Rethinking classic learning theory in deep neural networks

Hikaru Ibayashi@CSCI699 (Feb., 15th, 2023)

Today's papers

- Understanding deep learning requires rethinking generalization
  - An experiment suggests we need to "rethink" classic learning theory
- Uniform convergence may be unable to explain generalization in deep learning
  - Proposed a learning task where classic learning theory provably fails

Understanding deep learning requires rethinking generalization [2]

Notations for Classification Tasks

- $x \in \mathcal{X}$: Input
- $y \in \mathcal{Y}$: Label
- $S = \left((x_1, y_1), \ldots, (x_m, y_m) \right)$: Training set
- $\mathcal{H}$: Hypothesis class
- $\mathcal{A}\ (S \to \mathcal{H})$: Learning algorithm

- $\mathcal{L}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell \left(h\left(x_i\right), y_i \right)$: Train loss
- $\mathcal{L}_{\mathcal{D}}(h) = \mathrm{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$: Test loss
- $\Delta h = \mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_S(h)$: Generalization error

Can we give a bound?

Complexity measures and generalization error bound

- Complexity measures
  - Measures of how complex a hypothesis class is
  - The less complex, the more generalizing
- VC dimension: Roughly corresponds to the # of parameters
  - Bound: $\Delta h \leq \frac{1}{\delta}\sqrt{\frac{2VC(\mathscr{H})}{m}}$ with probability $1-\delta$ (Theorem 6.11 in [1])
- Rademacher complexity: How wrong a hypothesis can be
  - $RC(\ell \circ \mathscr{H}, S) = \frac{1}{m} \underset{\boldsymbol{\sigma}\sim\{\pm 1\}^m}{\mathrm{E}} \left[\sup_{h\in\mathscr{H}} \sum_{i=1}^{m} \sigma_i \, \ell\left(h\left(x_i\right), y_i\right) \right]$
  - Bound: $\Delta h \leq 2 \underset{S\sim D^m}{\mathrm{E}} \left[RC(\ell \circ \mathscr{H}, S) \right]$ (Theorem 26.3 in [1])

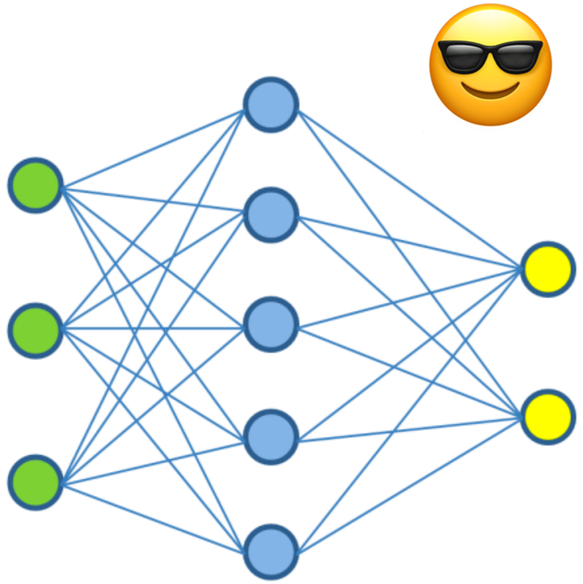
When it comes to DNN, we need to “rethink” those

Randomization test

CIFAR-10

horse bird truck

• • •

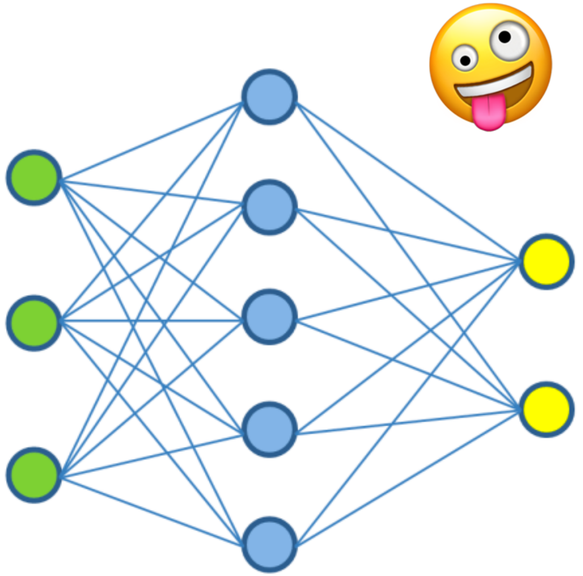


- Zero training error
- Strong generalization

CIFAR-10 with random labels

truck horse bird

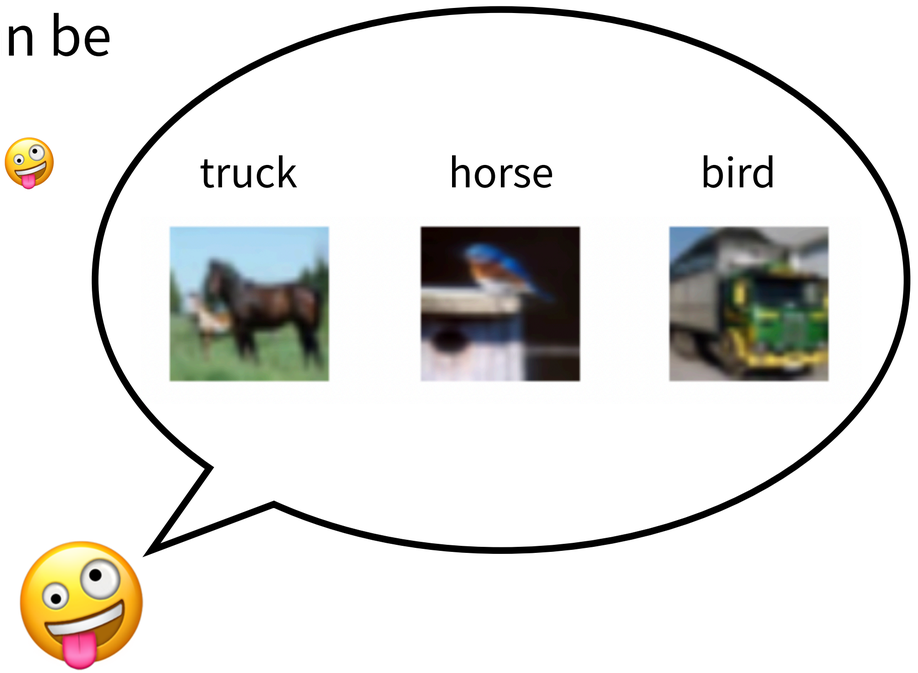
• • •



- Zero training error
- No generalization

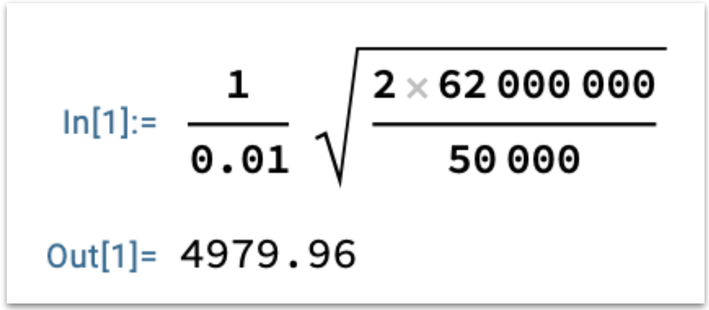
Failure of classic complexity measures

- VC dimension
  - Can't distinguish 😎 and 🤪 because they both have $VC(\mathscr{H})$
- Rademacher complexity
  - Recall that Rademacher complexity means how wrong h can be
  - Randomization test showed h can be super wrong, such as 🤪
  - $RC(\ell \circ \mathscr{H}, S) = \frac{1}{m} \underset{\boldsymbol{\sigma} \sim \{\pm 1\}^m}{\mathrm{E}} \left[\sup_{h \in \mathscr{H}} \sum_{i=1}^{m} \sigma_i \, \ell\left(h\left(x_i\right), y_i\right) \right] = 1$
  - $\Delta h \leq 2 \underset{S \sim D^m}{\mathrm{E}} \left[RC(\ell \circ \mathscr{H}, S) \right] = 2 \cdot 1$
  - Because $-1 \leq \Delta h \leq 1$, this bound is **vacuous**



Failure of classic complexity measures

- Generalization error bound via VC dimension
  - AlexNet trained with the CIFAR-10 dataset
  - $VC(\mathcal{H})$ ~ # of parameters = 62,000,000
  - $m = 50{,}000$
  - let $\delta = 0.01$
  - $\Delta h \leq \frac{1}{0.01}\sqrt{\frac{2 \times 62{,}000{,}000}{50{,}000}}$

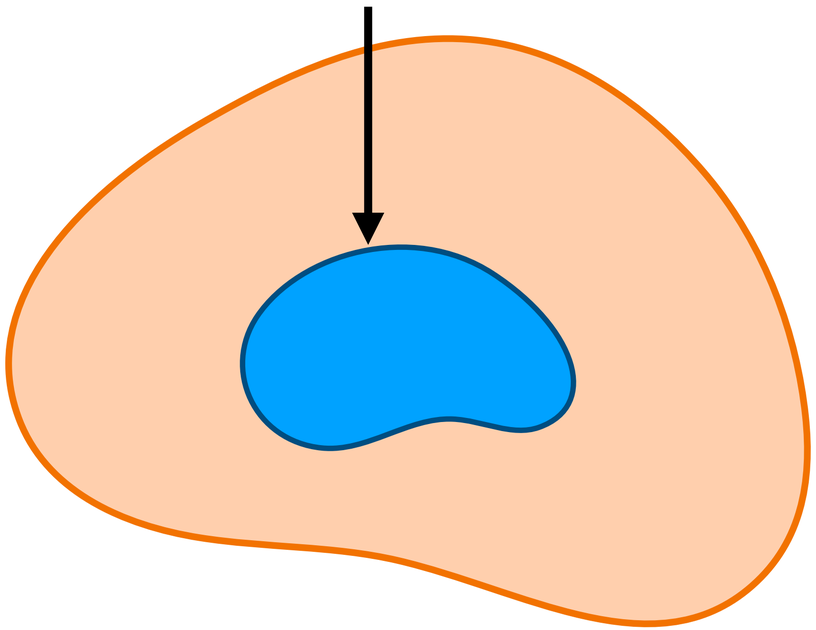


Role of regularization

- It's common to use regularization for generalization
  - Data Augmentation, Weight decay, Dropout
  - But they are neither necessary nor sufficient
- Also, there are tricks (not for generalization) that have **implicit regularization** effects
  - early stopping, batch normalization
- There could be some **unidentified implicit regularization** effect of $\mathscr{A}$

  → Invoked many research attempts trying to identify that

$\mathscr{H}_{\mathscr{A}}$: Hypothesis class returned by $\mathscr{A}$



Strong generalization without weight decan and dropout

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with the original labels						
yes	yes	yes	92.18	99.21	77.84	93.92
yes	no	no	92.33	99.17	72.95	90.43
no	no	yes	90.60	100.0	67.18 (72.57)	86.44 (91.31)
no	no	no	99.53	100.0	59.80 (63.16)	80.38 (84.49)
Alexnet (Krizhevsky et al., 2012)			-	-	-	83.6
ImageNet 1000 classes with random labels						
no	yes	yes	91.18	97.95	0.09	0.49
no	no	yes	87.81	96.15	0.12	0.50
no	no	no	95.20	99.14	0.11	0.56

Uniform convergence may be
unable to explain generalization in deep learning [3]

Key claim

- Classic generalization error bounds are all “uniform convergence bounds”

Definition 3.2. The **uniform convergence bound** with respect to loss $\mathcal{L}$ is the smallest value $\epsilon_{\text{unif}}(m, \delta)$ such that: $\Pr_{S \sim \mathcal{D}^m}\left[\sup_{h \in \mathcal{H}} \left|\mathcal{L}_{\mathcal{D}}(h) - \hat{\mathcal{L}}_S(h)\right| \leq \epsilon_{\text{unif}}(m, \delta)\right] \geq 1 - \delta.$

- The first paper, “The entire $\mathscr{H}$ is too big. Think about an algorithm-dependent $\mathscr{H}_{\mathscr{A}}$, otherwise, you will only get vacuous bounds”
- This paper, “Even if it’s the tightest possible algorithm-dependent one, uniform convergence bound is vacuous”
  - There is a learning task where $\mathscr{A}$ can find generalizing h, but the uniform convergence bound is vacuous.

Tightest algorithm-dependent uniform convergence bound

Training dataset

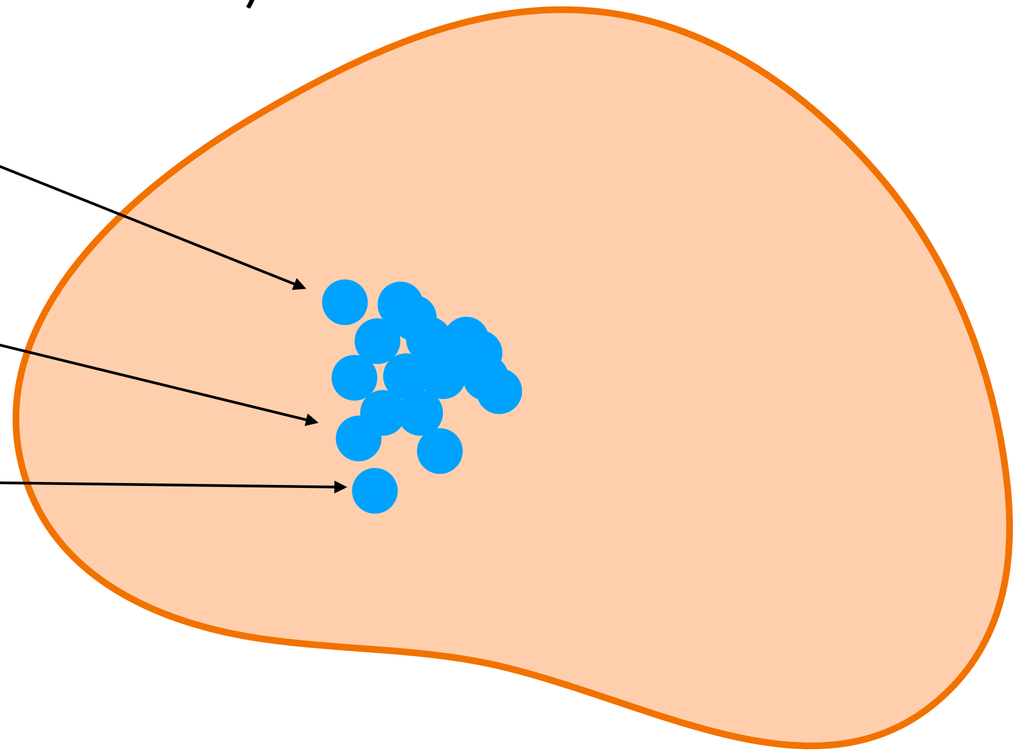
$S = \left((x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)\right)$

$S' = \left((x'_1, y'_1), (x'_2, y'_2), \ldots (x'_m, y'_m)\right)$

$S'' = \left((x''_1, y''_1), (x''_2, y''_2), \ldots (x''_m, y''_m)\right)$

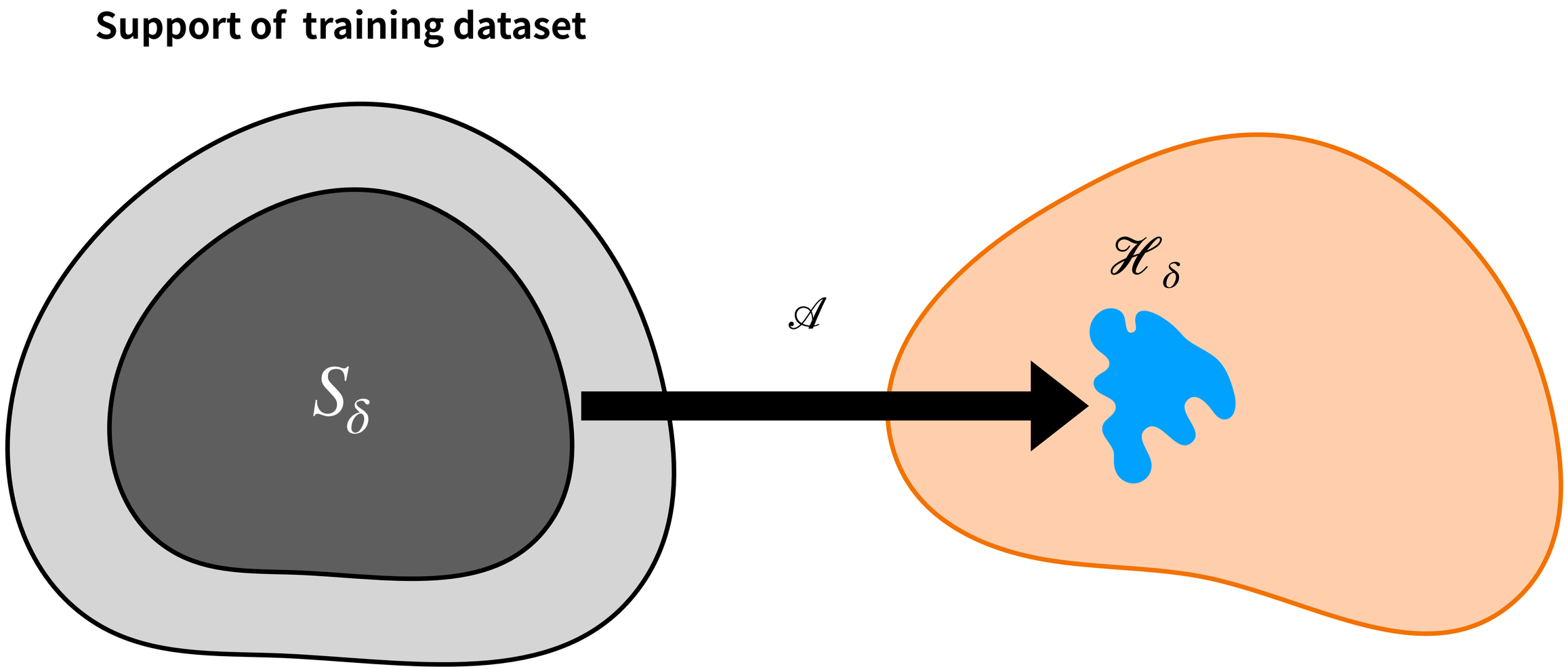
$\vdots$

$\mathscr{A}$ (deterministic)



$\mathscr{H}$: Entire hypothesis class

Tightest algorithm-dependent uniform convergence bound

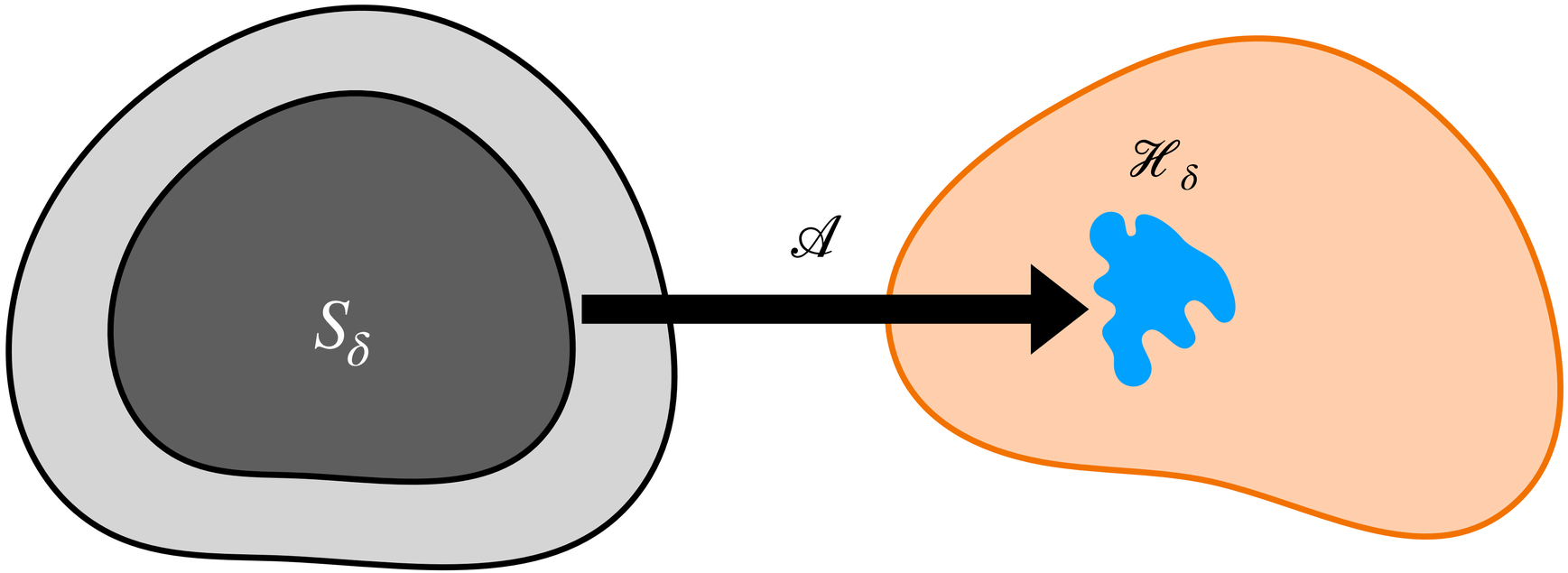


$\mathscr{H}$: Entire hypothesis class

Tightest algorithm-dependent uniform convergence bound

Definition 3.3. The **tightest algorithm-dependent uniform convergence bound** with respect to loss $\mathcal{L}$ is the smallest value $\epsilon_{\text{unif-alg}}(m, \delta)$ for which there exists a set of sample sets $\mathcal{S}_\delta$ such that $Pr_{S\sim\mathcal{D}^m}[S \in \mathcal{S}_\delta] \geq 1 - \delta$ and if we define the space of hypotheses explored by $\mathcal{A}$ on $\mathcal{S}_\delta$ as $\mathcal{H}_\delta := \bigcup_{S\in\mathcal{S}_\delta}\{h_S\} \subseteq \mathcal{H}$, the following holds: $\sup_{S\in\mathcal{S}_\delta} \sup_{h\in\mathcal{H}_\delta} \left|\mathcal{L}_\mathcal{D}(h) - \hat{\mathcal{L}}_S(h)\right| \leq \epsilon_{\text{unif-alg}}(m, \delta)$.

Largest possible generalization error in $\mathscr{H}_\delta$

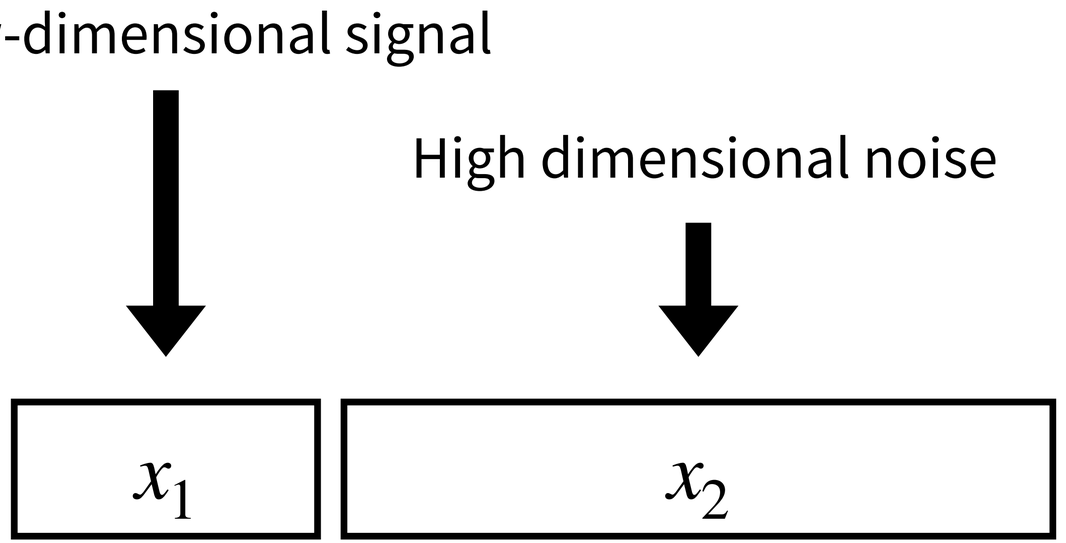


$\mathscr{H}$: Entire hypothesis class

A setup where UC bound provably fails

- $x \in \mathcal{X} : x = \left(x_1, x_2\right)\ (x_1 \in \mathbb{R}^K, x_2 \in \mathbb{R}^D)$
  - K is small constant, D is large
- $y \in \mathcal{Y} = \{-1, +1\}$
- Given a fixed vector u s.t. $\|u\|_2 = \frac{1}{\sqrt{m}}$

  $x_1 = 2 \cdot y \cdot u,\ x_2 \sim \mathcal{N}\left(0, \frac{32}{D} I\right),$
- $h \in \mathcal{H}$: $w = \left(w_1, w_2\right) \in \mathbb{R}^{K+D}$, $(h_w(x) = w_1 x_1 + w_2 x_2)$
- $\mathcal{A}$: Gradient descent



- $\mathcal{L}^{(\gamma)}\left(y', y\right) = \begin{cases} 1 & yy' \leq 0 \\ 1 - \frac{yy'}{\gamma} & yy' \in (0, \gamma) \\ 0 & yy' \geq \gamma \end{cases}$

Main theorem

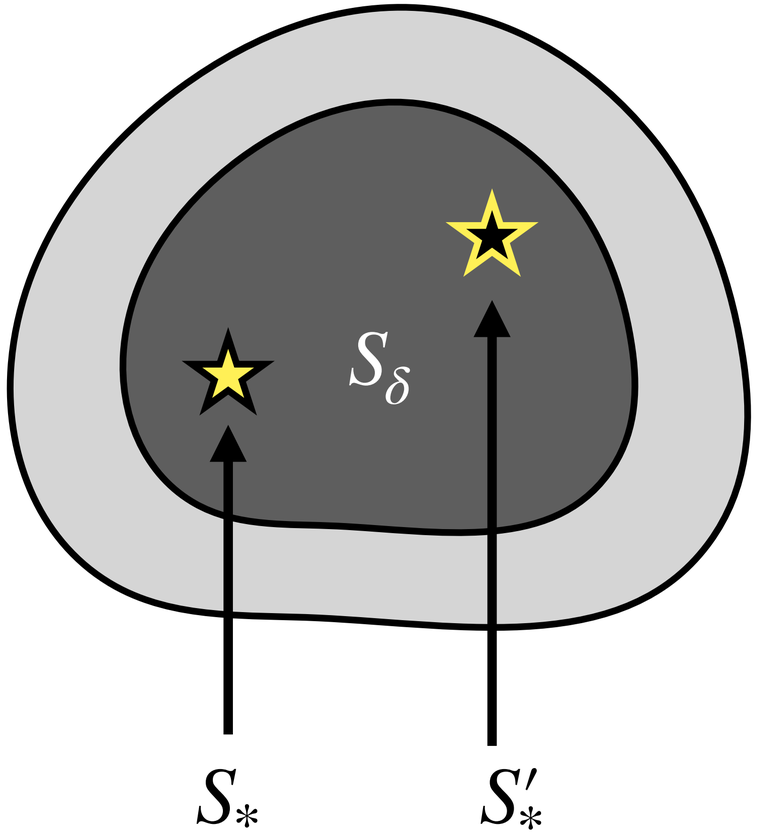
1. Gradient descent can find a generalizing h

2. But the “tightest algorithm-dependent uniform convergence bound” is vacuous

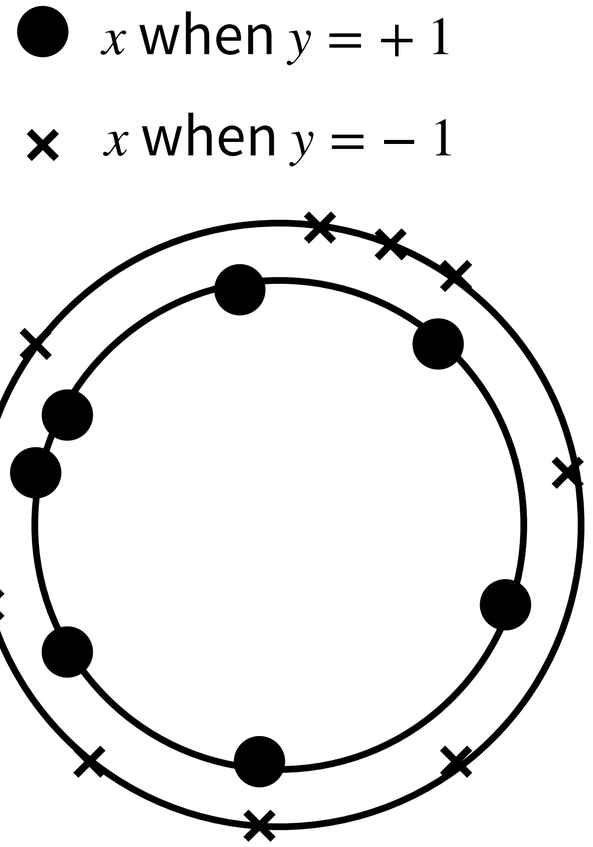
Theorem 3.1. *For any* $\epsilon, \delta > 0, \delta \leq 1/4$, *when* $D = \Omega\left(\max\left(m \ln \frac{m}{\delta}, m \ln \frac{1}{\epsilon}\right)\right)$, $\gamma \in [0,1]$, *the* $\mathcal{L}^{(\gamma)}$ *loss satisfies* $\epsilon_{gen}(m,\delta) \leq \epsilon$, *while* $\epsilon_{unif\text{-}alg}(m,\delta) \geq 1-\epsilon$. *Furthermore, for all* $\gamma \geq 0$, *for the* $\mathcal{L}^{(\gamma)}$ *loss,* $\epsilon_{unif\text{-}alg}(m,\delta) \geq 1 - \epsilon_{gen}(m,\delta)$.

Proof (Intuition)

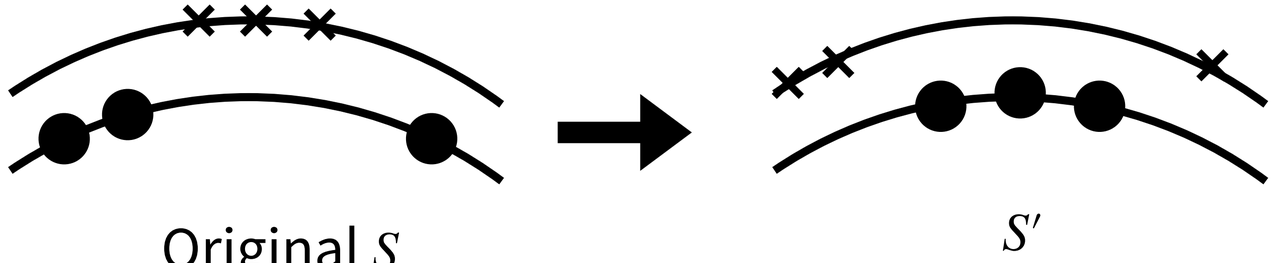
- $\epsilon_{\text{unif-alg}}$ always comes with some S_δ
- But for any S_δ, you can find the following S_*
  1. $S_* \in S_\delta$
  2. $S'_* \in S_\delta$, where $S'_* = \{((x_1, -x_2), y) \mid ((x_1, x_2), y) \in S_*\}$
  3. h_{S_*} has generalization error less than ϵ
  4. h_{S_*} completely misclassifies S'_*



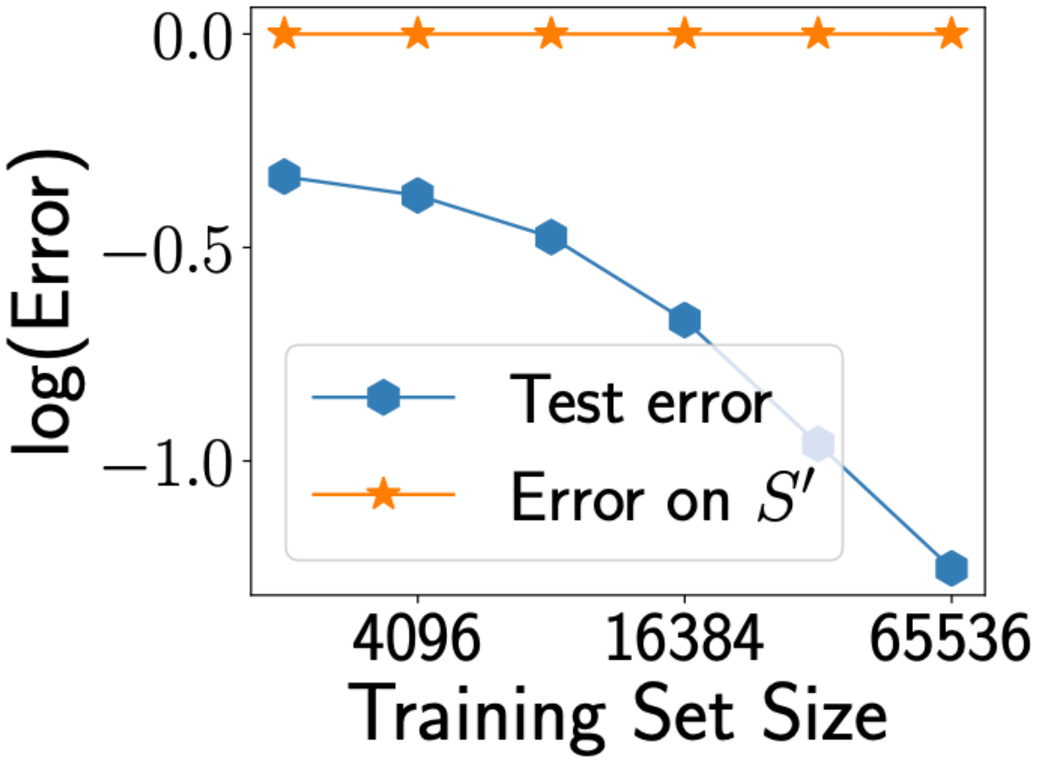
An experiment where UC bound fails

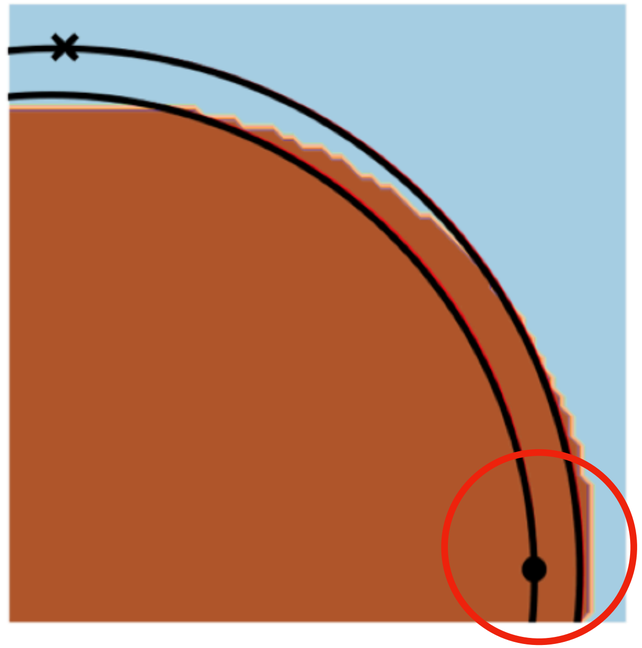


- $\mathscr{H}$: Two-layer ReLU networks (100k hidden units)
- $\mathscr{A}$: stochastic gradient descent
- $x \in \mathscr{X}$: hypersphere surface with radius 1 and 1.1
  (of 1000-dimensional)
- $y \in \mathscr{Y}$: $\{-1, +1\}$
- Generate S' by flipping the radius
  - Notice $S \sim \mathscr{D}$ and $S' \sim \mathscr{D}$



h_S generalizes well but performs poorly on S'





Because of this extra margin,
h_S misclassifies S'

My understanding: Should we focus on this?

Deep learning conjecture

- Over-parameterized deep networks mainly behave like a very simple model (such as a linear model) and roughly fit the training data
- Plenty of parameters are unused but some of them learn “unnecessary knowledge” from training data
- Such “unnecessary knowledge” does not affect generalization performance
  - Example: Even if I have knowledge that “the earth is flat,” I can have normal conversations in 99% of my daily life and few people think I’m strange.
- However, we can always find a dataset where such unnecessary knowledge seriously affect the performance, which establishes a loose uniform convergence bound.

References

1. Shalev-Shwartz, Shai, and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, (2014)

2. Zhang, Chiyuan, et al. "Understanding deep learning requires rethinking generalization." arXiv preprint arXiv:1611.03530 (2016).

3. Nagarajan, Vaishnavh, and J. Zico Kolter. "Uniform convergence may be unable to explain generalization in deep learning." Advances in Neural Information Processing Systems 32 (2019).