Some Statistical Properties of Deep Networks

Peter Bartlett

UC Berkeley

August 2, 2018

Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g., $$h_i : x \mapsto \sigma(W_i x)$$
$$\sigma(v)_i = \frac{1}{1 + \exp(-v_i)},$$

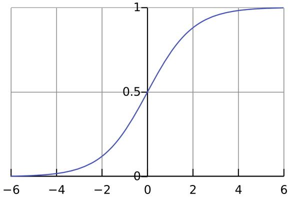
Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g., $$h_i : x \mapsto \sigma(W_i x)$$

$$\sigma(v)_i = \frac{1}{1 + \exp(-v_i)},$$



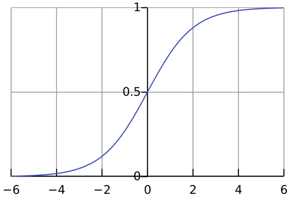
Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g., $h_i : x \mapsto \sigma(W_i x)$ $\qquad$ $h_i : x \mapsto r(W_i x)$

$$\sigma(v)_i = \frac{1}{1 + \exp(-v_i)}, \qquad r(v)_i = \max\{0, v_i\}$$



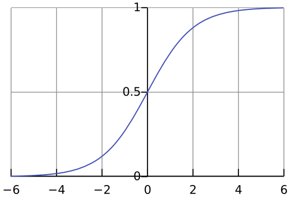
Deep Networks

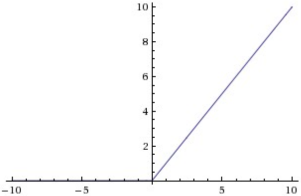
Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g., $\quad h_i : x \mapsto \sigma(W_i x) \qquad\qquad h_i : x \mapsto r(W_i x)$

$$\sigma(v)_i = \frac{1}{1+\exp(-v_i)}, \qquad\qquad r(v)_i = \max\{0, v_i\}$$





Deep Networks

Representation learning

Rich non-parametric family

Deep Networks

Representation learning
: Depth provides an effective way of representing useful features.

Rich non-parametric family

Deep Networks

Representation learning

Depth provides an effective way of representing useful features.

Rich non-parametric family

Depth provides parsimonious represention s.

Nonlinear parameterizations provide better rates of approximation. (Birman & Solomjak, 1967), (DeVore et al, 1991)

Deep Networks

Representation learning

Depth provides an effective way of representing useful features.

Rich non-parametric family

Depth provides parsimonious representions.

Nonlinear parameterizations provide better rates of approximation. (Birman & Solomjak, 1967), (DeVore et al, 1991)

Some functions require much more complexity for a shallow representation. (Telgarsky, 2015), (Eldan & Shamir, 2015)

Deep Networks

Representation learning

Depth provides an effective way of representing useful features.

Rich non-parametric family

Depth provides parsimonious represention.

Nonlinear parameterizations provide better rates of approximation. (Birman & Solomjak, 1967), (DeVore et al, 1991)

Some functions require much more complexity for a shallow representation. (Telgarsky, 2015), (Eldan & Shamir, 2015)

- Statistical complexity?

Outline

- VC theory: Number of parameters
- Margins analysis: Size of parameters
- Understanding generalization failures

VC Theory

VC Theory

- Assume network maps to $\{-1, 1\}$.
  (Threshold its output)

VC Theory

- Assume network maps to $\{-1, 1\}$.
  (Threshold its output)
- Data generated by a probability distribution P on $\mathcal{X} \times \{-1, 1\}$.

VC Theory

- Assume network maps to $\{-1, 1\}$.
  (Threshold its output)
- Data generated by a probability distribution P on $\mathcal{X} \times \{-1, 1\}$.
- Want to choose a function f such that $P(f(x) \neq y)$ is small (near optimal).

VC Theory

Theorem (Vapnik and Chervonenkis)

Suppose $\mathcal{F} \subseteq \{-1, 1\}^{\mathcal{X}}$.
For every prob distribution P on $\mathcal{X} \times \{-1, 1\}$,
with probability $1 - \delta$ over n iid examples $(x_1, y_1), \ldots, (x_n, y_n)$,
every f in $\mathcal{F}$ satisfies

$$P(f(x) \neq y) \leq \frac{1}{n} |\{i : f(x_i) \neq y_i\}| + \left(\frac{c}{n} \left(\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)\right)\right)^{1/2}.$$

VC Theory

Theorem (Vapnik and Chervonenkis)

Suppose $\mathcal{F} \subseteq \{-1, 1\}^{\mathcal{X}}$.
For every prob distribution P on $\mathcal{X} \times \{-1, 1\}$,
with probability $1 - \delta$ over n iid examples $(x_1, y_1), \ldots, (x_n, y_n)$,
every f in $\mathcal{F}$ satisfies

$$P(f(x) \neq y) \leq \frac{1}{n} |\{i : f(x_i) \neq y_i\}| + \left(\frac{c}{n} (\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta))\right)^{1/2}.$$

- For uniform bounds (that is, for all distributions and all $f \in \mathcal{F}$, proportions are close to probabilities), this inequality is tight—within a constant factor.

VC Theory

Theorem (Vapnik and Chervonenkis)

Suppose $\mathcal{F} \subseteq \{-1,1\}^{\mathcal{X}}$.
For every prob distribution P on $\mathcal{X} \times \{-1,1\}$,
with probability $1-\delta$ over n iid examples $(x_1,y_1),\ldots,(x_n,y_n)$,
every f in $\mathcal{F}$ satisfies

$$P(f(x) \neq y) \leq \frac{1}{n}\left|\{i : f(x_i) \neq y_i\}\right| + \left(\frac{c}{n}\left(\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)\right)\right)^{1/2}.$$

- For uniform bounds (that is, for all distributions and all $f \in \mathcal{F}$, proportions are close to probabilities), this inequality is tight—within a constant factor.
- For neural networks, VC-dimension:

VC Theory

Theorem (Vapnik and Chervonenkis)

Suppose $\mathcal{F} \subseteq \{-1, 1\}^{\mathcal{X}}$.
For every prob distribution P on $\mathcal{X} \times \{-1, 1\}$,
with probability $1 - \delta$ over n iid examples $(x_1, y_1), \ldots, (x_n, y_n)$,
every f in $\mathcal{F}$ satisfies

$$P(f(x) \neq y) \leq \frac{1}{n} |\{i : f(x_i) \neq y_i\}| + \left(\frac{c}{n} \left(\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)\right)\right)^{1/2}.$$

- For uniform bounds (that is, for all distributions and all $f \in \mathcal{F}$, proportions are close to probabilities), this inequality is tight—within a constant factor.
- For neural networks, VC-dimension:
  - increases with number of parameters

VC Theory

Theorem (Vapnik and Chervonenkis)

Suppose $\mathcal{F} \subseteq \{-1, 1\}^{\mathcal{X}}$.
For every prob distribution P on $\mathcal{X} \times \{-1, 1\}$,
with probability $1 - \delta$ over n iid examples $(x_1, y_1), \ldots, (x_n, y_n)$,
every f in $\mathcal{F}$ satisfies

$$P(f(x) \neq y) \leq \frac{1}{n} \left|\{i : f(x_i) \neq y_i\}\right| + \left(\frac{c}{n} \left(\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)\right)\right)^{1/2}.$$

- For uniform bounds (that is, for all distributions and all $f \in \mathcal{F}$, proportions are close to probabilities), this inequality is tight—within a constant factor.
- For neural networks, VC-dimension:
  - increases with number of parameters
  - depends on nonlinearity and depth

VC-Dimension of Neural Networks

Theorem

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

VC-Dimension of Neural Networks

Theorem

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}\left(p\right)$.

(Baum and Haussler, 1989)

VC-Dimension of Neural Networks

Theorem

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(p)$.

   (Baum and Haussler, 1989)

2. Piecewise linear (ReLUs): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(pL)$.

   (B., Harvey, Liaw, Mehrabian, 2017)

VC-Dimension of Neural Networks

Theorem

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(p)$.

   (Baum and Haussler, 1989)

2. Piecewise linear (ReLUs): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(pL)$.

   (B., Harvey, Liaw, Mehrabian, 2017)

3. Piecewise polynomial: $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(pL^2)$.

   (B., Maiorov, Meir, 1998)

VC-Dimension of Neural Networks

Theorem

Consider the class $\mathcal{F}$ of $\{-1,1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(p)$.
   (Baum and Haussler, 1989)
2. Piecewise linear (ReLUs): $\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(pL)$.
   (B., Harvey, Liaw, Mehrabian, 2017)
3. Piecewise polynomial: $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(pL^2\right)$.
   (B., Maiorov, Meir, 1998)
4. Sigmoid: $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(p^2k^2\right)$.
   (Karpinsky and MacIntyre, 1994)

Outline

- VC theory: Number of parameters
- **Margins analysis: Size of parameters**
- Understanding generalization failures

Spectrally-normalized margin bounds for neural networks.
B., Dylan J. Foster, Matus Telgarsky, NIPS 2017.
arXiv:1706.08498

Dylan Foster
Cornell

Matus Telgarsky
UIUC

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.
- The prediction on $x \in \mathcal{X}$ is $\arg\max_y f(x)_y$.

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.
- The prediction on $x \in \mathcal{X}$ is $\arg\max_y f(x)_y$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{1, \ldots, m\}$, define the margin $M(f(x), y) = f(x)_y - \max_{i \neq y} f(x)_i$.

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.
- The prediction on $x \in \mathcal{X}$ is $\arg\max_y f(x)_y$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{1, \ldots, m\}$, define the margin $M(f(x), y) = f(x)_y - \max_{i \neq y} f(x)_i$.
- If $M(f(x), y) > 0$ then f classifies x correctly.

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.
- The prediction on $x \in \mathcal{X}$ is $\arg\max_y f(x)_y$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{1, \ldots, m\}$, define the margin $M(f(x), y) = f(x)_y - \max_{i \neq y} f(x)_i$.
- If $M(f(x), y) > 0$ then f classifies x correctly.
- We can view a larger margin as a more confident correct classification.

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.
- The prediction on $x \in \mathcal{X}$ is $\arg\max_y f(x)_y$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{1, \ldots, m\}$, define the margin $M(f(x), y) = f(x)_y - \max_{i \neq y} f(x)_i$.
- If $M(f(x), y) > 0$ then f classifies x correctly.
- We can view a larger margin as a more confident correct classification.
- Minimizing a continuous loss, such as

$$\sum_{i=1}^{n} \|f(X_i) - Y_i\|^2 ,$$

encourages large margins.

Large-Margin Classifiers

- Consider a vector-valued function $f : \mathcal{X} \to \mathbb{R}^m$ used for classification, $y \in \{1, \ldots, m\}$.
- The prediction on $x \in \mathcal{X}$ is $\arg\max_y f(x)_y$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{1, \ldots, m\}$, define the margin $M(f(x), y) = f(x)_y - \max_{i \neq y} f(x)_i$.
- If $M(f(x), y) > 0$ then f classifies x correctly.
- We can view a larger margin as a more confident correct classification.
- Minimizing a continuous loss, such as

$$\sum_{i=1}^{n} \|f(X_i) - Y_i\|^2 ,$$

  encourages large margins.
- For large-margin classifiers, we should expect the fine-grained details of f to be less important.

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measure the size of functions computed by a network of ReLUs via operator norms.

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measure the size of functions computed by a network of ReLUs via operator norms.
- Large multiclass versus binary classification.

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measure the size of functions computed by a network of ReLUs via operator norms.
- Large multiclass versus binary classification.

Definitions

- Consider operator norms: For a matrix A_i,

  $\|A_i\|_* := \sup_{\|x\| \leq 1} \|A_i x\|$.

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measure the size of functions computed by a network of ReLUs via operator norms.
- Large multiclass versus binary classification.

Definitions

- Consider operator norms: For a matrix A_i,

  $\|A_i\|_* := \sup_{\|x\|\leq 1} \|A_i x\|$.
- Recall: Multiclass margin function for $f : \mathcal{X} \to \mathbb{R}^m$, $y \in \{1, \ldots, m\}$, is

$$M(f(x), y) = f(x)_y - \max_{i \neq y} f(x)_i.$$

Generalization in Deep Networks

Theorem

With high probability, every f_A

Generalization in Deep Networks

Theorem

With high probability, every f_A

Definitions

Network with L layers, parameters $A_1, \ldots, A_L$:

$$f_A(x) := \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

Generalization in Deep Networks

Theorem

With high probability, every f_A satisfies

$$\Pr(M(f_A(X), Y) \le 0) \le$$

Definitions

Network with L layers, parameters $A_1, \ldots, A_L$:

$$f_A(x) := \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

Generalization in Deep Networks

Theorem

With high probability, every f_A satisfies

$$\Pr(M(f_A(X), Y) \le 0) \le \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \le \gamma]$$

Definitions

Network with L layers, parameters $A_1, \ldots, A_L$:

$$f_A(x) := \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

Generalization in Deep Networks

Theorem

With high probability, every f_A satisfies

$$\Pr(M(f_A(X), Y) \leq 0) \leq \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \leq \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

Definitions

Network with L layers, parameters $A_1, \ldots, A_L$:

$$f_A(x) := \sigma_L(A_L\sigma_{L-1}(A_{L-1}\cdots\sigma_1(A_1x)\cdots)).$$

Generalization in Deep Networks

Theorem

With high probability, every f_A with $R_A \leq r$ satisfies

$$\Pr(M(f_A(X), Y) \leq 0) \leq \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \leq \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

Definitions

Network with L layers, parameters $A_1, \ldots, A_L$:

$$f_A(x) := \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

Scale of f_A: $R_A := \prod_{i=1}^{L} \|A_i\|_*$.

Generalization in Deep Networks

Theorem

With high probability, every f_A with $R_A \leq r$ satisfies

$$\Pr(M(f_A(X), Y) \leq 0) \leq \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \leq \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

Definitions

Network with L layers, parameters $A_1, \ldots, A_L$:

$$f_A(x) := \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

Scale of f_A: $R_A := \prod_{i=1}^{L} \|A_i\|_* \left(\sum_{i=1}^{L} \frac{\|A_i\|_{2,1}^{2/3}}{\|A_i\|_*^{2/3}}\right)^{3/2}$.

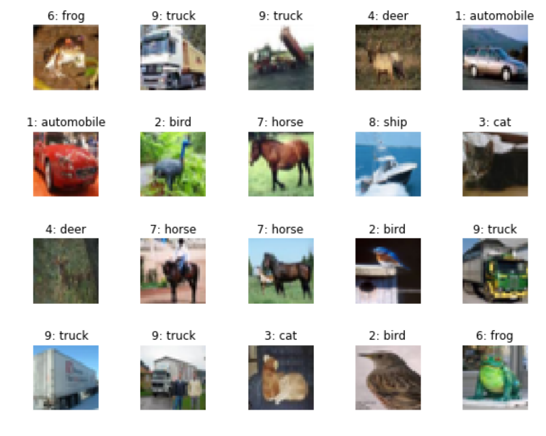
(Assume σ_i is 1-Lipschitz, inputs normalized.)

Outline

- VC theory: Number of parameters
- Margins analysis: Size of parameters
- **Understanding generalization failures**

Understanding Generalization Failures

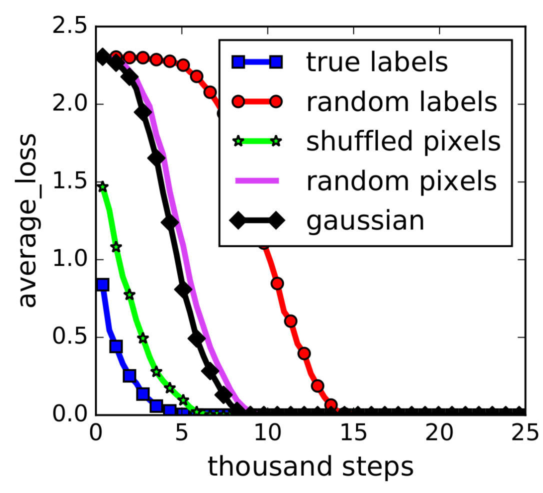
CIFAR10



http://corochann.com/

Understanding Generalization Failures

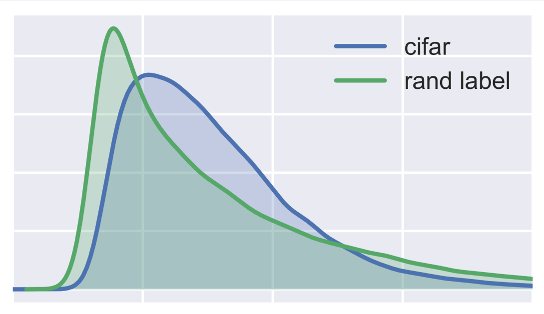
Stochastic Gradient Training Error on CIFAR10



(Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, 2017)

Understanding Generalization Failures

Training margins on CIFAR10 with true and random labels

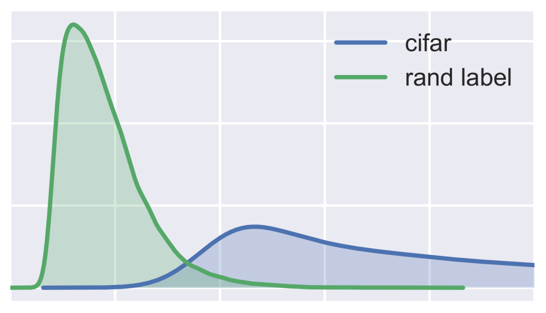


- How does this match the large margin explanation?

Understanding Generalization Failures

If we rescale the margins by R_A (the scale parameter):

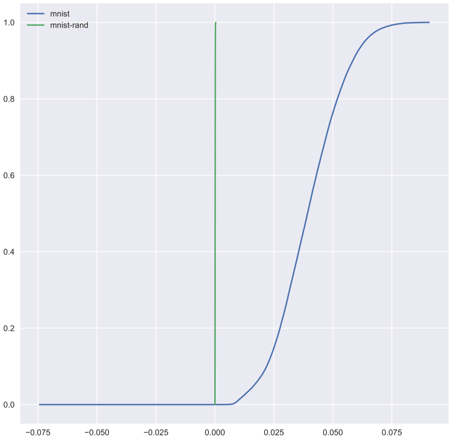
Rescaled margins on CIFAR10



Understanding Generalization Failures

If we rescale the margins by R_A (the scale parameter):

Rescaled cumulative margins on MNIST



Generalization in Deep Networks

Theorem

With high probability, every f_A with $R_A \leq r$ satisfies

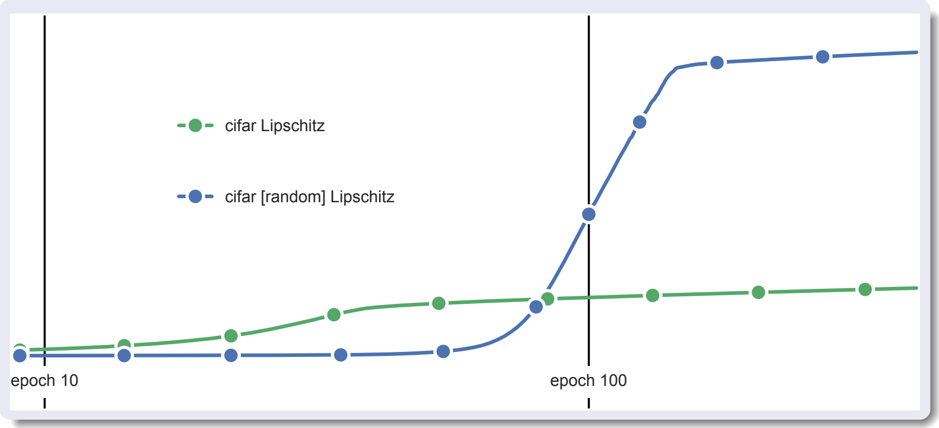
$$\Pr(M(f_A(X), Y) \leq 0) \leq \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \leq \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

Network with L layers, parameters $A_1, \ldots, A_L$:

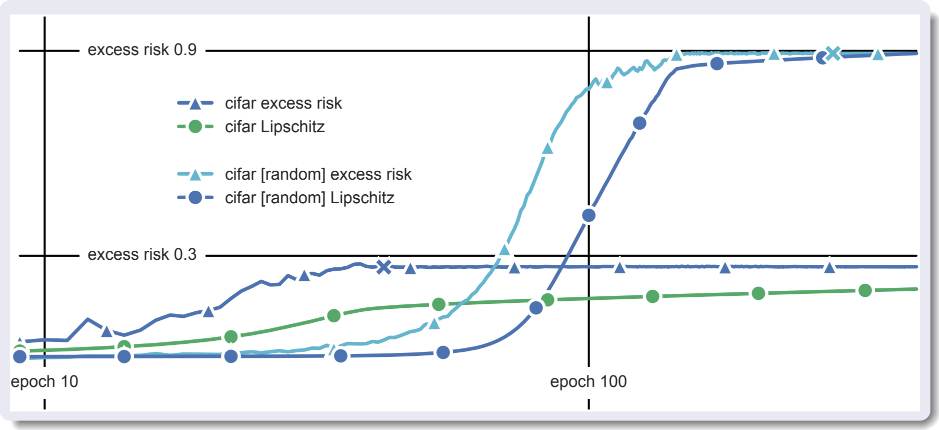
$$f_A(x) := \sigma(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

Scale of f_A: $R_A := \prod_{i=1}^{L} \|A_i\|_* \left(\sum_{i=1}^{L} \frac{\|A_i\|_{2,1}^{2/3}}{\|A_i\|_*^{2/3}}\right)^{3/2}$.

Understanding Generalization Failures



Understanding Generalization Failures



Generalization in Neural Networks

- With appropriate normalization, the margins analysis is qualitatively consistent with the generalization performance.

Generalization in Neural Networks

- With appropriate normalization, the margins analysis is qualitatively consistent with the generalization performance.
- Recent work by Golowich, Rakhlin, and Shamir give bounds with improved dependence on depth.

Generalization in Neural Networks

- With appropriate normalization, the margins analysis is qualitatively consistent with the generalization performance.
- Recent work by Golowich, Rakhlin, and Shamir give bounds with improved dependence on depth.
- Regularization and optimization: explicit control of operator norms?