Optimization Properties of Deep Residual Networks

Peter Bartlett

UC Berkeley

May 14, 2018

Deep Networks

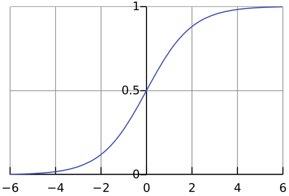
Deep compositions of nonlinear functions

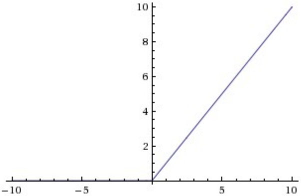
$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g.,

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

$$h_i : x \mapsto r(W_i x) \qquad r(v)_i = \max\{0, v_i\}$$





Deep Networks

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

Rich non-parametric family
: Depth provides parsimonious represions.
: 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.

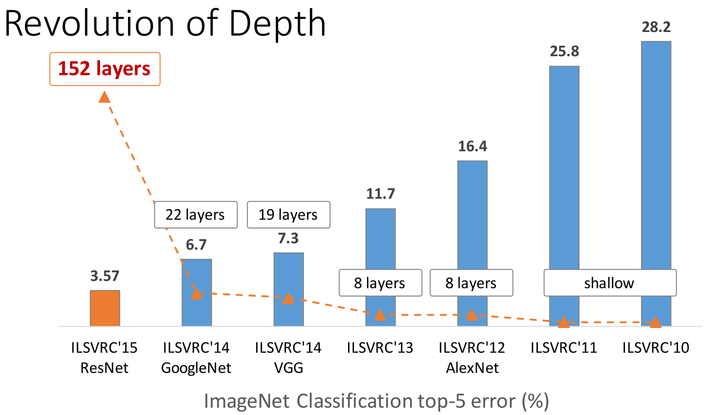
Outline

- 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

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

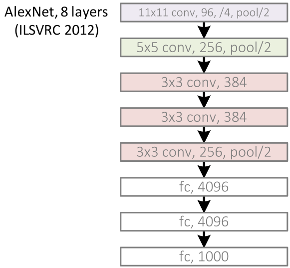
Deeper Networks



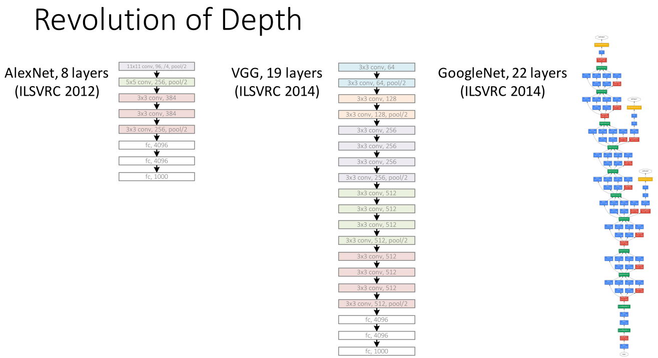
(Deep Residual Networks. Kaiming He. 2016)

Deeper Networks

Revolution of Depth

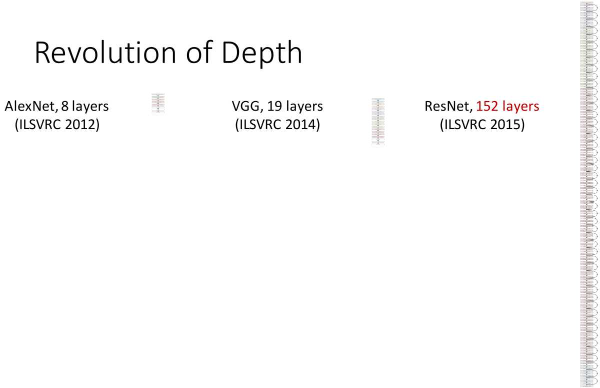


(Deep Residual Networks. Kaiming He. 2016)



(Deep Residual Networks. Kaiming He. 2016)

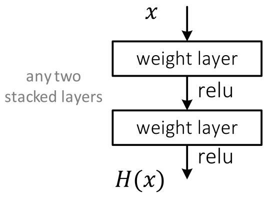
Deeper Networks



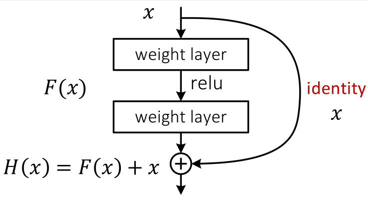
(Deep Residual Networks. Kaiming He. 2016)

Deep Residual Networks

Deep network component

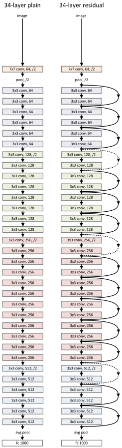


Residual network component



(Deep Residual Networks. Kaiming He. 2016)

Deep Residual Networks



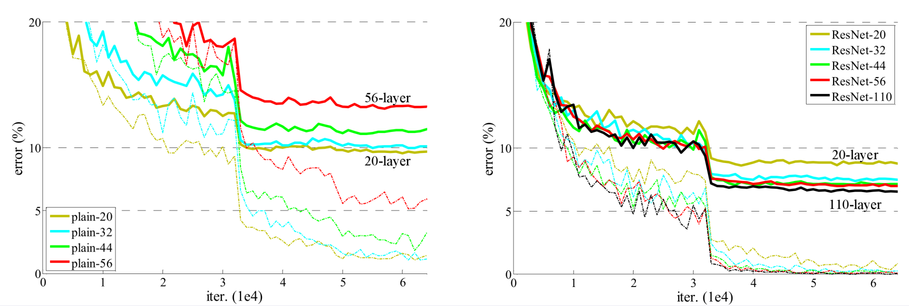
Advantages

- With zero weights, the network computes the identity.
- 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).

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

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

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

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

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**
- Optimization in deep linear residual networks

Stationary points

Result

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

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

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

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.

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

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

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

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

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

- 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

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

why wlog?

Define Φ as the minimizer of $\mathbb{E}\|\Phi x - y\|^2$ (the *least squares map*).
Then the projection theorem implies

$$\begin{aligned}\mathbb{E}\|\Theta x - y\|^2 &= \mathbb{E}\|\Theta x - \Phi x\|^2 + 2\mathbb{E}(\Theta x - \Phi x)^\top(\Phi x - y) + \mathbb{E}\|\Phi x - y\|^2 \\ &= \mathbb{E}\|\Theta x - \Phi x\|^2 + \mathbb{E}\|\Phi x - y\|^2,\end{aligned}$$

so wlog we can assume $y = \Phi x$ and define, for linear f_Θ,

$$\ell(\Theta) = \frac{1}{2}\mathbb{E}\|f_\Theta(x) - \Phi x\|^2.$$

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 $\ell(\Theta^{(t)}) \leq \epsilon$.

Gradient descent: proof idea

Lemma [Hardt and Ma] (Gradient is big when loss is big)

If, for all layers i, $\sigma_{\min}(\Theta_i) \geq 1 - a$, then $||\nabla_\Theta \ell(\Theta)||^2 \geq 4\ell(\Theta)L(1-a)^{2L}$.

Lemma (Hessian is small for near-identities)

For Θ with $||\Theta_i||_2 \leq 1 + z$ for all i,

$$\|\nabla^2_\Theta \ell(\Theta)\|_F \leq 3Ld^5(1+z)^{2L}.$$

Lemma (Stay close to the identity)

$$\mathcal{R}(t+1) \leq \mathcal{R}(t) + \eta(1+\mathcal{R}(t))^L \sqrt{2\ell(t)},$$

where $\mathcal{R}(t) := \max_i ||\Theta_i^{(t)} - I||_2$ and $\ell(t) := \frac{1}{2}||\Theta_{1:L}^{(t)} - \Phi||_F^2$.

Then for sufficiently small step-size η, the gradient update ensures that $\ell(t)$ decreases exponentially.

Outline

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

Optimization in deep linear residual networks

Definition (positive margin matrix)

A matrix A has *margin* $\gamma > 0$ if, for all unit length u, we have $u^\top Au > \gamma$.

Theorem

Suppose Φ is symmetric.

(a) There is an absolute positive constant c_3 such that if Φ has margin $0 < \gamma < 1$, $L \geq c_3 \ln(||\Phi||_2/\gamma)$, 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).

Symmetric linear functions

Proof idea

(a) A set of symmetric matrices $\mathcal{A}$ is *commuting normal* if there is a single unitary matrix U such that for all $A \in \mathcal{A}$, $U^\top AU$ is diagonal.
Clearly, $\{\Phi, \Theta_1^{(0)}, \Theta_2^{(0)}, \ldots, \Theta_L^{(0)}\} = \{\Phi, I\}$ is commuting normal.
The gradient update keeps $\bigcup_{i,t}\{\Phi, \Theta_i^{(t)}\}$ commuting normal.
So the dynamics decomposes:

$$\hat{\lambda}^{(t+1)} = \hat{\lambda}^{(t)} + \eta(\hat{\lambda}^{(t)})^{L-1}(\lambda^L - (\hat{\lambda}^{(t)})^L).$$

(b) The eigenvalues stay positive.

Outline

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

Positive (not necessarily symmetric) linear functions

Theorem

For any Φ with margin γ, 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 linear maps with margin γ.
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
  - 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