ECE826 Lecture 4:

Limitation of Rademacher Complexity; Moving forward with Stability

From This

To this

Contents

- Parameter count bounds for ERM
- VC dim and Rademacher Complexity generalization bounds
- Do these bounds explain generalization in modern ML?
- What are we missing?

Some Definitions

- Our goal is to find a hypothesis (classifier) h_S with small expected risk

$$R[h_S] = \mathbb{E}_{(x,y)\sim\mathscr{D}}\left[\ell(h_S(x); y)\right]$$

- The loss measures the disagreement between predictions and reality

- Since we can't directly measure $R[h_S]$ (our true cost function), we can consider optimizing its sample-average proxy, i.e., the empirical risk

$$\hat{R}[h_S] = \frac{1}{n}\sum_{i=1}^{n} \ell(h_S(x_i); y_i)$$

- Our hope is that $\hat{R}[h_S]$ is close to $R[h_S]$

The generalization gap

- The gap of the true cost function from the one we have access to

$$\epsilon_{gen} = |R[h_S] - \hat{R}[h_S]|$$

- <u>Question:</u> When is it possible to bound ϵ_{gen} by a small constant?

- The answer must depend on:

  1) n, the sample size
  2) $\mathcal{H}$, the hypothesis class (and its geometry)
  3) $\mathcal{D}$, the data distribution

  [4) the optimization algorithm that outputs our classifier]

Previously: parameter count bounds

- If Floats+parametric model => $n >>$ #params for good generalization (H.I.+Union bound over all classifiers)

- If Infinite class, then VC-dim can help in bounded the error, with not much better bound than $n >>$ #params for good generalization

- Compression arguments can lead to better results for nearly sparse/low-rank models

Back to complexity bounds: Rademacher Complexity

Rademacher Complexity

- Rademacher complexity: how much can your bag of classifiers fit random noise?

Definition:

The Rademacher Complexity of $\mathscr{H}$ with respect to a distribution D is equal to

$$\mathscr{R}_n(\mathscr{H}) = \frac{1}{n}\mathbb{E}_{S\sim D^n}\mathbb{E}_{\sigma}\left[\sup_{h\in\mathscr{H}}\sum_{i=1}^{n}\sigma_i h(x_i)\right]$$

where, σ_i are iid uniform ± 1 random variables.

Rademacher Complexity

- Rademacher complexity: how much can your bag of classifiers fit random noise?

Definition:

The Rademacher Complexity of $\mathscr{H}$ with respect to a distribution D is equal to

$$\mathscr{R}_n(\mathscr{H}) = \frac{1}{n}\mathbb{E}_{S\sim D^n}\mathbb{E}_\sigma \left[\sup_{h\in\mathscr{H}} \sum_{i=1}^{n} \sigma_i h(x_i) \right]$$

where, σ_i are iid uniform ± 1 random variables.

- Note that RC is between 0 and 1. 1 means my bag is expressive enough to fit random labels.
- What is RC used to bound the generalization gap?

Rademacher Complexity

- Rademacher complexity: how much can your bag of classifiers fit random noise?

Theorem:

The generalization error of $\mathscr{H}$ is bounded as

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq 2RC_m(\mathscr{H}) + \sqrt{\frac{\log(1/\delta)}{n}}$$

with probability $1 - \delta$

Rademacher Complexity

- Rademacher complexity: how much can your bag of classifiers fit random noise?

Theorem:

The generalization error of $\mathscr{H}$ is bounded as

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq 2RC_m(\mathscr{H}) + \sqrt{\frac{\log(1/\delta)}{n}}$$

with probability $1 - \delta$

- You'd like RC to scale like $\sim \dfrac{1}{\sqrt{n}}$

Generalization for smooth losses

- [Srebro, Sridharan, Tewari, 2012 https://arxiv.org/pdf/1009.3896.pdf]

Theorem:

Let L^* be the best risk achieved by $\mathscr{H}$ on an H-smooth, bounded loss function (i.e., Lipschitz derivative), then

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] = \tilde{O}\left(H \cdot RC_m^2(\mathscr{H}) + \sqrt{HL^*} RC_m(\mathscr{H}) \right)$$

- Meta-observation When the loss in the class is zero, you get a really fast rate of "learning"!

- Q: What is the RC of classifiers in practice??

Rademacher Complexity: Examples

- Linear classifiers

[source: https://arxiv.org/pdf/2007.11045.pdf]

Rademacher Complexity: Examples

- Linear classifiers

$$\|\mathbf{M}\|_{p,q} = \|(\|\mathbf{M}_1\|_1, \ldots, \|\mathbf{M}_d\|_p)\|_q, \text{ where } \mathbf{M}_i\text{s are the columns of } \mathbf{M}$$

Theorem 1 *Let $\mathcal{F}_p = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\|_p \le W\}$ be a family of linear functions defined over $\mathbb{R}^d$ with bounded weight in ℓ_p-norm. Then, the empirical Rademacher complexity of $\mathcal{F}_p$ for a sample $\mathcal{S} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ admits the following upper bounds:*

$$\widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_p) \le \begin{cases} \frac{W}{m}\sqrt{2\log(2d)}\,\|\mathbf{X}^\top\|_{2,p^*} & \text{if } p = 1 \\ \frac{\sqrt{2}W}{m}\left[\frac{\Gamma\left(\frac{p^*+1}{2}\right)}{\sqrt{\pi}}\right]^{\frac{1}{p^*}}\|\mathbf{X}^\top\|_{2,p^*} & \text{if } 1 < p \le 2 \\ \frac{W}{m}\|\mathbf{X}^\top\|_{2,p^*}, & \text{if } p \ge 2 \end{cases}$$

where $\mathbf{X}$ is the $d \times m$-matrix with $\mathbf{x}_i$s as columns: $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_m]$. Furthermore, the constant factor in the inequality for the case $1 < p \le 2$ can be bounded as follows:

$$e^{-\frac{1}{2}}\sqrt{p^*} \le \sqrt{2}\left[\frac{\Gamma(\frac{p^*+1}{2})}{\sqrt{\pi}}\right]^{\frac{1}{p^*}} \le e^{-\frac{1}{2}}\sqrt{p^* + 1}.$$

[source: https://arxiv.org/pdf/2007.11045.pdf]

Rademacher Complexity: Examples

- Linear classifiers

[redacted]

simpler bounds exist for linear classification, e.g.,

$$RC_m^2(\mathcal{H}) \leq \mathbb{E} \max_{i \in S} \|\mathbf{x}_i\|_2 / \sqrt{n}$$

$$\widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_p) \leq \begin{cases} \frac{\sqrt{2}W}{m}\left[\frac{\Gamma\left(\frac{p^*+1}{2}\right)}{\sqrt{\pi}}\right]^{\frac{1}{p^*}} \|\mathbf{X}^\top\|_{2,p^*} & \text{if } 1 < p \leq 2 \\ \frac{W}{m}\|\mathbf{X}^\top\|_{2,p^*}, & \text{if } p \geq 2 \end{cases}$$

where $\mathbf{X}$ *is the* $d \times m$*-matrix with* $\mathbf{x}_i$*s as columns:* $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_m]$*. Furthermore, the constant factor in the inequality for the case* $1 < p \leq 2$ *can be bounded as follows:*

$$e^{-\frac{1}{2}}\sqrt{p^*} \leq \sqrt{2}\left[\frac{\Gamma\left(\frac{p^*+1}{2}\right)}{\sqrt{\pi}}\right]^{\frac{1}{p^*}} \leq e^{-\frac{1}{2}}\sqrt{p^* + 1}.$$

[source: https://arxiv.org/pdf/2007.11045.pdf]

Rademacher Complexity: Examples

- Neural Networks

Theorem 18 *Suppose that* $\sigma : \mathbb{R} \to [-1, 1]$ *has Lipschitz constant* L *and satisfies* $\sigma(0) = 0$. *Define the class computed by a two-layer neural network with 1-norm weight constraints as*

$$F = \left\{ x \mapsto \sum_i w_i \sigma\left(v_i \cdot x\right) : \|w\|_1 \leq 1,\ \|v_i\|_1 \leq B \right\}.$$

Then for $x_1, \ldots, x_n$ *in* $\mathbb{R}^k$,

$$\hat{G}_n(F) \leq \frac{cLB(\ln k)^{1/2}}{n} \max_{j,j'} \sqrt{\sum_{i=1}^{n} \left(x_{ij} - x_{ij'}\right)^2},$$

where $x_i = (x_{i1}, \ldots, x_{ik})$.

[source: https://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf]

Rademacher Complexity: Examples

- Neural Networks

Theorem 18 *Suppose that* $\sigma : \mathbb{R} \to [-1, 1]$ *has Lipschitz constant* L *and satisfies* $\sigma(0) = 0$.

[redacted]

similar to fancy parameter bounds above

[redacted]

where $x_i = (x_{i1}, \ldots, x_{ik})$.

[source: https://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf]

Do the above explain generalization?

UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION

Chiyuan Zhang[*]
Massachusetts Institute of Technology
chiyuan@mit.edu

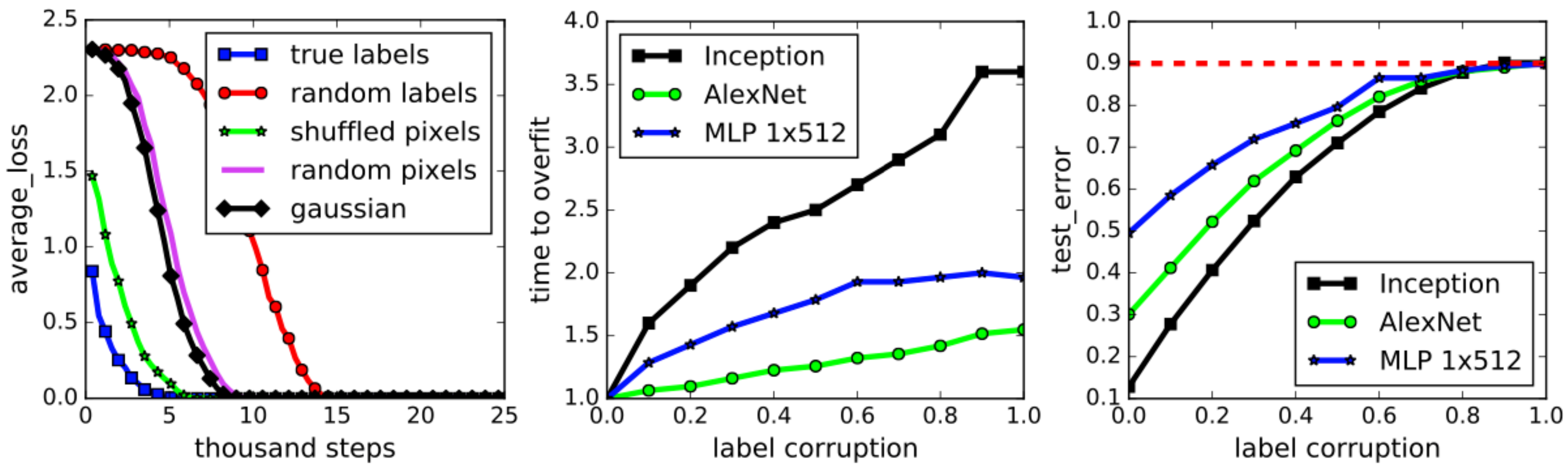
Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht[†]
University of California, Berkeley
brecht@berkeley.edu

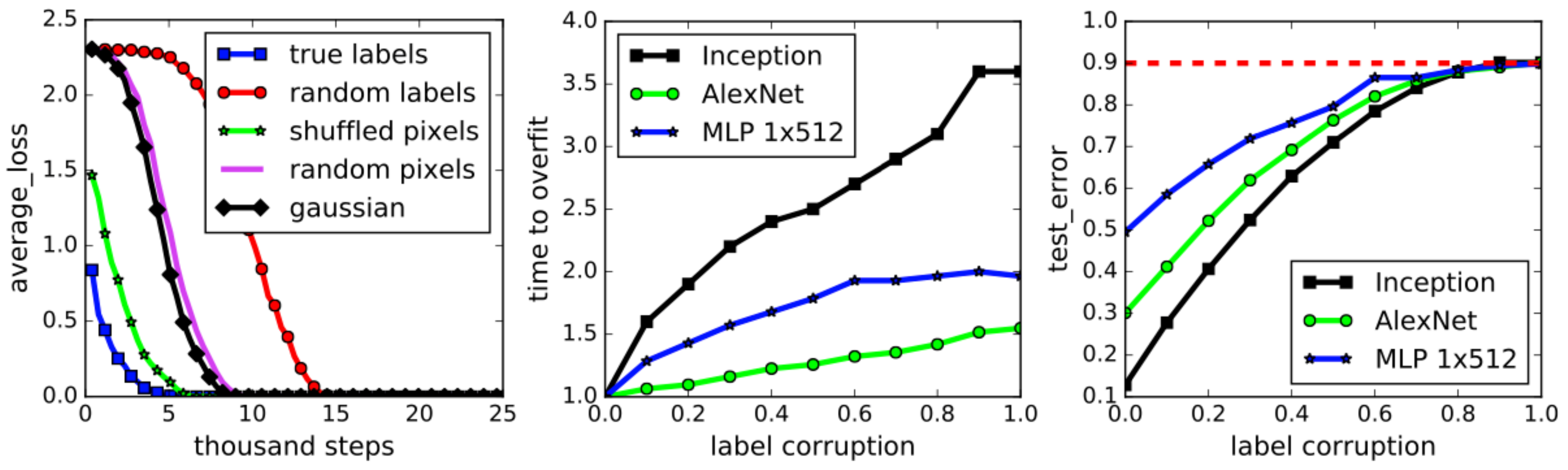
Oriol Vinyals
Google DeepMind
vinyals@google.com

ABSTRACT



(a) learning curves (b) convergence slowdown (c) generalization error growth

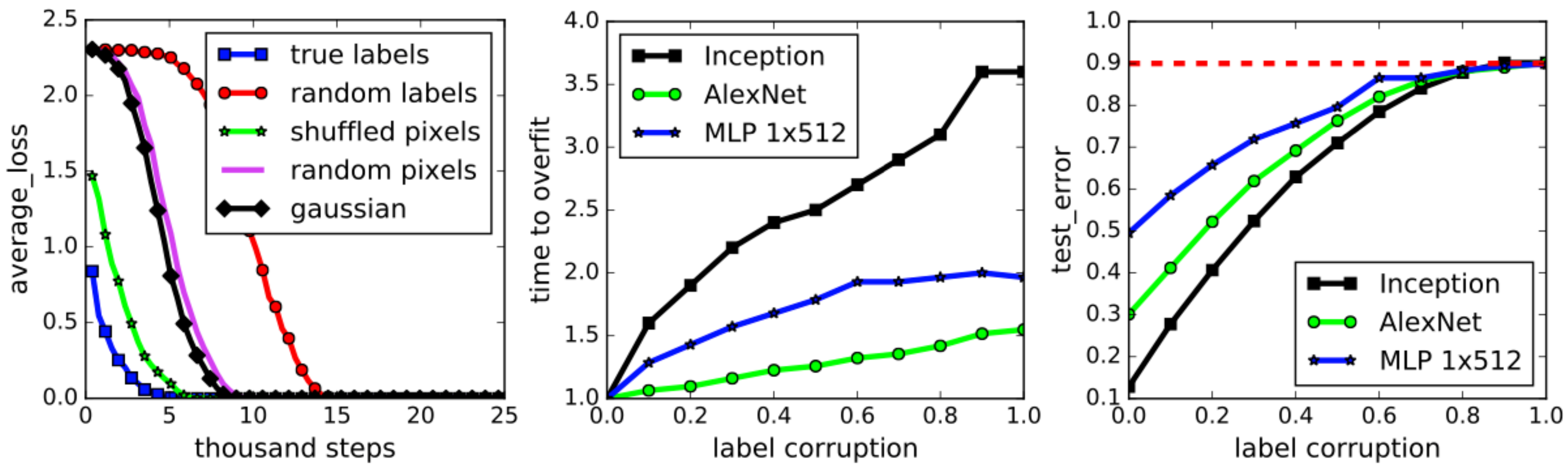
Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Modern models can fit random labels!

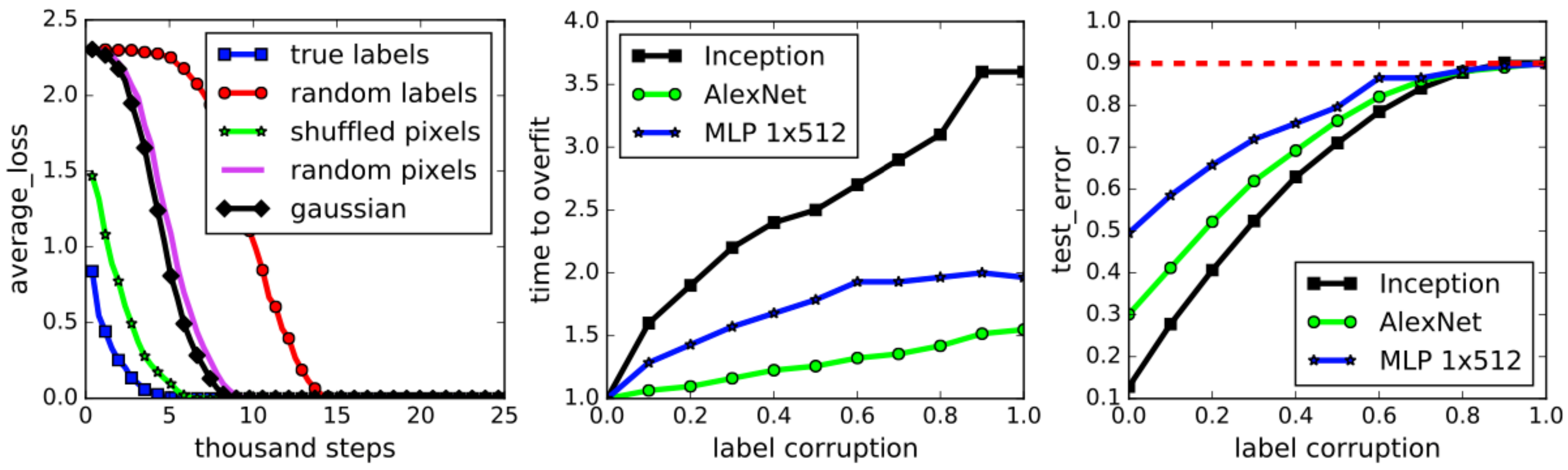


(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Rademacher complexity = 1

Ugh



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Modern models can fit random labels!

Ok so if uniform gen doesn't explain this, maybe regularization does

Maybe regularization helps?

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

Maybe regularization helps?

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	[illegible]	no	yes	[illegible]	[illegible]
		no	no	100.0	[illegible]
(fitting random labels)		no	no	99.34	10.61

Regularization is not the only key

How to make the algorithm part of the equation?

What algorithmic property begets generalization?

before implicit bias was cool

Stability of Learning Algorithms

Algorithmic Stability

- Learning algorithm $A(S)$ is stable if:
  "the trained classifier does not depend too much on one data point"

- Let S^i = original data set, but with z_i data point replaced by z_i'

- <u>*Def:*</u> Stability*

$$\mathbb{E}_{S,z_i} \left| \text{loss}(A(S); z_i) - \text{loss}(A(S^i); z_i) \right| \leq \delta$$

- Thm: (Bousquet and Elisseeff 2002) [amazing paper, please read]

δ-stable algorithms achieve δ generalization gap

Many stability notions

- Replace-one stability:

$$\mathbb{E}_{S,z_i}\left|\mathrm{loss}(A(S);z_i)-\mathrm{loss}(A(S^i);z_i)\right|\leq\delta$$

Many stability notions

- Replace-one stability:

$$\mathbb{E}_{S,z_i} \left| \text{loss}(A(S); z_i) - \text{loss}(A(S^i); z_i) \right| \leq \delta$$

- Hypothesis stability:

$$\mathbb{E}_{S,z} \left| \text{loss}(A(S); z) - \text{loss}(A(S^i); z) \right| \leq \delta$$

Many stability notions

- Replace-one stability:

$$\mathbb{E}_{S,z_i}\left|\text{loss}(A(S); z_i) - \text{loss}(A(S^i); z_i)\right| \leq \delta$$

- Hypothesis stability:

$$\mathbb{E}_{S,z}\left|\text{loss}(A(S); z) - \text{loss}(A(S^i); z)\right| \leq \delta$$

- Error stability:

$$\forall S, i \;\; \mathbb{E}_{z}\left|\text{loss}(A(S); z) - \text{loss}(A(S^i); z)\right| \leq \delta$$

Many stability notions

- Replace-one stability:

$$\mathbb{E}_{S,z_i}\left|\text{loss}(A(S);z_i) - \text{loss}(A(S^i);z_i)\right| \leq \delta$$

- Hypothesis stability:

$$\mathbb{E}_{S,z}\left|\text{loss}(A(S);z) - \text{loss}(A(S^i);z)\right| \leq \delta$$

- Error stability:

$$\forall S, i \ \ \mathbb{E}_{z}\left|\text{loss}(A(S);z) - \text{loss}(A(S^i);z)\right| \leq \delta$$

- Uniform stability:

$$\forall S, i, z, \quad \left|\text{loss}(A(S);z) - \text{loss}(A(S^i);z)\right| \leq \delta$$

Many stability notions

[redacted]

Stability depends on: Algorithm, Data, Loss function!

[redacted]

- Error stability:

$$\forall S, i \;\; \mathbb{E}_z \left| \text{loss}(A(S); z) - \text{loss}(A(S^i); z) \right| \leq \delta$$

- Uniform stability:

$$\forall S, i, z, \quad \left| \text{loss}(A(S); z) - \text{loss}(A(S^i); z) \right| \leq \delta$$

Many stability notions

[redacted]

Stability depends on: Algorithm, Data, Loss function!

[redacted]

[redacted]

Downside: it's tricky to establish

[redacted]

Stability <=> Generalization

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

gen gap = (empirical risk) – (true risk)

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

gen gap = (empirical risk) – (true risk)

$$= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z}\text{loss}(A(S); z)$$

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

$$\begin{aligned}
\text{gen gap} &= \text{(empirical risk)} - \text{(true risk)} \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z}\text{loss}(A(S); z) \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j')
\end{aligned}$$

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

$$
\begin{aligned}
\text{gen gap} &= \text{(empirical risk)} - \text{(true risk)} \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z}\text{loss}(A(S); z) \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j') \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{S,A,z_j'}\text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j')
\end{aligned}
$$

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

$$\begin{aligned}
\text{gen gap} &= (\text{empirical risk}) - (\text{true risk}) \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z}\text{loss}(A(S); z) \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j') \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{S,A,z_j'}\text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j') \\
&= \mathbb{E}_{S,A,z_j'}\text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j')
\end{aligned}$$

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

$$\text{gen gap} = (\text{empirical risk}) - (\text{true risk})$$

$$= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z}\text{loss}(A(S); z)$$

$$= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \text{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j')$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{S,A,z_j'}\text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j')$$

$$= \mathbb{E}_{S,A,z_j'}\text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\text{loss}(A(S); z_j')$$

$$= \mathbb{E}_{S,A,z_j'}\left[\text{loss}(A(S^j); z_j') - \text{loss}(A(S); z_j')\right]$$

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \dots, z_n\}, \quad S^j = \{z_1, \dots, z_j', \dots z_n\}$

$$
\begin{aligned}
\text{gen gap} &= \text{(empirical risk)} - \text{(true risk)} \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \mathrm{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z}\mathrm{loss}(A(S); z) \\
&= \mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{j=1}^{n} \mathrm{loss}(A(S); z_i)\right] - \mathbb{E}_{S,A,z_j'}\mathrm{loss}(A(S); z_j') \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{S,A,z_j'}\mathrm{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\mathrm{loss}(A(S); z_j') \\
&= \mathbb{E}_{S,A,z_j'}\mathrm{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'}\mathrm{loss}(A(S); z_j') \\
&= \mathbb{E}_{S,A,z_j'}\left[\mathrm{loss}(A(S^j); z_j') - \mathrm{loss}(A(S); z_j')\right]
\end{aligned}
$$

Boom, Stability

Stability = Generalization

- Proof by renaming
- Let $S = \{z_1, \ldots, z_n\}, \quad S^j = \{z_1, \ldots, z_j', \ldots z_n\}$

gen gap = (empirical risk) − (true risk)

[redacted]

Caveat: not a high probability result,
but possible to prove them with a bit more work

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{S,A,z_j'} \text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'} \text{loss}(A(S); z_j')$$

$$= \mathbb{E}_{S,A,z_j'} \text{loss}(A(S^j); z_j') - \mathbb{E}_{S,A,z_j'} \text{loss}(A(S); z_j')$$

$$= \mathbb{E}_{S,A,z_j'} \left[\text{loss}(A(S^j); z_j') - \text{loss}(A(S); z_j')\right]$$

Boom, Stability

Stable Algorithms generalize well

Q: Which algorithms are stable?

Example 0

- Trivial example of stable algorithm:

$$h(W; x) =$$

Example 1: k-NN

- Example training set:

- Resampled training set:

- Probability of difference in predictions:

$$P\left(h_S(x) \neq h_{S^i}(x)\right) =$$

- Stability:

$$\text{loss}(h_S(x); y) - \text{loss}(h_{S^i}(x); y) = \Pr\left(h_S(x) \neq y\right) - \Pr\left(h_{S^i}(x) \neq y\right)$$

Example 1: k-NN

- Example training set:

[redacted]

VC-dimension of kNN is infinite!

[redacted]

- Stability:

$$\text{loss}(h_S(x); y) - \text{loss}(h_{S^i}(x); y) = \Pr\left(h_S(x) \neq y\right) - \Pr\left(h_{S^i}(x) \neq y\right)$$

Example 2: Minimizers of PL Functions

- An empirical loss function is μ-PL if

$$\left\| \nabla \frac{1}{n} \sum_i \ell(w; z_i) \right\|^2 \geq \mu \|w - w^*\|$$

- Proof of stability:
  - $\ell(w^*; z) - \ell(w_i^*; z) =$

Example 2: Minimizers of PL Functions

- An empirical loss function is μ-PL if

[redacted]

Why is PL Interesting

[redacted]

Example 2: Minimizers of PL Functions

- An empirical loss function is μ-PL if

[redacted]

Why is PL Interesting

Strongly convex functions are PL!

Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path?

Samet Oymak* and Mahdi Soltanolkotabi†

Loss landscapes and optimization in over-parameterized non-linear systems and neural networks

Chaoyue Liu[a], Libin Zhu[b,c], and Mikhail Belkin[c]

[a]*Department of Computer Science and Engineering, The Ohio State University*
[b]*Department of Computer Science and Engineering, University of California, San Diego*
[c]*Halicioğlu Data Science Institute, University of California, San Diego*

May 28, 2021

On the Convergence Rate of Training Recurrent Neural Networks

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

October 28, 2018

A Convergence Theory for Deep Learning via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path?

Samet Oymak* and Mahdi Soltanolkotabi†

Loss landscapes and optimization in over-parameterized non-linear systems and neural networks

Chaoyue Liu[a], Libin Zhu[b,c], and Mikhail Belkin[c]

[a]*Department of Computer Science and Engineering, The Ohio State University*
[b]*Department of Computer Science and Engineering, University of California, San Diego*
[c]*Halicioğlu Data Science Institute, University of California, San Diego*

May 28, 2021

On the Convergence Rate of Training Recurrent Neural Networks

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

October 28, 2018

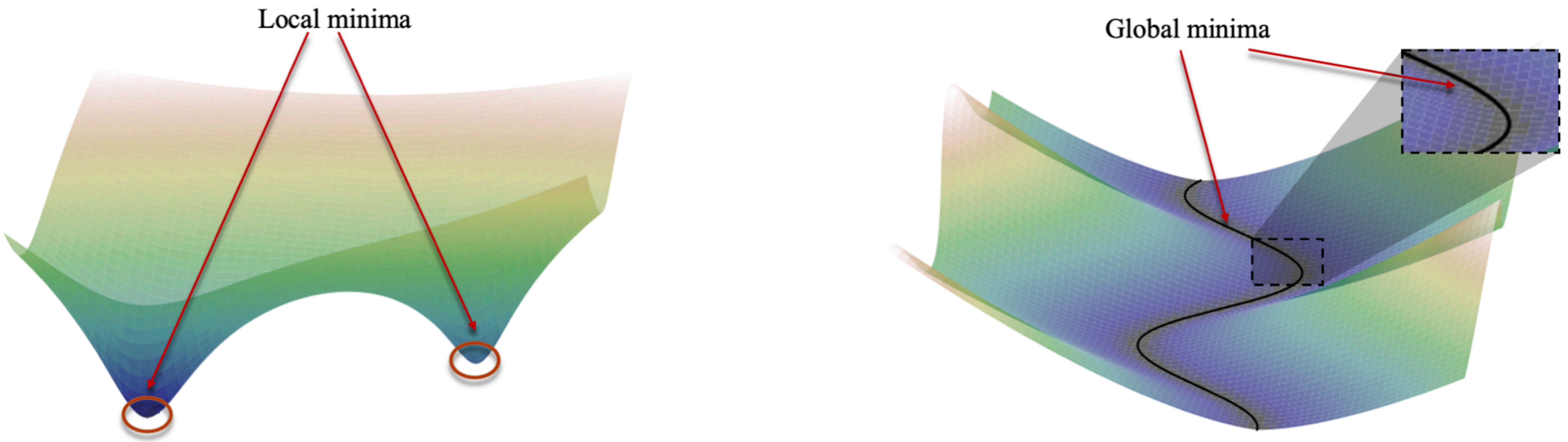
A Convergence Theory for Deep Learning via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

PL-like conditions hold in neighborhoods around initialization/optima.



(a) Loss landscape of under-parameterized models

(b) Loss landscape of over-parameterized models

Figure 1: Panel (a): Loss landscape is locally convex at local minima. Panel (b): Loss landscape incompatible with local convexity as the set of global minima is not locally linear.

PL-like conditions hold in neighborhoods around initialization/optima.

A Convergence Theory for Deep Learning via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

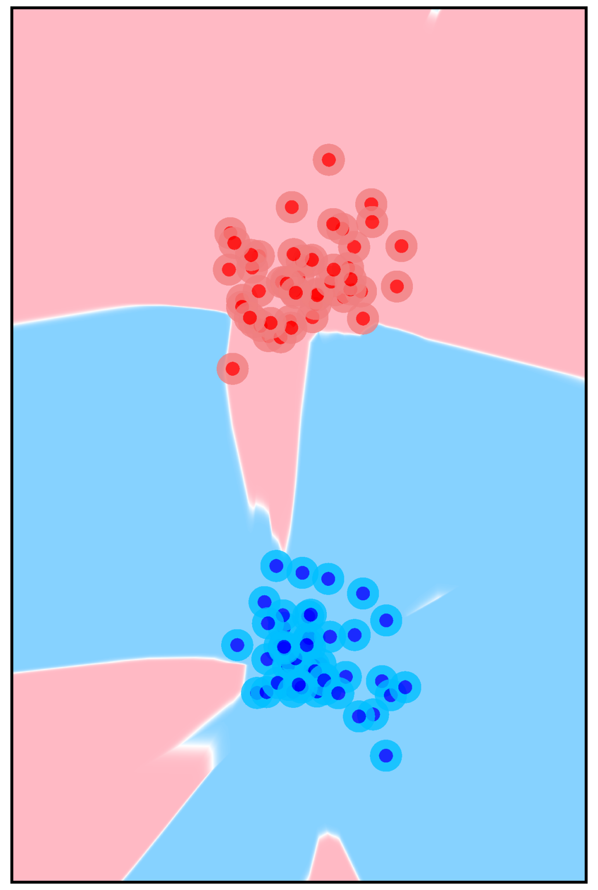
Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

October 28, 2018

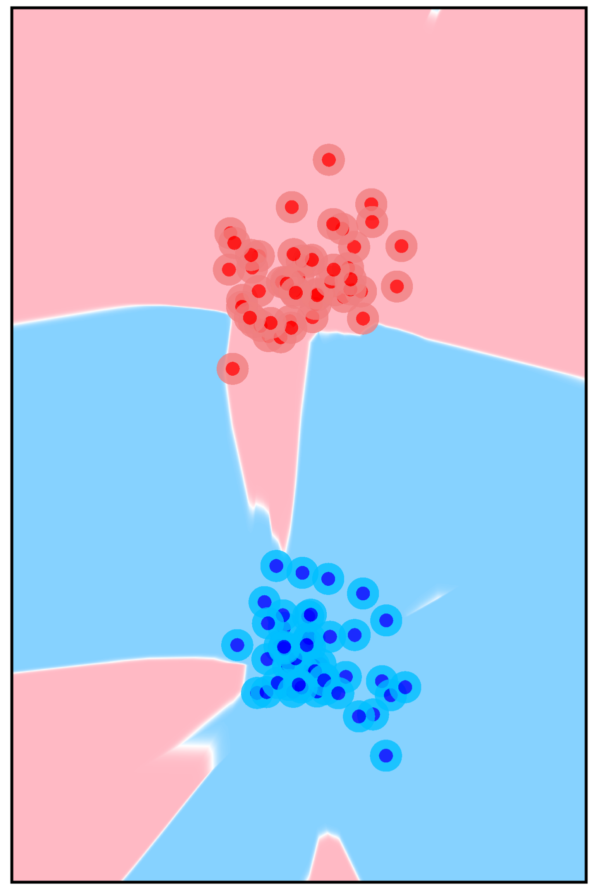
Final Remarks

- Rademacher complexity doesn't always give interesting bounds in practice
- Stability begets generalization!
- Many interesting minimizers are stable
- Open Qs:
  - Are the optimization algorithms stable?
  - Stability and loss geometry not well understood
  - Connections to implicit regularization?
  - How fast can we certify stability?
  - Combine with compression arguments from last lecture?



Final Remarks

- Rademacher complexity doesn't always give interesting bounds in practice
- Stability begets generalization!
- Many interesting minimizers are stable
- Open Qs:
  - Are the optimization algorithms stable?
  - Stability and loss geometry not well understood
  - Connections to implicit regularization?
  - How fast can we certify stability?
  - Combine with compression arguments from last lecture?



Why do SGD trained neural nets generalize so well?

reading list

Bousquet, Olivier, and André Elisseeff. "Stability and generalization." The Journal of Machine Learning Research 2 (2002): 499-526. https://www.jmlr.org/papers/volume2/bousquet02a/bousquet02a.pdf

(Stability and RC Chapters) Understanding Machine Learning: From Theory to Algorithms, https://www.cs.huji.ac.il/w~shais/UnderstandingMachineLearning/copy.html

Hardt, M., Recht, B. and Singer, Y., 2016, June. Train faster, generalize better: Stability of stochastic gradient descent. In International conference on machine learning (pp. 1225-1234). PMLR. Vancouver, http://proceedings.mlr.press/v48/hardt16.pdf

Srebro, N., Sridharan, K. and Tewari, A., 2010. Smoothness, low noise and fast rates. Advances in neural information processing systems, 23. https://proceedings.neurips.cc/paper/2010/file/76cf99d3614e23eabab16fb27e944bf9-Paper.pdf

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O., 2021. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3), pp.107-115. https://dl.acm.org/doi/pdf/10.1145/3446776

Bartlett, P.L. and Mendelson, S., 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3(Nov), pp.463-482. Vancouver , https://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf

Charles, Z. and Papailiopoulos, D., 2018, July. Stability and generalization of learning algorithms that converge to global optima. In International conference on machine learning (pp. 745-754). PMLR. http://proceedings.mlr.press/v80/charles18a/charles18a.pdf