Double Descent / Implicit Regularization
+ Neural Tangent Kernels

CPSC 532D: Modern Statistical Learning Theory
28 November 2022
cs.ubc.ca/~dsuth/532D/22w1/
degree 1

degree 3

Nakkiran et al. blog post's companion notebook
Important: this is the *minimum norm* solution, with the particular Legendre basis!
Important: this is the minimum norm solution, with the particular Legendre basis!
\( \mathbf{w}(t) = \frac{1}{2} \| \mathbf{x} \mathbf{w} - y \|_2^2 \quad \nabla f(w) = \mathbf{x}^\top (\mathbf{x} \mathbf{w} - y) \)

\( w^{(t)} = 0 \)

\[
\begin{align*}
w^{(t+1)}(w) &= \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}) \\
&= (\mathbf{I} - \eta \mathbf{x}^\top \mathbf{x}) \mathbf{w}^{(t)} + \eta \mathbf{x} \mathbf{y} \\
&= \eta \sum_{k=0}^{t} (\mathbf{I} - \eta \mathbf{e}^2 \mathbf{v}^\top) \mathbf{e} v u^\top \\
&= \eta \sum_{k=0}^{t} \mathbf{v} (\mathbf{I} - \eta \mathbf{e}^2 \mathbf{v}^\top)^k \mathbf{u} \mathbf{e}^\top \\
&= \eta \mathbf{V} \left( \mathbf{I} - (\mathbf{I} - \eta \mathbf{e}^2 \mathbf{V}^\top)^{-1} \mathbf{V} \mathbf{E} \mathbf{U}^\top \right) \\
&= \eta \mathbf{V} (\mathbf{I} - (\mathbf{I} - \eta \mathbf{e}^2 \mathbf{V}^\top)^{-1} \mathbf{V} \mathbf{E} \mathbf{U}^\top) \mathbf{E} \mathbf{V}^\top \\
&= \eta \mathbf{V} (\mathbf{I} - (\mathbf{I} - \eta \mathbf{e}^2 \mathbf{V}^\top)^{-1} \mathbf{V} \mathbf{E} \mathbf{U}^\top) \mathbf{E} \mathbf{V}^\top \\
&= \mathbf{x} \mathbf{y}^\top \mathbf{X} \mathbf{E} \mathbf{V}^\top \\
&= \mathbf{x} \mathbf{y}^\top \mathbf{X} \mathbf{E} \mathbf{V}^\top \\
&= \eta \mathbf{V} \mathbf{E} \mathbf{U}^\top \\
&= \eta \mathbf{V} \mathbf{E} \mathbf{U}^\top \\
&= \mathbf{x} \mathbf{y}^\top \mathbf{X} \mathbf{E} \mathbf{V}^\top \\
&= \mathbf{x} \mathbf{y}^\top \mathbf{X} \mathbf{E} \mathbf{V}^\top \\
&= \mathbf{x} \mathbf{y}^\top \mathbf{X} \mathbf{E} \mathbf{V}^\top \\
&= \mathbf{x} \mathbf{y}^\top \mathbf{X} \mathbf{E} \mathbf{V}^\top
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$

\[
\begin{align*}
\text{assume } xw &= y \\
\text{x} (x^\tau y + q) &= y \\
\text{u} \leq v^\tau (v \leq u^\tau y + q) &= y \\
uu^\tau y + uu^\tau q &= y \\
\text{u} uu^\tau xw &= uu^\tau y \\
xw &= y \\
\therefore y &= uu^\tau y
\end{align*}
\]

If rank$(X) = n$

\[
\begin{align*}
xq &= 0 = u \leq v^\tau q = 0 \\
&\Rightarrow \quad v^\tau q = 0
\end{align*}
\]

\[
\begin{align*}
\|v \leq u^\tau y + q\|^2 &= y^\tau u \leq u^\tau y + y^\tau u \leq v^\tau q + \|q\|^2 \\
&= 0
\end{align*}
\]
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$
- “Ridgeless” regression: $\lim_{\lambda \to 0} (X^\top X + \eta \lambda I)^{-1} X^\top y = X^\dagger y = \lim_{\lambda \to 0} X^\top (XX^\top + \eta \lambda I)^{-1} y$
Implicit regularization of gradient descent

• We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$

• “Ridgeless” regression: $\lim_{\lambda \to 0} (X^T X + \lambda I)^{-1} X^T y = X^\dagger y = \lim_{\lambda \to 0} X^T (X X^T + \lambda I)^{-1} y$

• If we track $w_0^{(t)} \neq 0$ in same analysis, get $w_0^{(\infty)} = (I - V V^T) w_0^{(1)} + X^\dagger y$ (proof)

$w_0^{(\infty)} = \arg\min_{w} \|x - w\|^2$
Implicit regularization of gradient descent

• We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < Cn / \sigma_{\max}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$.

  • “Ridgeless” regression: $\lim_{\lambda \to 0} (X^\top X + \lambda I)^{-1}X^\top y = X^\dagger y = \lim_{\lambda \to 0} X^\top (XX^\top + \lambda I)^{-1}y$

• If we track $w_0 \neq 0$ in same analysis, get $w_\infty = (I - VV^\top)w_0 + X^\dagger y$ (proof)

• So, the 1,000-degree polynomial picture is what (small-LR) GD would give
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < \frac{\sqrt{n}}{\sigma_{\max}(X)^2}$, converges to the minimum-norm interpolator $X^\dagger y$

  - “Ridgeless” regression: $\lim_{\lambda \to 0} (X^T X + \lambda I)^{-1} X^T y = X^\dagger y = \lim_{\lambda \to 0} X^T (XX^T + \lambda I)^{-1} y$

- If we track $w_0 \neq 0$ in same analysis, get $w_\infty = (I - VV^T)w_0 + X^\dagger y$ (proof)

- So, the 1,000-degree polynomial picture is what (small-LR) GD would give
- Does this same idea hold for other losses / models? Not necessarily.
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\max}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$
  - “Ridgeless” regression: $\lim_{\lambda \to 0} (X^T X + \lambda I)^{-1} X^T y = X^\dagger y = \lim_{\lambda \to 0} X^T (XX^T + \lambda I)^{-1} y$
  - If we track $w_0 \neq 0$ in same analysis, get $w_{\infty} = (I - VV^T)w_0 + X^\dagger y$ (proof)

- So, the 1,000-degree polynomial picture is what (small-LR) GD would give
- Does this same idea hold for other losses / models? **Not necessarily.**
  - Logistic regression:
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$

  - “Ridgeless” regression: $\lim_{\lambda \to 0} (X^\top X + \lambda I)^{-1}X^\top y = X^\dagger y = \lim_{\lambda \to 0} X^\top (XX^\top + \lambda I)^{-1}y$

- If we track $w_0 \neq 0$ in same analysis, get $w_\infty = (I - VV^\top)w_0 + X^\dagger y$ (proof)

- So, the 1,000-degree polynomial picture is what (small-LR) GD would give
- Does this same idea hold for other losses / models? **Not necessarily.**
  - Logistic regression:
    - Separable: norm diverges in direction of max-margin separator (Soudry et al.)
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$

  - “Ridgeless” regression: \( \lim_{\lambda \to 0} (X^\top X + \lambda I)^{-1}X^\top y = X^\dagger y = \lim_{\lambda \to 0} X^\top (XX^\top + \lambda I)^{-1}y \)

- If we track $w_0 \neq 0$ in same analysis, get $w_\infty = (I - VV^\top)w_0 + X^\dagger y$ (proof)

- So, the 1,000-degree polynomial picture is what (small-LR) GD would give

- Does this same idea hold for other losses / models? **Not necessarily.**
  - Logistic regression:
    - Separable: norm diverges in direction of max-margin separator (Soudry et al.)
    - Non-separable: biased towards max-margin, but complicated (Ji/Telgarsky)
Implicit regularization of gradient descent

• We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$
  - “Ridgeless” regression: $\lim_{\lambda \to 0} (X^T X + \lambda I)^{-1} X^T y = X^\dagger y = \lim_{\lambda \to 0} X^T (XX^T + \lambda I)^{-1} y$
  - If we track $w_0 \neq 0$ in same analysis, get $w_\infty = (I - VV^T)w_0 + X^\dagger y$ (proof)

• So, the 1,000-degree polynomial picture is what (small-LR) GD would give
• Does this same idea hold for other losses / models? Not necessarily.
  • Logistic regression:
    - Separable: norm diverges in direction of max-margin separator (Soudry et al.)
    - Non-separable: biased towards max-margin, but complicated (Ji/Telgarsky)
  • Also see Telgarsky notes section 10
Implicit regularization of gradient descent

- We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$
  - “Ridgeless” regression: $\lim_{\lambda \to 0} (X^\top X + \lambda I)^{-1}X^\top y = X^\dagger y = \lim_{\lambda \to 0} X^\top (XX^\top + \lambda I)^{-1}y$
  - If we track $\mathbf{w}_0 \neq 0$ in same analysis, get $\mathbf{w}_\infty = (I - VV^\top)\mathbf{w}_0 + X^\dagger y$ (proof)

- So, the 1,000-degree polynomial picture is what (small-LR) GD would give
- Does this same idea hold for other losses / models? **Not necessarily.**
  - Logistic regression:
    - Separable: norm diverges in direction of max-margin separator (**Soudry et al.**)
    - Non-separable: biased towards max-margin, but complicated (**Ji/Telgarsky**)
  - Also see **Telgarsky notes section 10**
  - Matrix factorization models: conjectured min nuclear norm, slightly controversial
Implicit regularization of gradient descent

• We just showed that gradient descent for OLS with $X$ of rank $n$, starting from zero with $\eta < 2n / \sigma_{\text{max}}(X)^2$, converges to the minimum-norm interpolator $X^\dagger y$
  
  • “Ridgeless” regression: $\lim \limits_{\lambda \to 0} (X^\top X + \lambda I)^{-1}X^\top y = X^\dagger y = \lim \limits_{\lambda \to 0} X^\top (XX^\top + \lambda I)^{-1}y$

• If we track $w_0 \neq 0$ in same analysis, get $w_\infty = (I - VV^\top)w_0 + X^\dagger y$ (proof)

• So, the 1,000-degree polynomial picture is what (small-LR) GD would give

• Does this same idea hold for other losses / models? Not necessarily.
  
  • Logistic regression:
    • Separable: norm diverges in direction of max-margin separator (Soudry et al.)
    • Non-separable: biased towards max-margin, but complicated (Ji/Telgarsky)
    • Also see Telgarsky notes section 10
  
  • Matrix factorization models: conjectured min nuclear norm, slightly controversial
  
  • Deep learning: ???
Double descent

Fig. 2. Double-descent risk curve for the RFF model on MNIST. Shown are test risks (log scale), coefficient $\ell_2$ norms (log scale), and training risks of the RFF model predictors $h_{n,N}$ learned on a subset of MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$. 

Belkin/Hsu/Ma/Mandal, PNAS 2019
Double descent

Classical regime (left of peak): unique ERM

Fig. 2. Double-descent risk curve for the RFF model on MNIST. Shown are test risks (log scale), coefficient \( \ell_2 \) norms (log scale), and training risks of the RFF model predictors \( h_{n,N} \) learned on a subset of MNIST \( (n = 10^4, 10 \text{ classes}) \). The interpolation threshold is achieved at \( N = 10^4 \).
Double descent

Classical regime (left of peak): unique ERM

Interpolating regime (right of peak): many possible interpolators

Fig. 2. Double-descent risk curve for the RFF model on MNIST. Shown are test risks (log scale), coefficient $\ell_2$ norms (log scale), and training risks of the RFF model predictors $h_{n,N}$ learned on a subset of MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$. 
Double descent

Classical regime (left of peak): unique ERM

Interpolating regime (right of peak): many possible interpolators which one we get depends on alg.’s implicit bias

Fig. 2. Double-descent risk curve for the RFF model on MNIST. Shown are test risks (log scale), coefficient $\ell_2$ norms (log scale), and training risks of the RFF model predictors $h_{n,N}$ learned on a subset of MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$. 

Belkin/Hsu/Ma/Mandal, PNAS 2019
Fig. 3. Double-descent risk curve for a fully connected neural network on MNIST. Shown are training and test risks of a network with a single layer of $H$ hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d + 1) \cdot H + (H + 1) \cdot K$. The interpolation threshold (black dashed line) is observed at $n \cdot K$.

Fig. 4. Double-descent risk curve for random forests on MNIST. The double-descent risk curve is observed for random forests with increasing model complexity trained on a subset of MNIST ($n = 10^4$, 10 classes). Its complexity is controlled by the number of trees $N_{\text{tree}}$ and the maximum number of leaves allowed for each tree $N_{\text{leaf}}^{\text{max}}$. 
More data hurts!
Definition 1 (Effective Model Complexity) The Effective Model Complexity (EMC) of a training procedure $\mathcal{T}$, with respect to distribution $\mathcal{D}$ and parameter $\epsilon > 0$, is defined as:

$$\text{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) := \max \{ n \mid \mathbb{E}_{S \sim \mathcal{D}^n} [\text{Error}_S(\mathcal{T}(S))] \leq \epsilon \}$$

where $\text{Error}_S(M)$ is the mean error of model $M$ on train samples $S$.

Our main hypothesis can be informally stated as follows:

Hypothesis 1 (Generalized Double Descent hypothesis, informal) For any natural data distribution $\mathcal{D}$, neural-network-based training procedure $\mathcal{T}$, and small $\epsilon > 0$, if we consider the task of predicting labels based on $n$ samples from $\mathcal{D}$ then:

Under-parameterized regime. If $\text{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ is sufficiently smaller than $n$, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.

Over-parameterized regime. If $\text{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ is sufficiently larger than $n$, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.

Critically parameterized regime. If $\text{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$, then a perturbation of $\mathcal{T}$ that increases its effective complexity might decrease or increase the test error.
(pause)
Neural Tangent Kernels (NTKs)
Neural Tangent Kernels (NTKs)

• Gradient descent for square loss finds min-norm interpolator ("ridgeless" regression)
Neural Tangent Kernels (NTKs)

• Gradient descent for square loss finds min-norm interpolator ("ridgeless" regression)

• As we’ll see, training an “ultrawide” deep network for square loss ends up being equivalent to “ridgeless” regression with a neural tangent kernel
Neural Tangent Kernels (NTKs)

- Gradient descent for square loss finds min-norm interpolator ("ridgeless" regression)

- As we'll see, training an “ultrawide” deep network for square loss ends up being equivalent to “ridgeless” regression with a neural tangent kernel

- So, in the infinite-width limit, we know things correspond to finding the solution that has small RKHS norm for the neural tangent kernel
Neural Tangent Kernels (NTKs)

• Gradient descent for square loss finds min-norm interpolator ("ridgeless" regression)

• As we’ll see, training an “ultrawide” deep network for square loss ends up being equivalent to “ridgeless” regression with a neural tangent kernel

• So, in the infinite-width limit, we know things correspond to finding the solution that has small RKHS norm for the neural tangent kernel
Neural Tangent Kernels (NTKs)

- Gradient descent for square loss finds min-norm interpolator ("ridgeless" regression)

- As we’ll see, training an “ultrawide” deep network for square loss ends up being equivalent to “ridgeless” regression with a neural tangent kernel

- So, in the infinite-width limit, we know things correspond to finding the solution that has small RKHS norm for the neural tangent kernel

- Another POV:

\[
L_{\mathcal{D}}(\mathcal{A}(S)) - L^* = L_{\mathcal{D}}(\mathcal{A}(S)) - L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) + L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L^*
\]

- \(L_{\mathcal{D}}(\mathcal{A}(S))\) optimization error
- \(L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S))\) estimation error
- \(\inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h)\) approximation error
Nonconvex optimization

- Neural nets are not convex

\[ \ell(w; (x, y)) = (f_w(x) - y)^2 \]
Nonconvex optimization

- Neural nets are not convex
- Even deep linear networks are not convex
Nonconvex optimization

• Neural nets are not convex
• Even deep linear networks are not convex

• But we do know that SGD converges to a critical point under fairly mild conditions
Nonconvex optimization

• Neural nets are not convex
• Even deep linear networks are not convex

• But we do know that SGD converges to a critical point under fairly mild conditions
  • e.g.: if $f \geq f^{\inf}$ is differentiable and $\beta$-smooth, and there are $A, B, C$ s.t. for all $x$, $\mathbb{E}[\|\hat{g}(x)\|^2] \leq 2A(f(x) - f^{\inf}) + B\|\nabla f(X)\|^2 + C$, then the best iterate from $O(\epsilon^{-4})$ steps has $\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon^2$ (Khaled/Richtárik 2020)
Nonconvex optimization

- Neural nets are not convex
- Even **deep linear networks** are not convex

- But we do know that SGD converges to a *critical point* under fairly mild conditions
  - e.g.: if $f \geq f^{\text{inf}}$ is differentiable and $\beta$-smooth, and there are $A, B, C$ s.t. for all $x$, $\mathbb{E}\left[\|\hat{g}(x)\|^2\right] \leq 2A(f(x) - f^{\text{inf}}) + B\|\nabla f(X)\|^2 + C$, then the best iterate from $O(\epsilon^{-4})$ steps has $\mathbb{E}\left[\|\nabla f(x)\|^2\right] \leq \epsilon^2$ (Khaled/Richtárik 2020)

- In deep linear nets, local minima are global minima (Kawaguchi 2016, Laurent/von Brecht 2019)
Nonconvex optimization

• Neural nets are not convex
• Even deep linear networks are not convex

• But we do know that SGD converges to a critical point under fairly mild conditions
  • e.g.: if $f \geq f^{\inf}$ is differentiable and $\beta$-smooth, and there are $A, B, C$ s.t. for all $x$, $\mathbb{E} \left[ \| \hat{g}(x) \|^2 \right] \leq 2A(f(x) - f^{\inf}) + B\| \nabla f(X) \|^2 + C$, then the best iterate from $\mathcal{O}(\varepsilon^{-4})$ steps has $\mathbb{E} \left[ \| \nabla f(x) \|^2 \right] \leq \varepsilon^2$ (Khaled/Richtárik 2020)

• In deep linear nets, local minima are global minima (Kawaguchi 2016, Laurent/von Brecht 2019)
  • …but there are saddle points, including “bad” ones where $\lambda_{\min}(\nabla^2 f) = 0$
Nonconvex optimization

• Neural nets are not convex
• Even **deep linear networks** are not convex

• But we do know that SGD converges to a *critical point* under fairly mild conditions
  
  e.g.: if $f \geq f^{\text{inf}}$ is differentiable and $\beta$-smooth, and
  
  there are $A, B, C$ s.t. for all $x$, $\mathbb{E} \left[ \| \hat{\nabla} g(x) \|^2 \right] \leq 2A(f(x) - f^{\text{inf}}) + B\| \nabla f(X) \|^2 + C$,

  then the *best* iterate from $O(\varepsilon^{-4})$ steps has $\mathbb{E} \left[ \| \nabla f(x) \|^2 \right] \leq \varepsilon^2$ *(Khaled/Richtárik 2020)*

• In deep linear nets, local minima are global minima *(Kawaguchi 2016, Laurent/von Brecht 2019)*
  
  …but there are saddle points, including “bad” ones where $\lambda_{\text{min}}(\nabla^2 f) = 0$

  …but gradient descent almost surely escapes saddles, reaches a local min *(Lee et al. 2016)*
Nonconvex optimization

• Neural nets are not convex
• Even deep linear networks are not convex

• But we do know that SGD converges to a critical point under fairly mild conditions
  • e.g.: if $f \geq f^{\inf}$ is differentiable and $\beta$-smooth, and
    there are $A, B, C$ s.t. for all $x$, $\mathbb{E} \left[ \| \hat{g}(x) \|_2^2 \right] \leq 2A (f(x) - f^{\inf}) + B \| \nabla f(X) \|_2^2 + C$,
    then the best iterate from $O(\epsilon^{-4})$ steps has $\mathbb{E} \left[ \| \nabla f(x) \|_2^2 \right] \leq \epsilon^2$ (Khaled/Richtárik 2020)

• In deep linear nets, local minima are global minima (Kawaguchi 2016, Laurent/von Brecht 2019)
  • …but there are saddle points, including “bad” ones where $\lambda_{\text{min}}(\nabla^2 f) = 0$
  • …but gradient descent almost surely escapes saddles, reaches a local min (Lee et al. 2016)
  • …but it can take exponential time to escape (Du et al. 2017)
Nonconvex optimization

• Neural nets are not convex
• Even **deep linear networks** are not convex

• But we do know that SGD converges to a *critical point* under fairly mild conditions
  • e.g.: if $f \geq f^{\inf}$ is differentiable and $\beta$-smooth, and there are $A$, $B$, $C$ s.t. for all $x$, $\mathbb{E}\left[\|\hat{g}(x)\|^2\right] \leq 2A(f(x) - f^{\inf}) + B\|\nabla f(X)\|^2 + C$, then the best iterate from $\mathcal{O}(\epsilon^{-4})$ steps has $\mathbb{E}\left[\|\nabla f(x)\|^2\right] \leq \epsilon^2$ (Khaled/Richtárik 2020)

• In deep linear nets, local minima are global minima (Kawaguchi 2016, Laurent/von Brecht 2019)
  • …but there are saddle points, including “bad” ones where $\lambda_{\min}(\nabla^2 f) = 0$
  • …but gradient descent almost surely escapes saddles, reaches a local min (Lee et al. 2016)
  • …but it can take exponential time to escape (Du et al. 2017)
  • …but that doesn’t happen on deep linear nets [under conditions] (Arora et al. 2019)
Bad local minima in ReLU nets

\[ h(x) = \text{ReLU}(wx) \text{ (reals to reals), square loss, } S = ((1,1)) : \]
Sub-Optimal Local Minima Exist for Neural Networks with Almost All Non-Linear Activations

Tian Ding*   Dawei Li †   Ruoyu Sun ‡

Nov 4, 2019