Understanding Generalization in Deep Learning

Gintarė Karolina Džiugaitė
Element AI, ServiceNow; Mila

Thanks to with Michael Carbin, Surya Ganguli, Ioannis Mitliagkas, Jonathan Frankle, Daniel M. Roy, Alexandre Drouin, Stanislav Fort, Waseem Gharbieh, Kyle Hsu, Mahdi Haghifam, Sepideh Kharaghani, Gabriel Arpino, Ethan Caballero, Ashish Khisti, Brady Neal, Jeffrey Negrea, Mansheej Paul, Nitarshan Rajkumar, and Linbo Wang.

Why does deep learning work?

Why does deep learning work?
Why does deep learning generalize?

Why does deep learning work?
Why does deep learning generalize?
Why does deep learning generalize in-distribution?

Why does deep learning work?
Why does deep learning generalize?
Why does deep learning generalize in-distribution?

There is, as yet, no satisfying theory explaining why overparameterized neural networks, trained by variants of stochastic gradient descent, generalize from training data to test data on standard benchmarks.

Why does deep learning work?
Why does deep learning generalize?
Why does deep learning generalize in-distribution?

There is, as yet, no satisfying theory explaining why overparameterized neural networks, trained by variants of stochastic gradient descent, generalize from training data to test data on standard benchmarks.

Without a solid foundation, we have little chance of understanding and taming more complex phenomena, such as out-of-distribution generalization.

Why does deep learning work?
Why does deep learning generalize?
Why does deep learning generalize in-distribution?

There is, as yet, no satisfying theory explaining why overparameterized neural networks, trained by variants of stochastic gradient descent, generalize from training data to test data on standard benchmarks.

Without a solid foundation, we have little chance of understanding and taming more complex phenomena, such as out-of-distribution generalization.

This gap between *practice* and *understanding* is a growing liability.

Why does deep learning work?
Why does deep learning generalize?
Why does deep learning generalize in-distribution?

There is, as yet, no satisfying theory explaining why overparameterized neural networks, trained by variants of stochastic gradient descent, generalize from training data to test data on standard benchmarks.

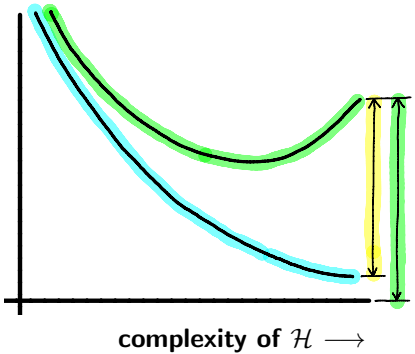
Without a solid foundation, we have little chance of understanding and taming more complex phenomena, such as out-of-distribution generalization.

This gap between *practice* and *understanding* is a growing liability.

I'll describe recent progress past barriers.

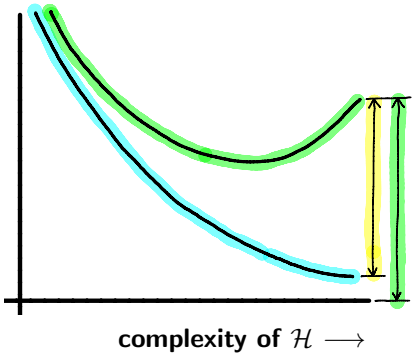
Generalization: textbook edition

Training data S drawn i.i.d. from data distribution $\mathcal{D}$.
Learning algorithm $\mathscr{A}(S)$ chooses a predictor from a space $\mathcal{H}$ of predictors.



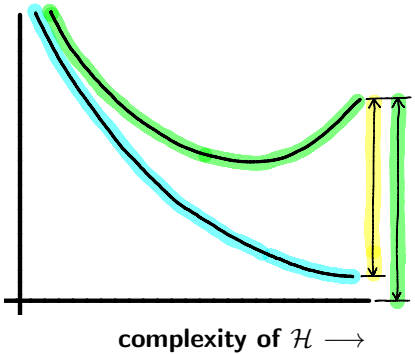
Generalization: textbook edition

Training data S drawn i.i.d. from data distribution $\mathcal{D}$.
Learning algorithm $\mathscr{A}(S)$ chooses a predictor from a space $\mathcal{H}$ of predictors.



Generalization: textbook edition

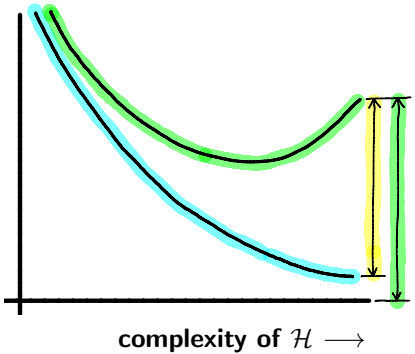
Training data S drawn i.i.d. from data distribution $\mathcal{D}$.
Learning algorithm $\mathscr{A}(S)$ chooses a predictor from a space $\mathcal{H}$ of predictors.



training error a.k.a. *empirical risk*
error a.k.a. *risk*

Generalization: textbook edition

Training data S drawn i.i.d. from data distribution $\mathcal{D}$.
Learning algorithm $\mathscr{A}(S)$ chooses a predictor from a space $\mathcal{H}$ of predictors.



training error a.k.a. *empirical risk*
error a.k.a. *risk*
generalization error

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Goal of classical machine learning: Choose a predictor $\hat{h}$ to minimize risk $L_{\mathcal{D}}(\hat{h})$.

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Goal of classical machine learning: Choose a predictor $\hat{h}$ to minimize risk $L_{\mathcal{D}}(\hat{h})$.

But we don't know $\mathcal{D}$ or $L_{\mathcal{D}}(\cdot)$! Use the training sample S and $L_S(\cdot)$.

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$
Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Goal of classical machine learning: Choose a predictor $\hat{h}$ to minimize risk $L_{\mathcal{D}}(\hat{h})$.

But we don't know $\mathcal{D}$ or $L_{\mathcal{D}}(\cdot)$! Use the training sample S and $L_S(\cdot)$.

$L_{\mathcal{D}}(\hat{h})$

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$
Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Goal of classical machine learning: Choose a predictor $\hat{h}$ to minimize risk $L_{\mathcal{D}}(\hat{h})$.

But we don't know $\mathcal{D}$ or $L_{\mathcal{D}}(\cdot)$! Use the training sample S and $L_S(\cdot)$.

$$L_{\mathcal{D}}(\hat{h}) = L_S(\hat{h}) + \underbrace{L_{\mathcal{D}}(\hat{h}) - L_S(\hat{h})}_{\text{Generalization gap/error}}$$

Basic notions from Statistical Learning Theory

Let $\mathcal{D}$ be a distribution on labeled examples.
Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}^m$ be m i.i.d. samples.

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Goal of classical machine learning: Choose a predictor $\hat{h}$ to minimize risk $L_{\mathcal{D}}(\hat{h})$.

But we don't know $\mathcal{D}$ or $L_{\mathcal{D}}(\cdot)$! Use the training sample S and $L_S(\cdot)$.

$$L_{\mathcal{D}}(\hat{h}) = L_S(\hat{h}) + \underbrace{L_{\mathcal{D}}(\hat{h}) - L_S(\hat{h})}_{\text{Generalization gap/error}}$$

$$L_{\mathcal{D}}(\hat{h}) = \underbrace{\inf_{h\in\mathcal{H}} L_D(h)}_{\text{Approximation error}} + \underbrace{L_{\mathcal{D}}(\hat{h}) - \inf_{h\in\mathcal{H}} L_D(h)}_{\text{Estimation error / excess risk}}$$

Two key ideas for understanding generalization

Two key ideas for understanding generalization

1. **Concentration of measure:**

Two key ideas for understanding generalization

1. **Concentration of measure:**

2. **Uniform convergence:**

Two key ideas for understanding generalization

1. **Concentration of measure:**
   Why heldout data allows us to accurately estimate risk for one predictor.
2. **Uniform convergence:**

Two key ideas for understanding generalization

1. **Concentration of measure:**
   Why heldout data allows us to accurately estimate risk for one predictor.
2. **Uniform convergence:**
   Why training data allows us to accurately estimate risk for every predictor.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

But that's not enough. Because S is i.i.d.,
$L_S(h)$ is guaranteed to be close to $L_{\mathcal{D}}(h)$.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

But that's not enough. Because S is i.i.d.,
$L_S(h)$ is guaranteed to be close to $L_{\mathcal{D}}(h)$.

Variance of $L_S(h)$ is $O(1/m)$.
But can say much more.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

But that's not enough. Because S is i.i.d.,
$L_S(h)$ is guaranteed to be close to $L_{\mathcal{D}}(h)$.

Variance of $L_S(h)$ is $O(1/m)$.
But can say much more.

Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon$ with probability no more than $2\exp(-2\epsilon^2 m)$.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

But that's not enough. Because S is i.i.d.,
$L_S(h)$ is guaranteed to be close to $L_{\mathcal{D}}(h)$.

Variance of $L_S(h)$ is $O(1/m)$.
But can say much more.

Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon$ with probability no more than $2\exp(-2\epsilon^2 m)$.

Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| \leq \sqrt{\frac{\ln 2/\delta}{2m}}$ with probability at least $1 - \delta$.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

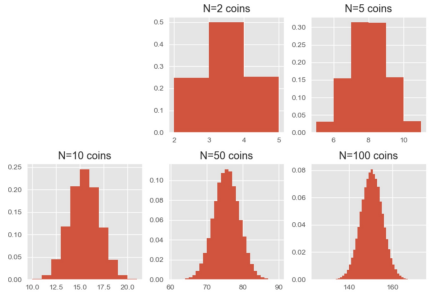
Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

But that's not enough. Because S is i.i.d.,
$L_S(h)$ is guaranteed to be close to $L_{\mathcal{D}}(h)$.

Variance of $L_S(h)$ is $O(1/m)$.
But can say much more.



Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon$ with probability no more than $2\exp(-2\epsilon^2 m)$.

Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| \leq \sqrt{\frac{\ln 2/\delta}{2m}}$ with probability at least $1-\delta$.

Concentration of measure

Risk/Error: $L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(x) \neq y]$

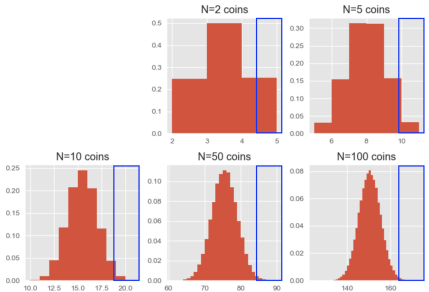
Empirical Risk/Error: $L_S(h) = \mathbb{P}_{(X,Y)\sim S}[h(x) \neq y] = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$

Why is the empirical risk a useful surrogate for the risk?

First, empirical risk is **unbiased**:
for all h, $\mathbb{E}_{S\sim\mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$.

But that's not enough. Because S is i.i.d.,
$L_S(h)$ is guaranteed to be close to $L_{\mathcal{D}}(h)$.

Variance of $L_S(h)$ is $O(1/m)$.
But can say much more.



Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon$ with probability no more than $2\exp(-2\epsilon^2 m)$.

Theorem (Hoeffding). $|L_S(h) - L_{\mathcal{D}}(h)| \leq \sqrt{\frac{\ln 2/\delta}{2m}}$ with probability at least $1-\delta$.

Uniform Convergence

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**
A: $\mathbb{E}[L_\eta] < 1/2$. Indeed, as K gets big, $\mathbb{E}[L_\eta]$ shrinks to 0! **No longer "unbiased"!**

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**
A: $\mathbb{E}[L_\eta] < 1/2$. Indeed, as K gets big, $\mathbb{E}[L_\eta]$ shrinks to 0! **No longer "unbiased"!**

Q: **What does this have to do with machine learning?**
A: This is the mechanism that leads to generalization error!

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**
A: $\mathbb{E}[L_\eta] < 1/2$. Indeed, as K gets big, $\mathbb{E}[L_\eta]$ shrinks to 0! **No longer "unbiased"!**

Q: **What does this have to do with machine learning?**
A: This is the mechanism that leads to generalization error!

Each *person* represents a *predictor*.

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**
A: $\mathbb{E}[L_\eta] < 1/2$. Indeed, as K gets big, $\mathbb{E}[L_\eta]$ shrinks to 0! **No longer "unbiased"!**

Q: **What does this have to do with machine learning?**
A: This is the mechanism that leads to generalization error!

Each *person* represents a *predictor*.
The *fraction of heads* is the *training error / empirical risk*.

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**
A: $\mathbb{E}[L_\eta] < 1/2$. Indeed, as K gets big, $\mathbb{E}[L_\eta]$ shrinks to 0! **No longer "unbiased"!**

Q: **What does this have to do with machine learning?**
A: This is the mechanism that leads to generalization error!

Each *person* represents a *predictor*.
The *fraction of heads* is the *training error / empirical risk*.
Then η is the predictor that does best on the training data.

Uniform Convergence

Thought experiment: Imagine everyone at this summer school flips a coin m times.
Let L_i be fraction of heads that person i got. (So $L_i \in \{0, \ldots, m\}/m$.)

Q: **How many heads do we expect each person to get?** (Assume $\mathbb{P}\{\text{heads}\} = \mathbb{P}\{\text{tails}\} = 1/2$.)
A: $\mathbb{E}[L_i] = 1/2$.

Q: **There are K of you. Let $\eta \in \{1, \ldots, K\}$ be the person with the least # of heads. (Note η is a random variable.) How does $\mathbb{E}[L_\eta]$ compare to $1/2$?**
A: $\mathbb{E}[L_\eta] < 1/2$. Indeed, as K gets big, $\mathbb{E}[L_\eta]$ shrinks to 0! **No longer "unbiased"!**

Q: **What does this have to do with machine learning?**
A: This is the mechanism that leads to generalization error!

Each *person* represents a *predictor*.
The *fraction of heads* is the *training error / empirical risk*.
Then η is the predictor that does best on the training data.
The *generalization error* of η is $|L_\eta - 1/2|$

Uniform Convergence, Part 2

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

$$\mathbb{P}\{|L_\eta - 1/2| > \epsilon\}$$

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

$$\mathbb{P}\{|L_\eta - 1/2| > \epsilon\} \leq \mathbb{P}(E_1 \cup \cdots \cup E_K) \qquad \text{uniform bound}$$

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

$$\begin{aligned} \mathbb{P}\{|L_\eta - 1/2| > \epsilon\} &\leq \mathbb{P}(E_1 \cup \cdots \cup E_K) && \text{uniform bound} \\ &\leq \textstyle\sum_{i=1}^{K} \mathbb{P}(E_i) && \text{union bound} \end{aligned}$$

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

$$\begin{aligned} \mathbb{P}\{|L_\eta - 1/2| > \epsilon\} &\leq \mathbb{P}(E_1 \cup \cdots \cup E_K) && \text{uniform bound} \\ &\leq \textstyle\sum_{i=1}^{K} \mathbb{P}(E_i) && \text{union bound} \\ &\leq \textstyle\sum_{i=1}^{K} O(\exp(-\epsilon^2 m)) && \text{concentration of measure} \end{aligned}$$

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

$$
\begin{aligned}
\mathbb{P}\{|L_\eta - 1/2| > \epsilon\} &\leq \mathbb{P}(E_1 \cup \cdots \cup E_K) && \text{uniform bound} \\
&\leq \textstyle\sum_{i=1}^{K} \mathbb{P}(E_i) && \text{union bound} \\
&\leq \textstyle\sum_{i=1}^{K} O(\exp(-\epsilon^2 m)) && \text{concentration of measure} \\
&\leq O(K \exp(-\epsilon^2 m)) && \text{no dependence on } i
\end{aligned}
$$

Uniform Convergence, Part 2

Q: **How do we ensure that L_η is still close to $1/2$?**

A: Make m bigger! *But how big?*

"Theorem" (Union bound). $\mathbb{P}(E_1 \cup E_2 \cup \cdots \cup E_K) \leq \sum_{i=1}^{K} \mathbb{P}(E_i)$.

Let E_i be the event $|L_i - 1/2| > \epsilon$.

$$\begin{aligned}
\mathbb{P}\{|L_\eta - 1/2| > \epsilon\} &\leq \mathbb{P}(E_1 \cup \cdots \cup E_K) && \text{uniform bound} \\
&\leq \textstyle\sum_{i=1}^{K} \mathbb{P}(E_i) && \text{union bound} \\
&\leq \textstyle\sum_{i=1}^{K} O(\exp(-\epsilon^2 m)) && \text{concentration of measure} \\
&\leq O(K \exp(-\epsilon^2 m)) && \text{no dependence on } i
\end{aligned}$$

In other words, $|L_\eta - 1/2| \leq \sqrt{\dfrac{\ln K + \ln 2/\delta}{2m}}$ with probability at least $1 - \delta$.

Can we study generalization under interpolation using standard tools?

Can we study generalization under interpolation using standard tools?

The classical approach to controlling generalization error exploits **uniform convergence**.

Can we study generalization under interpolation using standard tools?

The classical approach to controlling generalization error exploits **uniform convergence**.

Key idea:

$$\text{Risk(Learned Model)} = [\text{Empirical Risk (Learned Model)}] + [\text{Generalization Error (Learned Model)}]$$

Can we study generalization under interpolation using standard tools?

The classical approach to controlling generalization error exploits **uniform convergence**.

Key idea:

$$\begin{aligned}\text{Risk(Learned Model)} &= [\text{Empirical Risk (Learned Model)}] \\ &\quad + [\text{Generalization Error (Learned Model)}] \\ &\leq [\text{Empirical Risk (Learned Model)}] \\ &\quad + [\text{Worst Generalization Error of Any Possible Model}]\end{aligned}$$

Can we study generalization under interpolation using standard tools?

The classical approach to controlling generalization error exploits **uniform convergence**.

Key idea:

$$\begin{aligned}\text{Risk(Learned Model)} &= [\text{Empirical Risk (Learned Model)}] \\ &\quad + [\text{Generalization Error (Learned Model)}] \\ &\leq [\text{Empirical Risk (Learned Model)}] \\ &\quad + [\text{Worst Generalization Error of Any Possible Model}]\end{aligned}$$

This approach works well for "small model classes",

Can we study generalization under interpolation using standard tools?

The classical approach to controlling generalization error exploits **uniform convergence**.

Key idea:

$$\begin{aligned}\text{Risk(Learned Model)} &= [\text{Empirical Risk (Learned Model)}] \\ &\quad + [\text{Generalization Error (Learned Model)}] \\ &\leq [\text{Empirical Risk (Learned Model)}] \\ &\quad + [\text{Worst Generalization Error of Any Possible Model}]\end{aligned}$$

This approach works well for "small model classes",
but not for explaining modern machine learning.

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

- $\mathcal{H}$, containing the information about the neural network architecture;

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

- $\mathcal{H}$, containing the information about the neural network architecture;
- m the size of the training set;

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

- $\mathcal{H}$, containing the information about the neural network architecture;
- m the size of the training set;
- δ, the probability of failure of the bound;

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

- $\mathcal{H}$, containing the information about the neural network architecture;
- m the size of the training set;
- δ, the probability of failure of the bound;
- data distribution;

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

- $\mathcal{H}$, containing the information about the neural network architecture;
- m the size of the training set;
- δ, the probability of failure of the bound;
- data distribution;
- algorithm (like SGD);

"Explaining" deep learning with generalization bounds

Typical result: With probability at least $1 - \delta$ over the training data S,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \underbrace{\epsilon(\ldots)}_{\text{generalization bound}}$$

$\epsilon(\ldots)$ may depend on...

- $\mathcal{H}$, containing the information about the neural network architecture;
- m the size of the training set;
- δ, the probability of failure of the bound;
- data distribution;
- algorithm (like SGD);
- the classifier itself...

Does PAC learnability of Neural Networks explain generalization?

Classical generalization bound: with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \epsilon(\mathcal{H}, m, \delta)$$

Does PAC learnability of Neural Networks explain generalization?

Classical generalization bound: with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \overbrace{O\left(\sqrt{\frac{\mathrm{VCdim}(\mathcal{H}) + \ln 2/\delta}{m}}\right)}^{\epsilon(\mathcal{H}, m, \delta)}$$

Does PAC learnability of Neural Networks explain generalization?

Classical generalization bound: with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \epsilon(\mathcal{H}, m, \delta)$$

parameters in first layer *versus* # data

$784 \times 600 = 470400 > 55000$

$$O\left(\sqrt{\frac{\mathrm{VCdim}(\mathcal{H}) + \ln 2/\delta}{m}}\right)$$

Does PAC learnability of Neural Networks explain generalization?

Classical generalization bound: with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \epsilon(\mathcal{H}, m, \delta)$$

parameters in first layer *versus* # data
$784 \times 600 = 470400 > 55000$

$$\overbrace{O\left(\sqrt{\frac{\mathrm{VCdim}(\mathcal{H}) + \ln 2/\delta}{m}}\right)}$$

There's no distribution-free explanation of deep learning.

Does PAC learnability of Neural Networks explain generalization?

Classical generalization bound: with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(\hat{h}) - L_{\mathcal{D}}(\hat{h}) \leq \epsilon(\mathcal{H}, m, \delta)$$

parameters in first layer *versus* # data
$784 \times 600 = 470400 > 55000$

$$\overbrace{O\left(\sqrt{\frac{\mathrm{VCdim}(\mathcal{H}) + \ln 2/\delta}{m}}\right)}$$

There's no distribution-free explanation of deep learning.

And now, a brief history lesson...

Backstory – Zhang et al. (2016,2017)

Backstory – Zhang et al. (2016,2017)

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

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

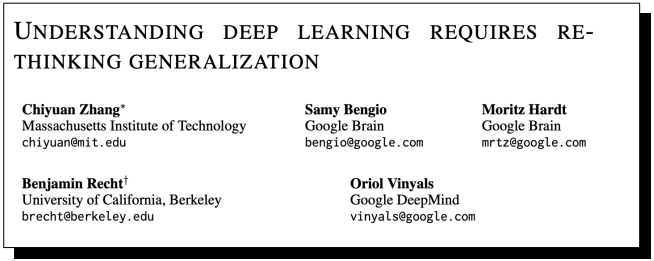
Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

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

Oriol Vinyals
Google DeepMind
vinyals@google.com

Backstory – Zhang et al. (2016,2017)



UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING 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

- Networks on standard data sets could achieve both zero training error and good test error, without regularization.
- Observed that networks had very large capacity—they could memorize random labels. Problematic for VC and Rademacher bounds for 0-1 loss class (error): the bounds are vacuous.

Backstory – Zhang et al. (2016,2017)

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING 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

- Networks on standard data sets could achieve both zero training error and good test error, without regularization.
- Observed that networks had very large capacity—they could memorize random labels. Problematic for VC and Rademacher bounds for 0-1 loss class (error): the bounds are vacuous.
- Reaction from many experts: can be explained via margin and implicit norm-based capacity control.

Backstory – Neyshabur, Tomioka, Srebro (2015)

Backstory – Neyshabur, Tomioka, Srebro (2015)

In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning

Behnam Neyshabur, Ryota Tomioka & Nathan Srebro
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA
{bneyshabur,tomioka,nati}@ttic.edu

Backstory – Neyshabur, Tomioka, Srebro (2015)

IN SEARCH OF THE REAL INDUCTIVE BIAS:
ON THE ROLE OF IMPLICIT REGULARIZATION IN DEEP LEARNING

Behnam Neyshabur, Ryota Tomioka & Nathan Srebro
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA
{bneyshabur,tomioka,nati}@ttic.edu

- Observed that SGD's generalization error (i.e., difference between training and test error) was not affected by increasing network size or lack of regularization.

Backstory – Neyshabur, Tomioka, Srebro (2015)

IN SEARCH OF THE REAL INDUCTIVE BIAS:
ON THE ROLE OF IMPLICIT REGULARIZATION IN DEEP LEARNING

Behnam Neyshabur, Ryota Tomioka & Nathan Srebro
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA
{bneyshabur,tomioka,nati}@ttic.edu

- Observed that SGD's generalization error (i.e., difference between training and test error) was not affected by increasing network size or lack of regularization.

Backstory – Neyshabur, Tomioka, Srebro (2015)

IN SEARCH OF THE REAL INDUCTIVE BIAS:
ON THE ROLE OF IMPLICIT REGULARIZATION IN DEEP LEARNING

Behnam Neyshabur, Ryota Tomioka & Nathan Srebro
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA
{bneyshabur,tomioka,nati}@ttic.edu

- Observed that SGD's generalization error (i.e., difference between training and test error) was not affected by increasing network size or lack of regularization.
- Motivated authors to propose *implicit regularization* as explanation.

Backstory – Neyshabur, Tomioka, Srebro (2015)

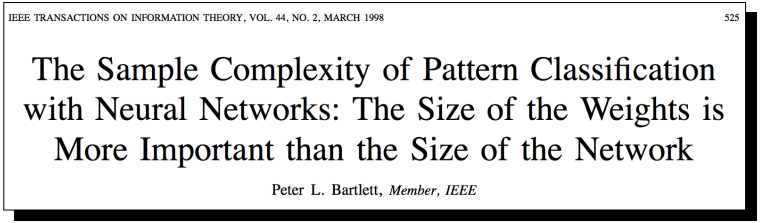
IN SEARCH OF THE REAL INDUCTIVE BIAS:
ON THE ROLE OF IMPLICIT REGULARIZATION IN DEEP LEARNING

Behnam Neyshabur, Ryota Tomioka & Nathan Srebro
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA
{bneyshabur,tomioka,nati}@ttic.edu

- Observed that SGD's generalization error (i.e., difference between training and test error) was not affected by increasing network size or lack of regularization.
- Motivated authors to propose *implicit regularization* as explanation.
- Likely explanation - no explicit regularization, SGD has implicit regularization, perhaps controlling the norms and maximizing the margin.

Backstory – Bartlett (1999)

Backstory – Bartlett (1999)

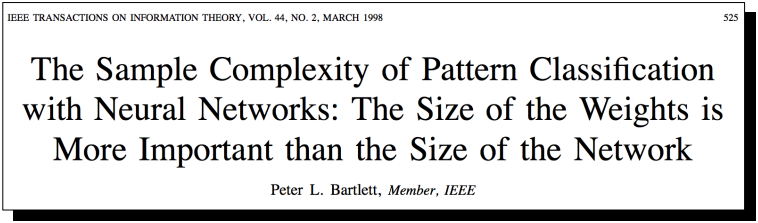


IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 2, MARCH 1998 525

The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network

Peter L. Bartlett, *Member, IEEE*

Backstory – Bartlett (1999)



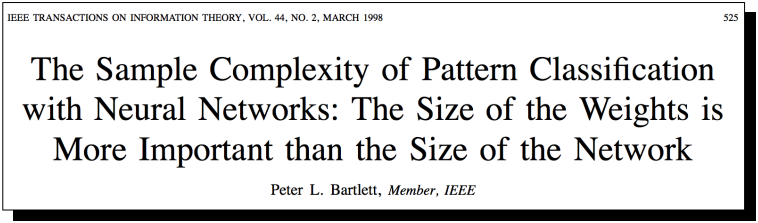
IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 2, MARCH 1998 525

The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network

Peter L. Bartlett, *Member, IEEE*

- Barlett set out to explain why "large" neural networks generalized.

Backstory – Bartlett (1999)

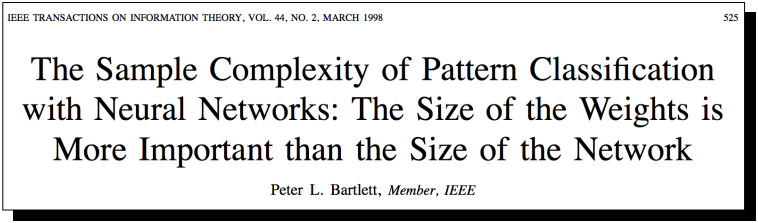

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 2, MARCH 1998 525

The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network

Peter L. Bartlett, *Member, IEEE*

- Barlett set out to explain why "large" neural networks generalized.
- It was common to use weight decay (implicit L2 regularization).

Backstory – Bartlett (1999)

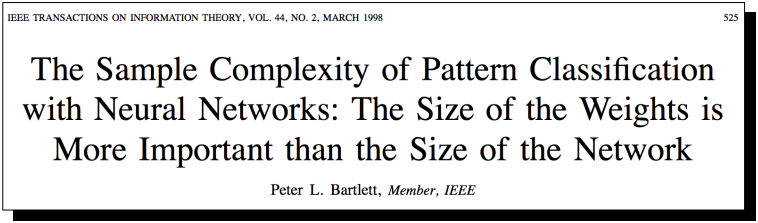

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 2, MARCH 1998 525

The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network

Peter L. Bartlett, *Member, IEEE*

- Barlett set out to explain why "large" neural networks generalized.
- It was common to use weight decay (implicit L2 regularization).
- Barlett derived generalization bounds in terms of the *norms* of the weights in the network, showing even infinite networks could generalize.

Backstory – Bartlett (1999)


IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 2, MARCH 1998

The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network

Peter L. Bartlett, *Member, IEEE*

- Barlett set out to explain why "large" neural networks generalized.
- It was common to use weight decay (implicit L2 regularization).
- Barlett derived generalization bounds in terms of the *norms* of the weights in the network, showing even infinite networks could generalize.
- Bartlett looked at sigmoid networks and a particular norm... technically, need a result for ReLU networks...

Backstory – Neyshabur, Tomioka, Srebro (2015)

Backstory – Neyshabur, Tomioka, Srebro (2015)

Norm-Based Capacity Control in Neural Networks

Behnam Neyshabur BNEYSHABUR@TTIC.EDU
Ryota Tomioka TOMIOKA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago, IL 60637, USA

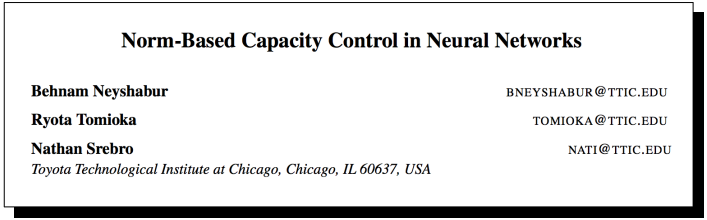
Backstory – Neyshabur, Tomioka, Srebro (2015)

Norm-Based Capacity Control in Neural Networks

Behnam Neyshabur BNEYSHABUR@TTIC.EDU
Ryota Tomioka TOMIOKA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago, IL 60637, USA

- Introduced "path-norm" based Rademacher bounds, custom built for ReLUs.

Backstory – Neyshabur, Tomioka, Srebro (2015)

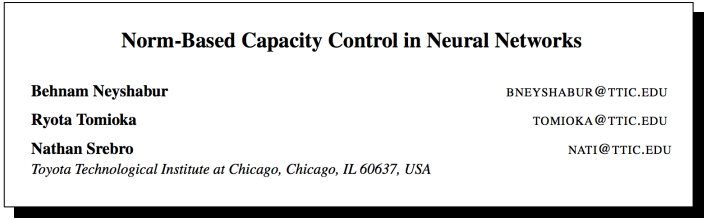


Norm-Based Capacity Control in Neural Networks

Behnam Neyshabur BNEYSHABUR@TTIC.EDU
Ryota Tomioka TOMIOKA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago, IL 60637, USA

- Introduced "path-norm" based Rademacher bounds, custom built for ReLUs.
- What the bounds say: no matter what size neural network, small norm + large margin $\Rightarrow$ small generalization error.

Backstory – Neyshabur, Tomioka, Srebro (2015)



Norm-Based Capacity Control in Neural Networks

Behnam Neyshabur BNEYSHABUR@TTIC.EDU
Ryota Tomioka TOMIOKA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago, IL 60637, USA

- Introduced "path-norm" based Rademacher bounds, custom built for ReLUs.
- What the bounds say: no matter what size neural network, small norm + large margin $\Rightarrow$ small generalization error.

- Do these bounds explain generalization in deep learning?

Backstory – Neyshabur, Tomioka, Srebro (2015)

Norm-Based Capacity Control in Neural Networks

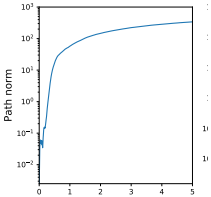
Behnam Neyshabur BNEYSHABUR@TTIC.EDU
Ryota Tomioka TOMIOKA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago, IL 60637, USA

- Introduced "path-norm" based Rademacher bounds, custom built for ReLUs.
- What the bounds say: no matter what size neural network, small norm + large margin $\Rightarrow$ small generalization error.

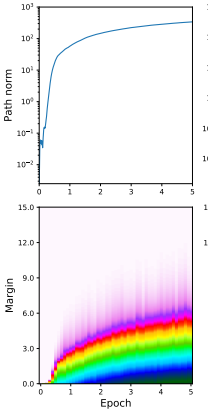
- Do these bounds explain generalization in deep learning?
- The norms are data-dependent, an empirical evaluation is needed!

Empirical evaluation (UAI 2017)

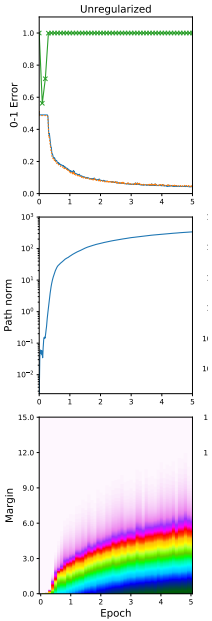
Empirical evaluation (UAI 2017)



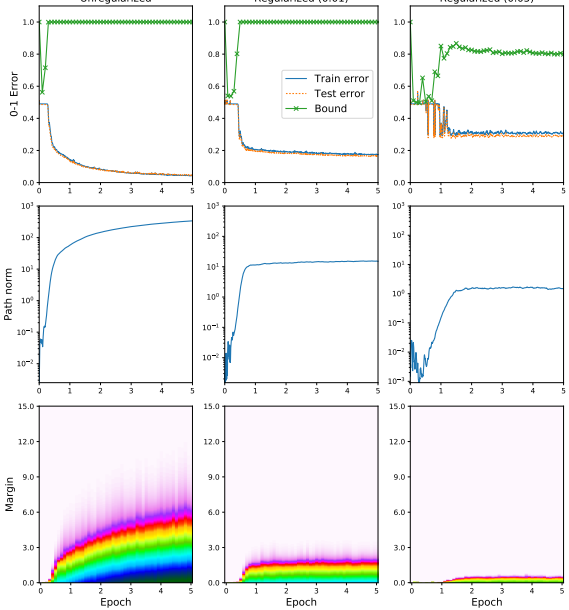
Empirical evaluation (UAI 2017)



Empirical evaluation (UAI 2017)

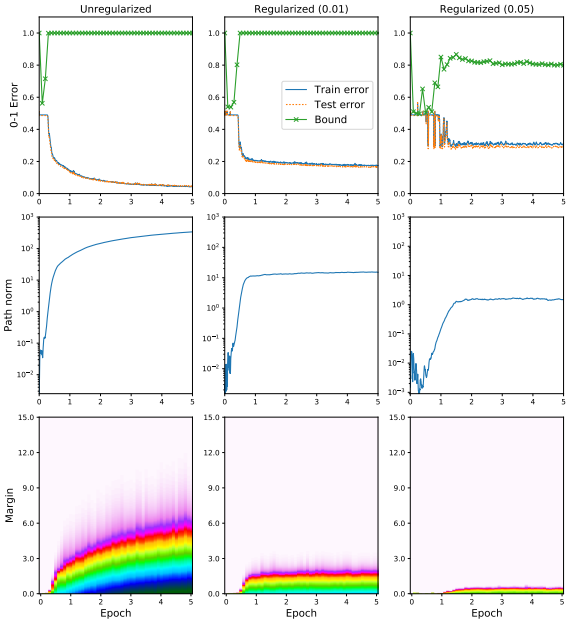


Empirical evaluation (UAI 2017)



Empirical evaluation (UAI 2017)

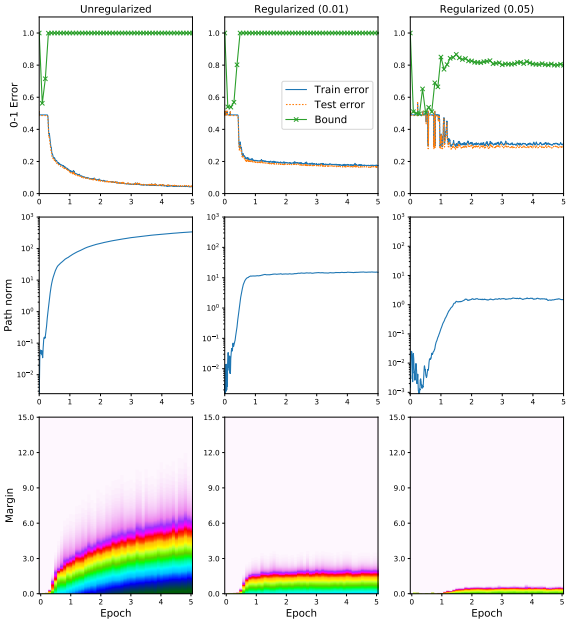
Empirically, norm not small enough relative to margin to explain generalization.



Empirical evaluation (UAI 2017)

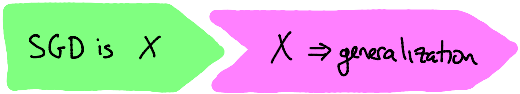
Empirically, norm not small enough relative to margin to explain generalization.

SGD is $\not\Rightarrow$ $\not\Rightarrow$ $\Rightarrow$ generalization

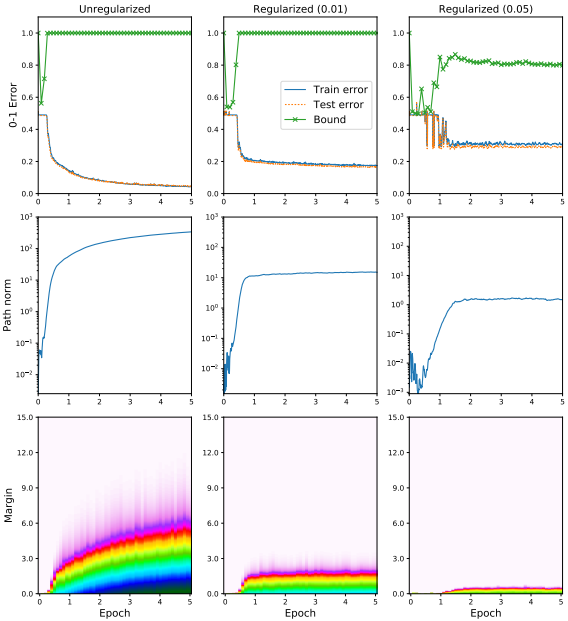


Empirical evaluation (UAI 2017)

Empirically, norm not small enough relative to margin to explain generalization.



What structure can we connect to generalization both theoretically and empirically?



Exploiting Flatness to Bound Generalization Error

If not margin and norm, what structure is sufficient for generalization?

Exploiting Flatness to Bound Generalization Error

If not margin and norm, what structure is sufficient for generalization?

On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima

Nitish Shirish Keskar*
Northwestern University
Evanston, IL 60208
keskar.nitish@u.northwestern.edu

Dheevatsa Mudigere
Intel Corporation
Bangalore, India
dheevatsa.mudigere@intel.com

Jorge Nocedal
Northwestern University
Evanston, IL 60208
j-nocedal@northwestern.edu

Mikhail Smelyanskiy
Intel Corporation
Santa Clara, CA 95054
mikhail.smelyanskiy@intel.com

Ping Tak Peter Tang
Intel Corporation
Santa Clara, CA 95054
peter.tang@intel.com

Exploiting Flatness to Bound Generalization Error

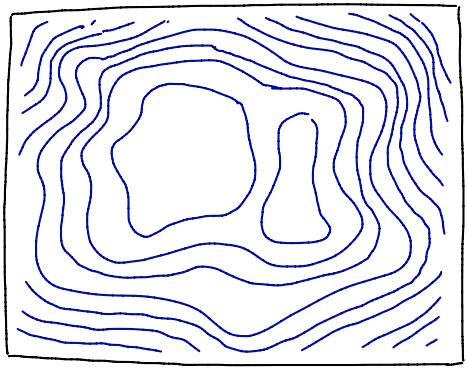
If not margin and norm, what structure is sufficient for generalization?

Our idea: flatness means that nontrivial random perturbation of weights have same loss.

Exploiting Flatness to Bound Generalization Error

If not margin and norm, what structure is sufficient for generalization?

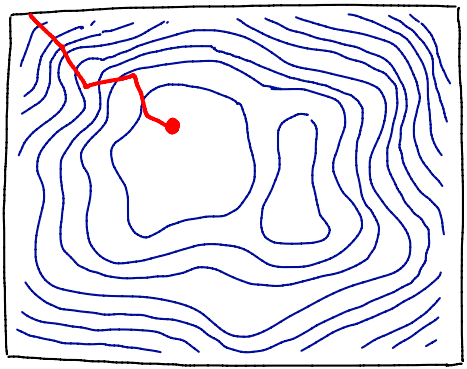
Our idea: flatness means that nontrivial random perturbation of weights have same loss.



Exploiting Flatness to Bound Generalization Error

If not margin and norm, what structure is sufficient for generalization?

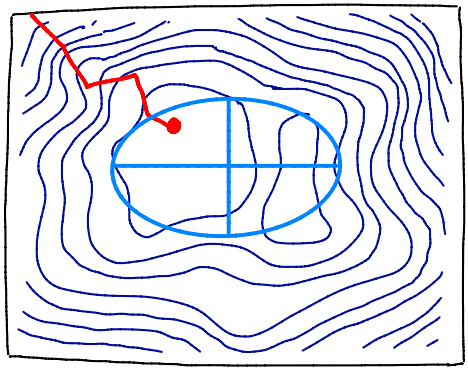
Our idea: flatness means that nontrivial random perturbation of weights have same loss.



Exploiting Flatness to Bound Generalization Error

If not margin and norm, what structure is sufficient for generalization?

Our idea: flatness means that nontrivial random perturbation of weights have same loss.

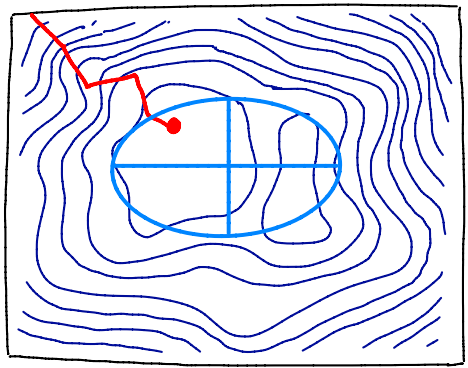


Exploiting Flatness to Bound Generalization Error

If not margin and norm, what structure is sufficient for generalization?

Our idea: flatness means that nontrivial random perturbation of weights have same loss.

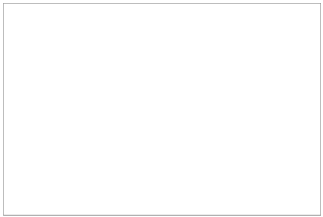
Old ideas: neural networks whose weights have low information generalize (Langford and Caruana, 2002; Hochreiter and Schmidhuber, 1997; Hinton and Van Camp, 1993)



PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.



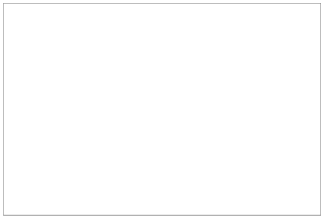
$\mathbb{R}^p$

PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.

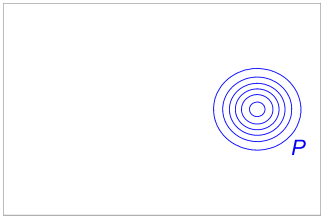


PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").

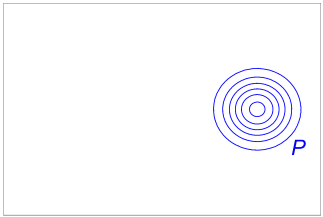


PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").
3. Nature gives us an i.i.d. data set $S \sim \mathcal{D}^m$.
   Now we know the empirical risk $L_S(\cdot) : \mathbb{R}^p \to \mathbb{R}$.

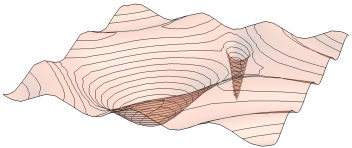


PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").
3. Nature gives us an i.i.d. data set $S \sim \mathcal{D}^m$.
   Now we know the empirical risk $L_S(\cdot) : \mathbb{R}^p \to \mathbb{R}$.



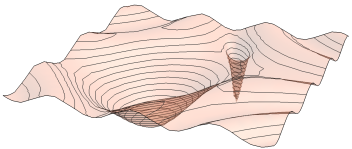
PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").
3. Nature gives us an i.i.d. data set $S \sim \mathcal{D}^m$.
   Now we know the empirical risk $L_S(\cdot) : \mathbb{R}^p \to \mathbb{R}$.
4. Then, with probability at least $(1 - \delta)$,

$$\forall Q,\ \text{GeneralizationError}(Q) \leq \sqrt{\frac{\mathrm{KL}(Q||P) + \ln m/\delta}{2m}}.$$



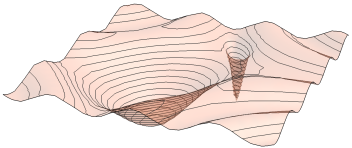
PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").
3. Nature gives us an i.i.d. data set $S \sim \mathcal{D}^m$.
   Now we know the empirical risk $L_S(\cdot) : \mathbb{R}^p \to \mathbb{R}$.
4. Then, with probability at least $(1 - \delta)$,

$$\forall Q,\ \text{GeneralizationError}(Q) \leq \sqrt{\frac{\mathrm{KL}(Q||P) + \ln m/\delta}{2m}}.$$



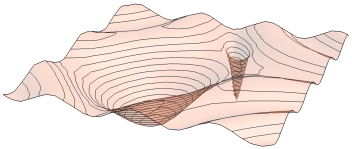
PAC-Bayes generalization bounds

Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").
3. Nature gives us an i.i.d. data set $S \sim \mathcal{D}^m$.
   Now we know the empirical risk $L_S(\cdot) : \mathbb{R}^p \to \mathbb{R}$.
4. Then, with probability at least $(1 - \delta)$,

$$\forall Q,\ \text{GeneralizationError}(Q) \leq \sqrt{\frac{\mathrm{KL}(Q||P) + \ln m/\delta}{2m}}.$$



PAC-Bayes generalization bounds

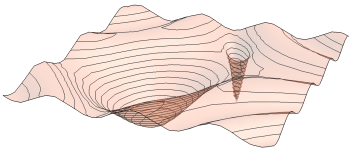
Theorem (PAC-Bayes) McAllester 1998, Shawe-Taylor and Williamson 1997

Assume loss is bounded.

1. Nature chooses a data distribution $\mathcal{D}$.
2. We choose a distribution P on weights (the "prior").
3. Nature gives us an i.i.d. data set $S \sim \mathcal{D}^m$.
   Now we know the empirical risk $L_S(\cdot) : \mathbb{R}^p \to \mathbb{R}$.
4. Then, with probability at least $(1 - \delta)$,

$$\forall Q,\ \text{GeneralizationError}(Q) \leq \sqrt{\frac{\mathrm{KL}(Q||P) + \ln m/\delta}{2m}}.$$

We study data-dependent $Q = Q(S) = \mathcal{N}(w_{\mathrm{SGD}}, \Sigma)$.



Computing nonvacuous generalization bounds... (UAI 2017)

Computing nonvacuous generalization bounds... (UAI 2017)

We exploit flatness via PAC-Bayes *plus nonconvex bound optimization* to obtain first nonvacuous bounds for overparameterized deep nets.

Computing nonvacuous generalization bounds... (UAI 2017)

We exploit flatness via PAC-Bayes *plus nonconvex bound optimization* to obtain first nonvacuous bounds for overparameterized deep nets.

Unlike contemporaneous work on bounds, *focused on empirical evaluation from the start.*

Computing nonvacuous generalization bounds... (UAI 2017)

We exploit flatness via PAC-Bayes *plus nonconvex bound optimization* to obtain first nonvacuous bounds for overparameterized deep nets.

Unlike contemporaneous work on bounds, *focused on empirical evaluation from the start.*

Weaknesses

Computing nonvacuous generalization bounds... (UAI 2017)

We exploit flatness via PAC-Bayes *plus nonconvex bound optimization* to obtain first nonvacuous bounds for overparameterized deep nets.

Unlike contemporaneous work on bounds, *focused on empirical evaluation from the start.*

Weaknesses

- Bounds are still numerically too loose to explain observed generalization.

Computing nonvacuous generalization bounds... (UAI 2017)

We exploit flatness via PAC-Bayes *plus nonconvex bound optimization* to obtain first nonvacuous bounds for overparameterized deep nets.

Unlike contemporaneous work on bounds, *focused on empirical evaluation from the start.*

Weaknesses

- Bounds are still numerically too loose to explain observed generalization.
- Bounds hold only for perturbed classifier, not original one.

Computing nonvacuous generalization bounds... (UAI 2017)

We exploit flatness via PAC-Bayes *plus nonconvex bound optimization* to obtain first nonvacuous bounds for overparameterized deep nets.

Unlike contemporaneous work on bounds, *focused on empirical evaluation from the start.*

Weaknesses

- Bounds are still numerically too loose to explain observed generalization.
- Bounds hold only for perturbed classifier, not original one.
- Bounds don't scale correctly — depend on the distance of the weights to initialization due to the choice of the prior.

What are we missing?
What are the barriers to explaining generalization?

What are we missing?
What are the barriers to explaining generalization?

1. **Statistical barriers**

What are we missing?
What are the barriers to explaining generalization?

1. **Statistical barriers**

2. **Computational barriers**

What are we missing?
What are the barriers to explaining generalization?

1. **Statistical barriers**

2. **Computational barriers**

3. **Analytical barriers**

What are we missing?
What are the barriers to explaining generalization?

1. **Statistical barriers**
   Generalization may rely on favorable properties of the data distribution, but real-world data distributions are unknown.
2. **Computational barriers**

3. **Analytical barriers**

What are we missing?
What are the barriers to explaining generalization?

1. **Statistical barriers**
   Generalization may rely on favorable properties of the data distribution, but real-world data distributions are unknown.
2. **Computational barriers**
   Generalization may rely on quantities that are known but hard to compute.

3. **Analytical barriers**

What are we missing?
What are the barriers to explaining generalization?

1. **Statistical barriers**
   Generalization may rely on favorable properties of the data distribution, but real-world data distributions are unknown.
2. **Computational barriers**
   Generalization may rely on quantities that are known but hard to compute.

3. **Analytical barriers**
   Generalization may happen in ways that our current mathematical tools cannot capture.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Tight PAC-Bayes bounds depend on the data distribution.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Tight PAC-Bayes bounds depend on the data distribution.

Consider a data-dependent posterior $Q = Q(S)$.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Tight PAC-Bayes bounds depend on the data distribution.

Consider a data-dependent posterior $Q = Q(S)$.

PAC-Bayes theorem bounds generalization error in terms of $\mathrm{KL}(Q(S)||P)$

$$\text{GeneralizationError}(Q(S)) \leq \sqrt{\frac{\mathrm{KL}(Q(S)||P) + \ln m/\delta}{2m}}.$$

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Tight PAC-Bayes bounds depend on the data distribution.

Consider a data-dependent posterior $Q = Q(S)$.

PAC-Bayes theorem bounds generalization error in terms of $\mathrm{KL}(Q(S)||P)$

$$\text{GeneralizationError}(Q(S)) \leq \sqrt{\frac{\mathrm{KL}(Q(S)||P) + \ln m/\delta}{2m}}.$$

The prior minimizing the KL term in expectation is a *distribution-dependent* prior $P^* = \mathbb{E}_{S\sim\mathcal{D}^m}[Q(S)]$.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Tight PAC-Bayes bounds depend on the data distribution.

Consider a data-dependent posterior $Q = Q(S)$.

PAC-Bayes theorem bounds generalization error in terms of $\mathrm{KL}(Q(S)||P)$

$$\text{GeneralizationError}(Q(S)) \leq \sqrt{\frac{\mathrm{KL}(Q(S)||P) + \ln m/\delta}{2m}}.$$

The prior minimizing the KL term in expectation is a *distribution-dependent* prior $P^* = \mathbb{E}_{S\sim\mathcal{D}^m}[Q(S)]$.

A prior is a prediction!

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Tight PAC-Bayes bounds depend on the data distribution.

Consider a data-dependent posterior $Q = Q(S)$.

PAC-Bayes theorem bounds generalization error in terms of $\mathrm{KL}(Q(S)||P)$

$$\text{GeneralizationError}(Q(S)) \leq \sqrt{\frac{\mathrm{KL}(Q(S)||P) + \ln m/\delta}{2m}}.$$

The prior minimizing the KL term in expectation is a *distribution-dependent* prior $P^* = \mathbb{E}_{S\sim\mathcal{D}^m}[Q(S)]$.

A prior is a prediction! Unknown data distribution is a statistical barrier.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

1. PAC-Bayes prior P **can** depend on data distribution $\mathcal{D}$ but "can't depend on the sample";

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

1. PAC-Bayes prior P **can** depend on data distribution $\mathcal{D}$ but "can't depend on the sample";
2. Our only handle on the unknown distribution $\mathcal{D}$ is the sample S.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

1. PAC-Bayes prior P **can** depend on data distribution $\mathcal{D}$ but "can't depend on the sample";
2. Our only handle on the unknown distribution $\mathcal{D}$ is the sample S.

In fact, (1) is wrong.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

1. PAC-Bayes prior P **can** depend on data distribution $\mathcal{D}$ but "can't depend on the sample";
2. Our only handle on the unknown distribution $\mathcal{D}$ is the sample S.

In fact, (1) is wrong.

- **Data-dependent PAC-Bayes priors via differential privacy (NeurIPS 2018)**
  Can use the data to learn a prior (predict Q) as long as prediction is differentially private

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

1. PAC-Bayes prior P **can** depend on data distribution $\mathcal{D}$ but "can't depend on the sample";
2. Our only handle on the unknown distribution $\mathcal{D}$ is the sample S.

In fact, (1) is wrong.

- **Data-dependent PAC-Bayes priors via differential privacy (NeurIPS 2018)**
  Can use the data to learn a prior (predict Q) as long as prediction is differentially private
- **On the role of data in PAC-Bayes bounds (AISTATS 2021)**
  Using some the data to approximate distribution-dependent prior P^* is not new (Ambroladze et al. 2007). But, you can outperform P^* using data.

Overcoming statistical barriers in PAC-Bayes via data-dependent priors

Fundamental tension

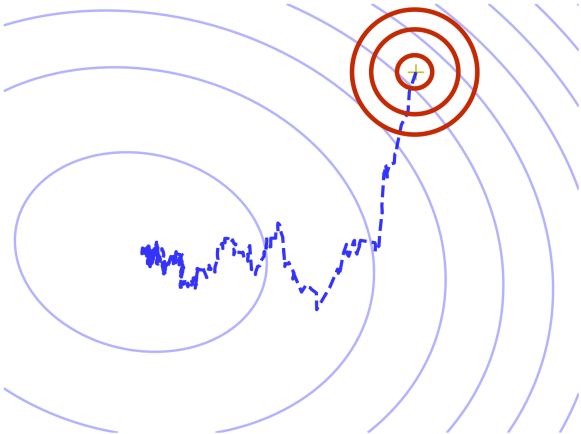
1. PAC-Bayes prior P **can** depend on data distribution $\mathcal{D}$ but "can't depend on the sample";
2. Our only handle on the unknown distribution $\mathcal{D}$ is the sample S.

In fact, (1) is wrong.

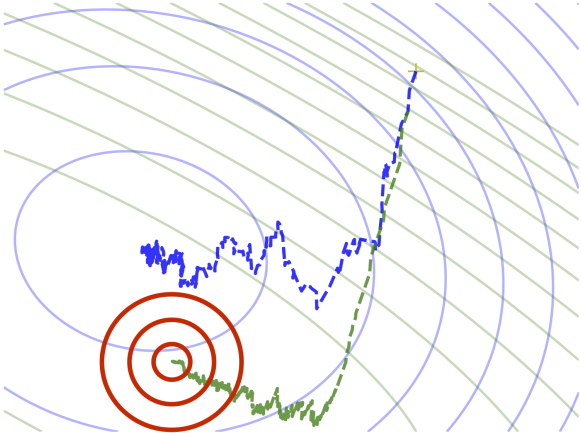
- **Data-dependent PAC-Bayes priors via differential privacy (NeurIPS 2018)**
  Can use the data to learn a prior (predict Q) as long as prediction is differentially private
- **On the role of data in PAC-Bayes bounds (AISTATS 2021)**
  Using some the data to approximate distribution-dependent prior P^* is not new (Ambroladze et al. 2007). But, you can outperform P^* using data.

Theorem (D., Roy, Hsu, Gharbieh, Arpino 2020). *Informally, there's a distribution, loss, and learning algorithm such that a PAC-Bayes bound with oracle prior $P^*(S') = \mathbb{E}[Q(S)]$ is vacuous, but same bound on a subset $S \setminus S'$ with data-dependent oracle prior $P^*(S') = \mathbb{E}[Q(S)|S']$ is nonvacuous.*

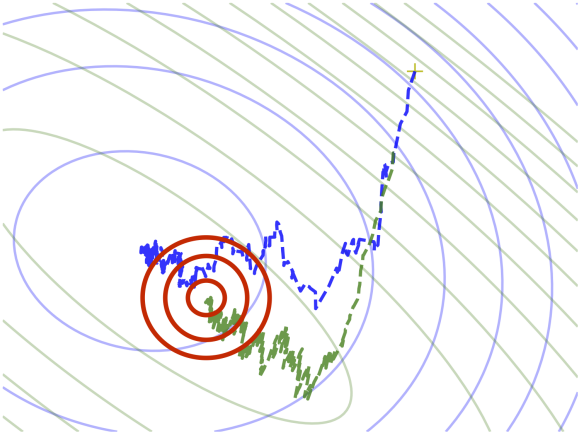
Empirical observation: SGD on S predicted by SGD on $S' \subseteq S$



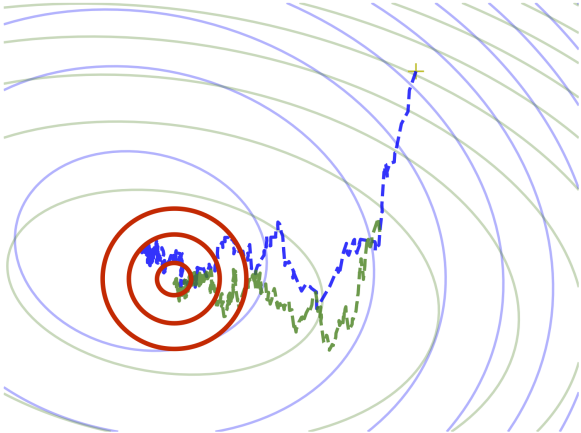
Empirical observation: SGD on S predicted by SGD on $S' \subseteq S$



Empirical observation: SGD on S predicted by SGD on $S' \subseteq S$



Empirical observation: SGD on S predicted by SGD on $S' \subseteq S$



Surrogates for SGDs

- **Up to this point:** Gaussian perturbations of the SGD-learned predictor.
  Capture role of flat minima in generalization.

Surrogates for SGDs

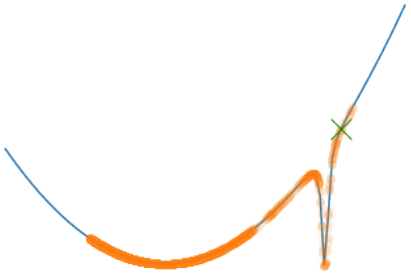
- **Up to this point:** Gaussian perturbations of the SGD-learned predictor.
  Capture role of flat minima in generalization.
- **Next:** Gaussian perturbations at every gradient update.
  Capture role of minibatch gradient covariance in generalization.

Noisy iterative variants of SGD

Noisy iterative variants of SGD

Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathsf{w}_{t+1} = \mathsf{w}_t - \eta_t \nabla F(\mathsf{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

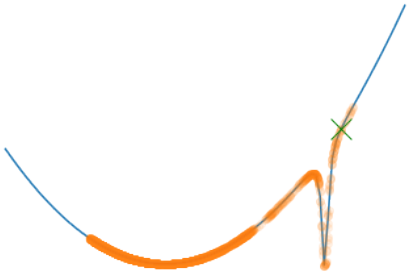


Noisy iterative variants of SGD

Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.

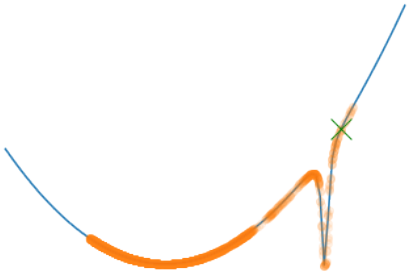


Noisy iterative variants of SGD

Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.



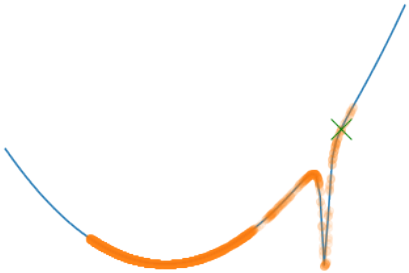
Noisy iterative variants of SGD

Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.

Two views:



Noisy iterative variants of SGD

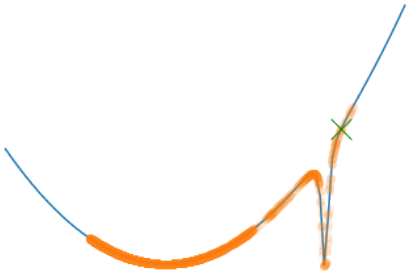
Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.

Two views:

1. Samples from Gibbs distribution $\exp\{-\beta F(\cdot)\}$...



Noisy iterative variants of SGD

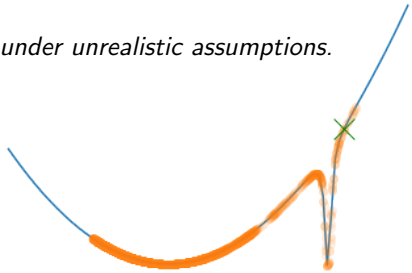
Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.

Two views:

1. Samples from Gibbs distribution $\exp\{-\beta F(\cdot)\}$... *but only under unrealistic assumptions.*



Noisy iterative variants of SGD

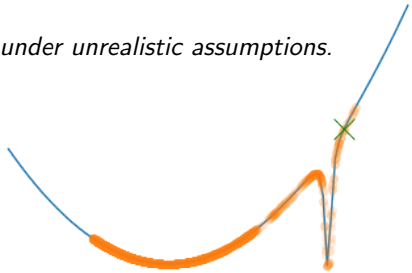
Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.

Two views:

1. Samples from Gibbs distribution $\exp\{-\beta F(\cdot)\}$... *but only under unrealistic assumptions.*
2. Optimizes...



Noisy iterative variants of SGD

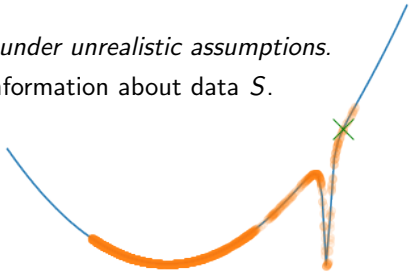
Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.

Two views:

1. Samples from Gibbs distribution $\exp\{-\beta F(\cdot)\}$... *but only under unrealistic assumptions.*
2. Optimizes... and resulting weights don't carry too much information about data S.



Noisy iterative variants of SGD

Stochastic Gradient Langevin Dynamics (Teh and Welling, 2011)

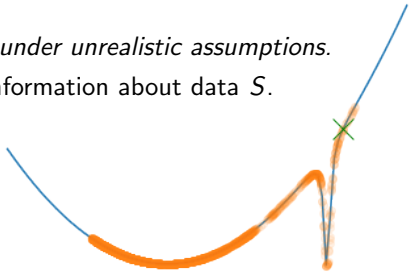
$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t) + \sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t$$

- Additional noise term makes SGLD easier to analyze compared to SGD.
- The inverse temperature β_t trades off exploration v. optimization.

Two views:

1. Samples from Gibbs distribution $\exp\{-\beta F(\cdot)\}$... *but only under unrealistic assumptions.*
2. Optimizes... and resulting weights don't carry too much information about data S.

Does this latter view "explain" generalization?



Mutual information bound

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathrm{w} = \mathscr{A}(S, U)$.

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathsf{w} = \mathscr{A}(S, U)$.
Expected Generalization Error (EGE):

$$\mathrm{EGE}(\mathsf{w}, S) = \mathbb{E}[\,\mathsf{Risk}_{\mathcal{D}}(\mathsf{w}) - \mathsf{EmpiricalRisk}_{S}(\mathsf{w})]$$

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathsf{w} = \mathscr{A}(S, U)$.
Expected Generalization Error (EGE):

$$\mathrm{EGE}(\mathsf{w}, S) = \mathbb{E}[\,\mathsf{Risk}_{\mathcal{D}}(\mathsf{w}) - \mathsf{EmpiricalRisk}_{S}(\mathsf{w})]$$

Assume the loss is bounded (in $[0, 1]$).

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathrm{w} = \mathscr{A}(S, U)$.
Expected Generalization Error (EGE):

$$\mathrm{EGE}(\mathrm{w}, S) = \mathbb{E}[\,\mathrm{Risk}_{\mathcal{D}}(\mathrm{w}) - \mathrm{EmpiricalRisk}_S(\mathrm{w})]$$

Assume the loss is bounded (in $[0, 1]$).

Theorem (XR17, RZ15). $|\mathrm{EGE}(\mathrm{w}, S)| \leq \sqrt{\dfrac{I(\mathrm{w}; S)}{2|S|}}$.

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathsf{w} = \mathscr{A}(S, U)$.
Expected Generalization Error (EGE):

$$\mathrm{EGE}(\mathsf{w}, S) = \mathbb{E}[\,\mathsf{Risk}_{\mathcal{D}}(\mathsf{w}) - \mathsf{EmpiricalRisk}_{S}(\mathsf{w})]$$

Assume the loss is bounded (in $[0, 1]$).

Theorem (XR17, RZ15). $|\mathrm{EGE}(\mathsf{w}, S)| \leq \sqrt{\dfrac{I(\mathsf{w}; S)}{2|S|}}$.

Does this theorem "explain" generalization of $\mathcal{A}(\cdot)$?

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathsf{w} = \mathscr{A}(S, U)$.
Expected Generalization Error (EGE):

$$\mathrm{EGE}(\mathsf{w}, S) = \mathbb{E}[\,\mathsf{Risk}_{\mathcal{D}}(\mathsf{w}) - \mathsf{EmpiricalRisk}_S(\mathsf{w})]$$

Assume the loss is bounded (in $[0, 1]$).

Theorem (XR17, RZ15). $|\mathrm{EGE}(\mathsf{w}, S)| \leq \sqrt{\dfrac{I(\mathsf{w}; S)}{2|S|}}$.

Does this theorem "explain" generalization of $\mathcal{A}(\cdot)$?

- Statistical barrier: $I(\mathsf{w}; S)$ depends on unknown $\mathcal{D}$.

Mutual information bound

Training data: $S \sim \mathcal{D}^n$
Learned weights (e.g., by SGLD): $\mathrm{w} = \mathscr{A}(S, U)$.
Expected Generalization Error (EGE):

$$\mathrm{EGE}(\mathrm{w}, S) = \mathbb{E}[\,\mathrm{Risk}_{\mathcal{D}}(\mathrm{w}) - \mathrm{EmpiricalRisk}_{S}(\mathrm{w})]$$

Assume the loss is bounded (in $[0, 1]$).

Theorem (XR17, RZ15). $|\mathrm{EGE}(\mathrm{w}, S)| \leq \sqrt{\dfrac{I(\mathrm{w}; S)}{2|S|}}$.

Does this theorem "explain" generalization of $\mathcal{A}(\cdot)$?

- Statistical barrier: $I(\mathrm{w}; S)$ depends on unknown $\mathcal{D}$.
- Computational barrier: even if $\mathcal{D}$ were known, $I(\mathrm{w}; S)$ intractable.

Getting around a computational barrier

Computational barrier: even if $\mathcal{D}$ were known, $I(S; \mathrm{w}_T)$ intractable.

Getting around a computational barrier

Computational barrier: even if $\mathcal{D}$ were known, $I(S; \mathrm{w}_T)$ intractable.

Let $S_0, S_1, S_2, \ldots, S_t, \ldots$ be random minibatches of S at time t.

Getting around a computational barrier

Computational barrier: even if $\mathcal{D}$ were known, $I(S; \mathrm{w}_T)$ intractable.

Let $S_0, S_1, S_2, \ldots, S_t, \ldots$ be random minibatches of S at time t.

Pensia, Jog, and Loh (2018) observe that chain rule implies

$$I(S; \mathrm{w}_T) \leq I(S; \mathrm{w}_{0:T}) \leq \sum_{t=1}^{T} \mathrm{I}(S_{t-1}; \mathrm{w}_t | \mathrm{w}_{0:t-1})$$

Getting around a computational barrier

Computational barrier: even if $\mathcal{D}$ were known, $I(S; \mathrm{w}_T)$ intractable.

Let $S_0, S_1, S_2, \ldots, S_t, \ldots$ be random minibatches of S at time t.

Pensia, Jog, and Loh (2018) observe that chain rule implies

$$I(S; \mathrm{w}_T) \leq I(S; \mathrm{w}_{0:T}) \leq \sum_{t=1}^{T} \mathrm{I}(S_{t-1}; \mathrm{w}_t | \mathrm{w}_{0:t-1})$$

In words: *"info leaked about data by final weights"* $\leq \sum_t$ *"info leaked about minibatch at step t"*

Getting around a computational barrier

Computational barrier: even if $\mathcal{D}$ were known, $I(S; \mathrm{w}_T)$ intractable.

Let $S_0, S_1, S_2, \ldots, S_t, \ldots$ be random minibatches of S at time t.

Pensia, Jog, and Loh (2018) observe that chain rule implies

$$I(S; \mathrm{w}_T) \leq I(S; \mathrm{w}_{0:T}) \leq \sum_{t=1}^{T} \mathrm{I}(S_{t-1}; \mathrm{w}_t | \mathrm{w}_{0:t-1})$$

In words: *"info leaked about data by final weights"* $\leq \sum_t$ *"info leaked about minibatch at step t"*

Next hurdle: $\mathrm{I}(S_{t-1}; \mathrm{w}_t | \mathrm{w}_{0:t-1})$ unknown and intractable (statistical + computational barrier)

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathsf{w}_{t+1} | \mathsf{w}_{0:t})$ is optimal expected loss in following prediction game

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathsf{w}_{t+1} | \mathsf{w}_{0:t})$ is optimal expected loss in following prediction game

You observe: $\mathsf{w}_{0:t}$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1} \mid \mathrm{w}_{0:t}, S_t]$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1} \mid \mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1} \mid \mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q \| P)$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1}|\mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q\|\|P)$
Optimal strategy:	$P^* = \mathbb{E}[Q|\mathrm{w}_{0:t}]$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1}|\mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q\|\|P)$
Optimal strategy:	$P^* = \mathbb{E}[Q|\mathrm{w}_{0:t}]$
... and its expected loss:	$\mathbb{E}[\mathrm{KL}(Q\|\|P^*)] = \mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1} \mid \mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q \mid\mid P)$
Optimal strategy:	$P^* = \mathbb{E}[Q \mid \mathrm{w}_{0:t}]$
... and its expected loss:	$\mathbb{E}[\mathrm{KL}(Q \mid\mid P^*)] = \mathrm{I}(S_t; \mathrm{w}_{t+1} \mid \mathrm{w}_{0:t})$

Key idea: Every prediction scheme P yields an upper bound on mutual information.

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1} \mid \mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q \| P)$
Optimal strategy:	$P^* = \mathbb{E}[Q \mid \mathrm{w}_{0:t}]$
... and its expected loss:	$\mathbb{E}[\mathrm{KL}(Q \| P^*)] = \mathrm{I}(S_t; \mathrm{w}_{t+1} \mid \mathrm{w}_{0:t})$

Key idea: Every prediction scheme P yields an upper bound on mutual information.

$$\forall P,\ \mathbb{E}[\mathrm{KL}(Q \| P(\mathrm{w}_{0:t}))] \geq \mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1}|\mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q\|\|P)$
Optimal strategy:	$P^* = \mathbb{E}[Q|\mathrm{w}_{0:t}]$
... and its expected loss:	$\mathbb{E}[\mathrm{KL}(Q\|\|P^*)] = \mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1}|\mathrm{w}_{0:t}, S_t]$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q\|\|P)$
Optimal strategy:	$P^* = \mathbb{E}[Q|\mathrm{w}_{0:t}]$
... and its expected loss:	$\mathbb{E}[\mathrm{KL}(Q\|\|P^*)] = \mathrm{I}(S_t; \mathrm{w}_{t+1}|\mathrm{w}_{0:t})$

Stochastic Gradient Langevin Dynamics adds Gaussian noise at every step:

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S_t) + \underbrace{\sqrt{\frac{2\eta_t}{\beta_t}}\,\varepsilon_t}_{\text{Gaussian noise}} \qquad \varepsilon_t \sim \mathcal{N}(0, \mathbb{I}_d) \text{ i.i.d.}$$

Bounding the one-step information via suboptimal predictions

Insight: $\mathrm{I}(S_t; \mathrm{w}_{t+1} | \mathrm{w}_{0:t})$ is optimal expected loss in following prediction game

You observe:	$\mathrm{w}_{0:t}$
You want to estimate:	$Q = Q(\mathrm{w}_{0:t}, S_t) = \mathbb{P}[\mathrm{w}_{t+1} \mid \mathrm{w}_{0:t}, S_t] = \mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S_t), \frac{2\eta_t}{\beta_t} \mathbb{I}_d)$
If your prediction is:	$P = P(\mathrm{w}_{0:t})$
You pay a loss of:	$\mathrm{KL}(Q \| P)$
Optimal strategy:	$P^* = \mathbb{E}[Q \mid \mathrm{w}_{0:t}]$
... and its expected loss:	$\mathbb{E}[\mathrm{KL}(Q \| P^*)] = \mathrm{I}(S_t; \mathrm{w}_{t+1} \mid \mathrm{w}_{0:t})$

Stochastic Gradient Langevin Dynamics adds Gaussian noise at every step:

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S_t) + \underbrace{\sqrt{\frac{2\eta_t}{\beta_t}}\, \varepsilon_t}_{\text{Gaussian noise}} \qquad \varepsilon_t \sim \mathcal{N}(0, \mathbb{I}_d) \text{ i.i.d.}$$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}}$$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})}$$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})}$$

Pensia et al. observe: $\mathsf{w}_{0:t}$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})}$$

Pensia et al. observe: $\mathsf{w}_{0:t}$

They want to estimate: $Q = \mathcal{N}(\mathsf{w}_t - \eta_t \nabla F(\mathsf{w}_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})}$$

Pensia et al. observe: $\mathsf{w}_{0:t}$

They want to estimate: $Q = \mathcal{N}(\mathsf{w}_t - \eta_t \nabla F(\mathsf{w}_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Their prediction is: $P = \mathcal{N}(\mathsf{w}_t, \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})}$$

Pensia et al. observe:	$\mathsf{w}_{0:t}$
They want to estimate:	$Q = \mathcal{N}(\mathsf{w}_t - \eta_t \nabla F(\mathsf{w}_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
Their prediction is:	$P = \mathcal{N}(\mathsf{w}_t, \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
They pay a loss of:	$\mathrm{KL}(Q\|\|P) = \frac{\eta_t\beta_t}{2}\|\nabla F(\mathsf{w}_t, S_t)\|^2$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})}$$

Pensia et al. observe:	$\mathsf{w}_{0:t}$
They want to estimate:	$Q = \mathcal{N}(\mathsf{w}_t - \eta_t \nabla F(\mathsf{w}_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
Their prediction is:	$P = \mathcal{N}(\mathsf{w}_t, \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
They pay a loss of:	$\mathrm{KL}(Q\|\|P) = \frac{\eta_t\beta_t}{2}\|\nabla F(\mathsf{w}_t, S_t)\|^2 \leq \frac{\eta_t\beta_t}{2}L^2$

Pensia, Jog, and Loh (2018) – a prediction perspective

$$|\mathrm{EGE}(\mathsf{w}_T, S)| \overset{[\mathrm{XR17}]}{\leq} \sqrt{\frac{I(S;\mathsf{w}_T)}{2|S|}} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{2|S|}\sum_{t=1}^{T} \mathrm{I}(S_{t-1};\mathsf{w}_t|\mathsf{w}_{0:t-1})} \overset{[\mathrm{PJL18}]}{\leq} \sqrt{\frac{1}{4n}\sum_{i=1}^{T}\eta_i\beta_i L^2}$$

Pensia et al. observe: $\mathsf{w}_{0:t}$

They want to estimate: $Q = \mathcal{N}(\mathsf{w}_t - \eta_t \nabla F(\mathsf{w}_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Their prediction is: $P = \mathcal{N}(\mathsf{w}_t, \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

They pay a loss of: $\mathrm{KL}(Q||P) = \frac{\eta_t\beta_t}{2}\|\nabla F(\mathsf{w}_t, S_t)\|^2 \leq \frac{\eta_t\beta_t}{2}L^2$

Data-dependent estimates (NeurIPS 2019)

Limitation of mutual information bound: must predict w_{t+1} given $w_{0:t}$ but no data.

Data-dependent estimates (NeurIPS 2019)

Limitation of mutual information bound: must predict w_{t+1} given $w_{0:t}$ but no data.

Solution: Use *conditional mutual information*, conditioning on all but one data point.

Data-dependent estimates (NeurIPS 2019)

Limitation of mutual information bound: must predict w_{t+1} given $w_{0:t}$ but no data.

Solution: Use *conditional mutual information*, conditioning on all but one data point.

We observe: $w_{0:t}, S_{-j}$

Data-dependent estimates (NeurIPS 2019)

Limitation of mutual information bound: must predict w_{t+1} given $w_{0:t}$ but no data.

Solution: Use *conditional mutual information*, conditioning on all but one data point.

We observe: $w_{0:t}, S_{-J}$

We want to estimate: $Q = \mathcal{N}(w_t - \eta_t \nabla F(w_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Data-dependent estimates (NeurIPS 2019)

Limitation of mutual information bound: must predict w_{t+1} given $w_{0:t}$ but no data.

Solution: Use *conditional mutual information*, conditioning on all but one data point.

We observe: $w_{0:t}, S_{-J}$

We want to estimate: $Q = \mathcal{N}(w_t - \eta_t \nabla F(w_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Our prediction is: $P = \begin{cases} Q, & \text{if } S_t \subseteq S_{-J} \\ \mathcal{N}(w_t - \eta_t \nabla F(w_t, S_{t,-J}), \frac{2\eta_t}{\beta_t}\mathbb{I}_d), & \text{o.w.} \end{cases}$

Data-dependent estimates (NeurIPS 2019)

Limitation of mutual information bound: must predict w_{t+1} given $w_{0:t}$ but no data.

Solution: Use *conditional mutual information*, conditioning on all but one data point.

We observe: $w_{0:t}, S_{-J}$

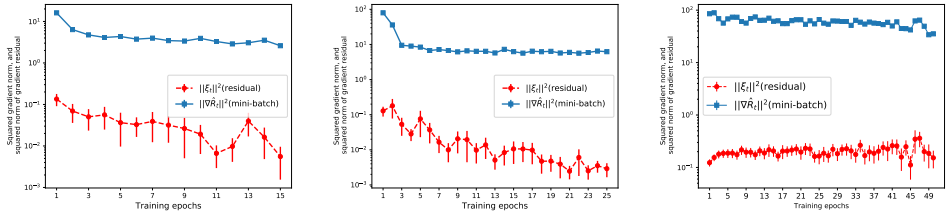
We want to estimate: $Q = \mathcal{N}(w_t - \eta_t \nabla F(w_t, S_t), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Our prediction is:
$$P = \begin{cases} Q, & \text{if } S_t \subseteq S_{-J} \\ \mathcal{N}(w_t - \eta_t \nabla F(w_t, S_{t,-J}), \frac{2\eta_t}{\beta_t}\mathbb{I}_d), & \text{o.w.} \end{cases}$$

We pay a loss of:
$$\mathrm{KL}(Q||P) = \begin{cases} 0, & \text{if } S_t \subseteq S_{-J} \\ \frac{\eta_t \beta_t}{2} \underbrace{\|\nabla F(w_t, S_{t,-J}) - \nabla F(w_t, S_t)\|^2}_{\text{gradient "incoherence"}}, & \text{o.w.} \end{cases}$$

Empirical evaluation of gradient norms

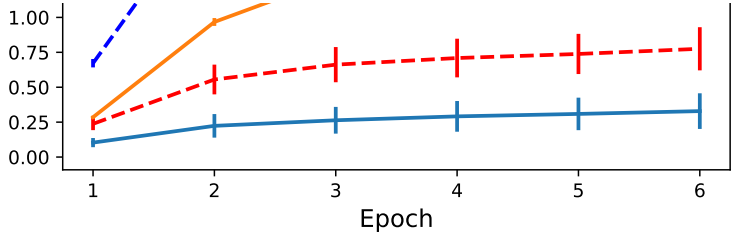
MNIST, Fashion MNIST and CIFAR10



Observation: Gradient norms don't vanish. Centered norms vanish, but not for CIFAR10.

Empirical evaluation on generalization bounds on MNIST

Comparison with Mou et al. (2019) in blue (large step size), orange (small)



Observation: Mou et al. bounds become vacuous within the first 3 epochs because the gradient norms do not vanish. Our bounds grow much slower, because norms vanish if you center them

Conditioning on a super sample (NeurIPS 2020)

Limitation of our 2019 bound: No control over missing data point.

Solution: Assume $S = S'$ or $S = S''$ with equal probability, where S', S'' differ by only a single data point. Condition on $S' \cup S''$ (Steinke and Zakynthinou 2020).

Conditioning on a super sample (NeurIPS 2020)

Limitation of our 2019 bound: No control over missing data point.

Solution: Assume $S = S'$ or $S = S''$ with equal probability, where S', S'' differ by only a single data point. Condition on $S' \cup S''$ (Steinke and Zakynthinou 2020).

Conditioning on a super sample (NeurIPS 2020)

Limitation of our 2019 bound: No control over missing data point.

Solution: Assume $S = S'$ or $S = S''$ with equal probability, where S', S'' differ by only a single data point. Condition on $S' \cup S''$ (Steinke and Zakynthinou 2020).

We observe: $w_{0:t}, S', S''$

Conditioning on a super sample (NeurIPS 2020)

Limitation of our 2019 bound: No control over missing data point.

Solution: Assume $S = S'$ or $S = S''$ with equal probability, where S', S'' differ by only a single data point. Condition on $S' \cup S''$ (Steinke and Zakynthinou 2020).

We observe:	$\mathrm{w}_{0:t}, S', S''$
We want to estimate:	$Q = \mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Conditioning on a super sample (NeurIPS 2020)

Limitation of our 2019 bound: No control over missing data point.

Solution: Assume $S = S'$ or $S = S''$ with equal probability, where S', S'' differ by only a single data point. Condition on $S' \cup S''$ (Steinke and Zakynthinou 2020).

We observe:	$\mathrm{w}_{0:t}, S', S''$
We want to estimate:	$Q = \mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
Our prediction is:	$P = \theta(w_{0:t})\,\mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S'), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
	$+ \big(1 - \theta(w_{0:t})\big)\,\mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S''), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Conditioning on a super sample (NeurIPS 2020)

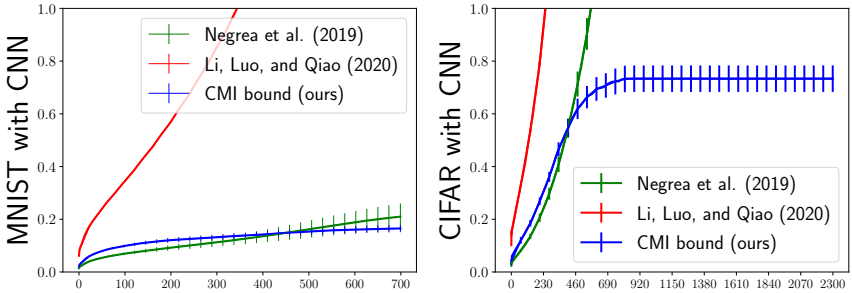
Limitation of our 2019 bound: No control over missing data point.

Solution: Assume $S = S'$ or $S = S''$ with equal probability, where S', S'' differ by only a single data point. Condition on $S' \cup S''$ (Steinke and Zakynthinou 2020).

We observe:	$\mathrm{w}_{0:t}, S', S''$
We want to estimate:	$Q = \mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
Our prediction is:	$P = \theta(w_{0:t})\,\mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S'), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$
	$+ \big(1 - \theta(w_{0:t})\big)\,\mathcal{N}(\mathrm{w}_t - \eta_t \nabla F(\mathrm{w}_t, S''), \frac{2\eta_t}{\beta_t}\mathbb{I}_d)$

Observation: Past iterates $\mathrm{w}_0, ..., \mathrm{w}_t$ provide evidence on $\{S = S'\}$ versus $\{S = S''\}$. Formally, we have a binary hypothesis test and $\theta(w_{0:t})$ is chosen based on conditional distribution of missing data point's identity.

Empirical Results



Bounds for SGD itself?

We've seen that:

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).
- We can use data-dependent priors to reduce impact of initial unstable optimization (and of not knowing $\mathcal{D}$).

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).
- We can use data-dependent priors to reduce impact of initial unstable optimization (and of not knowing $\mathcal{D}$).
- We can use conditional mutual information to reduce impact of measuring information leaked by entire optimization trajectory.

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).
- We can use data-dependent priors to reduce impact of initial unstable optimization (and of not knowing $\mathcal{D}$).
- We can use conditional mutual information to reduce impact of measuring information leaked by entire optimization trajectory.

Important caveat: We've bounded generalization error of **perturbations of SGD**

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).
- We can use data-dependent priors to reduce impact of initial unstable optimization (and of not knowing $\mathcal{D}$).
- We can use conditional mutual information to reduce impact of measuring information leaked by entire optimization trajectory.

Important caveat: We've bounded generalization error of **perturbations of SGD**
... not the same as obtaining bounds on learned weights directly!

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).
- We can use data-dependent priors to reduce impact of initial unstable optimization (and of not knowing $\mathcal{D}$).
- We can use conditional mutual information to reduce impact of measuring information leaked by entire optimization trajectory.

Important caveat: We've bounded generalization error of **perturbations of SGD**
... not the same as obtaining bounds on learned weights directly!

How can we bridge this gap?

Bounds for SGD itself?

We've seen that:

- We can exploit "flatness" by adding noise to the final weights.
- We can exploit gradient "coherence" by perturbing iterates with noise (SGLD).
- We can use data-dependent priors to reduce impact of initial unstable optimization (and of not knowing $\mathcal{D}$).
- We can use conditional mutual information to reduce impact of measuring information leaked by entire optimization trajectory.

Important caveat: We've bounded generalization error of **perturbations of SGD**
... not the same as obtaining bounds on learned weights directly!

How can we bridge this gap?

Bounds on SGD through a surrogate?

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{v}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{\nu}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

What we've done:

$$\text{Risk}(\tilde{\nu}) = \text{EmpiricalRisk}\ (\tilde{\nu}) + \text{GeneralizationError}\ (\tilde{\nu}, S)$$

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{\nu}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

What we've done:

$$\begin{aligned}\text{Risk}(\tilde{\nu}) &= \text{EmpiricalRisk}\ (\tilde{\nu}) + \text{GeneralizationError}\ (\tilde{\nu},\text{S}) \\ &\leq \text{EmpiricalRisk}\ (\tilde{\nu}) + \max_{\nu \in \mathcal{P}} \text{GeneralizationError}\ (\nu,\text{S})\end{aligned}$$

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{v}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

What we've done:

$$\begin{aligned}\text{Risk}(\tilde{v}) &= \text{EmpiricalRisk}\ (\tilde{v}) + \text{GeneralizationError}\ (\tilde{v},\text{S}) \\ &\leq \text{EmpiricalRisk}\ (\tilde{v}) + \max_{v \in \mathcal{P}} \text{GeneralizationError}\ (\text{v},\text{S})\end{aligned}$$

Direct approach via uniform convergence

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk}\ (\hat{w}) + \max_{w \in \mathcal{H}} \text{GeneralizationError}\ (\text{w},\text{S})$$

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{\nu}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

What we've done:

$$\begin{aligned}\text{Risk}(\tilde{\nu}) &= \text{EmpiricalRisk}\ (\tilde{\nu}) + \text{GeneralizationError}\ (\tilde{\nu},\text{S}) \\ &\leq \text{EmpiricalRisk}\ (\tilde{\nu}) + \max_{\nu \in \mathcal{P}} \text{GeneralizationError}\ (\text{v},\text{S})\end{aligned}$$

Direct approach via uniform convergence

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk}\ (\hat{w}) + \max_{w \in \mathcal{H}} \text{GeneralizationError}\ (\text{w},\text{S})$$

Indirect "surrogate" approach via uniform convergence of a surrogate class

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk}\ (\tilde{\nu}) + \max_{\nu \in \mathcal{P}} \text{GeneralizationError}\ (\text{v},\text{S}) + \underbrace{\text{Risk}(\hat{w}) - \text{Risk}(\tilde{\nu})}_{\text{or some bound}}$$

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{v}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

What we've done:

$$\begin{aligned}\text{Risk}(\tilde{v}) &= \text{EmpiricalRisk }(\tilde{v}) + \text{GeneralizationError }(\tilde{v},S) \\ &\leq \text{EmpiricalRisk }(\tilde{v}) + \max_{v \in \mathcal{P}} \text{GeneralizationError }(v,S)\end{aligned}$$

Direct approach via uniform convergence

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk }(\hat{w}) + \max_{w \in \mathcal{H}} \text{GeneralizationError }(w,S)$$

Indirect "surrogate" approach via uniform convergence of a surrogate class

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk }(\tilde{v}) + \max_{v \in \mathcal{P}} \text{GeneralizationError }(v,S) + \underbrace{\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})}_{\text{or some bound}}$$

Worth the trouble?

Bounds on SGD through a surrogate?

Let $\hat{w}$ be weights learned by SGD.
Let $\tilde{v}$ be a **perturbed** weight *distribution*, i.e., a "surrogate" classifier.

What we've done:

$$\begin{aligned}\text{Risk}(\tilde{v}) &= \text{EmpiricalRisk }(\tilde{v}) + \text{GeneralizationError }(\tilde{v},\text{S}) \\ &\leq \text{EmpiricalRisk }(\tilde{v}) + \max_{v \in \mathcal{P}} \text{GeneralizationError }(\text{v},\text{S})\end{aligned}$$

Direct approach via uniform convergence

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk }(\hat{w}) + \max_{w \in \mathcal{H}} \text{GeneralizationError }(\text{w},\text{S})$$

Indirect "surrogate" approach via uniform convergence of a surrogate class

$$\text{Risk}(\hat{w}) \leq \text{EmpiricalRisk }(\tilde{v}) + \max_{v \in \mathcal{P}} \text{GeneralizationError }(\text{v},\text{S}) + \underbrace{\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})}_{\text{or some bound}}$$

Worth the trouble? Empirically, no direct approach gives numerically nonvacuous bounds.

Analytical Barrier of UC bounds: Nagarajan and Kolter (2019)

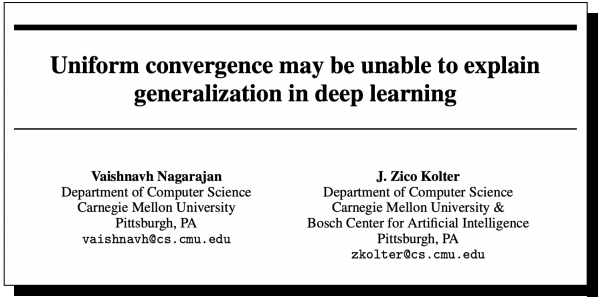
Analytical Barrier of UC bounds: Nagarajan and Kolter (2019)

Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

Analytical Barrier of UC bounds: Nagarajan and Kolter (2019)



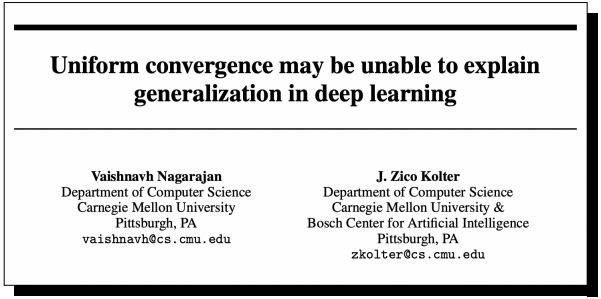
Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

- Observed that, in norm-based bounds, norm terms growing faster that $\sqrt{n}$

Analytical Barrier of UC bounds: Nagarajan and Kolter (2019)



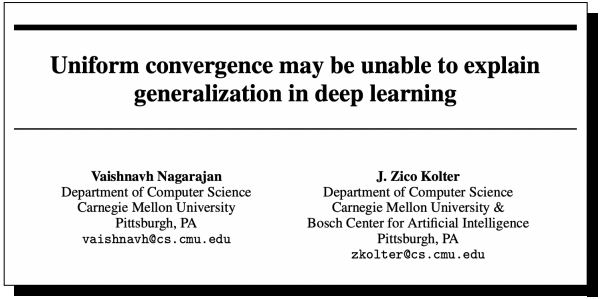
Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

- Observed that, in norm-based bounds, norm terms growing faster that $\sqrt{n}$
  Hence, norm-based bounds scale incorrectly with number of data.

Analytical Barrier of UC bounds: Nagarajan and Kolter (2019)



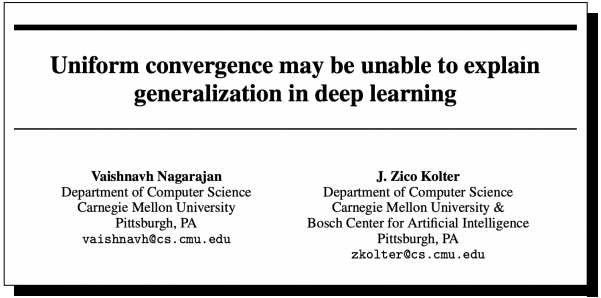
Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

- Observed that, in norm-based bounds, norm terms growing faster that $\sqrt{n}$
  Hence, norm-based bounds scale incorrectly with number of data.
- Gave toy examples where SGD learns a good classifier that, nonetheless, does NOT belong to a class with uniformly small generalization error.

Analytical Barrier of UC bounds: Nagarajan and Kolter (2019)



Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

- Observed that, in norm-based bounds, norm terms growing faster that $\sqrt{n}$
  Hence, norm-based bounds scale incorrectly with number of data.
- Gave toy examples where SGD learns a good classifier that, nonetheless, does NOT belong to a class with uniformly small generalization error.
  Hence, standard "uniform convergence" arguments may fail to explain deep learning.

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}
\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.} \\
&+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v} \\
&+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}
\end{aligned}$$

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

The generalization error of SGD can be broken down into:

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

The generalization error of SGD can be broken down into:

- the error from approximation by the surrogate, and

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

The generalization error of SGD can be broken down into:

- the error from approximation by the surrogate, and
- the generalization error of the surrogate
  ... and we can bound this uniformly over a **surrogate class**

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

The generalization error of SGD can be broken down into:

- the error from approximation by the surrogate, and
- the generalization error of the surrogate
  ... and we can bound this uniformly over a **surrogate class**

Our approach: Relate the generalization error for SGD to the generalization error of a better behaved *surrogate* classifier.

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}
\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.} \\
&+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v} \\
&+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}
\end{aligned}$$

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

- Toy example of Nagarajan and Kolter 2019 admits a surrogate analysis via uniform convergence.

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\mathrm{Risk}(\hat{w}) - \mathrm{Emp.Risk}(\hat{w})] &= \mathbb{E}[\mathrm{Risk}(\hat{w}) - \mathrm{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\mathrm{Risk}(\tilde{v}) - \mathrm{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\mathrm{Emp.Risk}(\tilde{v}) - \mathrm{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

- Toy example of Nagarajan and Kolter 2019 admits a surrogate analysis via uniform convergence.
- We show bias–variance analysis of overparametrized linear regression (Bartlett, Long, Lugosi, Tsigler, 2019) is a surrogate decomposition and can be made uniform.

Does [NK19] undermine the indirect surrogate approach? No.

Definition (Surrogate decomposition; NDR20). For any surrogate hypothesis $\tilde{v}$,

$$\begin{aligned}\mathbb{E}[\text{Risk}(\hat{w}) - \text{Emp.Risk}(\hat{w})] &= \mathbb{E}[\text{Risk}(\hat{w}) - \text{Risk}(\tilde{v})] && \text{risk diff.}\\ &+ \mathbb{E}[\text{Risk}(\tilde{v}) - \text{Emp.Risk}(\tilde{v})] && \text{generalization error of } \tilde{v}\\ &+ \mathbb{E}[\text{Emp.Risk}(\tilde{v}) - \text{Emp.Risk}(\hat{w})] && \text{empirical risk diff.}\end{aligned}$$

- Toy example of Nagarajan and Kolter 2019 admits a surrogate analysis via uniform convergence.
- We show bias–variance analysis of overparametrized linear regression (Bartlett, Long, Lugosi, Tsigler, 2019) is a surrogate decomposition and can be made uniform.
- We show that surrogates can be defined by probabilistic conditioning in order to ensure risk difference is zero in expectation.

Connecting surrogates back to SGD

We've discussed generalization bounds for two classes of surrogates:

Connecting surrogates back to SGD

We've discussed generalization bounds for two classes of surrogates:

- Gaussian perturbations of $\mathrm{w}_T = SGD(S)$;
- Gaussian perturbations of w_t, at every step t.

Connecting surrogates back to SGD

We've discussed generalization bounds for two classes of surrogates:

- Gaussian perturbations of $\mathrm{w}_T = SGD(S)$;
- Gaussian perturbations of w_t, at every step t.

Can we control the "risk difference" term to get a surrogate decomposition?

Connecting surrogates back to SGD

We've discussed generalization bounds for two classes of surrogates:

- Gaussian perturbations of $\mathrm{w}_T = SGD(S)$;
- Gaussian perturbations of w_t, at every step t.

Can we control the "risk difference" term to get a surrogate decomposition?

Empirical obs.: All proposals to "derandomize" PAC-Bayes bounds produce vacuous bounds.

Connecting surrogates back to SGD

We've discussed generalization bounds for two classes of surrogates:

- Gaussian perturbations of $w_T = SGD(S)$;
- Gaussian perturbations of w_t, at every step t.

Can we control the "risk difference" term to get a surrogate decomposition?

Empirical obs.: All proposals to "derandomize" PAC-Bayes bounds produce vacuous bounds.

Explanation: Arguments are "too uniform", would contradict NK19.

Connecting surrogates back to SGD

We've discussed generalization bounds for two classes of surrogates:

- Gaussian perturbations of $w_T = SGD(S)$;
- Gaussian perturbations of w_t, at every step t.

Can we control the "risk difference" term to get a surrogate decomposition?

Empirical obs.: All proposals to "derandomize" PAC-Bayes bounds produce vacuous bounds.

Explanation: Arguments are "too uniform", would contradict NK19.

What structure / empirical phenomenon might surrogates exploit?

Conclusion

- Good generalization is a product of favourable properties of the data, algorithm and the interaction between the two: bounds have to depend on the training samples and/or data distribution.

Conclusion

- Good generalization is a product of favourable properties of the data, algorithm and the interaction between the two: bounds have to depend on the training samples and/or data distribution.
- Empirical evaluation of the generalization bounds and measures is critical: Complexity terms are data-dependent and empirical evaluation is the only way to know how they grow, where the bounds may fail, and where we need to improve to make progress.

Conclusion

- Good generalization is a product of favourable properties of the data, algorithm and the interaction between the two: bounds have to depend on the training samples and/or data distribution.
- Empirical evaluation of the generalization bounds and measures is critical: Complexity terms are data-dependent and empirical evaluation is the only way to know how they grow, where the bounds may fail, and where we need to improve to make progress.
- Capturing data-distribution dependence by using training samples allows us to circumvent statistical and computational barriers when evaluating theories.

Conclusion

- Good generalization is a product of favourable properties of the data, algorithm and the interaction between the two: bounds have to depend on the training samples and/or data distribution.
- Empirical evaluation of the generalization bounds and measures is critical: Complexity terms are data-dependent and empirical evaluation is the only way to know how they grow, where the bounds may fail, and where we need to improve to make progress.
- Capturing data-distribution dependence by using training samples allows us to circumvent statistical and computational barriers when evaluating theories.
- Instead of bounding the generalization error uniformly over a hypothesis class containing the predictor, we should think about related predictors – surrogates, that may belong to a class where uniform convergence holds.

Conclusion

- Good generalization is a product of favourable properties of the data, algorithm and the interaction between the two: bounds have to depend on the training samples and/or data distribution.
- Empirical evaluation of the generalization bounds and measures is critical: Complexity terms are data-dependent and empirical evaluation is the only way to know how they grow, where the bounds may fail, and where we need to improve to make progress.
- Capturing data-distribution dependence by using training samples allows us to circumvent statistical and computational barriers when evaluating theories.
- Instead of bounding the generalization error uniformly over a hypothesis class containing the predictor, we should think about related predictors – surrogates, that may belong to a class where uniform convergence holds.
- We have made progress on studying some surrogates (by perturbing the classifier stepwise or at the end), but there is still a lot of work to be done.