Representation, Optimization and Generalization in Deep Learning

Peter Bartlett

UC Berkeley

25 January, 2018

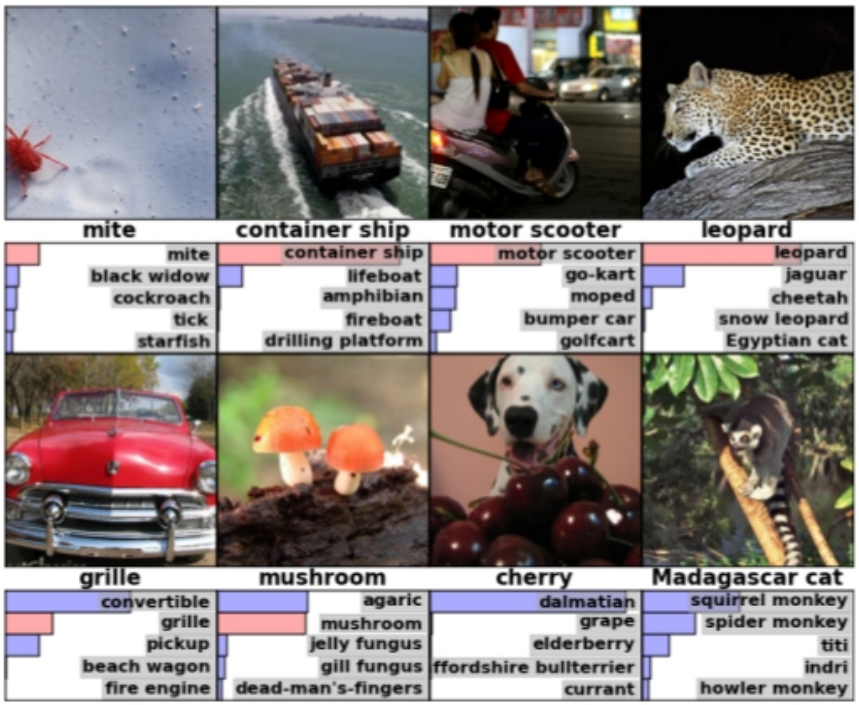
Deep neural networks

Game playing

(Jung Yeon-Je/AFP/Getty Images)

Deep neural networks

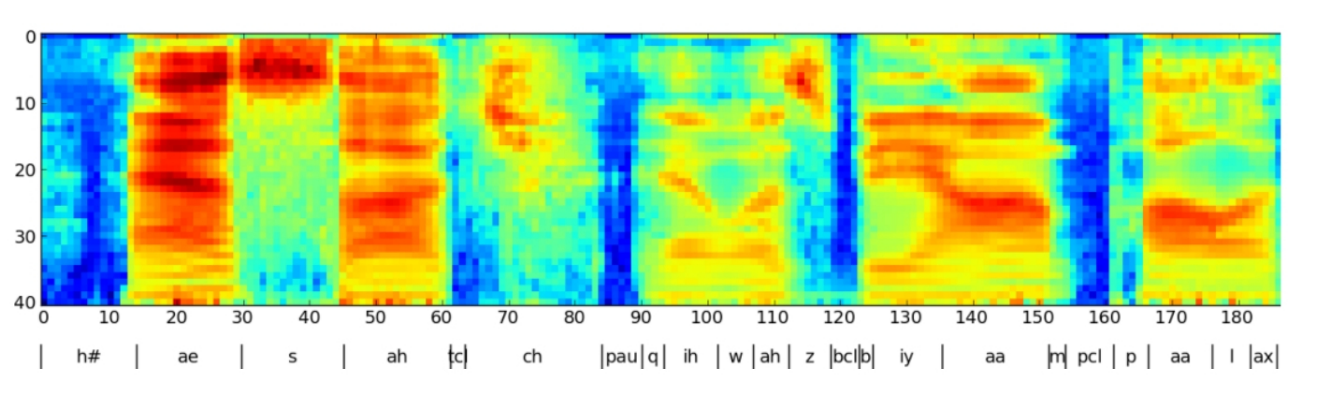
Image recognition



(Krizhevsky et al, 2012)

Deep neural networks

Speech recognition



(Graves et al, 2013)

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

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

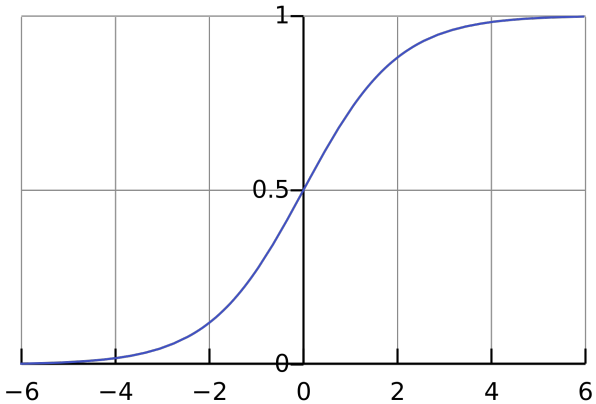
Deep Networks

Deep compositions of nonlinear functions

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

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

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



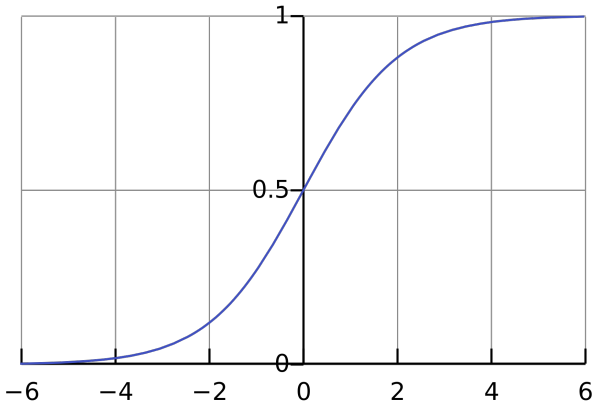
Deep Networks

Deep compositions of nonlinear functions

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

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

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



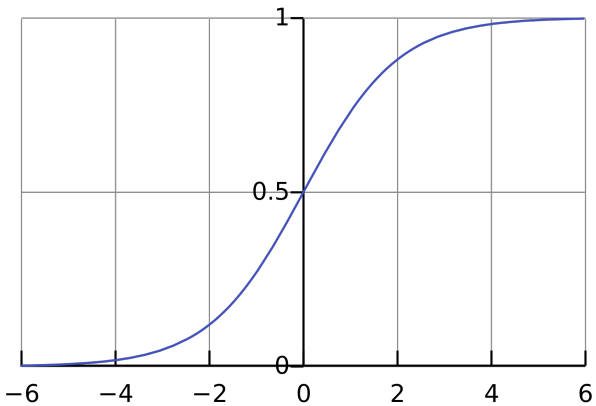
Deep Networks

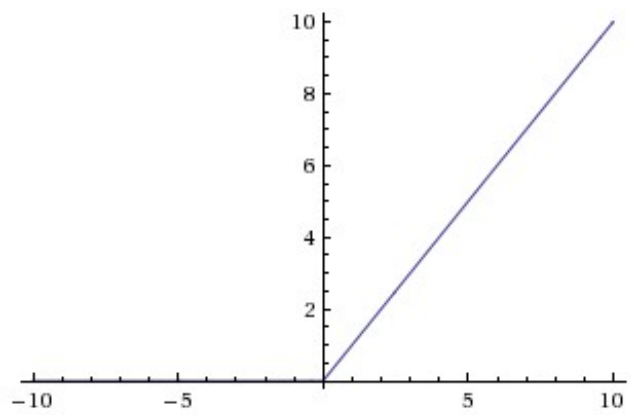
Deep compositions of nonlinear functions

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

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

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





Deep Networks

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 representions.
Nonlinear parameterizations provide better rates of approximation.

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.

Some functions require much more complexity for a shallow representation.

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.

Some functions require much more complexity for a shallow representation.

But...

- Optimization?

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.

Some functions require much more complexity for a shallow representation.

But...

- Optimization?
  - Nonlinear parameterization.

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.
Some functions require much more complexity for a shallow representation.

But...

- Optimization?
  - Nonlinear parameterization.
  - Apparently worse as the depth increases.

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.
Some functions require much more complexity for a shallow representation.

But...

- Optimization?
  - Nonlinear parameterization.
  - Apparently worse as the depth increases.
- Generalization?

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.

Some functions require much more complexity for a shallow representation.

But...

- Optimization?
  - Nonlinear parameterization.
  - Apparently worse as the depth increases.
- Generalization?
  - What determines the statistical complexity of a deep network?

Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points

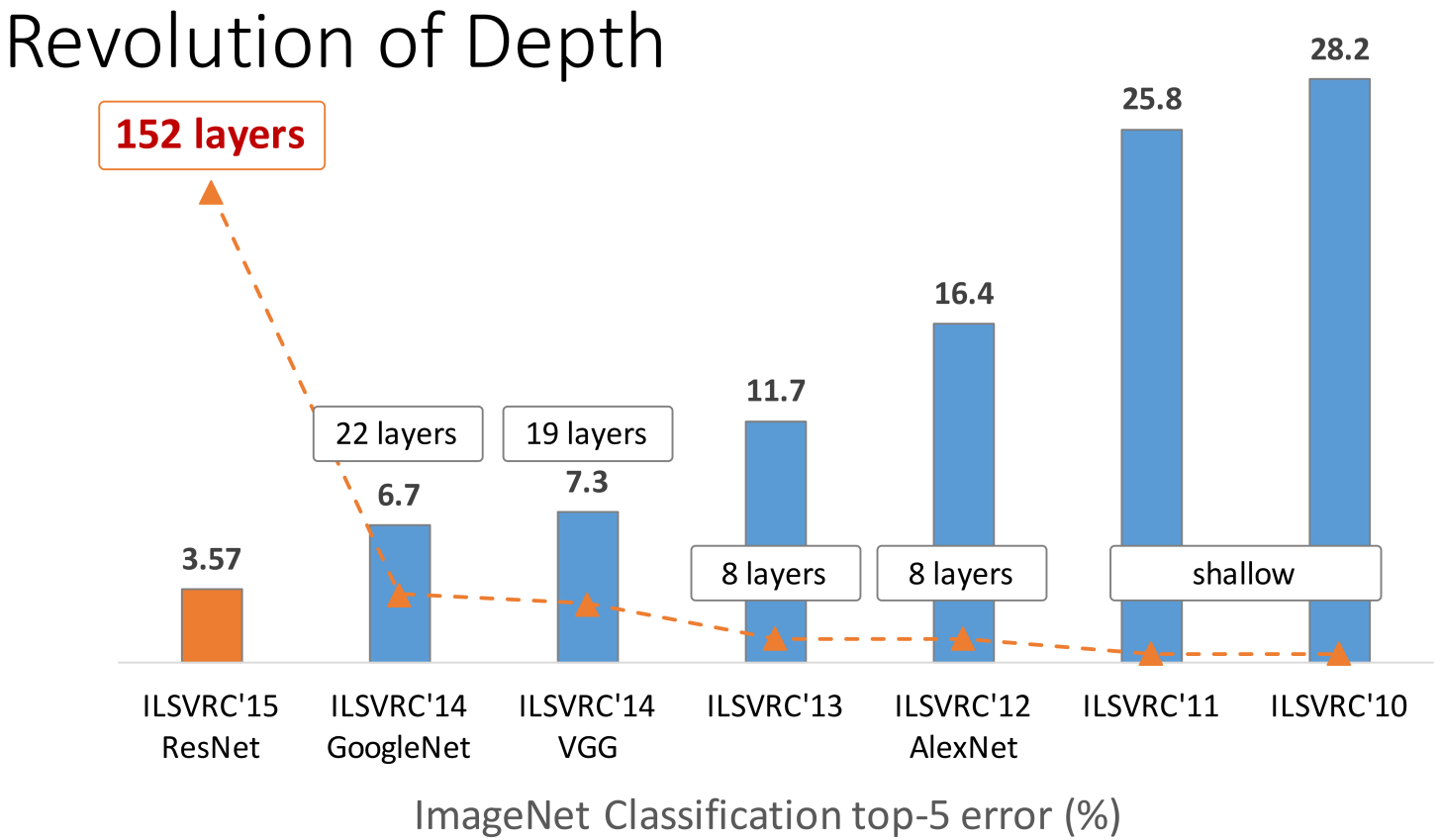
Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points
- What determines the statistical complexity of a deep network?
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - Understanding generalization failures

Outline

- **Deep residual networks**
  - Representing with near-identities
  - Global optimality of stationary points
- What determines the statistical complexity of a deep network?
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - Understanding generalization failures

Deeper Networks

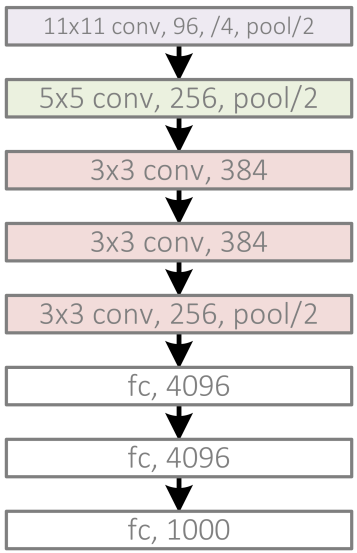


(Deep Residual Networks. Kaiming He. 2016)

Deeper Networks

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

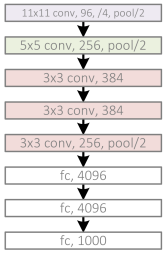


(Deep Residual Networks. Kaiming He. 2016)

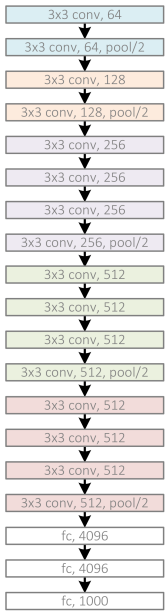
Deeper Networks

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

(Deep Residual Networks. Kaiming He. 2016)

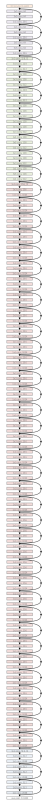
Deeper Networks

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

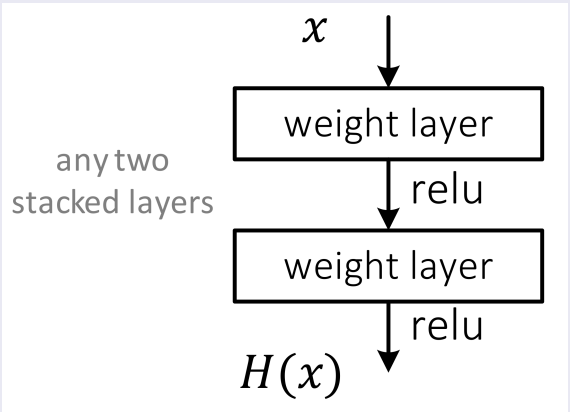
ResNet, 152 layers
(ILSVRC 2015)



(Deep Residual Networks. Kaiming He. 2016)

Deep Residual Networks

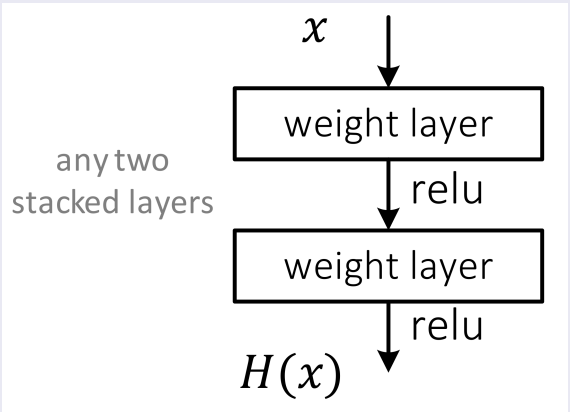
Deep network component



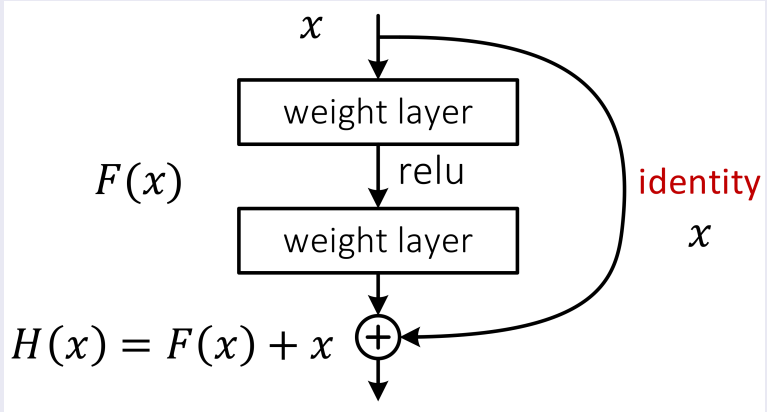
(Deep Residual Networks. Kaiming He. 2016)

Deep Residual Networks

Deep network component



Residual network component



(Deep Residual Networks. Kaiming He. 2016)

Deep Residual Networks

Advantages

- With zero weights, the network computes the identity.

Deep Residual Networks

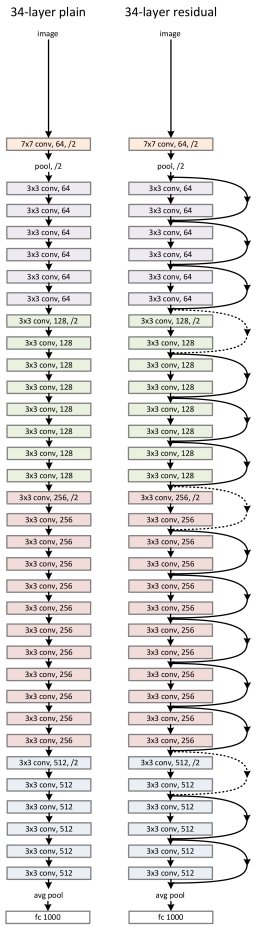
Advantages

- With zero weights, the network computes the identity.
- Identity connections provide useful feedback throughout the network.

Deep Residual Networks

Advantages

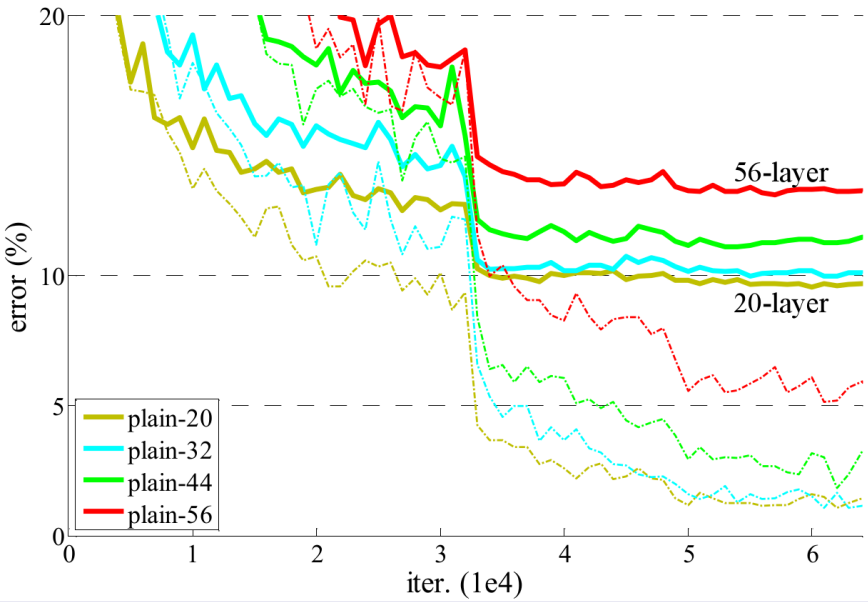
- With zero weights, the network computes the identity.
- Identity connections provide useful feedback throughout the network.

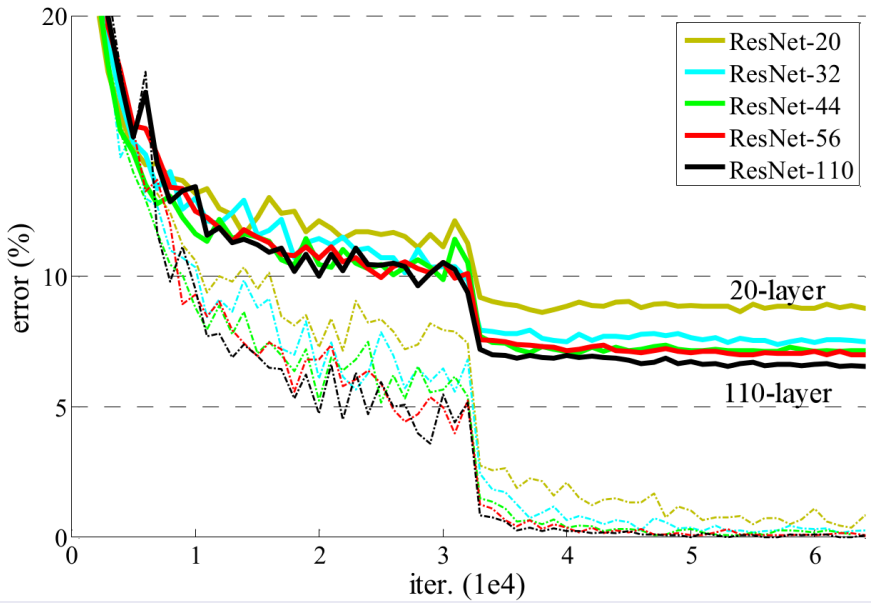


(Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016)

Deep Residual Networks

Training deep plain nets vs deep residual nets: CIFAR-10

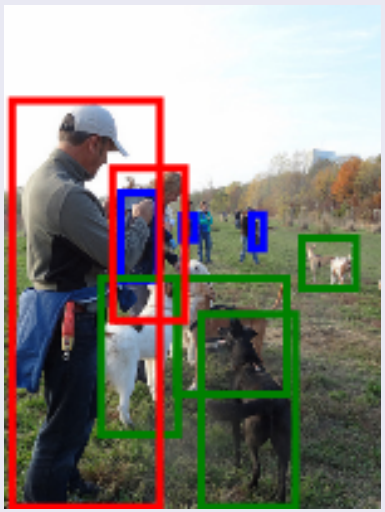


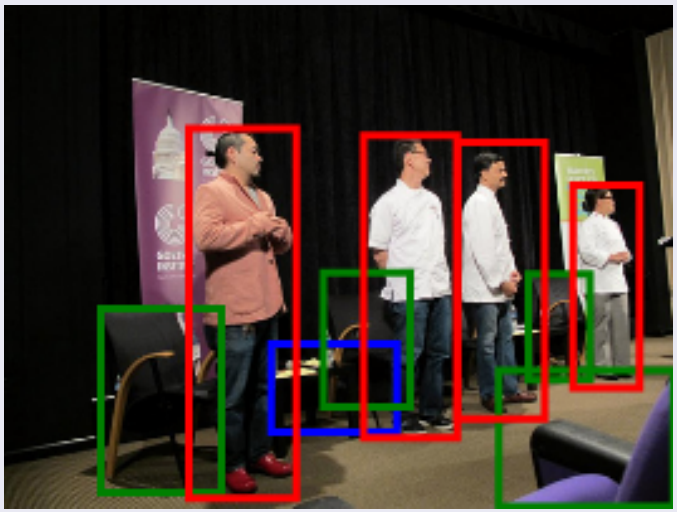


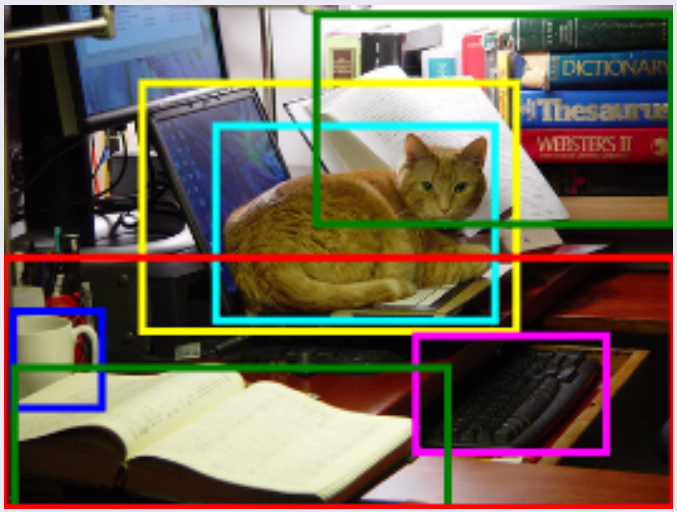
(Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016)

Deep Residual Networks: Competition Successes

ImageNet Large Scale Visual Recognition Challenge



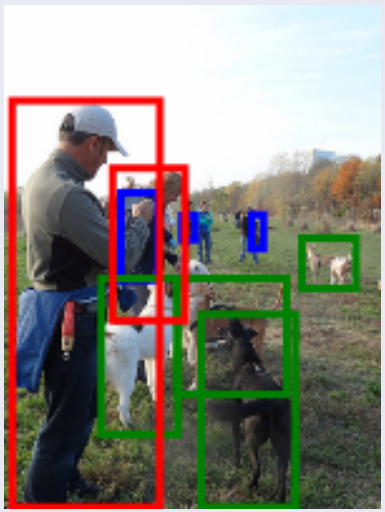
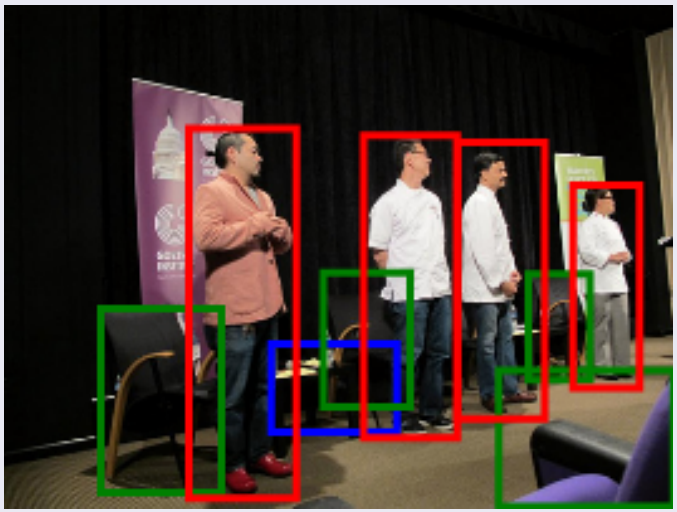
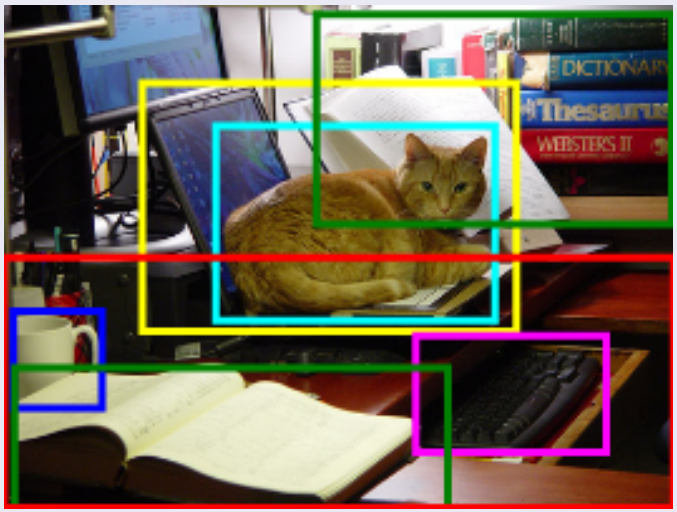




(http://image-net.org/)

Deep Residual Networks: Competition Successes

ImageNet Large Scale Visual Recognition Challenge

(http://image-net.org/)

First place:

- Object detection: 16% better than next best
- Object localization: 27% better than next best

Deep Residual Networks: Competition Successes

COCO (Common Objects in Context)



(http://mscoco.org/)

Deep Residual Networks: Competition Successes

COCO (Common Objects in Context)



(http://mscoco.org/)

First place:

- Detection: 11% better than next best
- Segmentation: 12% better than next best

Deep Residual Networks

Why?

- What is behind the success of residual networks?
- What is important for their performance?

Some intuition: linear functions

Products of near-identity matrices

1. Every invertible* A can be written as

$$A = (I + A_m) \cdots (I + A_1),$$

where $\|A_i\| = O(1/m)$.

(Hardt and Ma, 2016)

* Provided $\det(A) > 0$.

Some intuition: linear functions

Products of near-identity matrices

2. For a linear Gaussian model,

$$y = Ax + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I),$$

(Hardt and Ma, 2016)

Some intuition: linear functions

Products of near-identity matrices

2. For a linear Gaussian model,

$$y = Ax + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I),$$

consider choosing $A_1, \ldots, A_m$ to minimize quadratic loss:

$$\mathbb{E}\|(I + A_m) \cdots (I + A_1)x - y\|^2.$$

(Hardt and Ma, 2016)

Some intuition: linear functions

Products of near-identity matrices

2. For a linear Gaussian model,

$$y = Ax + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I),$$

consider choosing $A_1, \ldots, A_m$ to minimize quadratic loss:

$$\mathbb{E}\|(I + A_m) \cdots (I + A_1)x - y\|^2.$$

If $\|A_i\| < 1$, every stationary point of the quadratic loss is a global optimum:

$$\forall i,\ \nabla_{A_i} \mathbb{E}\|(I + A_m) \cdots (I + A_1)x - y\|^2 = 0$$
$$\Rightarrow \qquad A = (I + A_m) \cdots (I + A_1).$$

(Hardt and Ma, 2016)

Outline

- Deep residual networks
  - **Representing with near-identities**
  - Global optimality of stationary points
- What determines the statistical complexity of a deep network?

Steve Evans
Berkeley, Stat/Math

Phil Long
Google

Representing with near-identities

Result

The computation of a smooth invertible map h can be spread throughout a deep network,

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

so that all layers compute near-identity functions:

$$\|h_i - \mathrm{Id}\|_L = O\left(\frac{\log m}{m}\right).$$

Representing with near-identities

Result

The computation of a smooth invertible map h can be spread throughout a deep network,

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

so that all layers compute near-identity functions:

$$\|h_i - \mathrm{Id}\|_L = O\left(\frac{\log m}{m}\right).$$

Definition: the *Lipschitz seminorm* of f satisfies, for all x, y,

$$\|f(x) - f(y)\| \leq \|f\|_L \|x - y\|.$$

Representing with near-identities

Result

The computation of a smooth invertible map h can be spread throughout a deep network,

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

so that all layers compute near-identity functions:

$$\|h_i - \mathrm{Id}\|_L = O\left(\frac{\log m}{m}\right).$$

Think of the functions h_i as near-identity maps that might be computed as

$$h_i(x) = x + \underbrace{A\sigma(Bx)}.$$

Representing with near-identities

Result

The computation of a smooth invertible map h can be spread throughout a deep network,

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

so that all layers compute near-identity functions:

$$\|h_i - \mathrm{Id}\|_L = O\left(\frac{\log m}{m}\right).$$

Think of the functions h_i as near-identity maps that might be computed as

$$h_i(x) = x + \underbrace{A\sigma(Bx)}.$$

As the network gets deeper, the functions $x \mapsto A\sigma(Bx)$ can get flatter.

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,
3. Smooth: For some $\alpha > 0$ and all x, y, u, $\|Dh(y) - Dh(x)\| \leq \alpha \|y - x\|$.

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,
3. Smooth: For some $\alpha > 0$ and all x, y, u,
   $\|Dh(y) - Dh(x)\| \leq \alpha \|y - x\|$.

- Dh is the derivative; $\|Dh(y)\|$ is the induced norm:
  $\|f\| := \sup \left\{ \frac{\|f(x)\|}{\|x\|} : \|x\| > 0 \right\}$.

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,
3. Smooth: For some $\alpha > 0$ and all x, y, u, $\|Dh(y) - Dh(x)\| \leq \alpha \|y - x\|$.
4. Lipschitz inverse: For some $M > 0$, $\|h^{-1}\|_L \leq M$.

- Dh is the derivative; $\|Dh(y)\|$ is the induced norm:
  $\|f\| := \sup \left\{ \frac{\|f(x)\|}{\|x\|} : \|x\| > 0 \right\}$.

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,
3. Smooth: For some $\alpha > 0$ and all x, y, u, $\|Dh(y) - Dh(x)\| \leq \alpha \|y - x\|$.
4. Lipschitz inverse: For some $M > 0$, $\|h^{-1}\|_L \leq M$.
5. Positive orientation: For some x_0, $\det(Dh(x_0)) > 0$.

- Dh is the derivative; $\|Dh(y)\|$ is the induced norm:
  $\|f\| := \sup \left\{ \frac{\|f(x)\|}{\|x\|} : \|x\| > 0 \right\}$.

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,
3. Smooth: For some $\alpha > 0$ and all x, y, u, $\|Dh(y) - Dh(x)\| \leq \alpha \|y - x\|$.
4. Lipschitz inverse: For some $M > 0$, $\|h^{-1}\|_L \leq M$.
5. Positive orientation: For some x_0, $\det(Dh(x_0)) > 0$.

Then for all m, there are m functions $h_1, \ldots, h_m : \mathbb{R}^d \to \mathbb{R}^d$ satisfying $\|h_i - \mathrm{Id}\|_L = O(\log m / m)$

- Dh is the derivative; $\|Dh(y)\|$ is the induced norm:
  $\|f\| := \sup \left\{ \frac{\|f(x)\|}{\|x\|} : \|x\| > 0 \right\}$.

Representing with near-identities

Theorem

Consider a function $h : \mathbb{R}^d \to \mathbb{R}^d$ on a bounded domain $\mathcal{X} \subset \mathbb{R}^d$.
Suppose that it is

1. Differentiable,
2. Invertible,
3. Smooth: For some $\alpha > 0$ and all x, y, u, $\|Dh(y) - Dh(x)\| \leq \alpha \|y - x\|$.
4. Lipschitz inverse: For some $M > 0$, $\|h^{-1}\|_L \leq M$.
5. Positive orientation: For some x_0, $\det(Dh(x_0)) > 0$.

Then for all m, there are m functions $h_1, \ldots, h_m : \mathbb{R}^d \to \mathbb{R}^d$ satisfying $\|h_i - \mathrm{Id}\|_L = O(\log m/m)$ and $h_m \circ h_{m-1} \circ \cdots \circ h_1 = h$ on $\mathcal{X}$.

- Dh is the derivative; $\|Dh(y)\|$ is the induced norm:
  $\|f\| := \sup \left\{ \frac{\|f(x)\|}{\|x\|} : \|x\| > 0 \right\}$.

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that
$$h_1(x) = \frac{h(a_1 x)}{a_1}$$

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that

$$h_1(x) = \frac{h(a_1 x)}{a_1}$$

$$h_2(h_1(x)) = \frac{h(a_2 x)}{a_2}$$

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that

$$h_1(x) = \frac{h(a_1 x)}{a_1}$$

$$h_2(h_1(x)) = \frac{h(a_2 x)}{a_2}$$

$$\vdots$$

$$h_m(\cdots(h_1(x))\cdots) = \frac{h(a_m x)}{a_m},$$

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that

$$h_1(x) = \frac{h(a_1 x)}{a_1}$$
$$h_2(h_1(x)) = \frac{h(a_2 x)}{a_2}$$
$$\vdots$$
$$h_m(\cdots(h_1(x))\cdots) = \frac{h(a_m x)}{a_m},$$

3. Pick $a_m = 1$ so $h_m \circ \cdots \circ h_1 = h$.

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that

$$h_1(x) = \frac{h(a_1 x)}{a_1}$$

$$h_2(h_1(x)) = \frac{h(a_2 x)}{a_2}$$

$$\vdots$$

$$h_m(\cdots(h_1(x))\cdots) = \frac{h(a_m x)}{a_m},$$

3. Pick $a_m = 1$ so $h_m \circ \cdots \circ h_1 = h$.
4. Ensure that a_1 is small enough that $h_1 \approx Dh(0) = \mathrm{Id}$.

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that

$$h_1(x) = \frac{h(a_1 x)}{a_1}$$

$$h_2(h_1(x)) = \frac{h(a_2 x)}{a_2}$$

$$\vdots$$

$$h_m(\cdots(h_1(x))\cdots) = \frac{h(a_m x)}{a_m},$$

3. Pick $a_m = 1$ so $h_m \circ \cdots \circ h_1 = h$.
4. Ensure that a_1 is small enough that $h_1 \approx Dh(0) = \mathrm{Id}$.
5. Ensure that a_i and a_{i+1} are sufficiently close that $h_i \approx \mathrm{Id}$.

Representing with near-identities

Key ideas

1. Assume $h(0) = 0$ and $Dh(0) = \mathrm{Id}$ (else shift and linearly transform).
2. Construct the h_i so that

$$h_1(x) = \frac{h(a_1 x)}{a_1}$$

$$h_2(h_1(x)) = \frac{h(a_2 x)}{a_2}$$

$$\vdots$$

$$h_m(\cdots(h_1(x))\cdots) = \frac{h(a_m x)}{a_m},$$

3. Pick $a_m = 1$ so $h_m \circ \cdots \circ h_1 = h$.
4. Ensure that a_1 is small enough that $h_1 \approx Dh(0) = \mathrm{Id}$.
5. Ensure that a_i and a_{i+1} are sufficiently close that $h_i \approx \mathrm{Id}$.
6. Show $\|h_i - \mathrm{Id}\|_L$ is small on small and large scales (c.f. $a_i - a_{i-1}$).

Representing with near-identities

Result

The computation of a smooth invertible map h can be spread throughout a deep network,

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

so that all layers compute near-identity functions:

$$\|h_i - \mathrm{Id}\|_L = O\left(\frac{\log m}{m}\right).$$

- Deeper networks allow flatter nonlinear functions at each layer.

Outline

- Deep residual networks
  - Representing with near-identities
  - **Global optimality of stationary points**
- What determines the statistical complexity of a deep network?

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.

- e.g., if (X, Y) is uniform on a training sample,

then Q is empirical risk and h^* an empirical risk minimizer.

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.
Consider a function $h = h_m \circ \cdots \circ h_1$, where $\|h_i - \mathrm{Id}\|_L \leq \epsilon < 1$.

- e.g., if (X, Y) is uniform on a training sample,

then Q is empirical risk and h^* an empirical risk minimizer.

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.
Consider a function $h = h_m \circ \cdots \circ h_1$, where $\|h_i - \mathrm{Id}\|_L \leq \epsilon < 1$.
Then for all i,

$$\|D_{h_i} Q(h)\| \geq \frac{(1-\epsilon)^{m-1}}{\|h - h^*\|}\,(Q(h) - Q(h^*))\,.$$

- e.g., if (X, Y) is uniform on a training sample,

then Q is empirical risk and h^* an empirical risk minimizer.

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.
Consider a function $h = h_m \circ \cdots \circ h_1$, where $\|h_i - \mathrm{Id}\|_L \leq \epsilon < 1$.
Then for all i,

$$\|D_{h_i} Q(h)\| \geq \frac{(1-\epsilon)^{m-1}}{\|h - h^*\|}\,(Q(h) - Q(h^*))\,.$$

- e.g., if (X, Y) is uniform on a training sample,

then Q is empirical risk and h^* an empirical risk minimizer.

- $D_{h_i} Q$ is a Fréchet derivative; $\|h\|$ is the induced norm.

Stationary points

What the theorem says

Stationary points

What the theorem says

- If the composition h is sub-optimal and each function h_i is a near-identity, then there is a downhill direction in function space: the functional gradient of Q wrt h_i is non-zero.

Stationary points

What the theorem says

- If the composition h is sub-optimal and each function h_i is a near-identity, then there is a downhill direction in function space: the functional gradient of Q wrt h_i is non-zero.
- Thus every stationary point is a global optimum.

Stationary points

What the theorem says

- If the composition h is sub-optimal and each function h_i is a near-identity, then there is a downhill direction in function space: the functional gradient of Q wrt h_i is non-zero.
- Thus every stationary point is a global optimum.
- There are no local minima and no saddle points.

Stationary points

What the theorem says

Stationary points

What the theorem says

- The theorem does not say there are no local minima of a deep residual network of ReLUs or sigmoids with a fixed architecture.

Stationary points

What the theorem says

- The theorem does not say there are no local minima of a deep residual network of ReLUs or sigmoids with a fixed architecture.
- Except at the global minimum, there is a downhill direction in function space. But this direction might be orthogonal to functions that can be computed with this fixed architecture.

Stationary points

What the theorem says

- The theorem does not say there are no local minima of a deep residual network of ReLUs or sigmoids with a fixed architecture.
- Except at the global minimum, there is a downhill direction in function space. But this direction might be orthogonal to functions that can be computed with this fixed architecture.
- We should expect suboptimal stationary points in the ReLU or sigmoid parameter space, but these cannot arise because of interactions between parameters in different layers; they arise only within a layer.

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.
Consider a function $h = h_m \circ \cdots \circ h_1$, where $\|h_i - \mathrm{Id}\|_L \leq \epsilon < 1$.
Then for all i,

$$\|D_{h_i} Q(h)\| \geq \frac{(1-\epsilon)^{m-1}}{\|h - h^*\|}\,(Q(h) - Q(h^*))\,.$$

- e.g., if (X, Y) is uniform on a training sample,

then Q is empirical risk and h^* an empirical risk minimizer.

- $D_{h_i} Q$ is a Fréchet derivative; $\|h\|$ is the induced norm.

Stationary points

Proof ideas (1)

If $\|f - \mathrm{Id}\|_L \leq \alpha < 1$ then

Stationary points

Proof ideas (1)

If $\|f - \mathrm{Id}\|_L \leq \alpha < 1$ then

1. f is invertible.

Stationary points

Proof ideas (1)

If $\|f - \mathrm{Id}\|_L \leq \alpha < 1$ then

1. f is invertible.
2. $\|f\|_L \leq 1 + \alpha$ and $\|f^{-1}\|_L \leq 1/(1-\alpha)$.

Stationary points

Proof ideas (1)

If $\|f - \mathrm{Id}\|_L \leq \alpha < 1$ then

1. f is invertible.
2. $\|f\|_L \leq 1 + \alpha$ and $\|f^{-1}\|_L \leq 1/(1-\alpha)$.
3. For $F(g) = f \circ g$, $\|DF(g) - \mathrm{Id}\| \leq \alpha$.

Stationary points

Proof ideas (1)

If $\|f - \mathrm{Id}\|_L \leq \alpha < 1$ then

1. f is invertible.
2. $\|f\|_L \leq 1 + \alpha$ and $\|f^{-1}\|_L \leq 1/(1-\alpha)$.
3. For $F(g) = f \circ g$, $\|DF(g) - \mathrm{Id}\| \leq \alpha$.

- $\|f\|$ denotes the induced norm: $\|g\| := \sup\left\{\frac{\|g(x)\|}{\|x\|} : \|x\| > 0\right\}$.

Stationary points

Proof ideas (1)

If $\|f - \mathrm{Id}\|_L \leq \alpha < 1$ then

1. f is invertible.
2. $\|f\|_L \leq 1 + \alpha$ and $\|f^{-1}\|_L \leq 1/(1-\alpha)$.
3. For $F(g) = f \circ g$, $\|DF(g) - \mathrm{Id}\| \leq \alpha$.
4. For a linear map h (such as $DF(g) - \mathrm{Id}$), $\|h\| = \|h\|_L$.

- $\|f\|$ denotes the induced norm: $\|g\| := \sup\left\{\frac{\|g(x)\|}{\|x\|} : \|x\| > 0\right\}$.

Proof ideas (2)

1. Projection theorem implies

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - h^*(X)\|_2^2 + \text{constant}.$$

Proof ideas (2)

1. Projection theorem implies

$$Q(h) = \frac{1}{2}\mathbb{E}\left\|h(X) - h^*(X)\right\|_2^2 + \text{constant.}$$

2. Then

$$D_{h_i} Q(h) = \mathbb{E}\left[\left(h(X) - h^*(X)\right) \cdot \mathrm{ev}_X \circ D_{h_i} h\right].$$

Stationary points

Proof ideas (2)

1. Projection theorem implies

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - h^*(X)\|_2^2 + \text{constant}.$$

2. Then

$$D_{h_i} Q(h) = \mathbb{E}\left[(h(X) - h^*(X)) \cdot \mathrm{ev}_X \circ D_{h_i} h\right].$$

- ev_x is the evaluation functional, $\mathrm{ev}_x(f) = f(x)$.

Stationary points

Proof ideas (2)

1. Projection theorem implies

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - h^*(X)\|_2^2 + \text{constant}.$$

2. Then

$$D_{h_i} Q(h) = \mathbb{E}\left[(h(X) - h^*(X)) \cdot \mathrm{ev}_X \circ D_{h_i} h\right].$$

3. It is possible to choose a direction Δ s.t. $\|\Delta\| = 1$ and

$$D_{h_i} Q(h)(\Delta) = c\mathbb{E}\,\|h(X) - h^*(X)\|_2^2.$$

- ev_x is the evaluation functional, $\mathrm{ev}_x(f) = f(x)$.

Stationary points

Proof ideas (2)

1. Projection theorem implies

$$Q(h) = \frac{1}{2}\mathbb{E}\left\|h(X) - h^*(X)\right\|_2^2 + \text{constant}.$$

2. Then

$$D_{h_i} Q(h) = \mathbb{E}\left[\left(h(X) - h^*(X)\right) \cdot \mathrm{ev}_X \circ D_{h_i} h\right].$$

3. It is possible to choose a direction Δ s.t. $\|\Delta\| = 1$ and

$$D_{h_i} Q(h)(\Delta) = c\mathbb{E}\left\|h(X) - h^*(X)\right\|_2^2.$$

4. Because the h_js are near-identities,

$$c \geq \frac{(1-\epsilon)^{m-1}}{\|h - h^*\|}.$$

- ev_x is the evaluation functional, $\mathrm{ev}_x(f) = f(x)$.

Stationary points

Result

For (X, Y) with an arbitrary joint distribution, define the squared error,

$$Q(h) = \frac{1}{2}\mathbb{E}\,\|h(X) - Y\|_2^2\,,$$

define the minimizer $h^*(x) = \mathbb{E}[Y|X = x]$.
Consider a function $h = h_m \circ \cdots \circ h_1$, where $\|h_i - \mathrm{Id}\|_L \leq \epsilon < 1$.
Then for all i,

$$\|D_{h_i} Q(h)\| \geq \frac{(1-\epsilon)^{m-1}}{\|h - h^*\|}\,(Q(h) - Q(h^*))\,.$$

- e.g., if (X, Y) is uniform on a training sample,

then Q is empirical risk and h^* an empirical risk minimizer.

- $D_{h_i} Q$ is a Fréchet derivative; $\|h\|$ is the induced norm.

Deep compositions of near-identities

Questions

- If the mapping is not invertible?

Deep compositions of near-identities

Questions

- If the mapping is not invertible?
  e.g., $h : \mathbb{R}^d \to \mathbb{R}$?

Deep compositions of near-identities

Questions

- If the mapping is not invertible?
  e.g., $h : \mathbb{R}^d \to \mathbb{R}$?
  If h can be extended to a bi-Lipschitz mapping to $\mathbb{R}^d$, it can be represented with flat functions at each layer.

Deep compositions of near-identities

Questions

- If the mapping is not invertible?
  e.g., $h : \mathbb{R}^d \to \mathbb{R}$?
  If h can be extended to a bi-Lipschitz mapping to $\mathbb{R}^d$, it can be represented with flat functions at each layer.
  What if it cannot?

Deep compositions of near-identities

Questions

- If the mapping is not invertible?
  e.g., $h : \mathbb{R}^d \to \mathbb{R}$?
  If h can be extended to a bi-Lipschitz mapping to $\mathbb{R}^d$, it can be represented with flat functions at each layer.
  What if it cannot?
- Implications for optimization?

Deep compositions of near-identities

Questions

- If the mapping is not invertible?
  e.g., $h : \mathbb{R}^d \to \mathbb{R}$?
  If h can be extended to a bi-Lipschitz mapping to $\mathbb{R}^d$, it can be represented with flat functions at each layer.
  What if it cannot?
- Implications for optimization?
  Related to Polyak-Łojasiewicz function classes; proximal algorithms for these classes converge quickly to stationary points.

Deep compositions of near-identities

Questions

- If the mapping is not invertible?
  e.g., $h : \mathbb{R}^d \to \mathbb{R}$?
  If h can be extended to a bi-Lipschitz mapping to $\mathbb{R}^d$, it can be represented with flat functions at each layer.
  What if it cannot?
- Implications for optimization?
  Related to Polyak-Łojasiewicz function classes; proximal algorithms for these classes converge quickly to stationary points.
- Do stochastic gradient methods produce near-identities?

Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points
- **What determines the statistical complexity of a deep network?**
  - 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).

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

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

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.

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.
- 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

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

- 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.
- 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): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(p)$.

(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): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(p)$.

   (Baum and Haussler, 1989)

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

   (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): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(p)$.
   (Baum and Haussler, 1989)
2. Piecewise linear (ReLUs): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(pL)$.
   (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): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(p)$.
   (Baum and Haussler, 1989)
2. Piecewise linear (ReLUs): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(pL)$.
   (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)

Generalization in Neural Networks: Number of Parameters

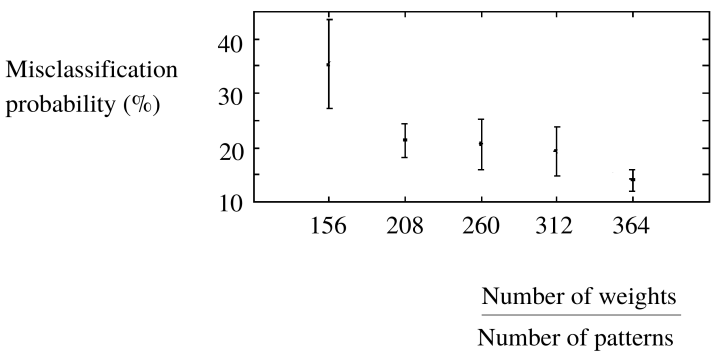
NIPS 1996

Experimental Results

Neural networks with many parameters, trained on small data sets, sometimes generalize well.

Eg: Face recognition (Lawrence *et al*, 1996)

$m = 50$ training patterns.



Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points
- What determines the statistical complexity of a deep network?
  - VC theory: Number of parameters
  - **Margins analysis: Size of parameters**
  - Understanding generalization failures

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\text{sign}(f(x)) \in \{-1, 1\}$.

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\operatorname{sign}(f(x)) \in \{-1, 1\}$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, if $yf(x) > 0$ then f classifies x correctly.

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\mathrm{sign}(f(x)) \in \{-1, 1\}$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, if $yf(x) > 0$ then f classifies x correctly.
- We call $yf(x)$ the *margin* of f on x.

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\operatorname{sign}(f(x)) \in \{-1, 1\}$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, if $yf(x) > 0$ then f classifies x correctly.
- We call $yf(x)$ the *margin* of f on x.
- We can view a larger margin as a more confident correct classification.

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\text{sign}(f(x)) \in \{-1, 1\}$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, if $yf(x) > 0$ then f classifies x correctly.
- We call $yf(x)$ the *margin* of f on x.
- 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 real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\operatorname{sign}(f(x)) \in \{-1, 1\}$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, if $yf(x) > 0$ then f classifies x correctly.
- We call $yf(x)$ the *margin* of f on x.
- 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: Margins and Size of Parameters

Theorem (B., 1996)

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

n training examples

$(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

n training examples

$(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$ $\quad$ $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

n training examples

$(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$ $\quad$ $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$

$$\Pr(\operatorname{sign}(f(X)) \neq Y)$$

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\operatorname{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

- The bound depends on the margin loss plus an error term.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\operatorname{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

- The bound depends on the margin loss plus an error term.
- Minimizing quadratic loss or cross-entropy loss leads to large margins.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\operatorname{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

- The bound depends on the margin loss plus an error term.
- Minimizing quadratic loss or cross-entropy loss leads to large margins.
- $\operatorname{fat}_{\mathcal{F}}(\gamma)$ is a scale-sensitive version of VC-dimension. Unlike the VC-dimension, it need not grow with the number of parameters.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\mathrm{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

2. If functions in $\mathcal{F}$ are computed by two-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\mathrm{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^2).$$

- The bound depends on the margin loss plus an error term.
- Minimizing quadratic loss or cross-entropy loss leads to large margins.
- $\mathrm{fat}_{\mathcal{F}}(\gamma)$ is a scale-sensitive version of VC-dimension. Unlike the VC-dimension, it need not grow with the number of parameters.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\mathrm{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

2. If functions in $\mathcal{F}$ are computed by two-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\mathrm{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^2).$$

- Same ideas used to give rigorous dimension-independent generalization bounds for SVMs (B. and Shawe-Taylor, 1999)

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\mathrm{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

2. If functions in $\mathcal{F}$ are computed by two-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\mathrm{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^2).$$

- Same ideas used to give rigorous dimension-independent generalization bounds for SVMs (B. and Shawe-Taylor, 1999)
- ... and margins analysis of AdaBoost. (Schapire, Freund, B., Lee, 1998)

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\operatorname{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

2. If functions in $\mathcal{F}$ are computed by two-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\operatorname{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^2).$$

- The scale of functions $f \in \mathcal{F}$ is important.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\operatorname{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

2. If functions in $\mathcal{F}$ are computed by two-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\operatorname{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^2).$$

- The scale of functions $f \in \mathcal{F}$ is important.
- Bigger fs give bigger margins, so $\operatorname{fat}_{\mathcal{F}}(\gamma)$ should be bigger.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\operatorname{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\operatorname{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

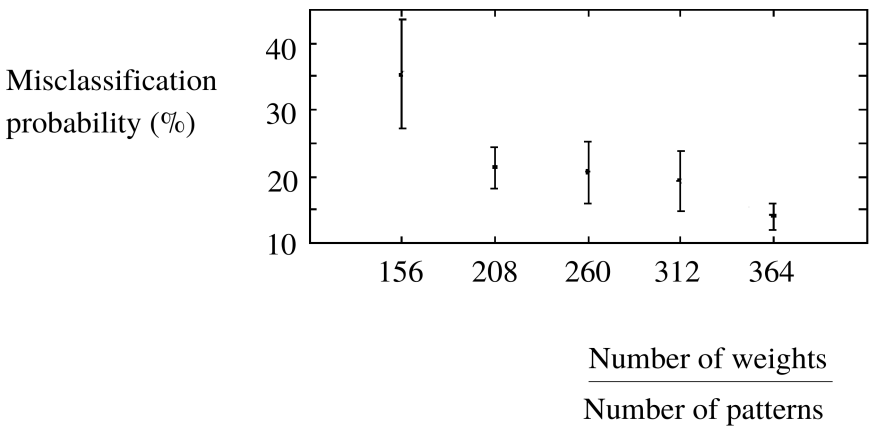
2. If functions in $\mathcal{F}$ are computed by two-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\operatorname{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^2).$$

- The scale of functions $f \in \mathcal{F}$ is important.
- Bigger fs give bigger margins, so $\operatorname{fat}_{\mathcal{F}}(\gamma)$ should be bigger.
- The output y of a sigmoid layer has $\|y\|_\infty \leq 1$, so $\|w\|_1 \leq B$ controls the scale of f.

Generalization: Margins and Size of Parameters

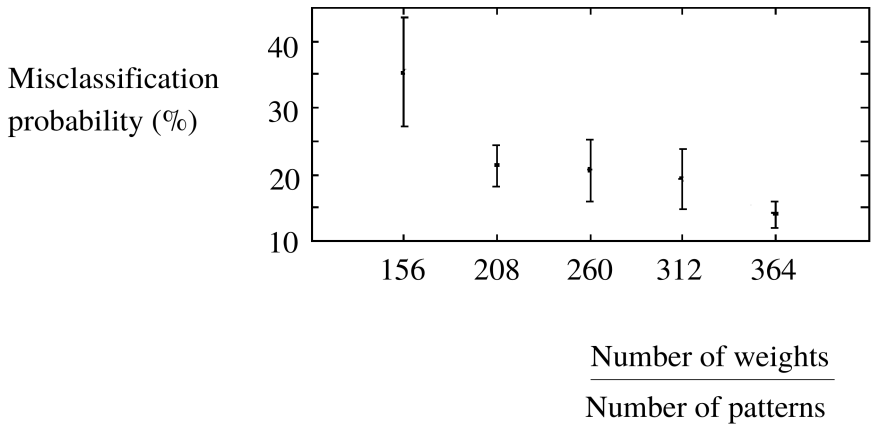
1996: Sigmoid networks



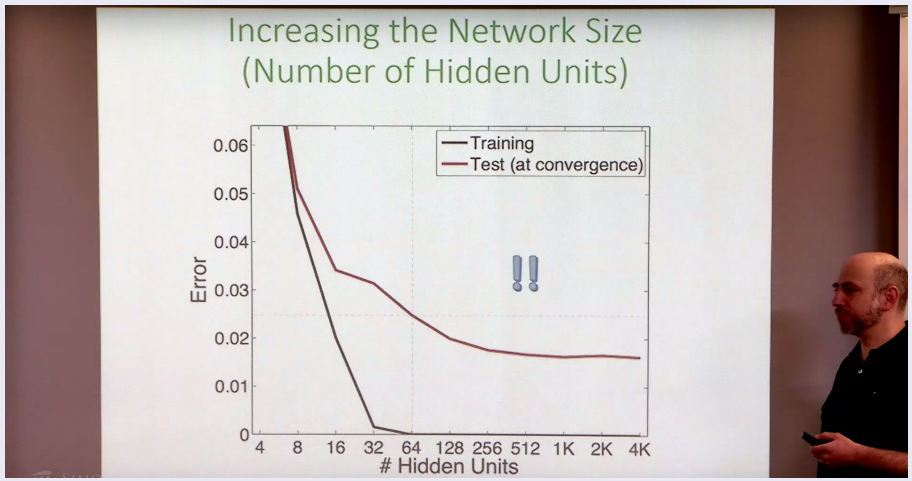
- Qualitative behavior explained by small weights theorem.

Generalization: Margins and Size of Parameters

1996: Sigmoid networks



2017: Deep ReLU networks

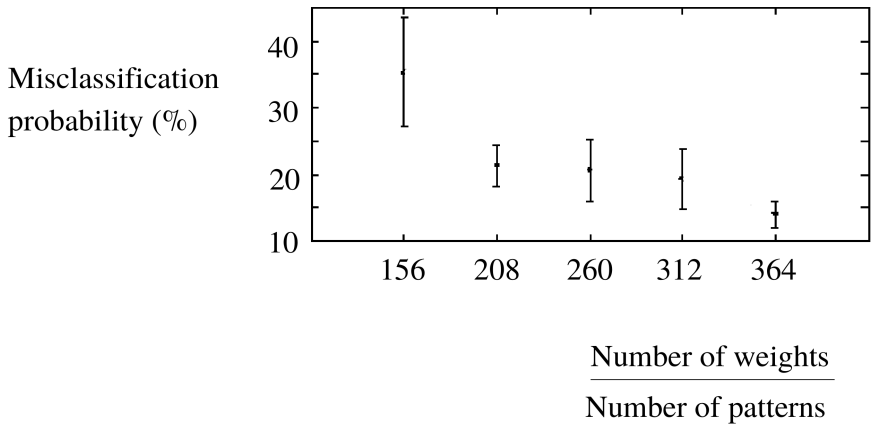


simons.berkeley.edu

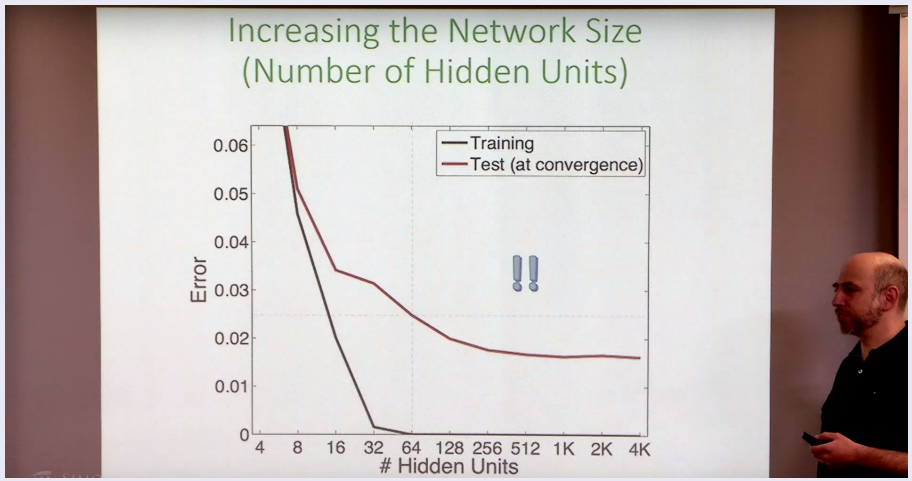
- Qualitative behavior explained by small weights theorem.

Generalization: Margins and Size of Parameters

1996: Sigmoid networks



2017: Deep ReLU networks



simons.berkeley.edu

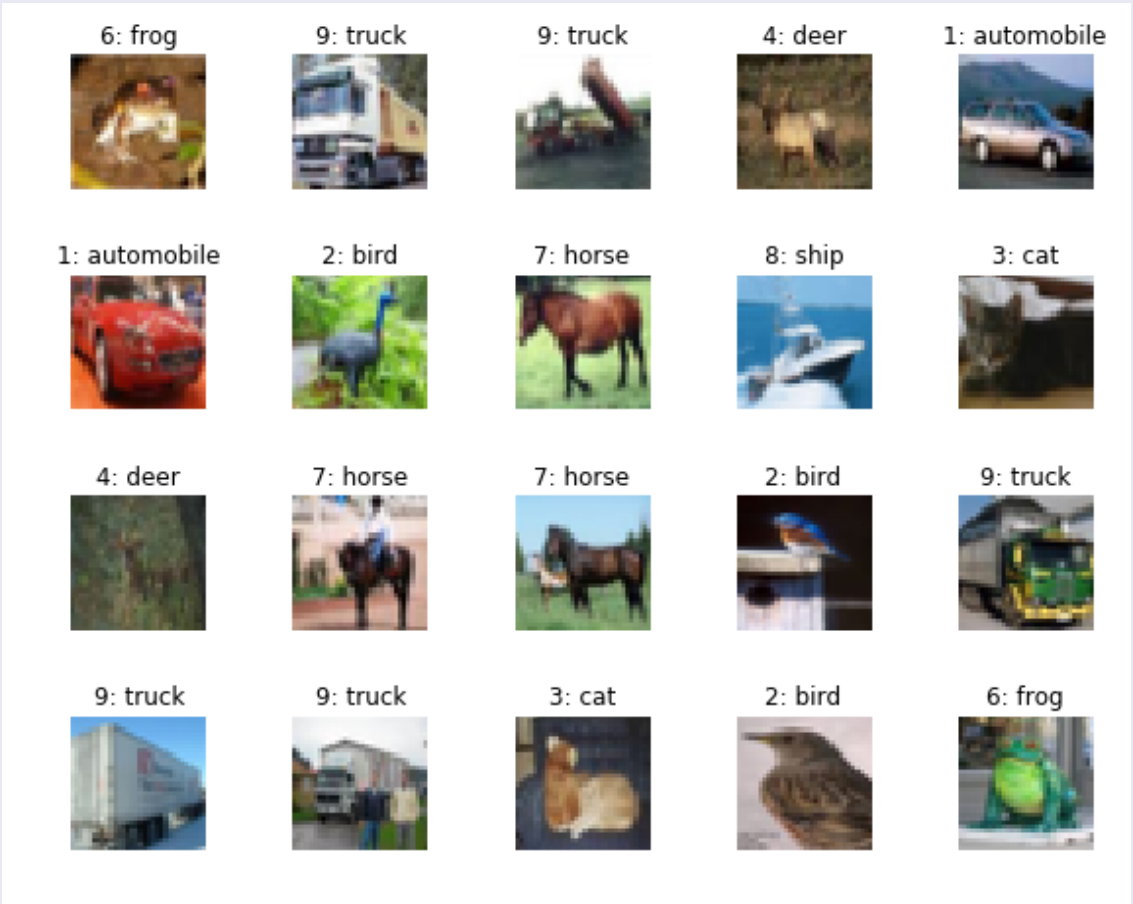
- Qualitative behavior explained by small weights theorem.
- How to measure the complexity of a ReLU network?

Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points
- What determines the statistical complexity of a deep network?
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - **Understanding generalization failures**

Explaining Generalization Failures

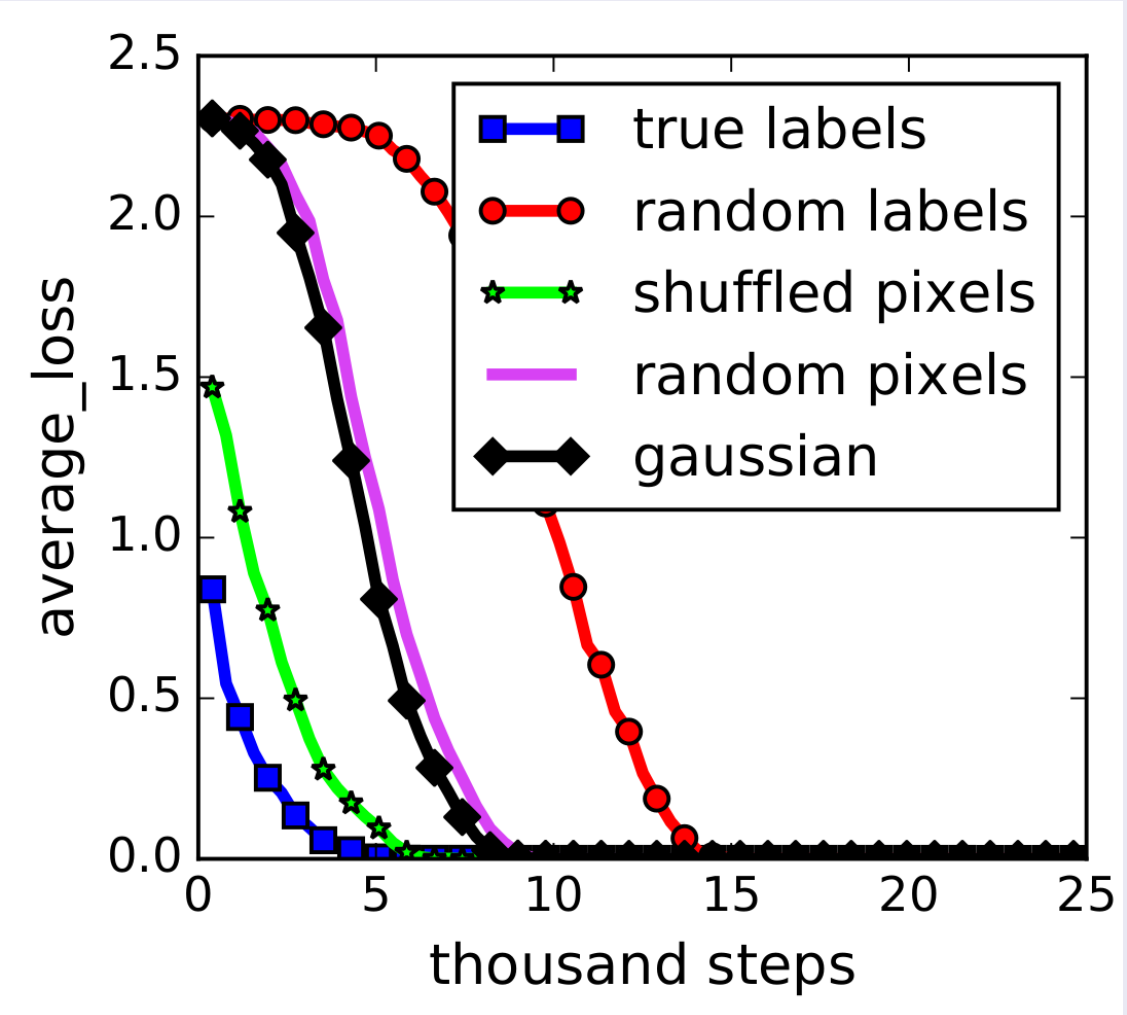
CIFAR10



http://corochann.com/

Explaining Generalization Failures

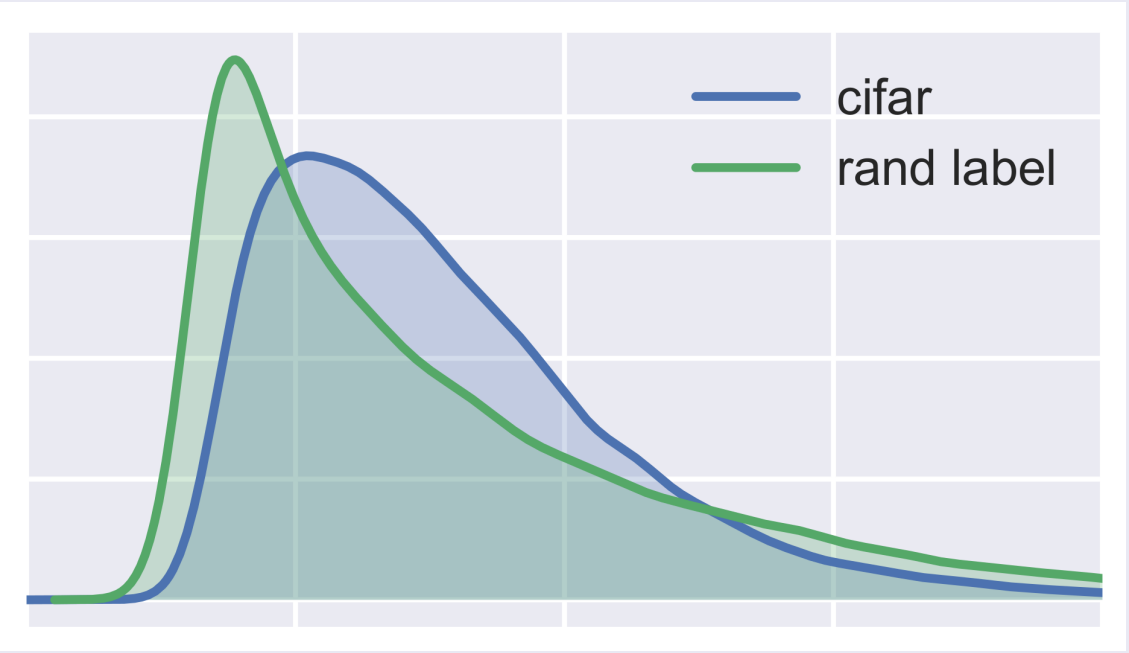
Stochastic Gradient Training Error on CIFAR10



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

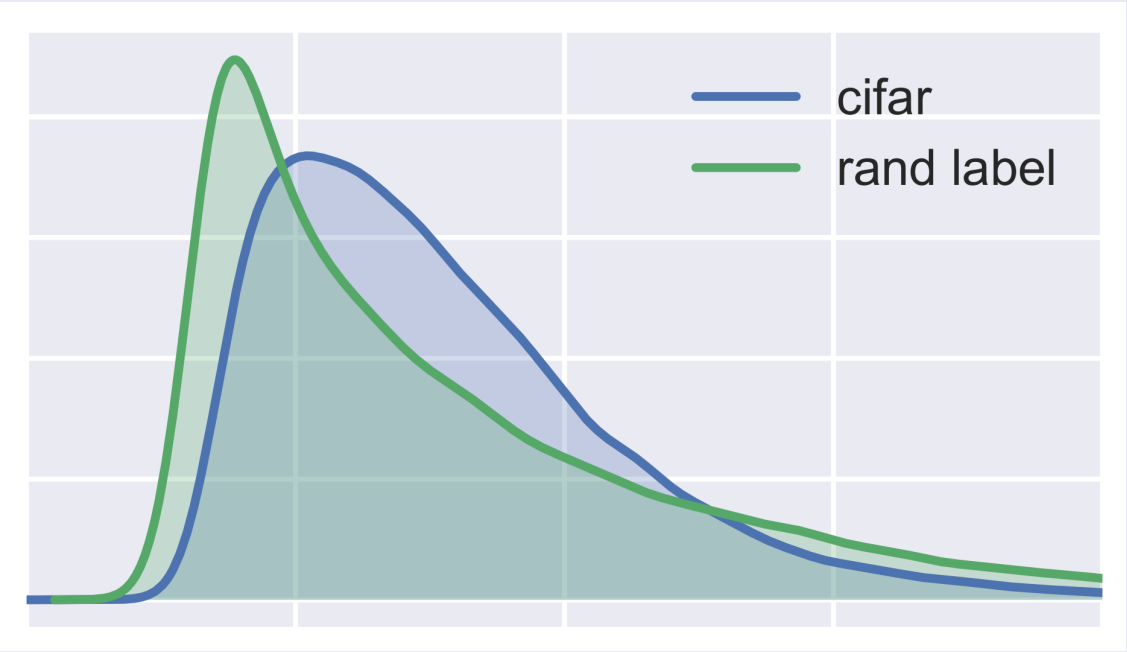
Explaining Generalization Failures

Training margins on CIFAR10 with true and random labels



Explaining Generalization Failures

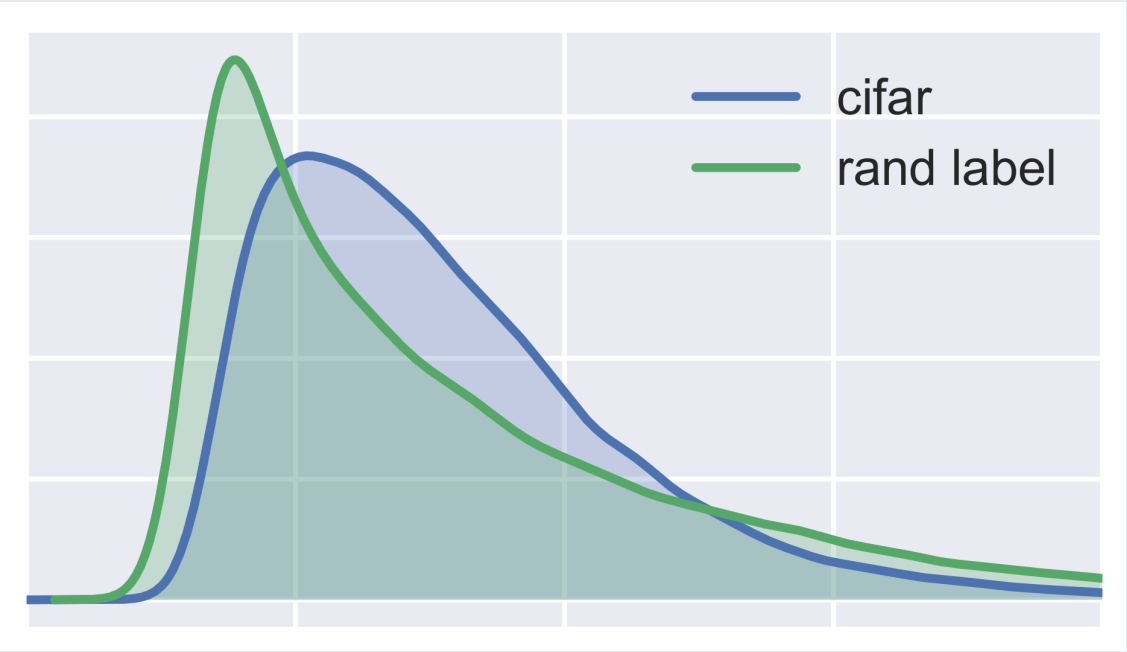
Training margins on CIFAR10 with true and random labels



- How does this match the large margin explanation?

Explaining Generalization Failures

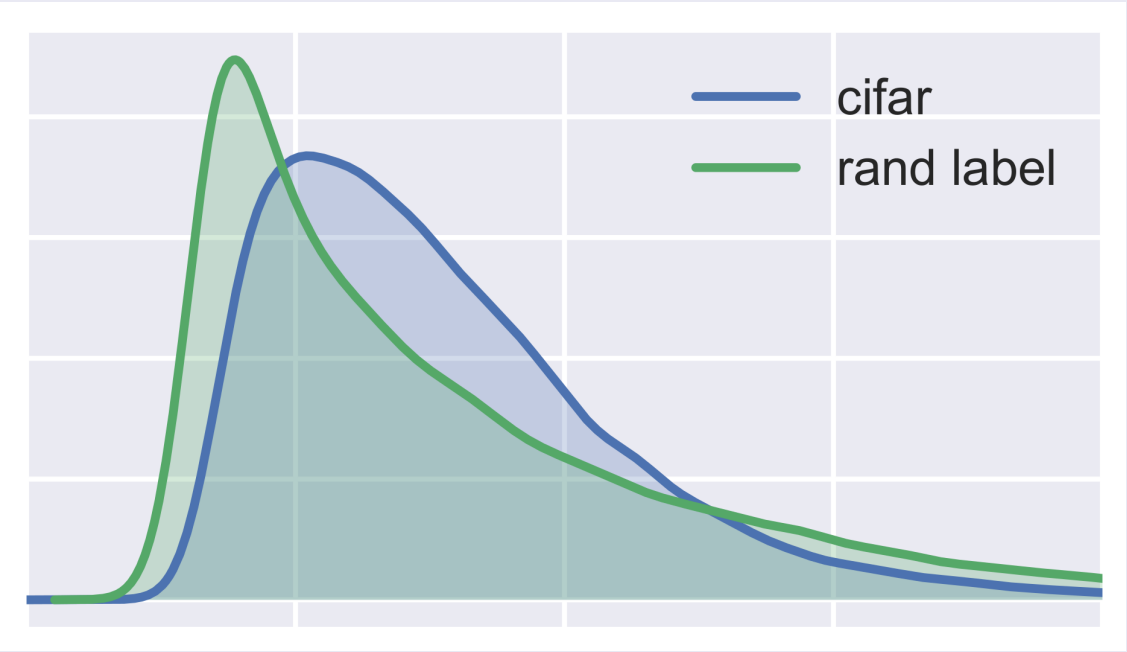
Training margins on CIFAR10 with true and random labels



- How does this match the large margin explanation?
- Need to account for the *scale* of the neural network functions.

Explaining Generalization Failures

Training margins on CIFAR10 with true and random labels



- How does this match the large margin explanation?
- Need to account for the *scale* of the neural network functions.
- What is the appropriate notion of the size of these functions?

Generalization in Deep Networks

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

Dylan Foster
Cornell

Matus Telgarsky
UIUC

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measuring the size of functions computed by a network of ReLUs.

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measuring the size of functions computed by a network of ReLUs. (c.f. sigmoid networks: the output y of a layer has $\|y\|_\infty \leq 1$, so $\|w\|_1 \leq B$ keeps the scale under control.)

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measuring the size of functions computed by a network of ReLUs. (c.f. sigmoid networks: the output y of a layer has $\|y\|_\infty \leq 1$, so $\|w\|_1 \leq B$ keeps the scale under control.)
- Large multiclass versus binary classification.

Generalization in Deep Networks

New results for generalization in deep ReLU networks

- Measuring the size of functions computed by a network of ReLUs. (c.f. sigmoid networks: the output y of a layer has $\|y\|_\infty \leq 1$, so $\|w\|_1 \leq B$ keeps the scale under control.)
- 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

- Measuring the size of functions computed by a network of ReLUs. (c.f. sigmoid networks: the output y of a layer has $\|y\|_\infty \leq 1$, so $\|w\|_1 \leq B$ keeps the scale under control.)
- Large multiclass versus binary classification.

Definitions

- Consider operator norms: For a matrix A_i,

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

- Multiclass margin function for $f : \mathcal{X} \to \mathbb{R}^m$, $y \in \{1, \ldots, m\}$:

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

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

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

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

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

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 $A_1, \ldots, A_L$:

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

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 $A_1, \ldots, A_L$:

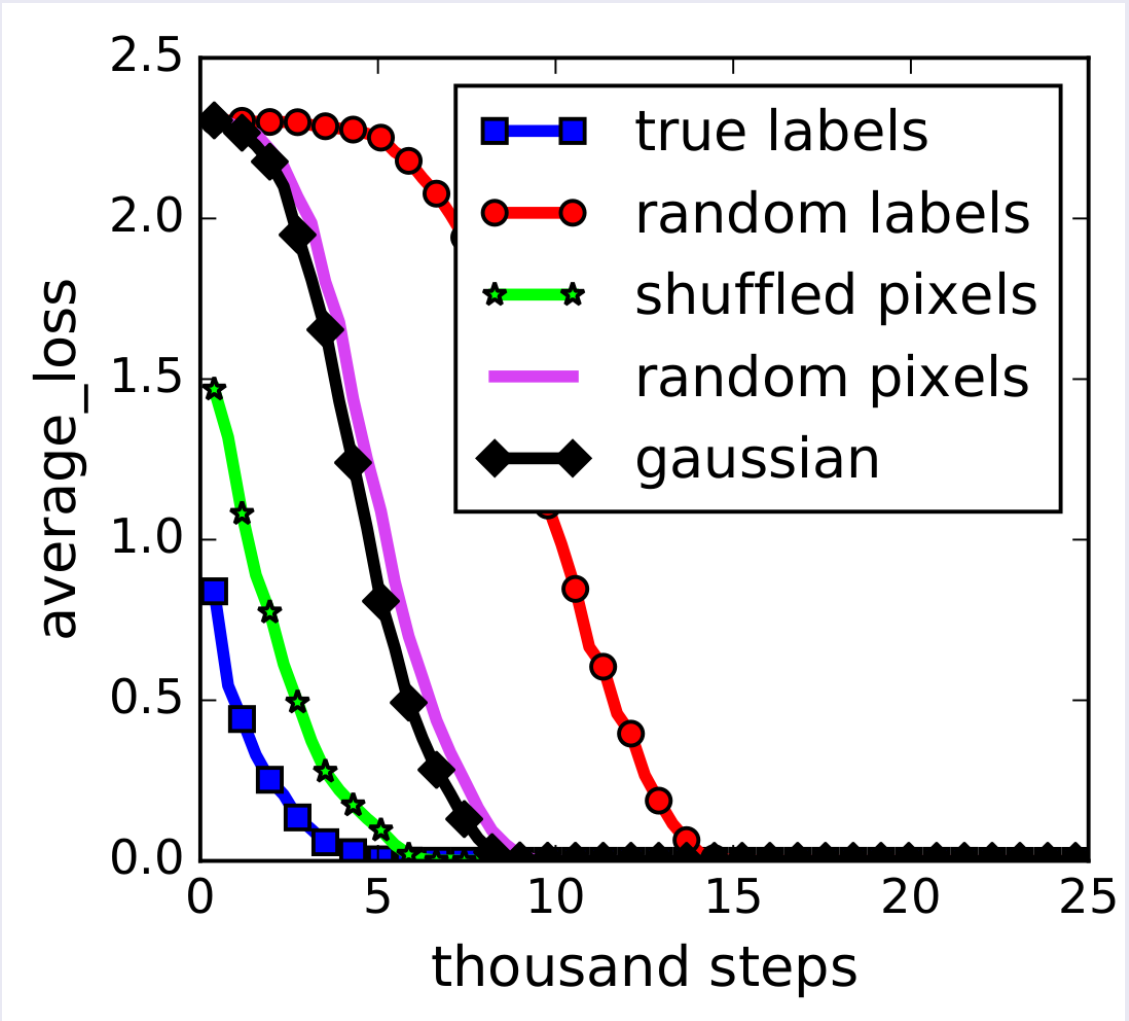
$$f_A(x) := \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

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.)

Explaining Generalization Failures

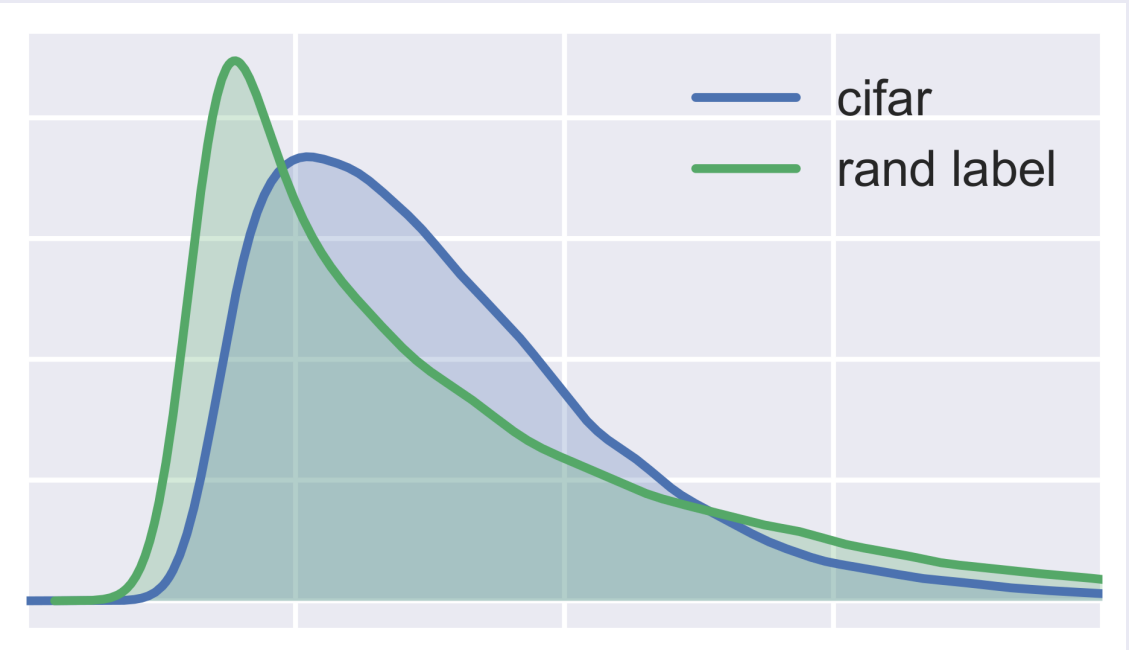
Stochastic Gradient Training Error on CIFAR10



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

Explaining Generalization Failures

Training margins on CIFAR10 with true and random labels

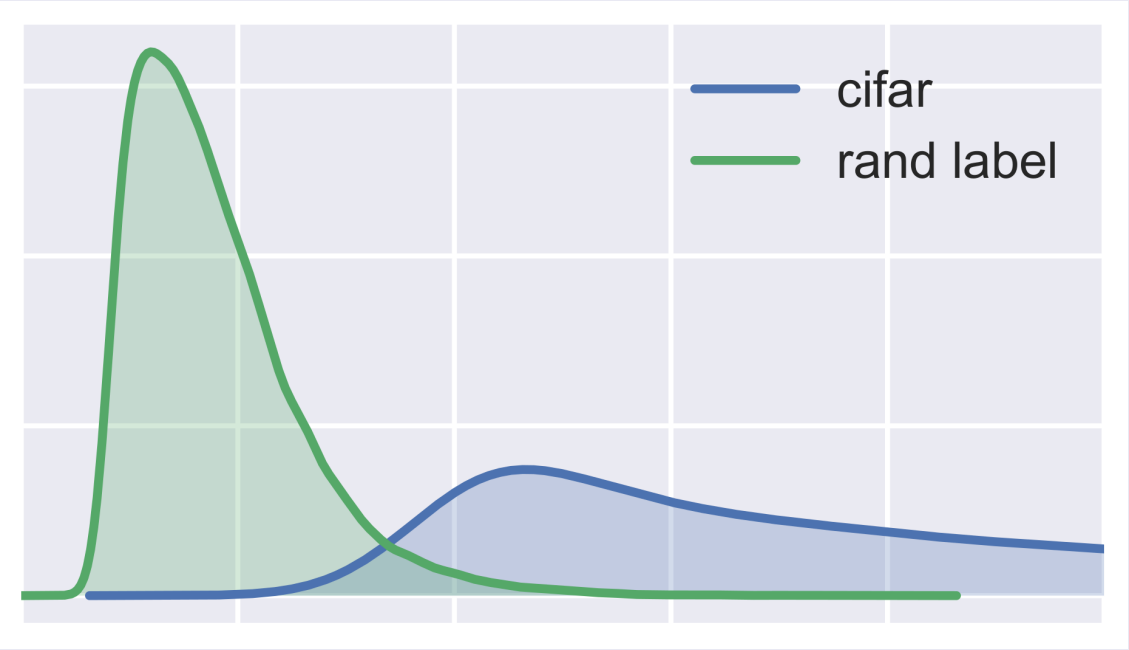


- How does this match the large margin explanation?

Explaining Generalization Failures

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

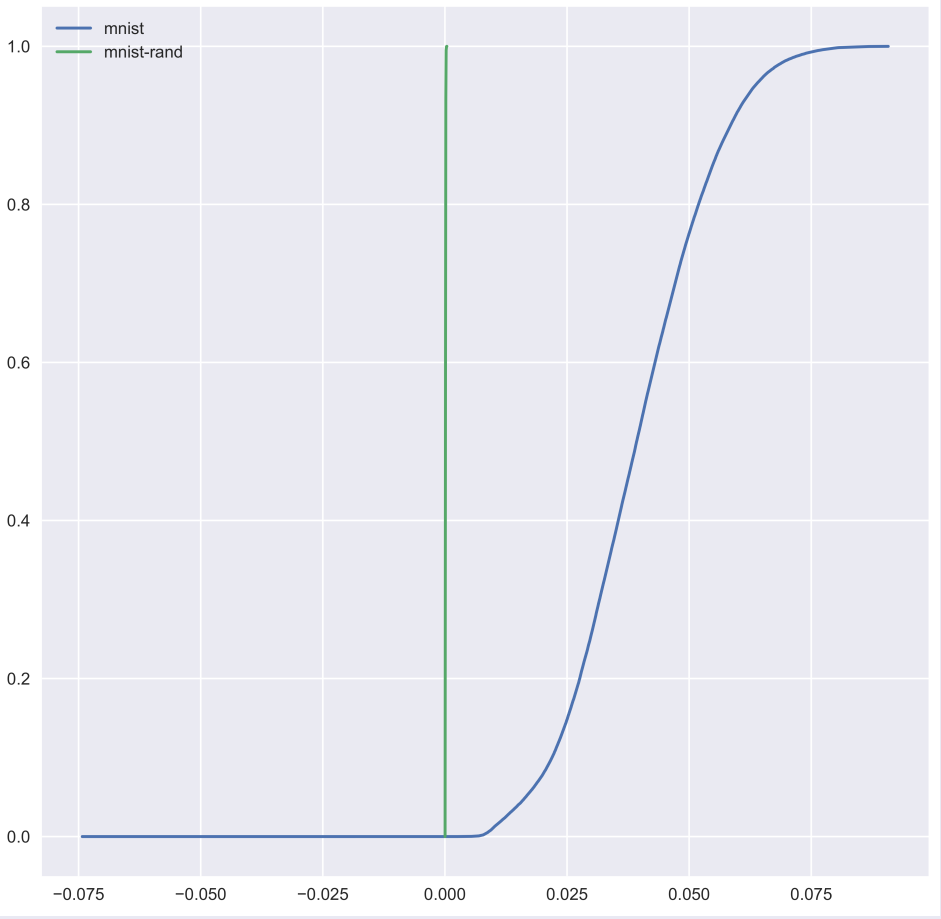
Rescaled margins on CIFAR10



Explaining 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 $R_A \le r$ satisfies

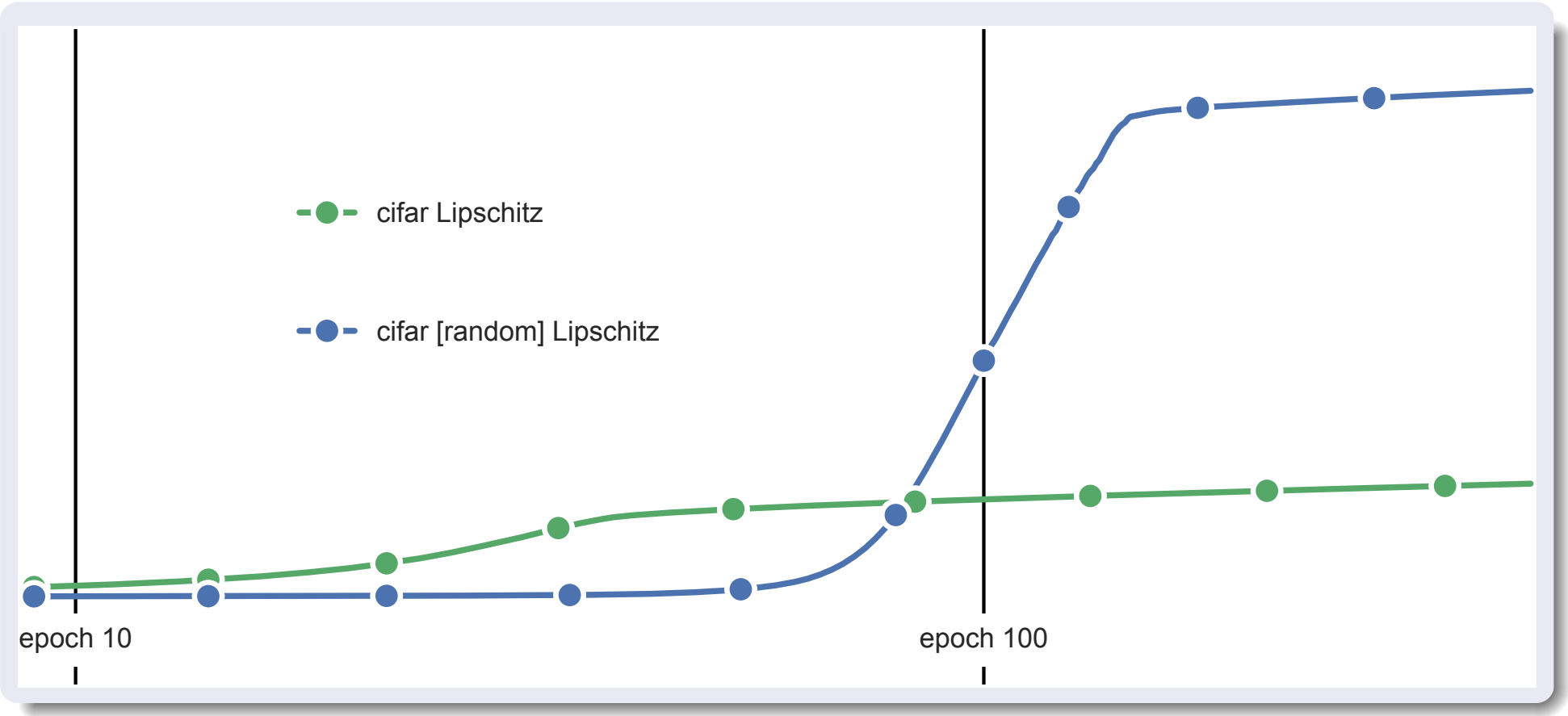
$$\Pr(M(f_A(X), Y) \le 0) \le \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \le \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

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

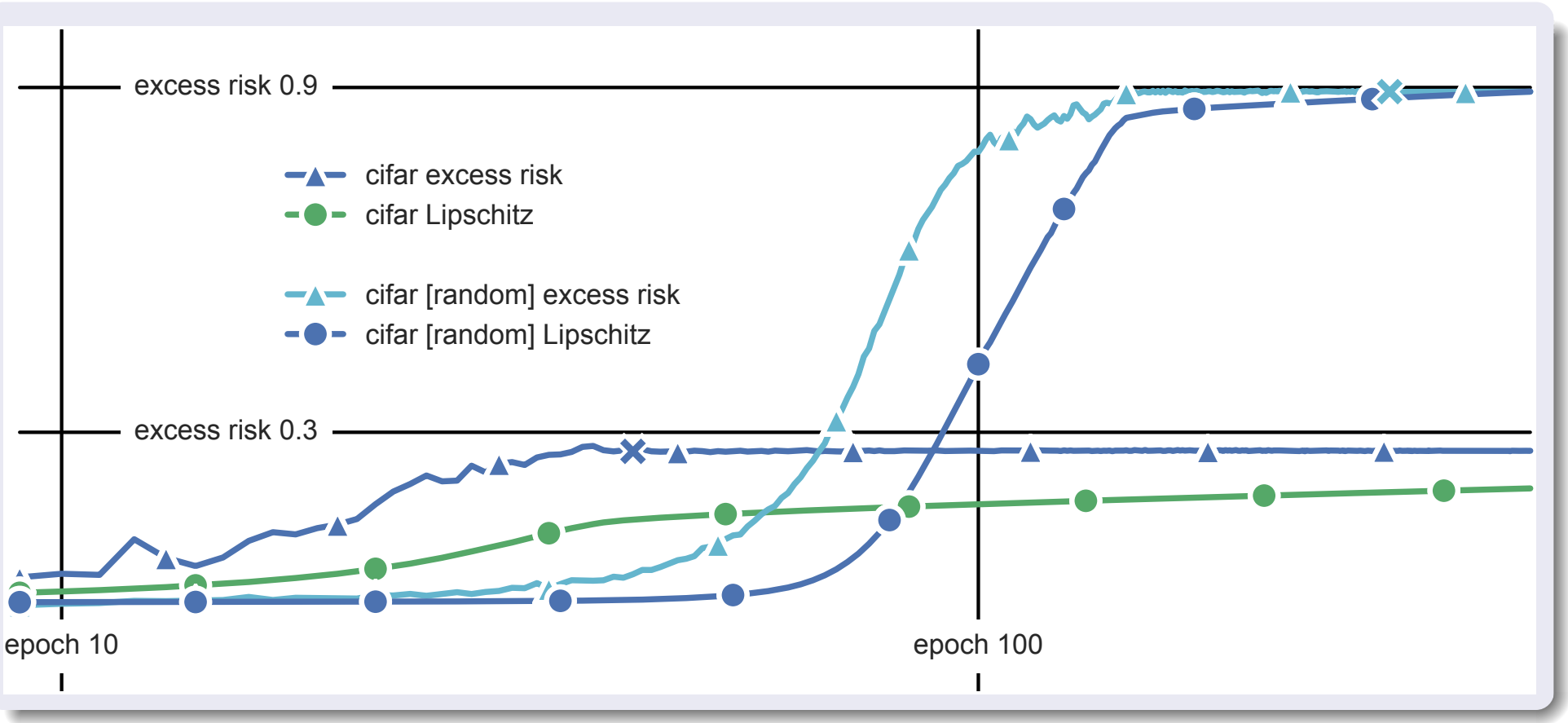
$$f_A(x) := \sigma(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

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

Explaining Generalization Failures



Explaining 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.
- Margin bounds extend to residual networks.

Generalization in Neural Networks

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

Generalization in Neural Networks

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

Generalization in Neural Networks

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

Generalization in Neural Networks

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

Outline

- Deep residual networks
  - Representing with near-identities
    - Deeper networks allow flatter functions at each layer.
  - Global optimality of stationary points
    - With flat functions, stationary points are global minima.
- What determines the statistical complexity of a deep network?
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - Understanding generalization failures