Some Representation, Optimization and Generalization Properties of Deep Networks

Peter Bartlett

UC Berkeley

August 13, 2018

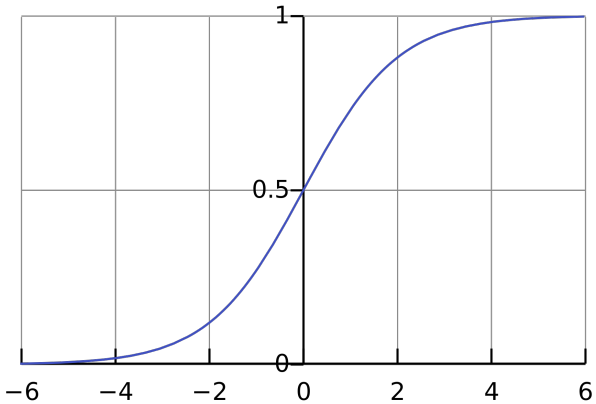
Deep Networks

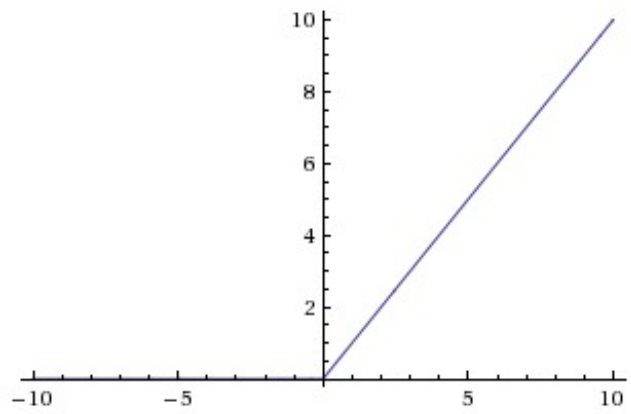
Deep compositions of nonlinear functions

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

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

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)

But...

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

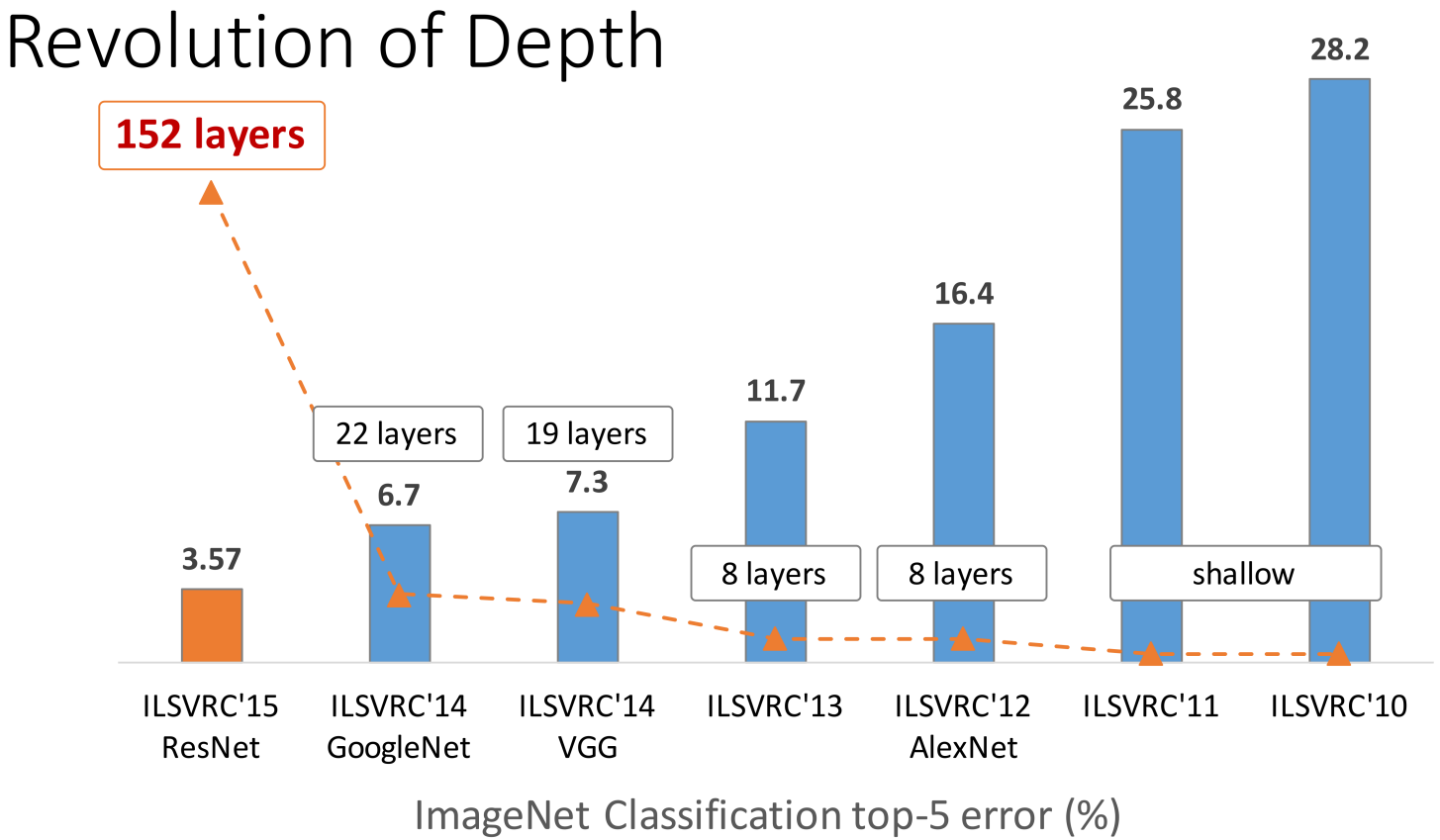
Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points
- Optimization in deep linear residual networks
  - Gradient descent
  - Symmetric maps and positivity
  - Regularized gradient descent and positive maps
- Statistical complexity of deep networks
  - 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
- Optimization in deep linear residual networks
  - Gradient descent
  - Symmetric maps and positivity
  - Regularized gradient descent and positive maps
- Statistical complexity of deep networks
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - Understanding generalization failures

Deeper Networks

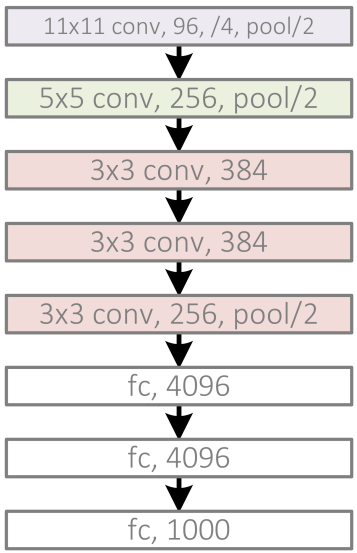


(Deep Residual Networks. Kaiming He. 2016)

Deeper Networks

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

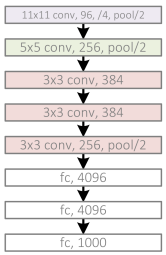


(Deep Residual Networks. Kaiming He. 2016)

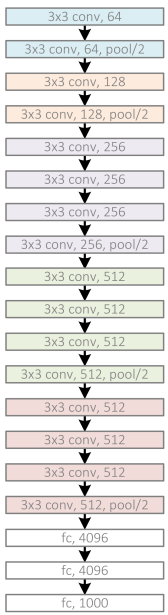
Deeper Networks

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

(Deep Residual Networks. Kaiming He. 2016)

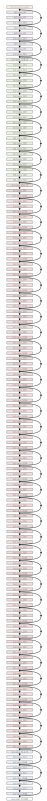
Deeper Networks

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

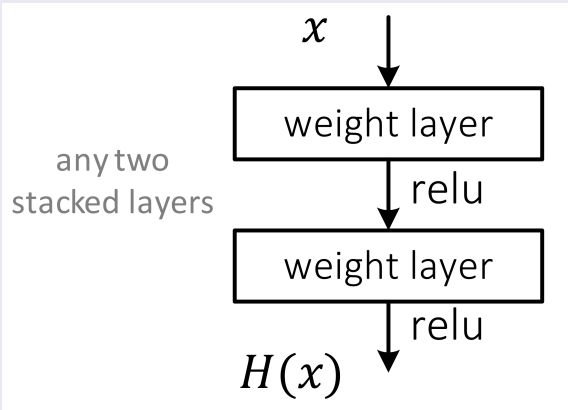
ResNet, 152 layers
(ILSVRC 2015)



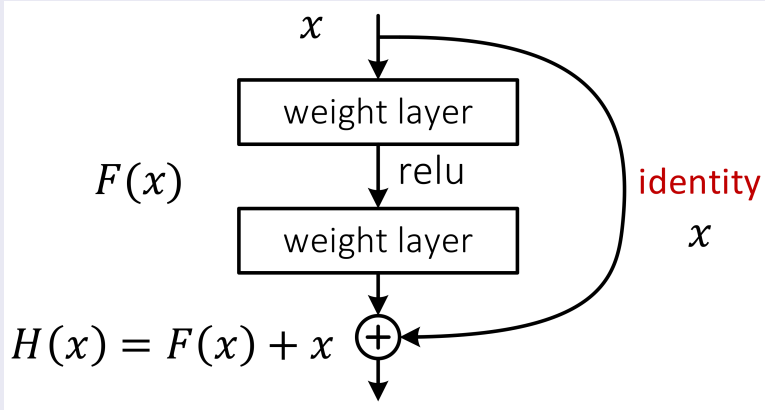
(Deep Residual Networks. Kaiming He. 2016)

Deep Residual Networks

Deep network component



Residual network component



(Deep Residual Networks. Kaiming He. 2016)

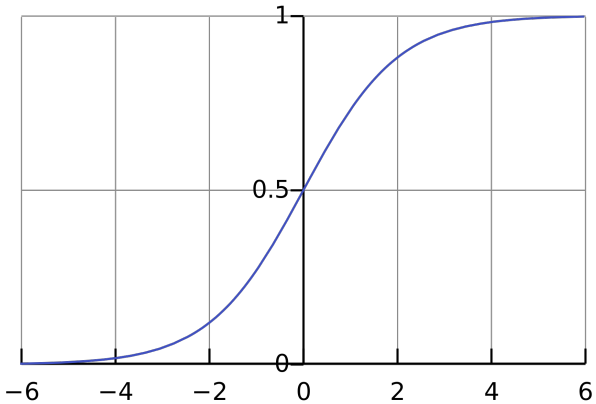
Deep Networks

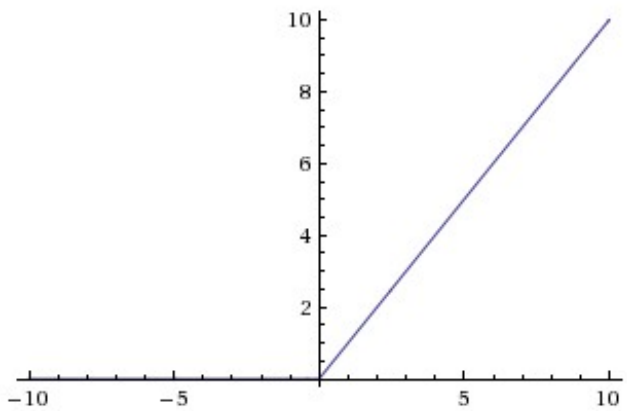
Deep compositions of nonlinear functions

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

e.g., $h_i : x \mapsto x + A_i \sigma(B_i x)$ $\qquad$ $h_i : x \mapsto x + A_i r(B_i x)$

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

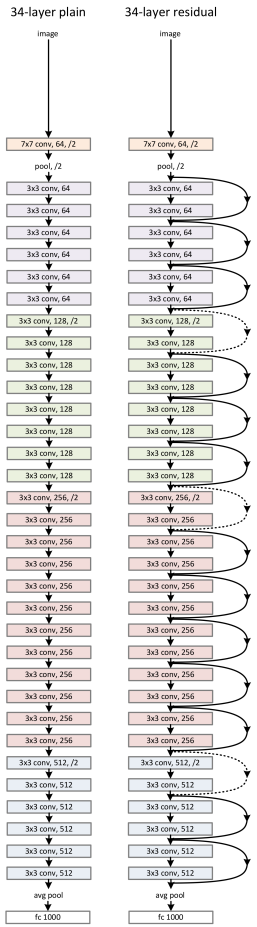




Deep Residual Networks

Advantages

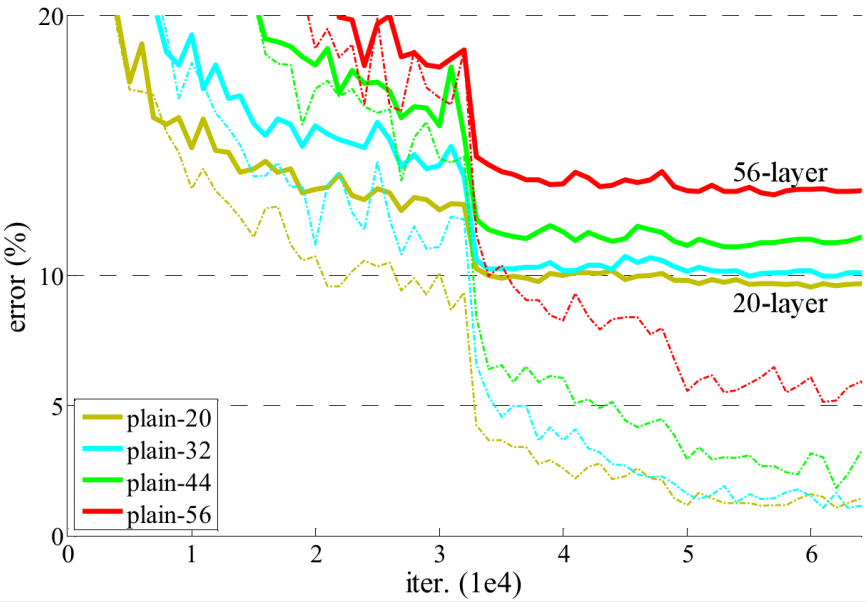
- With zero-valued parameters, the network computes the identity.
- Identity connections provide useful feedback throughout the network.

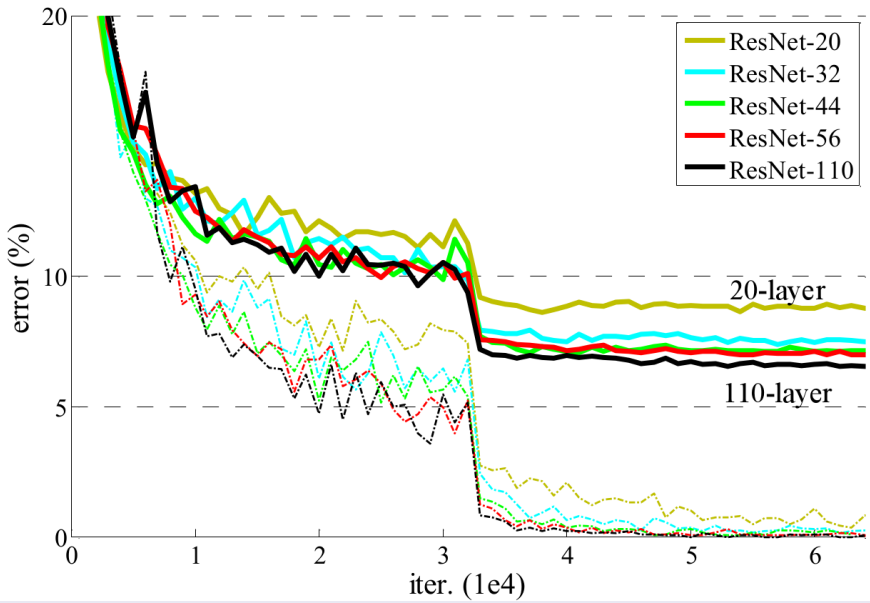


(Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016)

Deep Residual Networks

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





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

Large improvements over plain nets (e.g., ImageNet Large Scale Visual Recognition Challenge, Common Objects in Context Detection Challenge).

Related work

- Deep linear compositions: $(I + A_m) \cdots (I + A_1)$. (Hardt & Ma, 2016)
- Residual nets: $a^\top (x + Bf_\theta(x))$. (Shamir, 2018)
- Empirical risk landscape for $n > p$. e.g.,(Soudry and Carmon, 2016), (Kawaguchi, 2016)
- Optimization landscape and gradient descent

(Du & Lee, 2018), (Du, Lee, Tian, Poczos, Singh, 2017), (Soltanolkotabi, Javanmard, Lee, 2017)

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,

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

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
- Optimization in deep linear residual networks
- Statistical complexity of deep networks

Steve Evans
Berkeley, Stat/Math

Phil Long
Google

arXiv:1804.05012

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,

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

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

$$h_i(x) = x + \underbrace{A_i \sigma(B_i x)}.$$

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 h 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:
  $\|f\| := \sup \left\{ \frac{\|f(x)\|}{\|x\|} : \|x\| > 0 \right\}$.

Outline

- Deep residual networks
  - Representing with near-identities
  - **Global optimality of stationary points**
- Optimization in deep linear residual networks
- Statistical complexity of deep networks

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

- 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

- 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,

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

- 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?
  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.
- Regularized gradient methods for near-identity maps?

Outline

- Deep residual networks
- **Optimization in deep linear residual networks**
  - Gradient descent
  - Symmetric maps and positivity
  - Regularized gradient descent and positive maps
- Statistical complexity of deep networks

Dave Helmbold
UCSC

Phil Long
Google

arXiv:1802.06093

Optimization in deep linear residual networks

Linear networks

- Consider $f_\Theta : \mathbb{R}^d \to \mathbb{R}^d$ defined by $f_\Theta(x) = \Theta_L \cdots \Theta_1 x$.
- Suppose $(x, y) \sim P$, and consider using gradient methods to choose Θ to minimize $\ell(\Theta) = \frac{1}{2}\mathbb{E}\|f_\Theta(x) - y\|^2$.

Assumptions

1. $\mathbb{E}xx^\top = I$
2. $y = \Phi x$ for some matrix Φ (wlog, because of projection theorem)

Optimization in deep linear residual networks

Recall $f_\Theta(x) = \Theta_L \cdots \Theta_1 x = \Theta_{1:L} x$,
where we use the notation $\Theta_{i:j} = \Theta_j \Theta_{j-1} \cdots \Theta_i$.

Gradient descent

$$\Theta^{(0)} = \left(\Theta_1^{(0)}, \Theta_2^{(0)}, \ldots, \Theta_L^{(0)}\right) := (I, I, \ldots, I)$$

$$\Theta_i^{(t+1)} := \Theta_i^{(t)} - \eta(\Theta_{i+1:L})^\top \left(\Theta_{1:L}^{(t)} - \Phi\right)(\Theta_{1:i-1}^{(t)})^\top,$$

where η is a step-size.

Gradient descent in deep linear residual networks

Theorem

There is a positive constant c_0 and polynomials p_1 and p_2 such that if $\ell(\Theta^{(0)}) \leq c_0$ and $\eta \leq 1/p_1(d, L)$, after $p_2(d, L, 1/\eta) \log(1/\epsilon)$ iterations, gradient descent achieves $\ell(\Theta^{(t)}) \leq \epsilon$.

Outline

- Deep residual networks
- Optimization in deep linear residual networks
  - Gradient descent
  - **Symmetric maps and positivity**
  - Regularized gradient descent and positive maps
- Statistical complexity of deep networks

Optimization in deep linear residual networks

Definition (γ-positive matrix)

A matrix A is *γ-positive* for $\gamma > 0$ if, for all unit length u, we have $u^\top Au > \gamma$.

Theorem

Suppose that the least squares map Φ is symmetric.

(a) There is an absolute positive constant c_3 such that
if Φ is γ-positive $(0 < \gamma < 1)$, $L \geq c_3 \ln(||\Phi||_2/\gamma)$, and $\eta \leq \frac{1}{L(1+||\Phi||_2^2)}$,
after $t = \mathrm{poly}(L, ||\Phi||_2/\gamma, 1/\eta)\log(d/\epsilon)$ iterations,
gradient descent achieves $\ell(f_{\Theta(t)}) \leq \epsilon$.

(b) If Φ has a negative eigenvalue $-\lambda$ and L is even, then gradient descent satisfies $\ell(\Theta^{(t)}) \geq \lambda^2/2$ (as does any penalty-regularized version of gradient descent).

Outline

- Deep residual networks
- Optimization in deep linear residual networks
  - Gradient descent
  - Symmetric maps and positivity
  - **Regularized gradient descent and positive maps**
- Statistical complexity of deep networks

Positive (not necessarily symmetric) linear functions

Theorem

For any γ-positive Φ, there is an algorithm (*power projection*) that, after $t = \mathrm{poly}(d, ||\Phi||_F, \frac{1}{\gamma}) \log(1/\epsilon)$ iterations, produces $\Theta^{(t)}$ with $\ell(\Theta^{(t)}) \leq \epsilon$.

Power projection algorithm idea

1. Take a gradient step for each Θ_i.
2. Project $\Theta_{1:L}$ onto the set of γ-positive linear maps.
3. Set $\Theta_1^{(t+1)}, \ldots, \Theta_L^{(t+1)}$ as the *balanced factorization* of $\Theta_{1:L}$.

Positive (not necessarily symmetric) linear functions

Balanced factorization

We can write any matrix A, with singular values $\sigma_1, \ldots, \sigma_d$, as $A = A_L \cdots A_1$, where the singular values of each A_i are $\sigma_1^{1/L}, \ldots, \sigma_d^{1/L}$.

(Idea: Write the polar decomposition $A = RP$ (i.e., R unitary, P psd); set $A_i = R^{1/L} P_i$, with $P_i = R^{(i-1)/L} P^{1/L} R^{-(i-1)/L}$.)

Optimization in deep linear residual networks

- Gradient descent
  - converges if $\ell(0)$ sufficiently small,
  - converges for a positive symmetric map,
  - cannot converge for a symmetric map with a negative eigenvalue.
- Regularized gradient descent converges for a positive map.
- Convergence is linear in all cases.
- Deep nonlinear residual networks?

Outline

- Deep residual networks
- Optimization in deep linear residual networks
- **Statistical complexity of deep networks**
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - Understanding generalization failures

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

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

- 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
  - 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:

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)
4. Sigmoid: $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(p^2k^2)$.
   (Karpinsky and MacIntyre, 1994)

Outline

- Deep residual networks
- Optimization in deep linear residual networks
- Statistical complexity of deep networks
  - VC theory: Number of parameters
  - **Margins analysis: Size of parameters**
  - Understanding generalization failures

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

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

- Measure the size of functions computed by a deep network 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 with $R_A \le r$ satisfies

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

Definitions

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

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

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

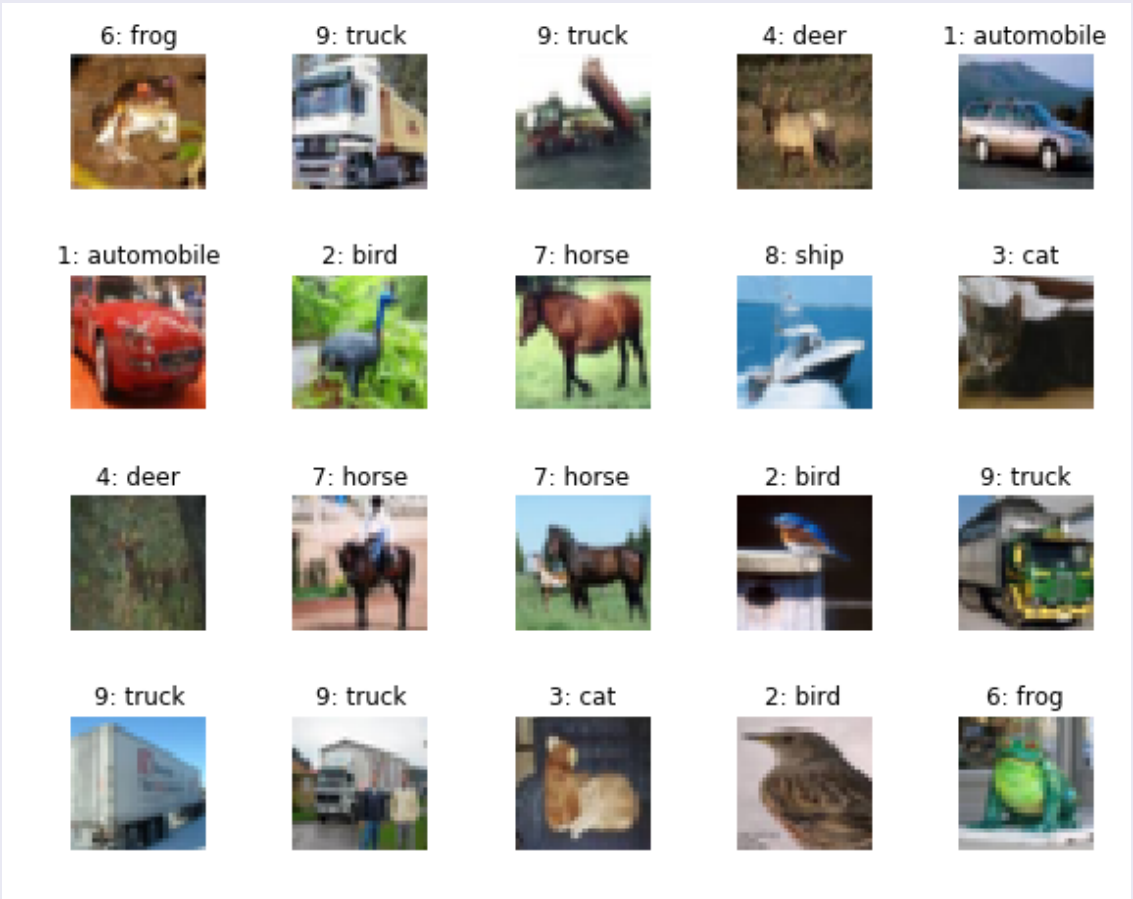
(Assume σ_i is 1-Lipschitz, inputs normalized.)

Outline

- Deep residual networks
- Optimization in deep linear residual networks
- Statistical complexity of deep networks
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - **Understanding generalization failures**

Understanding Generalization Failures

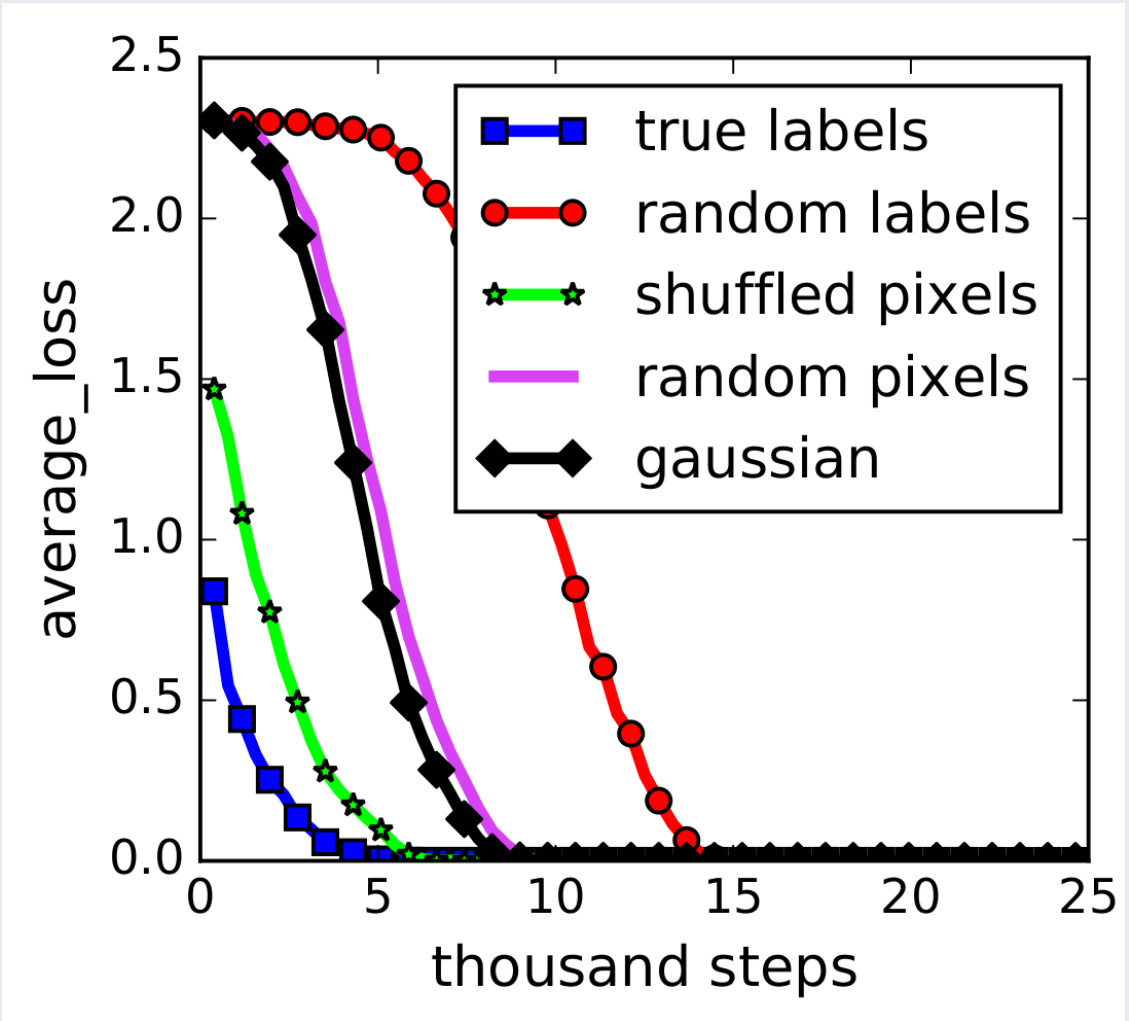
CIFAR10



http://corochann.com/

Understanding Generalization Failures

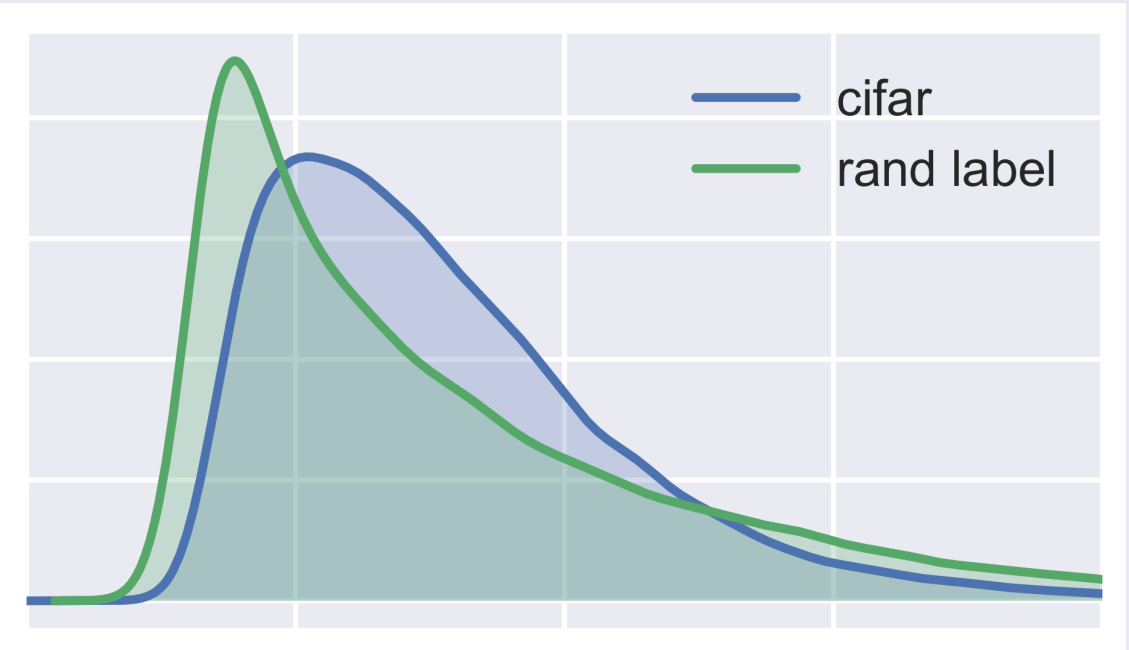
Stochastic Gradient Training Error on CIFAR10



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

Understanding Generalization Failures

Training margins on CIFAR10 with true and random labels

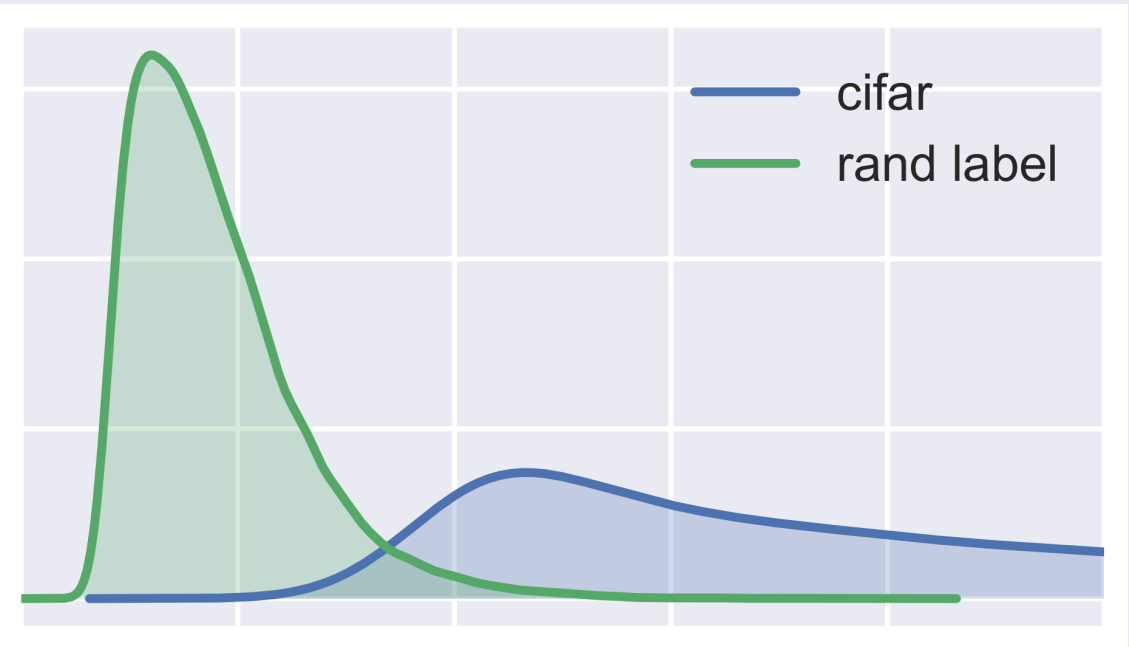


- How does this match the large margin explanation?

Understanding Generalization Failures

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

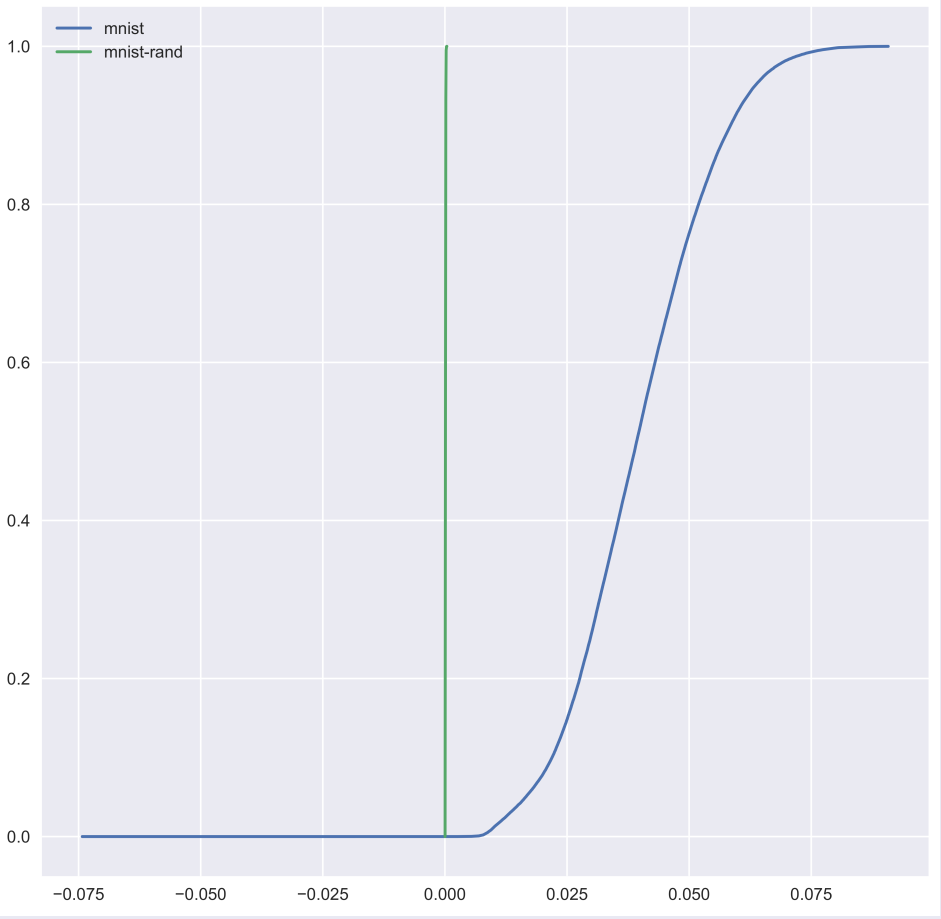
Rescaled margins on CIFAR10



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 $R_A \le r$ satisfies

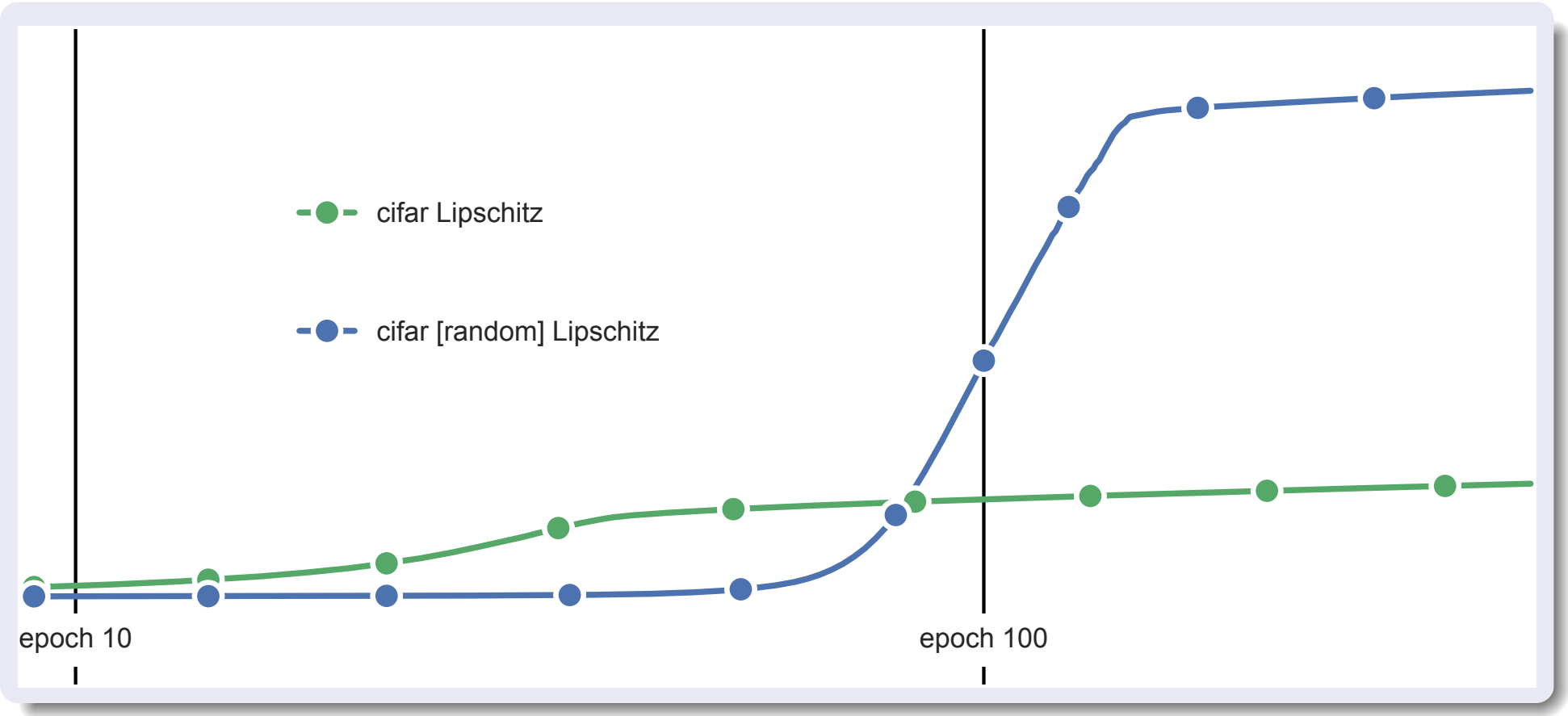
$$\Pr(M(f_A(X), Y) \le 0) \le \frac{1}{n}\sum_{i=1}^{n} 1[M(f_A(X_i), Y_i) \le \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

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

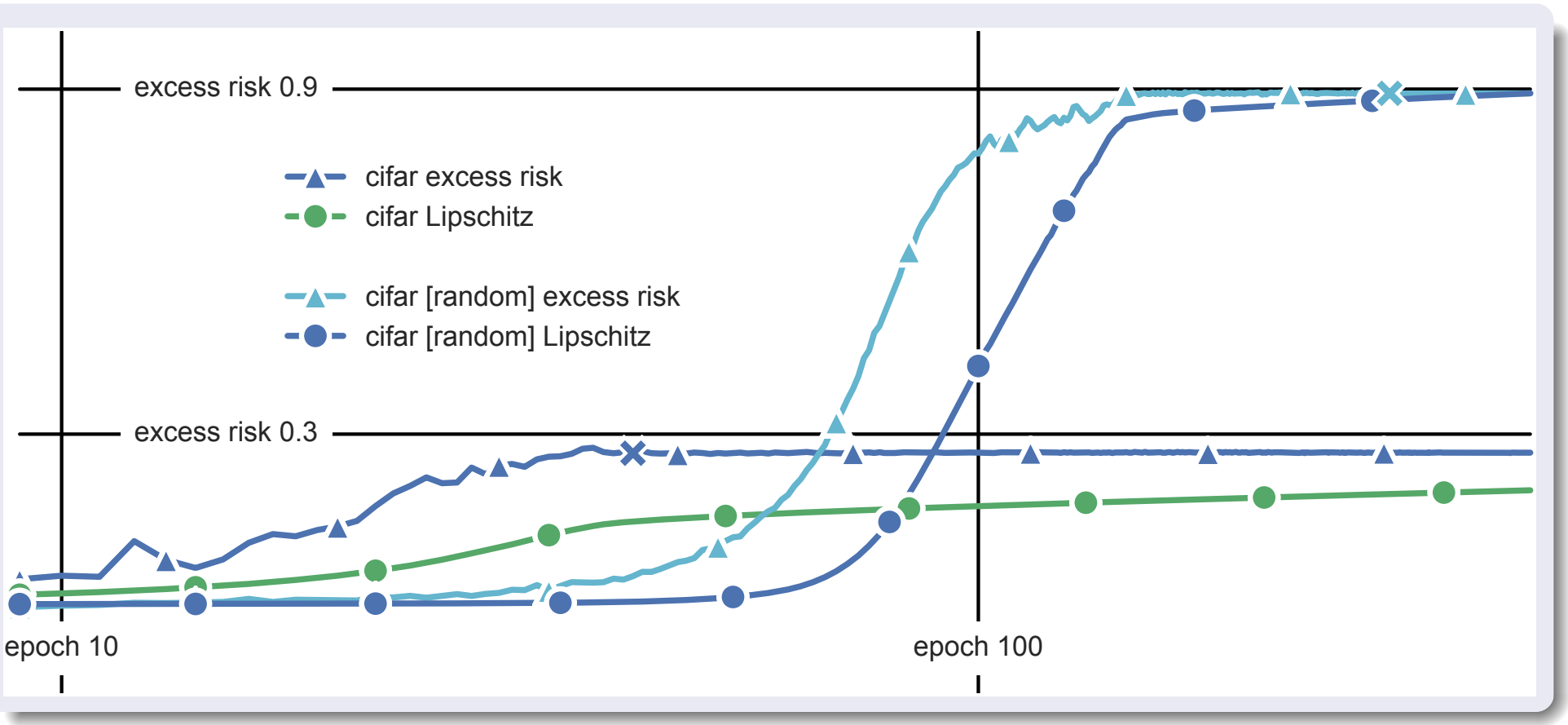
$$f_A(x) := \sigma(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 x) \cdots)).$$

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

Understanding Generalization Failures



Understanding Generalization Failures



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.
- Regularization and optimization: explicit control of operator norms?

Outline

- Deep residual networks
  - Representing with near-identities
  - Global optimality of stationary points
- Optimization in deep linear residual networks
  - Gradient descent
  - Symmetric maps and positivity
  - Regularized gradient descent and positive maps
- Statistical complexity of deep networks
  - VC theory: Number of parameters
  - Margins analysis: Size of parameters
  - Understanding generalization failures