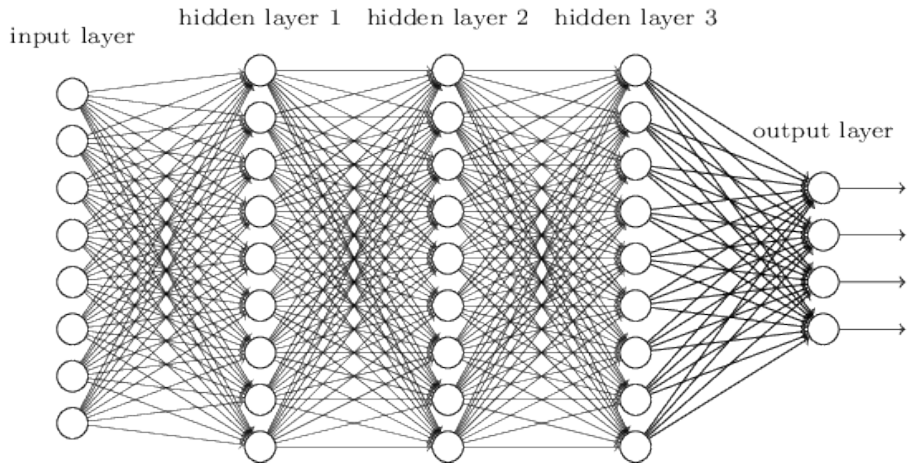
Size-Independent Sample Complexity of Neural Networks

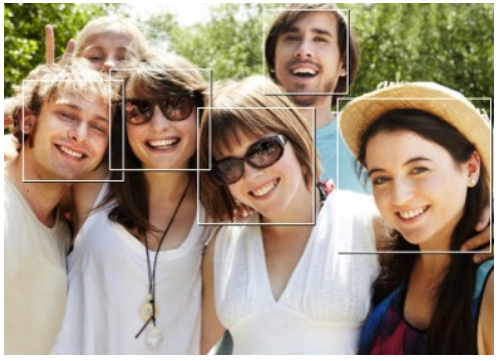
Ohad Shamir
Weizmann Institute and Microsoft Research

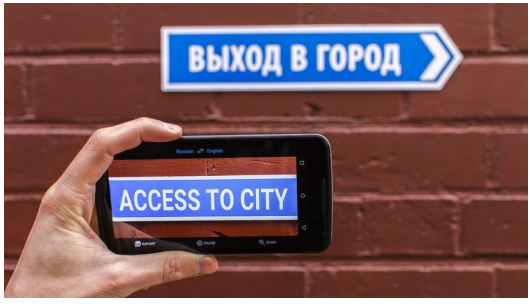
Joint work with Noah Golowich and Alexander Rakhlin

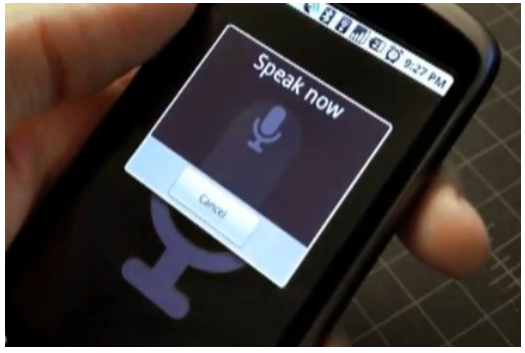
Simons Institute, June 2018

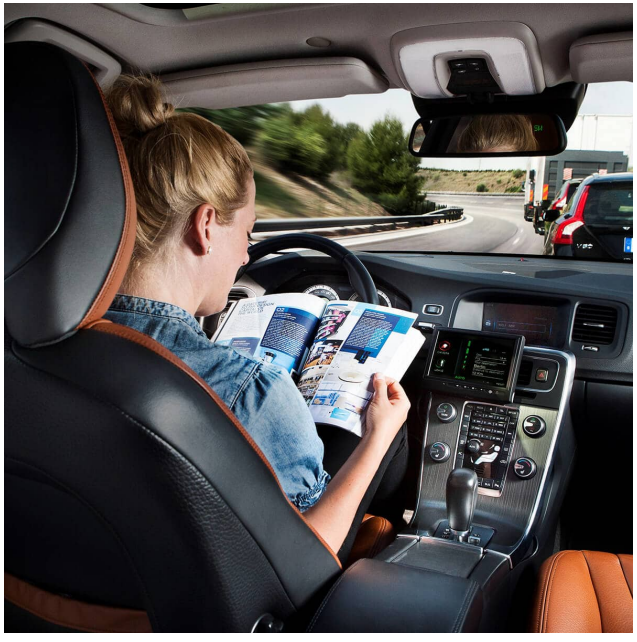


Neural Networks (a.k.a. Deep Learning)

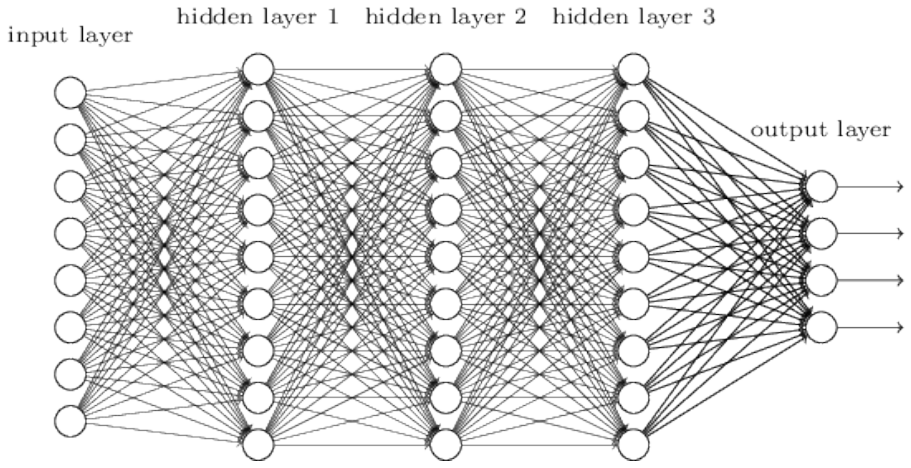








Setting



$$\mathbf{x} \mapsto W_d \sigma_d(\ldots \sigma_2(W_2 \sigma_1(W_1 \mathbf{x}))\ldots)$$

Running example: σ_i is ReLU ($z \mapsto \max\{z, 0\}$)

- Network depth (number of W_i matrices): d
- Network width (max dimension of W_i matrices): n

Inputs with $\mathcal{O}(1)$ norm; $\mathcal{O}(1)$ Lipschitz losses (over scalars or vectors)

What Makes Deep Neural Networks Generalize?

What Makes Deep Neural Networks Generalize?

- # of parameters? No, much larger than data size

What Makes Deep Neural Networks Generalize?

- # of parameters? No, much larger than data size
- Architecture? No, standard architectures can fit random labels

What Makes Deep Neural Networks Generalize?

- # of parameters? No, much larger than data size
- Architecture? No, standard architectures can fit random labels
- Dynamics of the training algorithm? Maybe, but doesn't answer the question (dynamics lead to what kind of network?)

What Makes Deep Neural Networks Generalize?

- # of parameters? No, much larger than data size
- Architecture? No, standard architectures can fit random labels
- Dynamics of the training algorithm? Maybe, but doesn't answer the question (dynamics lead to what kind of network?)

This Talk

Study how parameter norms affect generalization error
Which norms can lead to learnability, even if # of parameters is large?

Norms depend on training dynamics, and their choice can have algorithmic implications

An Aside

An Aside

Goal: Understand whether norms can control statistical complexity
not whether resulting bound is $\leq X$ for network Y on dataset Z

An Aside

Goal: Understand whether norms can control statistical complexity
not whether resulting bound is $\leq X$ for network Y on dataset Z

- If *just* want a numerically tight estimate, use a validation set
- Don't need numerical tightness to understand/motivate algorithms
  - Example: SVM generalization bounds

A Neural Network That We Understand Well

Linear Predictors:

$$\mathbf{x} \mapsto \mathbf{w}^\top \mathbf{x}$$

A Neural Network That We Understand Well

Linear Predictors:

$$\mathbf{x} \mapsto \mathbf{w}^\top \mathbf{x}$$

Generalize if the Euclidean norm of $\mathbf{w}$ is small (independent of # of parameters!)

$$\sup_{\mathbf{w}:\|\mathbf{w}\|\leq B} \left|\mathrm{Err}(\mathbf{w}) - \widehat{\mathrm{Err}}(\mathbf{w})\right| \leq \mathcal{O}\left(\sqrt{\frac{B^2}{m}}\right)$$

- Explicit regularization
- Implicit regularization: Training with gradient descent encourages small-norm solution

What is the analogous norm for neural networks?
Can we get such norm-based (and size-independent) bounds?

Existing Results

All have strong dependence on network size (width/depth)

Existing Results

All have strong dependence on network size (width/depth)
Generalization error given m examples, ignoring logs:

Existing Results

All have strong dependence on network size (width/depth)
Generalization error given m examples, ignoring logs:

VC bounds

$$\sqrt{\frac{n^2 d^2}{m}}$$

Existing Results

All have strong dependence on network size (width/depth)
Generalization error given m examples, ignoring logs:

VC bounds

$$\sqrt{\frac{n^2 d^2}{m}}$$

Lipschitz bounds

$$\frac{\prod_{j=1}^{d} \|W_j\|_{op}}{m^{1/n}}$$

Existing Results

All have strong dependence on network size (width/depth)
Generalization error given m examples, ignoring logs:

VC bounds

$$\sqrt{\frac{n^2 d^2}{m}}$$

Lipschitz bounds

$$\frac{\prod_{j=1}^{d} \|W_j\|_{op}}{m^{1/n}}$$

Scale-sensitive bounds
(e.g. [NS15])

$$2^d \sqrt{\frac{\prod_{j=1}^{d} \|W_j\|_F^2}{m}}$$

Existing Results

All have strong dependence on network size (width/depth)
Generalization error given m examples, ignoring logs:

VC bounds

$$\sqrt{\frac{n^2 d^2}{m}}$$

Lipschitz bounds

$$\frac{\prod_{j=1}^{d} \|W_j\|_{op}}{m^{1/n}}$$

Scale-sensitive bounds
(e.g. [NS15])

$$2^d \sqrt{\frac{\prod_{j=1}^{d} \|W_j\|_F^2}{m}}$$

$$\sqrt{\frac{d^2 n \left(\prod_{j=1}^{d} \|W_j\|_{op}^2\right) \sum_{j=1}^{d} \frac{\|W_j\|_F^2}{\|W_j\|^2}}{m}}$$

[NS17]

$$\sqrt{\frac{\left(\prod_{j=1}^{d} \|W_j\|_{op}^2\right) \left(\sum_{j=1}^{d} \left(\frac{\|W_j\|_{2,1}}{\|W_j\|_{op}}\right)^{2/3}\right)^3}{m}}$$

[BFT17]

Is Depth Dependence Inevitable?

Is Depth Dependence Inevitable?

Consider norm-bounded linear predictors composed with ReLU:

$$\left\{ \mathbf{x} \mapsto \sigma(\mathbf{w}^\top \mathbf{x}) \ : \ \|\mathbf{w}\| \leq B \right\} \ .$$

Uniform convergence bound: $\sqrt{B^2/m}$

Is Depth Dependence Inevitable?

Consider norm-bounded linear predictors composed with ReLU:

$$\left\{ \mathbf{x} \mapsto \sigma(\mathbf{w}^\top \mathbf{x}) \ : \ \|\mathbf{w}\| \leq B \right\} \ .$$

Uniform convergence bound: $\sqrt{B^2/m}$

Observation

Class equivalent to class of "thin" neural networks of form

$$\left\{ \mathbf{x} \mapsto \sigma(w_d \sigma(w_{d-1} \cdots \sigma(w_2 \sigma(\mathbf{w}_1^\top \mathbf{x})) \cdots)) \ : \ \|\mathbf{w}_1\| \cdot \prod_{j=2}^{d} |w_j| \leq B \right\} .$$

Same $\sqrt{B^2/m}$ bound, independent of depth/width!

Is Depth Dependence Inevitable?

Consider norm-bounded linear predictors composed with ReLU:

$$\left\{ \mathbf{x} \mapsto \sigma(\mathbf{w}^\top \mathbf{x}) \ : \ \|\mathbf{w}\| \leq B \right\} \ .$$

Uniform convergence bound: $\sqrt{B^2/m}$

Observation

Class equivalent to class of "thin" neural networks of form

$$\left\{ \mathbf{x} \mapsto \sigma(w_d \sigma(w_{d-1} \cdots \sigma(w_2 \sigma(\mathbf{w}_1^\top \mathbf{x})) \cdots)) \ : \ \|\mathbf{w}_1\| \cdot \prod_{j=2}^{d} |w_j| \leq B \right\} .$$

Same $\sqrt{B^2/m}$ bound, independent of depth/width!

- But: Existing analyses lead to size dependence.
  A "satisfying" analysis should do better

General Networks

- Scalars w_j are now parameter matrices W_j

General Networks

- Scalars w_j are now parameter matrices W_j
- Plausible approach: Since error was controlled by $\prod_{j=1}^{d} |w_j|$ for the "thin" networks, try to control $\prod_{j=1}^{d} \|W_j\|$ for some matrix norm $\|\cdot\|$. But which norm?

General Networks

- Scalars w_j are now parameter matrices W_j
- Plausible approach: Since error was controlled by $\prod_{j=1}^{d} |w_j|$ for the "thin" networks, try to control $\prod_{j=1}^{d} \|W_j\|$ for some matrix norm $\|\cdot\|$. But which norm?
- First attempt: Spectral norm

Theorem

Generalization error over neural networks s.t. $\prod_{j=1}^{d} \|W_j\|_{op} \leq B$ *can be at least* $\Omega\left(\sqrt{B^2 n/m}\right)$.

General Networks

- Scalars w_j are now parameter matrices W_j
- Plausible approach: Since error was controlled by $\prod_{j=1}^{d} |w_j|$ for the "thin" networks, try to control $\prod_{j=1}^{d} \|W_j\|$ for some matrix norm $\|\cdot\|$. But which norm?
- First attempt: Spectral norm

Theorem

Generalization error over neural networks s.t. $\prod_{j=1}^{d} \|W_j\|_{op} \leq B$ *can be at least* $\Omega\left(\sqrt{B^2 n/m}\right)$.

More generally, $\Omega\left(\sqrt{B^2 n^{\max\left\{0, \frac{1}{2}-\frac{1}{p}\right\}}/m}\right)$ *with any p-Schatten norm (p-norm of singular values)*

General Networks

- Scalars w_j are now parameter matrices W_j
- Plausible approach: Since error was controlled by $\prod_{j=1}^{d} |w_j|$ for the "thin" networks, try to control $\prod_{j=1}^{d} \|W_j\|$ for some matrix norm $\|\cdot\|$. But which norm?
- First attempt: Spectral norm

Theorem

Generalization error over neural networks s.t. $\prod_{j=1}^{d} \|W_j\|_{op} \leq B$ *can be at least* $\Omega\left(\sqrt{B^2 n/m}\right)$.

More generally, $\Omega\left(\sqrt{B^2 n^{\max\left\{0, \frac{1}{2}-\frac{1}{p}\right\}}/m}\right)$ *with any p-Schatten norm (p-norm of singular values)*

Proof idea (spectral norm): $\{\mathbf{x} \mapsto \mathrm{diag}(\mathbf{w})\mathbf{x} : \|\mathbf{w}\|_\infty \leq B\}$

General Networks

What about $p = 2$ (Frobenius norm)?

General Networks

What about $p = 2$ (Frobenius norm)?

Theorem

If $\prod_{j=1}^{d} \|W_j\|_F \leq B$, *generalization error is*

$$\tilde{\mathcal{O}}\left(\min\left\{\frac{B}{m^{1/4}}, \sqrt{\frac{dB^2}{m}}\right\}\right).$$

- $\leq B/m^{1/4}$: Independent of network depth or width
- $\leq \sqrt{dB^2/m}$ (compared to $2^d\sqrt{B^2/m}$ in [NS15])

Our Work

Technical Contributions

- An algebraic trick to avoid exponential depth dependencies
- Generic technique to eliminate depth dependencies, if product-of-p-Schatten-norms is controlled (for any $p < \infty$)
- New lower bound

Applications

- Improving/removing depth dependencies in previous bounds in the literature
- First explicit, fully size-independent generalization error bounds for standard neural networks, using Schatten norms

Rademacher Complexity

Given a function class $\mathcal{H}$ (e.g. neural networks with certain norm constraints) and inputs $\mathbf{x}_1, \ldots, \mathbf{x}_m$

$$\hat{\mathcal{R}}_m(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i h(\mathbf{x}_i) \right] \quad , \quad \boldsymbol{\epsilon} \sim \mathcal{U}(\{-1, +1\}^m)$$

Convertible to uniform generalization error bound of $\mathcal{H}$ w.r.t. training set of size m

$$\sup_{h \in \mathcal{H}} \left| \frac{1}{m} \sum_{i=1}^{m} \ell(h(\mathbf{x}_i), y_i) - \mathbb{E}_{\mathbf{x},y} \left[\ell(h(\mathbf{x}), y) \right] \right|$$

Analyzing Neural Networks with Rademacher Complexity

Analyzing Neural Networks with Rademacher Complexity

Let

$$\mathcal{H}_k = \{\mathbf{x} \mapsto W_k\sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

$$\mathcal{H}'_k = \{\mathbf{x} \mapsto \sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

Bounding via Peeling

Analyzing Neural Networks with Rademacher Complexity

Let

$$\mathcal{H}_k = \{\mathbf{x} \mapsto W_k\sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

$$\mathcal{H}'_k = \{\mathbf{x} \mapsto \sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

Bounding via Peeling

$$\hat{\mathcal{R}}_m(\mathcal{H}_d)$$

Analyzing Neural Networks with Rademacher Complexity

Let

$$\mathcal{H}_k = \{\mathbf{x} \mapsto W_k\sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

$$\mathcal{H}'_k = \{\mathbf{x} \mapsto \sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

Bounding via Peeling

$$\hat{\mathcal{R}}_m(\mathcal{H}_d) \ \leq \ C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}'_d)$$

Analyzing Neural Networks with Rademacher Complexity

Let

$$\mathcal{H}_k = \{\mathbf{x} \mapsto W_k\sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

$$\mathcal{H}'_k = \{\mathbf{x} \mapsto \sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

Bounding via Peeling

$$\begin{aligned}\hat{\mathcal{R}}_m(\mathcal{H}_d) &\leq C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}'_d) \\ &\overset{*}{\leq} 2C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}_{d-1})\end{aligned}$$

(*) Due to the contraction lemma. Tight!

Analyzing Neural Networks with Rademacher Complexity

Let

$$\mathcal{H}_k = \{\mathbf{x} \mapsto W_k\sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

$$\mathcal{H}'_k = \{\mathbf{x} \mapsto \sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \ : \ \forall i \ W_i \in \mathcal{W}_i\}$$

Bounding via Peeling

$$\begin{aligned}\hat{\mathcal{R}}_m(\mathcal{H}_d) &\leq C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}'_d) \\ &\overset{*}{\leq} 2C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}_{d-1}) \\ &\leq 2C_{\mathcal{W}_k}C_{\mathcal{W}_{d-1}}\hat{\mathcal{R}}_m(\mathcal{H}'_{d-1})\end{aligned}$$

(*) Due to the contraction lemma. Tight!

Analyzing Neural Networks with Rademacher Complexity

Let

$$\mathcal{H}_k = \{\mathbf{x} \mapsto W_k\sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \;:\; \forall i\; W_i \in \mathcal{W}_i\}$$

$$\mathcal{H}'_k = \{\mathbf{x} \mapsto \sigma(W_{k-1}\dots W_2\sigma(W_1\mathbf{x})\dots) \;:\; \forall i\; W_i \in \mathcal{W}_i\}$$

Bounding via Peeling

$$\begin{aligned}
\hat{\mathcal{R}}_m(\mathcal{H}_d) &\leq C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}'_d) \\
&\overset{*}{\leq} 2C_{\mathcal{W}_d}\hat{\mathcal{R}}_m(\mathcal{H}_{d-1}) \\
&\leq 2C_{\mathcal{W}_k}C_{\mathcal{W}_{d-1}}\hat{\mathcal{R}}_m(\mathcal{H}'_{d-1}) \\
&\cdots \leq 2^d\left(\prod_{i=1}^{d} C_{\mathcal{W}_i}\right)\cdot\sqrt{\frac{1}{m}}
\end{aligned}$$

(*) Due to the contraction lemma. Tight!

A Log-sum-exp Trick

Introducing a parameter $\lambda > 0$,

$$m \cdot \hat{\mathcal{R}}_m(\mathcal{H}_d) = \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}_d} \sum_i \varepsilon_i h(\mathbf{x}_i) \right]$$

A Log-sum-exp Trick

Introducing a parameter $\lambda > 0$,

$$
\begin{aligned}
m \cdot \hat{\mathcal{R}}_m(\mathcal{H}_d) &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}_d} \sum_i \varepsilon_i h(\mathbf{x}_i) \right] \\
&= \frac{1}{\lambda} \log \exp \left(\lambda \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \sum_i \epsilon_i h(\mathbf{x}_i) \right)
\end{aligned}
$$

A Log-sum-exp Trick

Introducing a parameter $\lambda > 0$,

$$
\begin{aligned}
m \cdot \hat{\mathcal{R}}_m(\mathcal{H}_d) &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}_d} \sum_i \varepsilon_i h(\mathbf{x}_i) \right] \\
&= \frac{1}{\lambda} \log \exp \left(\lambda \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \sum_i \epsilon_i h(\mathbf{x}_i) \right) \\
&\leq \frac{1}{\lambda} \log \left(\mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \exp \left(\lambda \sum_i \epsilon_i h(\mathbf{x}_i) \right) \right)
\end{aligned}
$$

A Log-sum-exp Trick

Introducing a parameter $\lambda > 0$,

$$
\begin{aligned}
m \cdot \hat{\mathcal{R}}_m(\mathcal{H}_d) &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}_d} \sum_i \varepsilon_i h(\mathbf{x}_i) \right] \\
&= \frac{1}{\lambda} \log \exp \left(\lambda \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \sum_i \epsilon_i h(\mathbf{x}_i) \right) \\
&\leq \frac{1}{\lambda} \log \left(\mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \exp \left(\lambda \sum_i \epsilon_i h(\mathbf{x}_i) \right) \right)
\end{aligned}
$$

Using a contraction lemma variant and peeling as before:

$$
\frac{1}{\lambda} \log \left(2^d \mathbb{E}_{\epsilon} \exp \left(\lambda \prod_i C_{\mathcal{W}_i} \cdot f(\mathbf{x}_1, \ldots, \mathbf{x}_m) \right) \right)
$$

A Log-sum-exp Trick

Introducing a parameter $\lambda > 0$,

$$
\begin{aligned}
m \cdot \hat{\mathcal{R}}_m(\mathcal{H}_d) &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{h \in \mathcal{H}_d} \sum_i \varepsilon_i h(\mathbf{x}_i) \right] \\
&= \frac{1}{\lambda} \log \exp \left(\lambda \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \sum_i \epsilon_i h(\mathbf{x}_i) \right) \\
&\leq \frac{1}{\lambda} \log \left(\mathbb{E}_{\boldsymbol{\epsilon}} \sup_{h \in \mathcal{H}_d} \exp \left(\lambda \sum_i \epsilon_i h(\mathbf{x}_i) \right) \right)
\end{aligned}
$$

Using a contraction lemma variant and peeling as before:

$$
\frac{1}{\lambda} \log \left(2^d \mathbb{E}_{\epsilon} \exp \left(\lambda \prod_i C_{\mathcal{W}_i} \cdot f(\mathbf{x}_1, \ldots, \mathbf{x}_m) \right) \right)
$$

Tuning λ and simplifying: $\sqrt{d} \left(\prod_i C_{\mathcal{W}_i} \right) \sqrt{m}$.

Applications

Theorem

If $\prod_{j=1}^{d} \|W_j\|_F \leq B$, *generalization error is*

$$\mathcal{O}\left(\sqrt{\frac{dB^2}{m}}\right)$$

Theorem

If $\prod_{j=1}^{d} \|W_j\|_{1,\infty} \leq B$, *generalization error is*

$$\mathcal{O}\left(\sqrt{\frac{(d+\log(n)) \cdot B^2}{m}}\right)$$

Depth Independence

Depth Independence

- Consider a network $\mathfrak{N}$ s.t. $\prod_{j=1}^{d} \|W_j\|_p \leq B$, where $\|\cdot\|_p$ is a p-Schatten norm

Depth Independence

- Consider a network $\mathfrak{N}$ s.t. $\prod_{j=1}^{d} \|W_j\|_p \leq B$, where $\|\cdot\|_p$ is a p-Schatten norm
- Suppose $\mathfrak{N}$ is nontrivial: exists $\mathbf{x}$ in domain s.t. $|\mathfrak{N}(\mathbf{x})| \geq \Gamma$ for some $\Gamma = \Omega(1)$

Depth Independence

- Consider a network $\mathfrak{N}$ s.t. $\prod_{j=1}^{d} \|W_j\|_p \leq B$, where $\|\cdot\|_p$ is a p-Schatten norm
- Suppose $\mathfrak{N}$ is nontrivial: exists $\mathbf{x}$ in domain s.t. $|\mathfrak{N}(\mathbf{x})| \geq \Gamma$ for some $\Gamma = \Omega(1)$

Claim

For any r, one of the first r parameter matrices is close to rank 1:

$$\min_{j \in \{1,\ldots,r\}} \frac{\|W\|_p}{\|W\|_{op}} \leq \left(\frac{B}{\Gamma}\right)^{1/r}.$$

Depth Independence

- Consider a network $\mathfrak{N}$ s.t. $\prod_{j=1}^{d} \|W_j\|_p \leq B$, where $\|\cdot\|_p$ is a p-Schatten norm
- Suppose $\mathfrak{N}$ is nontrivial: exists $\mathbf{x}$ in domain s.t. $|\mathfrak{N}(\mathbf{x})| \geq \Gamma$ for some $\Gamma = \Omega(1)$

Claim

For any r, one of the first r parameter matrices is close to rank 1:

$$\min_{j \in \{1,\ldots,r\}} \frac{\|W\|_p}{\|W\|_{op}} \leq \left(\frac{B}{\Gamma}\right)^{1/r}.$$

Proof:

$$\frac{B}{\Gamma} \geq \frac{\prod_{j=1}^{d} \|W_j\|_p}{\prod_{j=1}^{d} \|W_j\|_{op}} \geq \prod_{j=1}^{r} \frac{\|W_j\|_p}{\|W_j\|_{op}} \geq \left(\min_{j \in \{1,\ldots,r\}} \frac{\|W_j\|_p}{\|W_j\|_{op}}\right)^{r}.$$

Implication

Implication

Can approximate network by replacing one of first r matrices by its rank-1 SVD approximation:

$$\mathbf{x} \mapsto W_d \sigma(W_{d-1} \dots W_k \sigma(\dots \sigma(W_1 \mathbf{x}) \dots) \dots)$$

$$\approx$$

$$\mathbf{x} \mapsto W_d \sigma(W_{d-1} \dots s\mathbf{u}\mathbf{v}^\top \sigma(\dots \sigma(W_1 \mathbf{x}) \dots) \dots)$$

Implication

Can approximate network by replacing one of first r matrices by its rank-1 SVD approximation:

$$\mathbf{x} \mapsto W_d \sigma(W_{d-1} \dots W_k \sigma(\dots \sigma(W_1 \mathbf{x}) \dots) \dots)$$

$$\approx$$

$$\mathbf{x} \mapsto \underbrace{W_d \sigma(W_{d-1} \dots s\mathbf{u})}_{\text{Univariate Lipschitz func.}} \underbrace{\mathbf{v}^\top \sigma(\dots \sigma(W_1 \mathbf{x}) \dots) \dots)}_{\text{Depth} \leq r \text{ network}}$$

Implication

Can approximate network by replacing one of first r matrices by its rank-1 SVD approximation:

$$\mathbf{x} \mapsto W_d \sigma(W_{d-1} \dots W_k \sigma(\dots \sigma(W_1 \mathbf{x}) \dots) \dots)$$

$$\approx$$

$$\mathbf{x} \mapsto \underbrace{W_d \sigma(W_{d-1} \dots s\mathbf{u})}_{\text{Univariate Lipschitz func.}} \underbrace{\mathbf{v}^\top \sigma(\dots \sigma(W_1 \mathbf{x}) \dots) \dots)}_{\text{Depth} \leq r \text{ network}}$$

- Networks with bounded product-of-Schatten-norms $\approx$ Networks of depth $\leq r$ composed with univariate Lipschitz functions
  - Original depth no longer appears explicitly!
- r is tunable: Trades-off approximation and statistical complexity

A Recipe

Given a Rademacher complexity bound for depth-d network,
Can get a depth-independent bound by combining

A Recipe

Given a Rademacher complexity bound for depth-d network,
Can get a depth-independent bound by combining

1. Corresponding bound for depth r ($\ll d$) networks...

A Recipe

Given a Rademacher complexity bound for depth-d network,
Can get a depth-independent bound by combining

1. Corresponding bound for depth r ($\ll d$) networks...
2. composed with univariate Lipschitz functions...

A Recipe

Given a Rademacher complexity bound for depth-d network,
Can get a depth-independent bound by combining

1. Corresponding bound for depth r ($\ll d$) networks...
2. composed with univariate Lipschitz functions...

Theorem

If $\mathcal{H}$ is a class with Rademacher complexity case $\hat{R}_m(\mathcal{H})$, and $\mathcal{F}_L$ is class of univariate L-Lipschitz functions,

$$\hat{R}_m(\mathcal{F}_L \circ \mathcal{H}) \leq \mathcal{O}\left(\frac{1}{\sqrt{m}} + \log^{3/2}(m) \cdot \hat{R}_m(\mathcal{H})\right) \cdot L$$

Proof via covering numbers

A Recipe

Given a Rademacher complexity bound for depth-d network,
Can get a depth-independent bound by combining

1. Corresponding bound for depth r ($\ll d$) networks...
2. composed with univariate Lipschitz functions...
3. $+$ error in replacing layer r by rank-1 approximation,

A Recipe

Given a Rademacher complexity bound for depth-d network,
Can get a depth-independent bound by combining

1. Corresponding bound for depth r ($\ll d$) networks...
2. composed with univariate Lipschitz functions...
3. $+$ error in replacing layer r by rank-1 approximation,

and tuning r

Applications

Applications

Theorem

If $\prod_{j=1}^{d} \|W_j\|_F \leq B$, *generalization error is*

$$\tilde{\mathcal{O}}\left(B \cdot \min\left\{\frac{\log(B/\Gamma)}{m^{1/4}}, \sqrt{\frac{d}{m}}\right\}\right)$$

Applications

Theorem

If $\prod_{j=1}^{d} \|W_j\|_F \leq B$, *generalization error is*

$$\tilde{\mathcal{O}}\left(B \cdot \min\left\{\frac{\log(B/\Gamma)}{m^{1/4}}, \sqrt{\frac{d}{m}}\right\}\right)$$

Theorem (Depth-Independent Version of BFT17)

If $\prod_{j=1}^{d} \|W_j\|_{op} \leq B$, $\prod_{j=1}^{d} \|W_j\|_p \leq B_p$ *and* $\max_j \frac{\|W_j\|_{2,1}}{\|W_j\|} \leq L$, *generalization error is*

$$\tilde{\mathcal{O}}\left(BL \cdot \min\left\{\frac{(\log(B_p/\Gamma))^{\frac{1}{\frac{3}{2}+p}}}{m^{\frac{1}{2+3p}}}, \sqrt{\frac{d^3}{m}}\right\}\right)$$

Summary and Further Open Questions

- First explicit size-independent generalization bounds for standard neural networks
- Techniques can be used to get depth-improved versions of existing bounds; applicable to any "deep" function class
- Results use Schatten norms (spectral norm is not enough)

Summary and Further Open Questions

- First explicit size-independent generalization bounds for standard neural networks
- Techniques can be used to get depth-improved versions of existing bounds; applicable to any "deep" function class
- Results use Schatten norms (spectral norm is not enough)

- Size-independence comes at cost of rates worse than $1/\sqrt{m}$. Can be avoided?
- Is dependence on $\prod_{j=1}^{d} \|W_j\|$ inevitable?
  - In worst-case, yes
  - Still, seems too conservative
- Algorithmic implications?
  - Standard gradient descent does not seem aligned with product-of-norms inductive bias