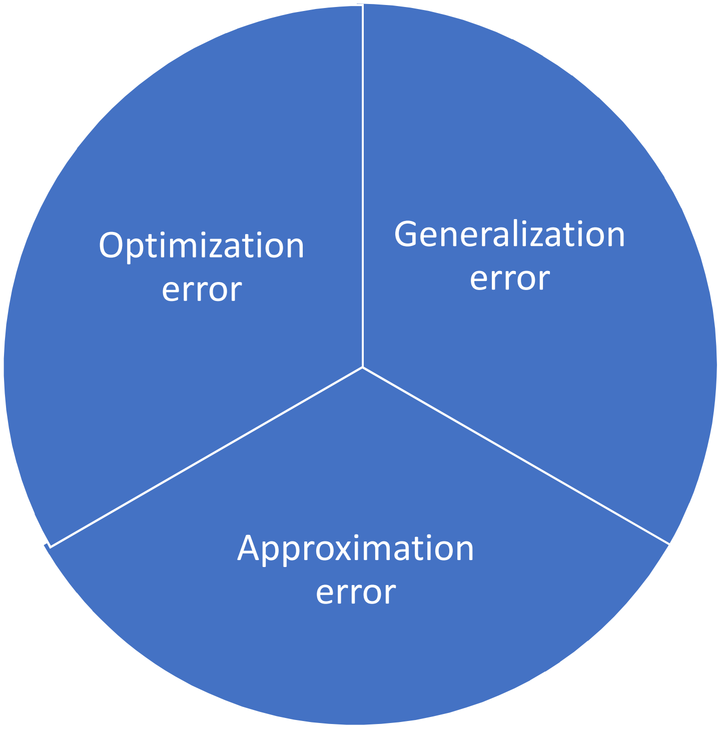
刘铁岩、张辉帅
微软亚洲研究院

Microsoft

清華大學
Tsinghua University

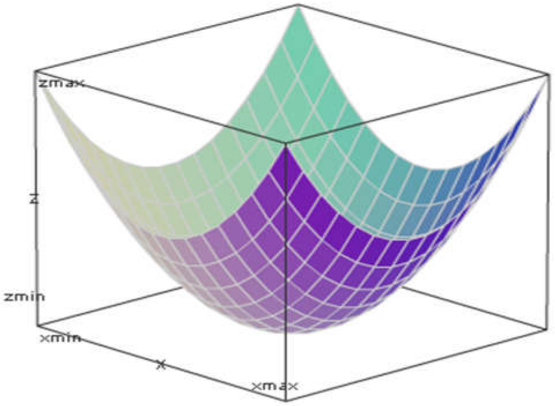
Deep Learning Theory

Deep Learning Theory



Recap of learning theory and deep learning

- What is strongly convex function?
- What is the convergence rate of gradient descent when minimizing a strongly convex function?



- Is deep neural network **convex**? Why?

Recap of learning theory and deep learning

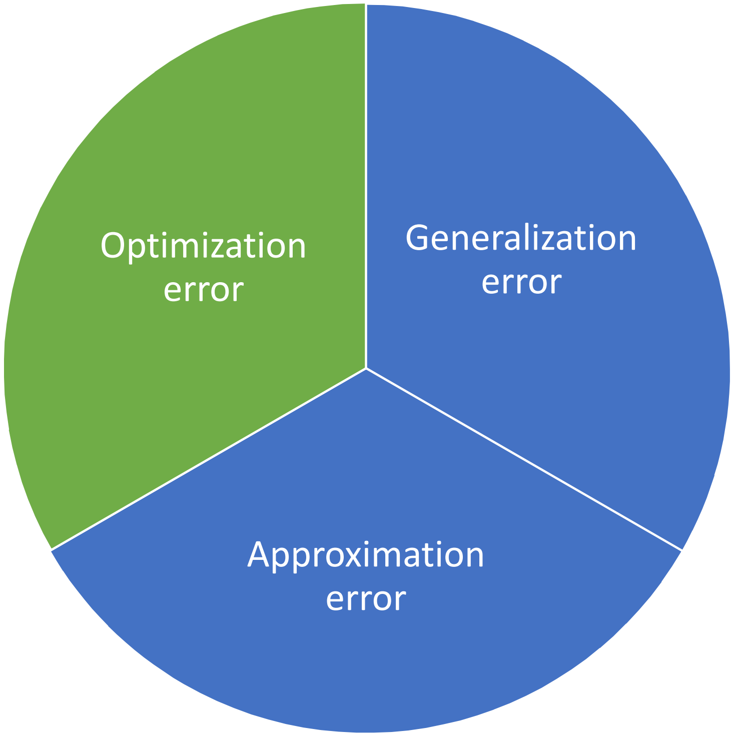
- What is generalization?
- How do we usually characterize it?

- Next, we will show what happens to the generalization bound for deep neural network

Theoretical Challenge of Deep Learning

- Optimization: Non-convex with exponentially many critical points (gradient=0)
- Generalization: Successful deep networks are big models (more parameters than samples).
- They are intertwined together:
  - The choice of optimization algorithm affects the generalization error
  - Regularizers for generalization also change the algorithm dynamics and loss landscape.
  - Practical observation: gradient methods find high quality solution.
- Approximation: show the benefit of depth.

Deep Learning Theory



Over-parameterization Eases Optimization

- *Understand how over-parametrization helps training*
  - Though the loss of DNN is nonconvex, over-parameterized NN behaves like “convex” near the initialization
  - Coincide with practical observation: gradient methods find high quality solutions for wide DNN
  - Gradient descent converges to global minima for training over-parametrized neural network

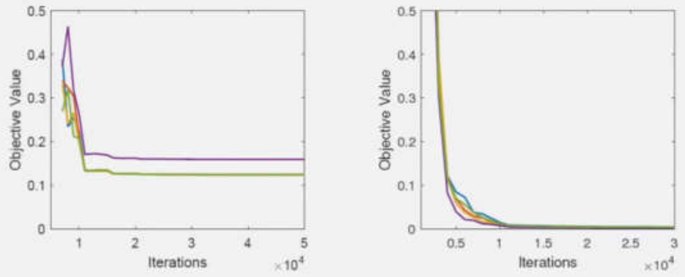
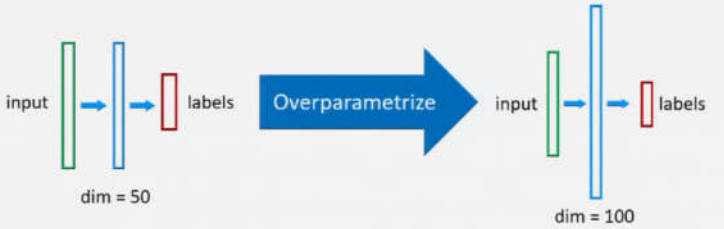


Figure: Experiment first done by Livni-Shalev-Shwartz-Shamir 2014



Theory of Optimizing Deep Neural Network

Question:
When gradient methods (GD/SGD) are successful for optimizing over-parameterized deep neural network?

Convergence Theory for Over-parameterized DNN (Allen-Zhu et al. 2019)

Problem Setup:

- Network Input: $x \in \mathbb{R}^p$. Network Output: $y \in \mathbb{R}^d$
- Network structure
  - Input layer: $h_0 = \phi(Ax)$; $L-1$ layers: $h_l = \phi(W_l h_{l-1})$; Output layer: $y = Bh_L$;
  - $\phi(\cdot)$ is ReLU activation
- Random initialization

$$[W_l]_{i,j} \sim \mathcal{N}\left(0, \frac{2}{m}\right); A_{i,j} \sim \mathcal{N}\left(0, \frac{2}{m}\right); B_{i,j} \sim \mathcal{N}\left(0, \frac{1}{d}\right)$$

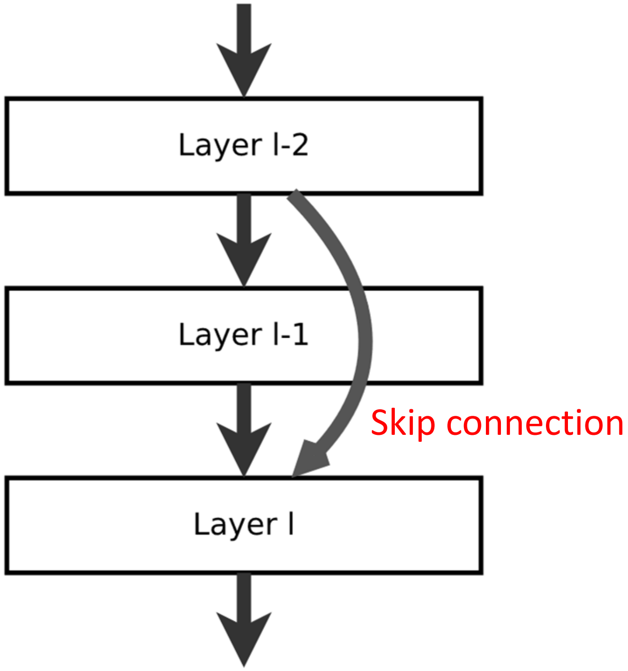
1. The wider the network, the easier the optimization;
2. The deeper the network, the harder the optimization.

starting from random initialization, with high probability, gradient descent with learning rate $\eta = \Theta\left(\frac{d}{\text{poly}(n,L)\cdot m}\right)$ finds a solution with error less than ϵ in $T = \Theta\left(\text{poly}(n, L)\log\frac{1}{\epsilon}\right)$ iterations.

What about ResNet?

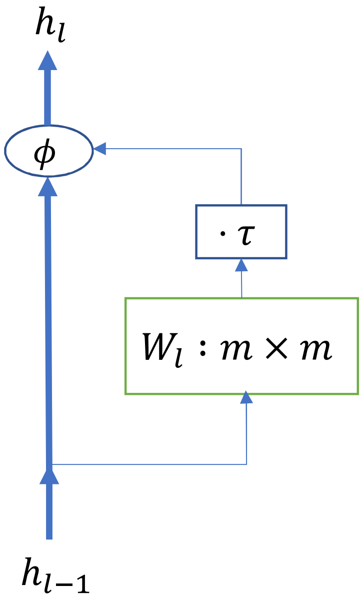
- In practice, ResNet behaves differently from PlainNet:
  - one can train 1000-layer ResNet, but fail to train 30-layer PlainNet.

Does the skip-connection in ResNet help with the convergence of gradient descent?



Problem Setup

- Network Input: $x \in \mathbb{R}^p$. Network Output: $y \in \mathbb{R}^d$
- Network structure
  - $L-1$ residual layers: $h_l = \phi(h_{l-1} + \tau W_l h_{l-1})$,
  - $\phi(\cdot)$ is ReLU nonlinear activation function.
- Random initialization
  - $[W_l]_{i,j} \sim \mathcal{N}\left(0, \frac{2}{m}\right)$;



Convergence Theory for Over-parameterized ResNet

Informal theorem (Zhang et al. 2019): For ResNet defined above and $\tau = O(\sqrt{1/L})$, if the network width

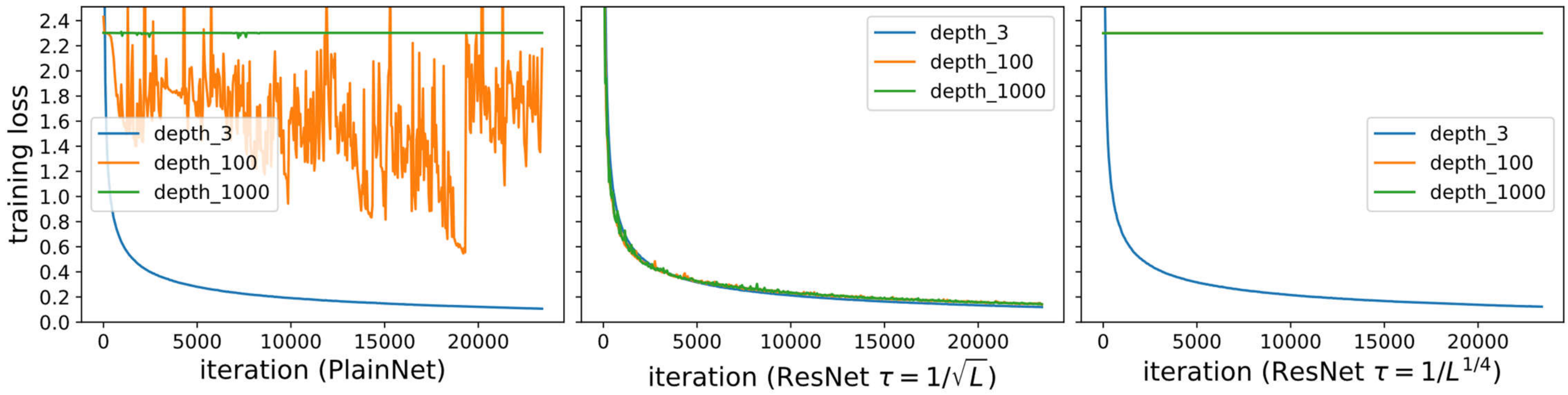
$$m \geq \Omega(\mathrm{poly(n, L)} \log m),$$

with high probability, GD finds an ϵ-optimal solution in $\Theta\left(\mathrm{poly}(n)\log\frac{1}{\epsilon}\right)$ steps.

For $\tau > \Omega(\sqrt{1/L})$, the output of network explodes in expectation as L increases.

- For $\tau = O(\sqrt{1/L})$, the width requirement is smaller than DNN
- The bound $\tau = O(\sqrt{1/L})$ is sharp for bounded network output with arbitrary depth.

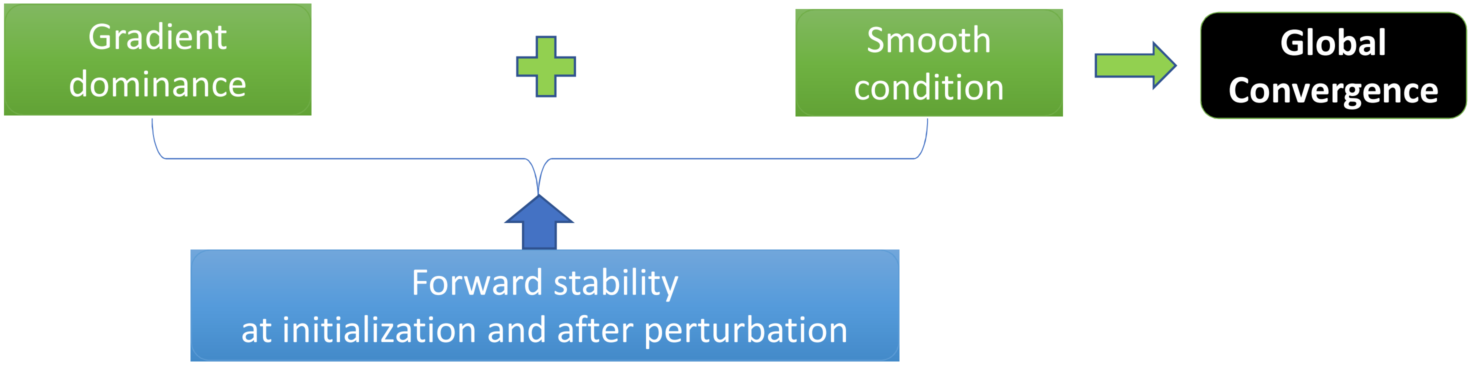
MNIST classification, network width 128



Remark 1: ResNets with $\tau \leq O(1/\sqrt{L})$ are easier to train than PlainNet.

Remark 2: ResNet explodes for $\tau = 1/L^{1/4}$ for depth L = 100 and 1000.

Overall picture proof



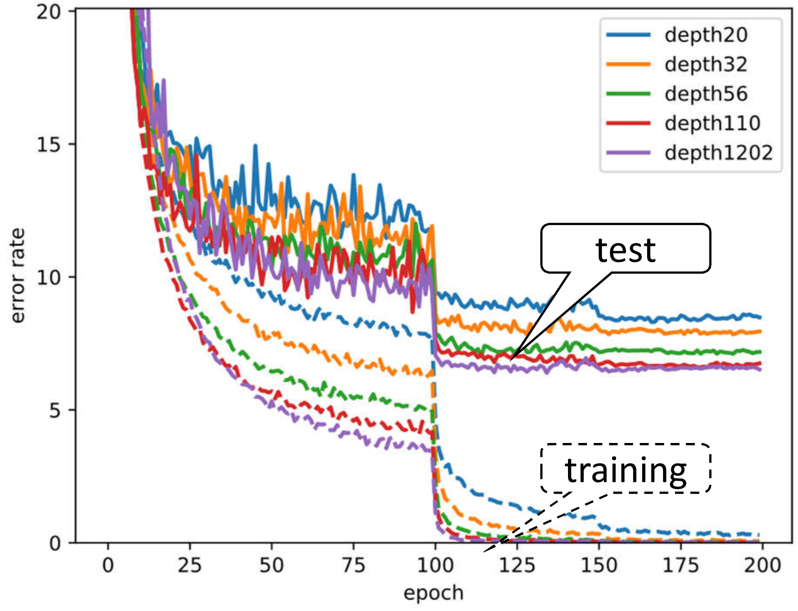
- For ResNet proof: To keep the output $f_\theta(x) = O(1)$, need a spectral norm bound for $\tau = 1/\sqrt{L}$

$$\|(I + \tau W_b) \cdots (I + \tau W_a)\|_2 \leq 1 + c$$

New practice for ResNet design

- With $\tau \sim 1/\sqrt{L}$, we can train very deep ResNet even without batch normalization.
- The output norm explodes after several residual blocks if without τ.

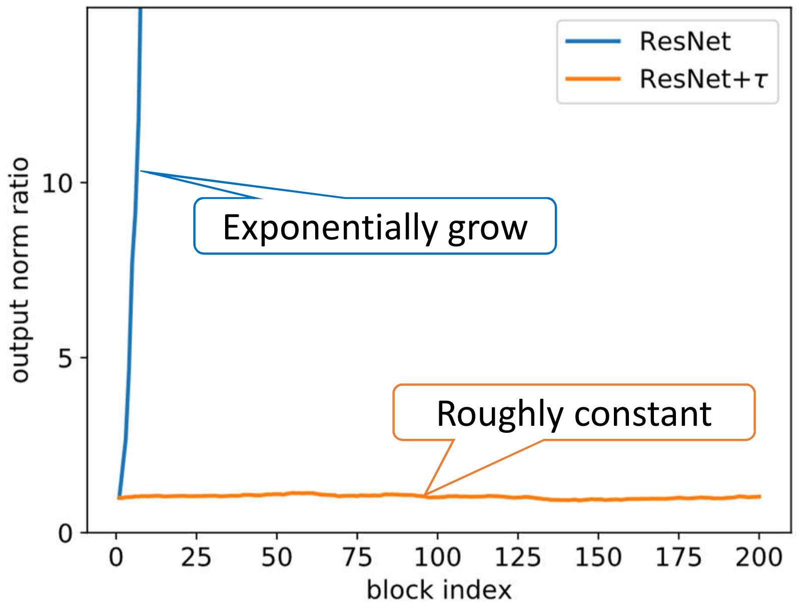
Training/test curves for CIFAR10



test

training

Layer output norm of ResNet 1202

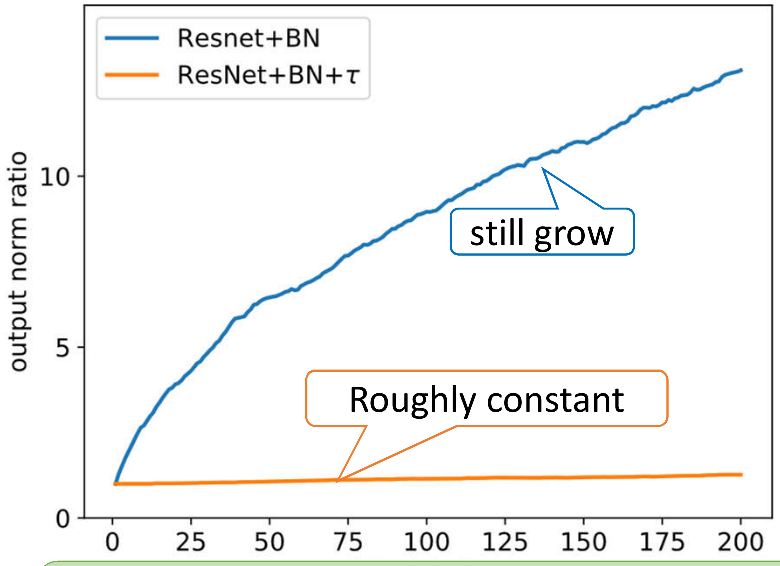


Exponentially grow

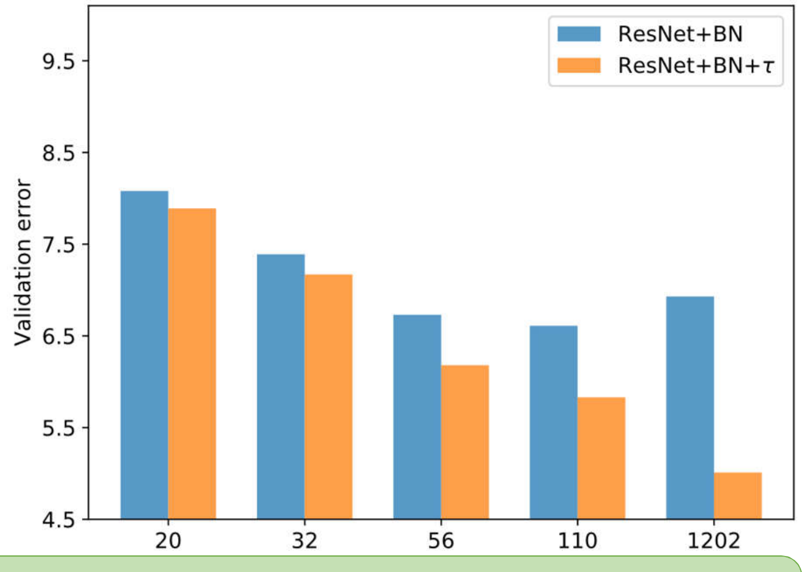
Roughly constant

What if there is *batch normalization (BN)* ?

- The output norm of each layer for ResNet 1202 (with BN)



Validation error on CIFAR10



With $\tau \sim 1/\sqrt{L}$, the deeper the network, the bigger the performance gain.

Class project

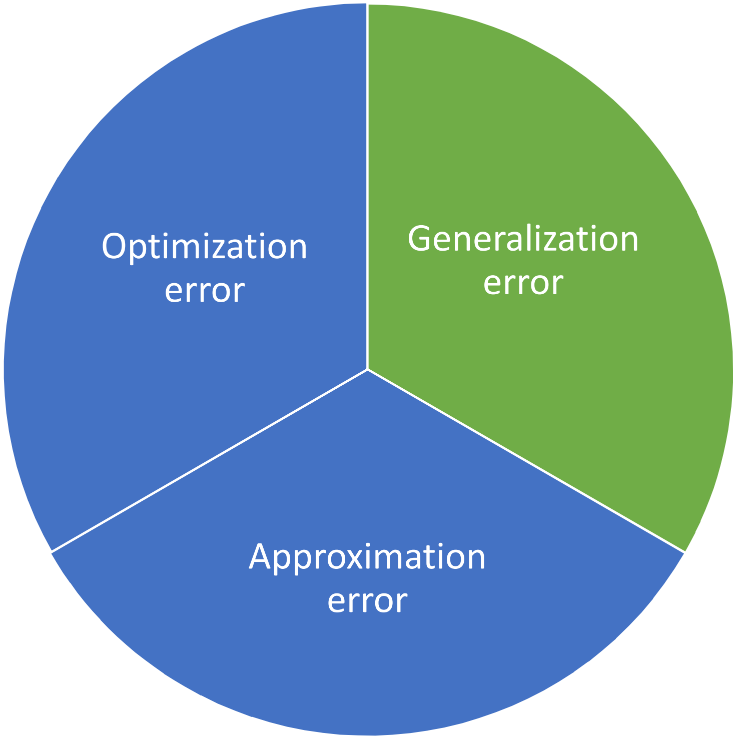
- Background: Two lines of over-parameterization analysis for studying NN.
  - Neural Tangent Kernel (NTK, what we have talked about), where both intermediate layers and last layer are trained with (stochastic) gradient descent. We have shown that gradient descent converges to the global minima if the network width is polynomial with the sample size and the network depth.
  - Random feature kernel, where the intermediate layers are fixed and only the last layer are trained. Recently [Kawaguchi and Huang, 2019] reduced the over-parameterization requirement (to linear dependence of the sample size and depth) by using the random feature approach.
- Problem: Reduce the over-parameterization requirement with all layers trained. Is there similar improvement for the ResNet case?
  - You can choose ReLU or other activation function, with or without ResNet.

Over-parameterization: not the end

- It turns out the above proofs rely on the fact the gradient descent does not move far from the initialization.

- There is a gap between practical training and the above regime
  - In practice, gradient descent moves a moderate distance from the initialization
  - It is now not clear how to characterize the real NN regime. But tons of ideas pour in.

Deep Learning Theory



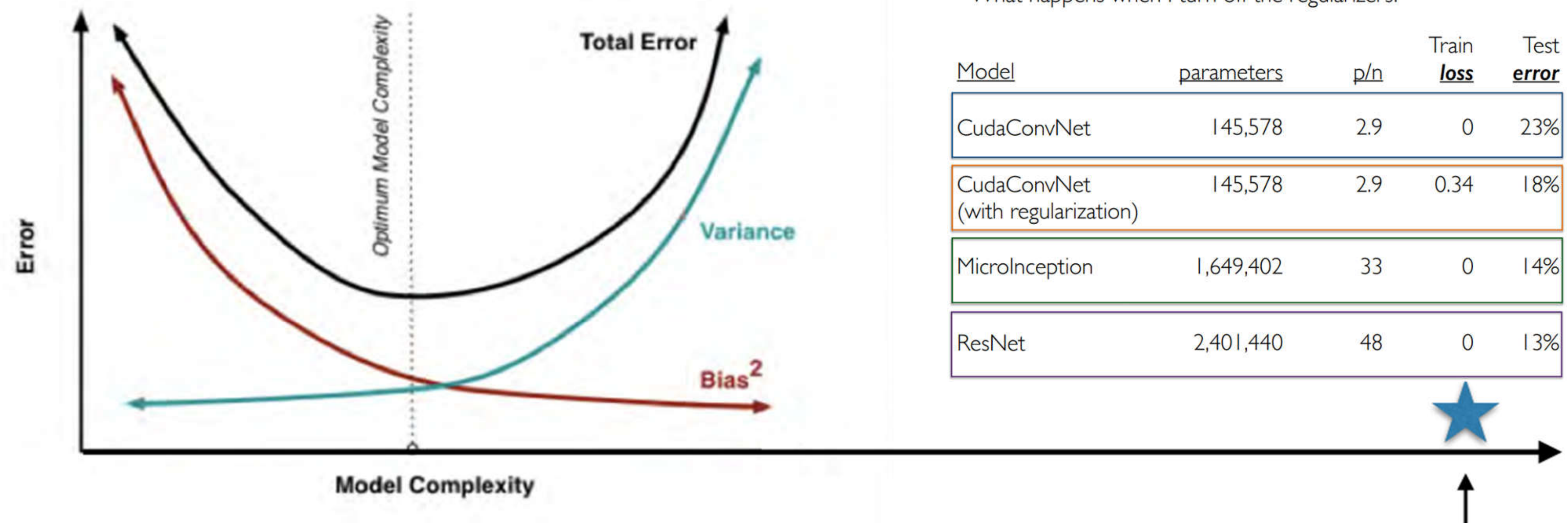
Generalization of Deep Neural Networks

CIFAR10

n=50,000
d=3,072
k=10

What happens when I turn off the regularizers?

Model	parameters	p/n	Train ***loss***	Test ***error***
CudaConvNet	145,578	2.9	0	23%
CudaConvNet (with regularization)	145,578	2.9	0.34	18%
MicroInception	1,649,402	33	0	14%
ResNet	2,401,440	48	0	13%



Deep models

VC-dimension of Deep Neural Networks

Suppose $\mathcal{F}$ contains neural networks with p parameters, m nodes and L layers, then $d_{VC}(\mathcal{F})$ has the following bounds:

Activation Function	$d_{VC}(\mathcal{F})$
Linear threshold (Baum 1989)	$\tilde{O}(p)$
Piecewise polynomial (Bartlett 1998)	$\tilde{O}(pL^2)$
Piecewise linear (Bartlett et al. 2017)	$\tilde{O}(pL)$

- The role of depth L is not very clear in some cases.
- Since $\boldsymbol{p}$ could be very large in big networks, the VC-based generalization bounds are very **loose**!

Rademacher Average of Deep Neural Networks

- Let $\mathcal{F}_{L,\boldsymbol{W}}$ denote the functional class of L-layer neural network parameterized by $\boldsymbol{W} = \{\boldsymbol{W}_1, \dots, \boldsymbol{W}_L\}$, i.e., $\forall\, f \in \mathcal{F}_{L,\boldsymbol{W}}$,

$$f(x) = \boldsymbol{W}_L \sigma(\boldsymbol{W}_{L-1}(\dots \sigma(\boldsymbol{W}_1 x) \dots),$$

where $\|\boldsymbol{W}_i\|_F \leq B$, then Rademacher Average has an upper bound (Neyshabur et al. 2015):

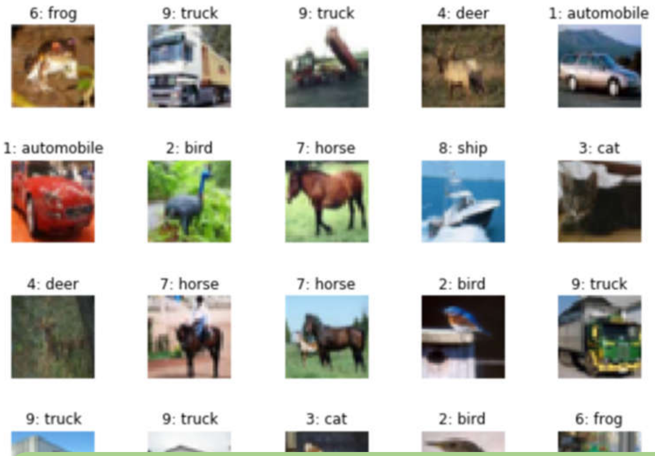
$$RA(\mathcal{F}_{L,\boldsymbol{W}}; n) \leq O\left(\frac{B^L}{\sqrt{n}}\right)$$

- Not rely on the number of parameters.
- Grow exponentially with L.

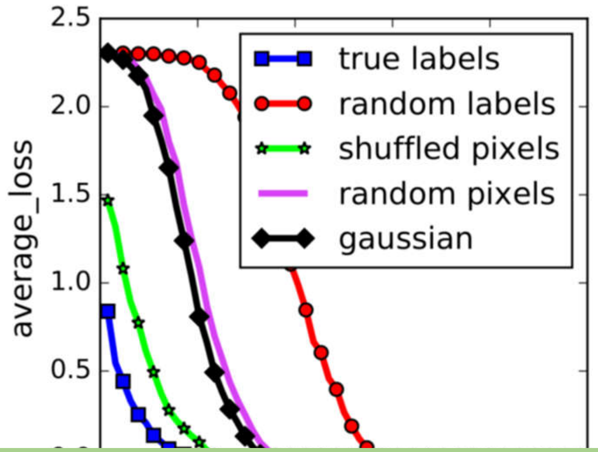
Margin bound

- Define the margin as $f(x)y$, we have $L(f) \lesssim \hat{L}_\gamma(f) + \frac{RA(\mathcal{F};n)}{\gamma}$.

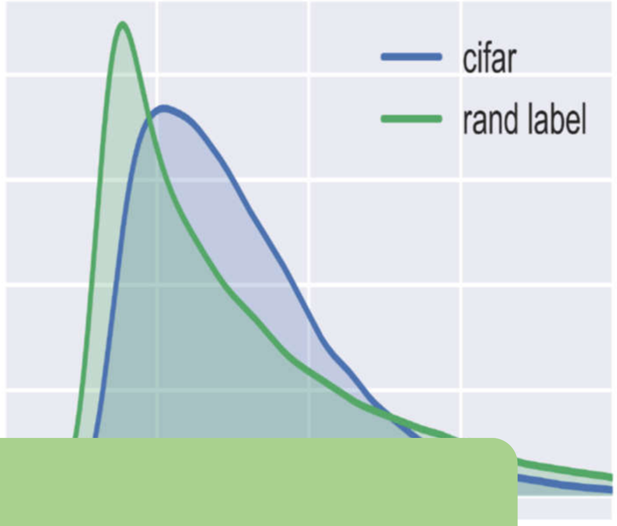
CIFAR 10 dataset



Training error (Zhang et al. 2017)



Margin (Bartlett et al. 2017)

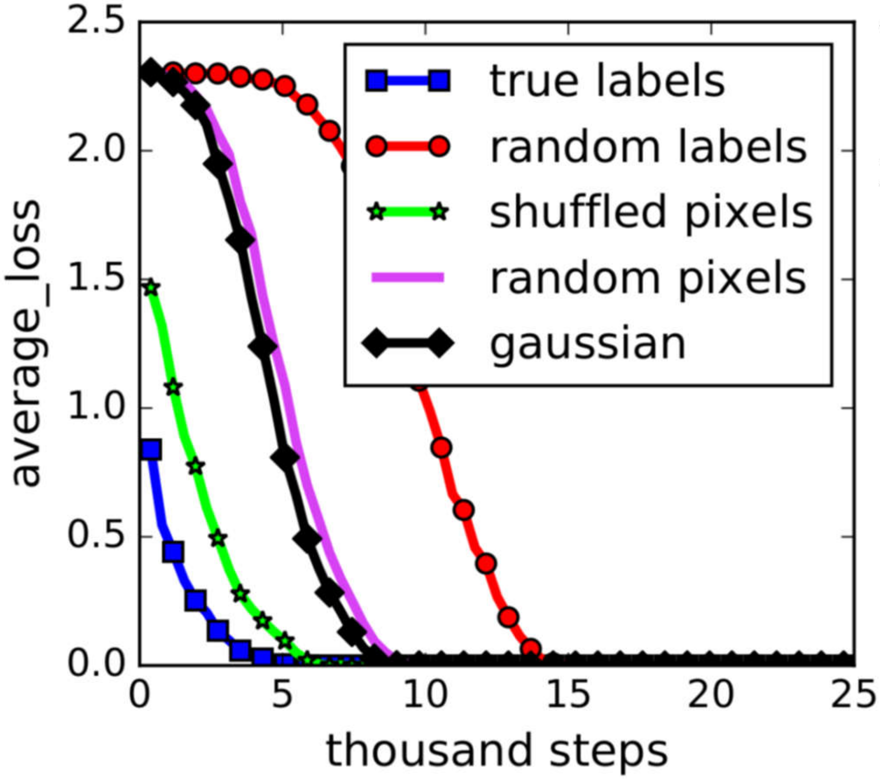


Margin bound cannot explain generalization

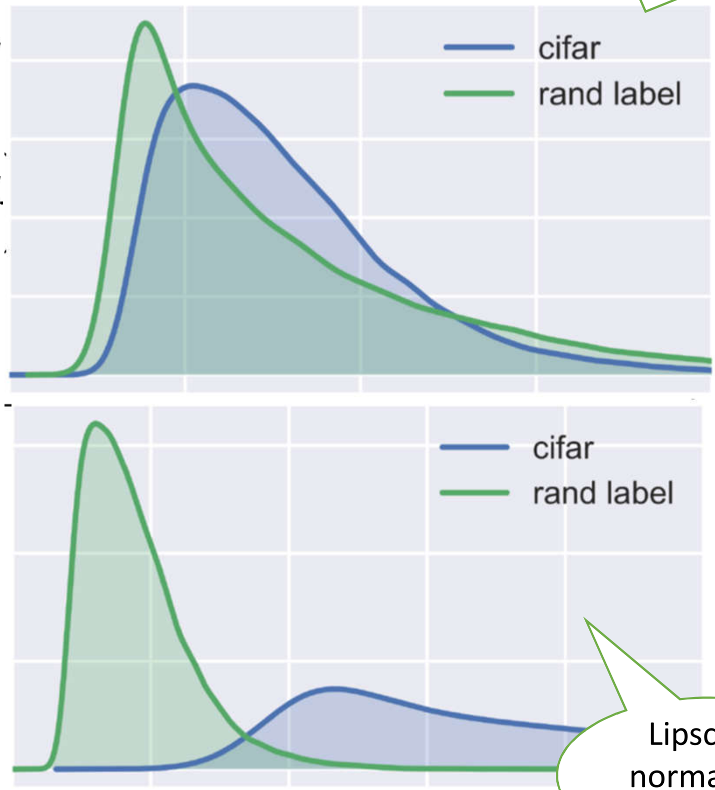
Network Lipschitz normalized Margin Bound (Bartlett et al. 2017)

Theorem. F

where R_W



Margin



Lipschitz normalized margin

Recall what is generalization

$$L(f_n^*) - \widehat{L_n}(f_n^*) \leq \sup_{f \in \mathcal{F}} \left| \widehat{L_n}(f) - L(f) \right| \leq \epsilon(Cap(\mathcal{F}), n)$$

- For any f, $L(f) - \widehat{L_n}(f)$ is small w.h.p. because of the Large Number Law
- The union bound breaks down because of the large capacity of the function class.

Can we bypass the union bound?

PAC Bayes Generalization Bound

- Before data, we have a prior distribution on the hypothesis space: Q_0.
- Then after learning procedure, we have a posterior distribution: Q.
- Predictions:
  - Draw $h \sim Q$ and predict with the chosen h.
  - Each prediction with a fresh random draw.
- The risk measure is now given by integral:

$$L(Q) \coloneqq \int_{\mathcal{H}} L(h) dQ(h), \qquad \hat{L}(Q) \coloneqq \int_{\mathcal{H}} \hat{L}(h) dQ(h),$$

- PAC-Bayes bound:
  - Choose a Q_0. For any sample size n, for any $\delta \in (0,1), w.p. \geq 1 - \delta$,

$$\forall Q, \qquad KL(L(Q)||\hat{L}(Q)) \leq \frac{1}{n} KL(Q||Q_0) + \frac{1}{n} \log \frac{m+1}{\delta}$$

Class project: PAC Bayes bound and (normalized) flat minima

$$L(Q) \le \hat{L}(f) + \hat{L}(Q) - \hat{L}(f) + \lambda^{-1} KL(Q||Q_0) + \lambda^{-1}\left(\ln 1/\delta + \psi(\lambda, n)\right)$$

- Original Flatness (at minima):

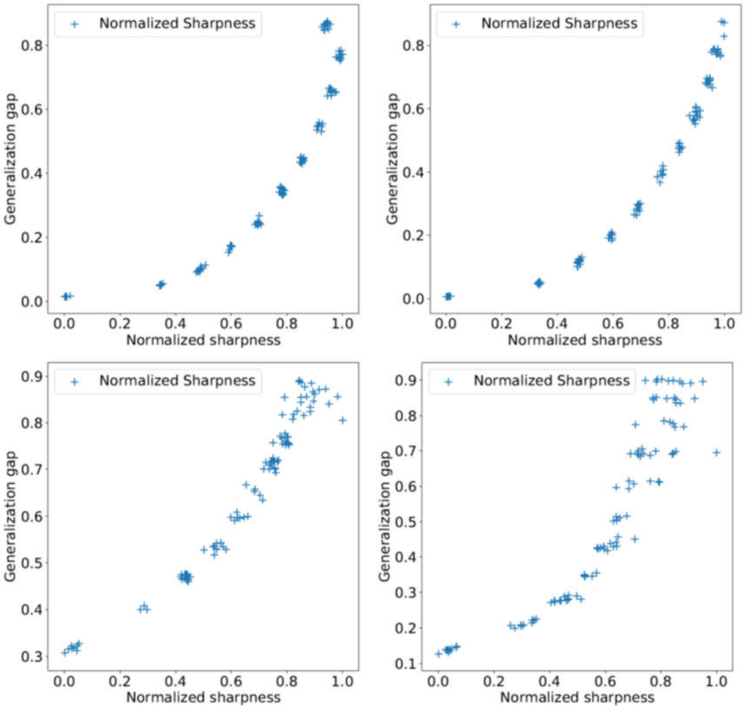
$$\hat{L}(Q) - \hat{L}(f) \approx \mathbb{E}_\epsilon \hat{L}(f_{W+\epsilon}) - \hat{L}(f_W) \approx$$

- KL term: $KL(Q||Q_0) = \sum_l \ln \frac{\delta^2_{Q_0,l}}{\delta^2_{Q,l}} + \frac{\|W_l\|_F^2 + \delta^2_{Q,l}}{2\delta^2_{Q_0,l}} \ge \sum_l$

  $\delta^2_{Q_0,l} = \delta^2_{Q,l} := \delta^2_l$

- Minimizing flatness and KL term jointly gives an invar

$$\sqrt{\frac{2}{\lambda} \sum_l \sqrt{\|W_l\|_F^2 \mathrm{Tr}(H_l) +}}$$



Key technical idea.

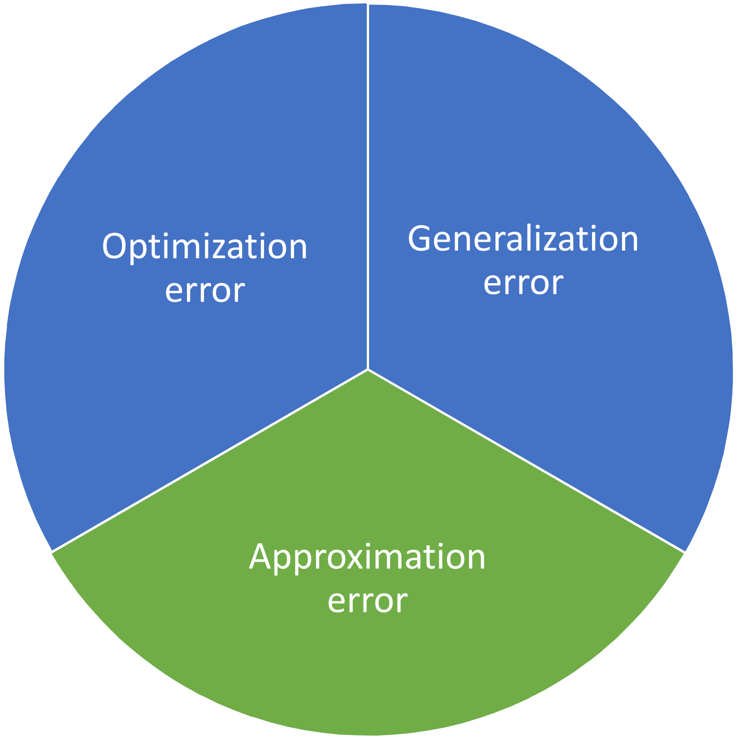
- C could be the number of layers or neurons or weights (for finer-grained invariant flatness). The bound could be as bad as parameter counts.

- How to reduce C and give a tighter PAC-Bayes bound?
  - Employing the equivalence of NN, find equivalent models $\hat{f}$ such that for all k, l

$$\frac{\left\|\widehat{W}_l\right\|_F^2}{\widehat{H}_l} = \frac{\left\|\widehat{W}_k\right\|_F^2}{\widehat{H}_k}$$

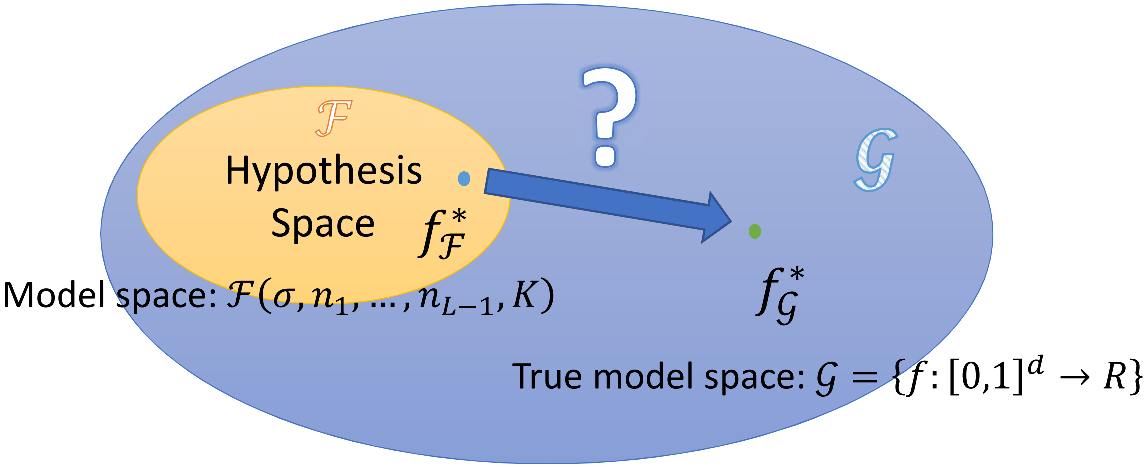
- Moreover, Hessian is not PSD and could be very small for certain loss. We may consider other invariant flatness measure: Fisher norm, JJ^T norm.

Deep Learning Theory



Approximation Theory

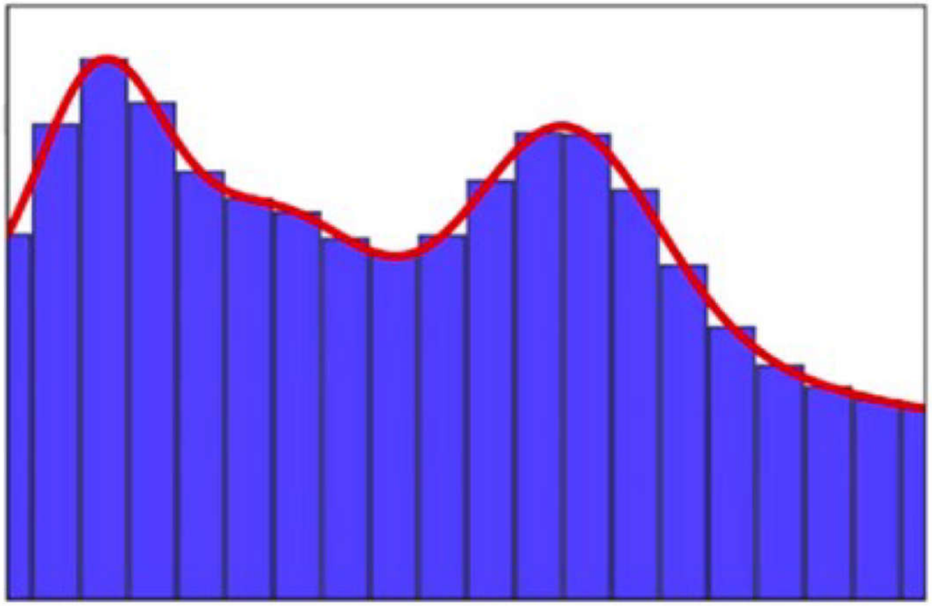
- Representation power of the function class of deep neural networks

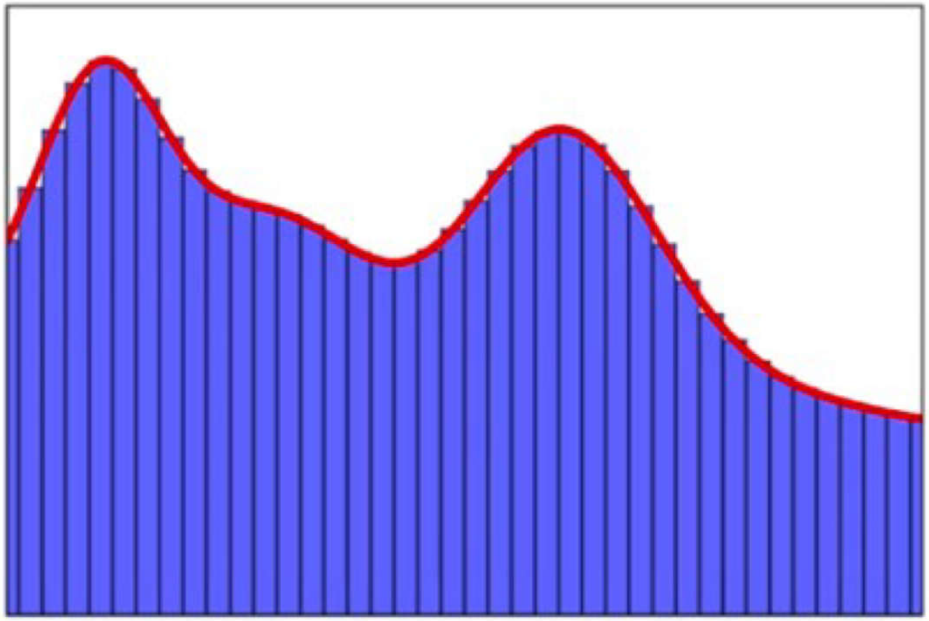

$\mathcal{F}$
Hypothesis Space
$f_{\mathcal{F}}^*$
?
$\mathcal{G}$
$f_{\mathcal{G}}^*$
Model space: $\mathcal{F}(\sigma, n_1, \dots, n_{L-1}, K)$
True model space: $\mathcal{G} = \{f: [0,1]^d \to R\}$

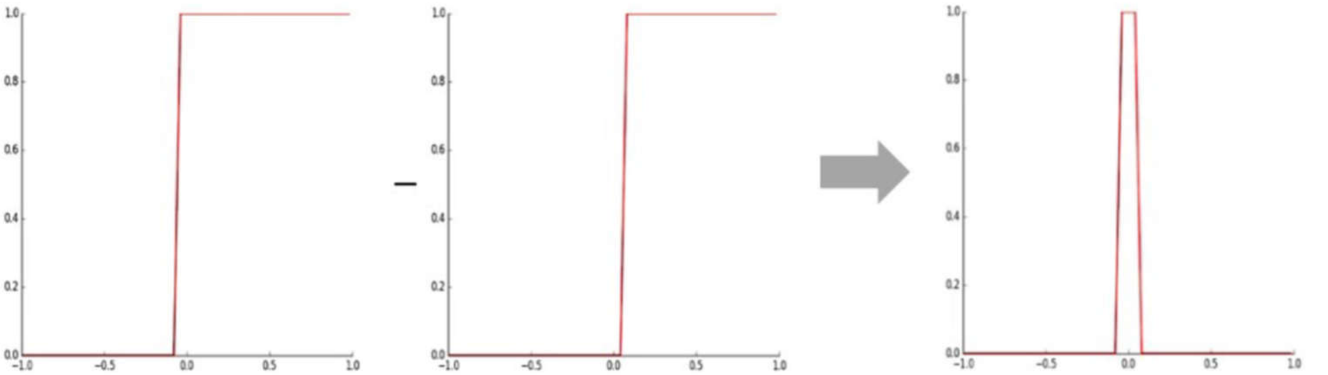
Universal Approximation of Neural Networks

- (Hornik 1989) Feedforward networks with only a single hidden layer can approximate any continuous function **uniformly** on any compact set and any measurable function arbitrarily well.
- For example, $\forall f \in C\left([0,1]^d\right), \forall \epsilon > 0, \exists$ 2-layer neural network $NN, s.t.$

$$\forall x \in [0,1]^d, |NN(x) - f(x)| \leq \epsilon.$$






1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
–
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0

Universal Approximation of Deep NN

	Function class	Activation function	#Hidden layers	#neurons	Approximation error
Barron (1993)	Certain smoothness	Sigmoid	1	ϵ^{-d}	ϵ
Liang & Srikant (2017)	functions with enough piecewise smoothness	ReLU and binary step unit	$\Theta\left(\log\left(\frac{1}{\epsilon}\right)\right)$	$O\left(poly\log\left(\frac{1}{\epsilon}\right)\right)$	ϵ
	differentiable and strongly convex functions	ReLU and binary step unit		$\Omega(\log(1/\epsilon))$	ϵ

Shallow networks require exponentially more neurons than a deep network to achieve the level of accuracy for function approximation.

Benefit of Depth v.s. benefit of width

Depth efficiency:

- Cohen et al. (2016): Exist deep CNN ReLU networks that cannot be approximated by shallow ones unless its size is exponentially large.
- Telgarsky (2016): Construct a network with $O(k^3)$ layers and $O(1)$ neurons per layer that cannot be approximated by any network with $O(k)$ layers and $O(2^k)$ neurons per layer.

- Width efficiency:
  - Lu et al. (2017): Exist networks of width $O(k^2)$ and depth 2 that cannot be approximately by any width $O(k^{1.5})$ and depth k network.

This efficiency of depth and width are not universal.

Reference

- Z. Allen-Zhu, Y. Li and Z. Song. Convergence theory for learning DNN via over-parameterization. ICML 2019
- H. Zhang, D. Yu, M. Yi, W. Chen and T-Y Liu. Convergence theory for learning over-parameterized ResNet: a full characterization. ArXiv 2019
- Baum, Eric B., and David Haussler. "What size net gives valid generalization?." *Advances in neural information processing systems*. 1989.
- Bartlett, Peter L., Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks." *Advances in Neural Information Processing Systems*. 1999.
- B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. COLT, 2015
- Bartlett P L, Harvey N, Liaw C, et al. Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks[J]. Journal of Machine Learning Research, 2019, 20(63): 1-17.
- Bartlett, Peter L., Dylan J. Foster, and Matus J. Telgarsky. "Spectrally-normalized margin bounds for neural networks." *NIPS* 2017.
- John Shawe-Taylor and Omar Ravisplata. Statistical Learning Theory: A Hitchhiker’s Guide. NIPS 2018.
- Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural networks, 1989, 2(5): 359-366.
- Barron A R. Universal approximation bounds for superpositions of a sigmoidal function[J]. IEEE Transactions on Information theory, 1993, 39(3): 930-945.
- S. Liang and R. Srikant. WHY DEEP NEURAL NETWORKS FOR FUNCTION APPROXIMATION? ICLR 2017
- Nadav Cohen, Or Sharir and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. COLT, 2016.
- Matus Telgarsky. Benefits of depth in neural networks. JMLR 2016
- Z. Lu, H. Pu, F. Wang, Z. Hu and L. Wang. The Expressive Power of Neural Networks: A View from the Width, NIPS 2017