ECE826 Lecture 9:

The PL Land of Nonconvexity

Contents

- GD on general non convex functions

- PL makes things faster

- Linear & Non-linear least Squares

- 1-layer Neural Networks

Minimizing the Empirical Risk

- The empirical cost function that we have access to

$$\min_{h \in \mathscr{H}} \left(R_S[h] = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i); y_i) \right)$$

- <u>Question</u>: Can we approximate the solution to this minimization? If so how fast?

- The answer must depend on:

  1) $\boldsymbol{n}$, the sample size
  2) $\mathscr{H}$, the hypothesis class and loss function
  3) $\mathscr{D}$, the data distribution
  4) the optimization algorithm that outputs our classifier

Last time: From GD to SGD

Last time: Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

- Note: we haven't used the fact that $f(w) = \dfrac{1}{n}\sum_{i=1}^{n} f_i(w)$
- Idea ('50s, '60s [Robbins, Monro], [Widrow, Hoff]):
  instead of computing $\nabla f(w)$ we can sample one f_i at random and compute its gradient
- Why does that make sense? In "expectation" it's the same algorithm, i.e.,

$$E_{i \sim \text{uniform}} \nabla f_i = \sum_i \frac{1}{n} \nabla f_i = \nabla f(w)$$

SGD: $w_{k+1} = w_k - \gamma \nabla f_{i_k}(w_k)$

The Uber-Algorithm

Convergence rates for SGD

Corollary:

SGD with constant stepwise achieves exponential convergence till error an error floor of $\mathbb{E}\|w_{k+1} - w^*\|^2 \geq \epsilon \cdot O\left(\frac{L^2}{\lambda^2}\right)$ and after that achieves a rate of $O(1/T)$ for arbitrary errors.

How does SGD compare with GD?

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Total GD computational cost

$$O\left(T_\epsilon^{\text{GD}} \cdot \text{cost}(\nabla f)\right) = O\left(\text{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total GD computational cost

$$O\left(T_\epsilon^{\text{SGD}} \cdot \mathbb{E}\text{cost}(\nabla f_i)\right) = O\left(\frac{\text{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$

$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

- SGD is faster than GD (for regularized logistic regression and in the worst case) as long as

$$nd^2 \log\left(\frac{d}{\epsilon}\right) \geq \frac{d^2}{\epsilon}\log\left(\frac{d}{\epsilon}\right)$$

$$\implies \epsilon \geq \frac{1}{n}$$

OK what do I do in practice?

The SGD quick start guide

Newcomers to stochastic gradient descent often find all of these design choices daunting, and it's useful to have simple rules of thumb to get going. We recommend the following:

1. Pick as large a minibatch size as you can given your computer's RAM.
2. Set your momentum parameter to either 0 or 0.9. Your call!
3. Find the largest constant stepsize such that SGD doesn't diverge. This takes some trial and error, but you only need to be accurate to within a factor of 10 here.
4. Run SGD with this constant stepsize until the empirical risk plateaus.
5. Reduce the stepsize by a constant factor (say, 10)
6. Repeat steps 4 and 5 until you converge.

While this approach may not be the most optimal in all cases, it's a great starting point and is good enough for probably 90% of applications we've encountered.

[Recht, Hardt, book 2021]

Convergence for Nonconvex functions?

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \frac{R}{\beta L^2 T}$ satisfy

$$\min_{k \in [T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

Proof:

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k \in [T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

Proof:

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k \rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\langle \nabla f(w_k), \nabla f_{s_k}(w_k)\rangle \leq \frac{\beta}{2}\|\gamma \nabla f_{s_k}(w_k)\|^2$$

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k\in[T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

Proof:

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k\rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\langle \nabla f(w_k), \nabla f_{s_k}(w_k)\rangle \leq \frac{\beta}{2}\|\gamma \nabla f_{s_k}(w_k)\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\beta\gamma^2}{2}\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2$$

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k\in[T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

Proof:

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k\rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\Longrightarrow \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\langle \nabla f(w_k), \nabla f_{s_k}(w_k)\rangle \leq \frac{\beta}{2}\|\gamma \nabla f_{s_k}(w_k)\|^2$$

$$\Longrightarrow \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\beta\gamma^2}{2}\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2$$

$$\Longrightarrow \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k)}{\gamma} + \frac{\gamma L^2\beta}{2}$$

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k\in[T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

Proof:

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k\rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\langle \nabla f(w_k), \nabla f_{s_k}(w_k)\rangle \leq \frac{\beta}{2}\|\gamma \nabla f_{s_k}(w_k)\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\beta\gamma^2}{2}\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2$$

$$\implies \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k)}{\gamma} + \frac{\gamma L^2\beta}{2}$$

$$\implies \min_k \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_1) - f(w_0)}{\gamma T} + \frac{\gamma L^2\beta}{2T}$$

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k\in[T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

Proof:

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k\rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma\langle \nabla f(w_k), \nabla f_{s_k}(w_k)\rangle \leq \frac{\beta}{2}\|\gamma \nabla f_{s_k}(w_k)\|^2$$

This is a very slow rate, that is very conservative. Makes sense!

$$\implies \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k)}{\gamma} + \frac{\gamma L^2\beta}{2}$$

It also doesn't tell us anything about the quality of the solution that SGD finds

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\| \nabla_w L(w) \|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \dfrac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\left\| \nabla_w L(w) \right\|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \dfrac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2} \|w_k - w_{k+1}\|^2$$

$$\leq -\gamma \|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2} \|\nabla f(w_k)\|^2$$

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\| \nabla_w L(w) \|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \frac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2} \|w_k - w_{k+1}\|^2$$

$$\leq -\gamma \|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2} \|\nabla f(w_k)\|^2$$

$$\leq -\frac{1}{\beta} \|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*)$$

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\| \nabla_w L(w) \|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \dfrac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\leq -\gamma\|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2}\|\nabla f(w_k)\|^2$$

$$\leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*)$$

$$\implies f(w_{k+1}) - f(w_k) - f^* \leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - f^*$$

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\| \nabla_w L(w) \|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \frac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\leq -\gamma\|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2}\|\nabla f(w_k)\|^2$$

$$\leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*)$$

$$\Longrightarrow f(w_{k+1}) - f(w_k) - f^* \leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - f^*$$

$$f(w_{k+1}) - f^* \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - (f^* - f(w_k))$$

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\| \nabla_w L(w) \|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \dfrac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\leq -\gamma\|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2}\|\nabla f(w_k)\|^2$$

$$\leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*)$$

$$\implies f(w_{k+1}) - f(w_k) - f^* \leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - f^*$$

$$f(w_{k+1}) - f^* \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - (f^* - f(w_k))$$

$$f(w_{k+1}) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)(f(w_k) - f^*)$$

GD on Polyak-Łojasiewicz functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\left\| \nabla_w L(w) \right\|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \dfrac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

Proof:

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\leq -\gamma\|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2}\|\nabla f(w_k)\|^2$$

much faster rate, when is PL satisfied?

$$\implies f(w_{k+1}) - f(w_k) - f^* \leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - f^*$$

$$f(w_{k+1}) - f^* \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - (f^* - f(w_k))$$

$$f(w_{k+1}) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)(f(w_k) - f^*)$$

GD on PL functions

Theorem

Let $f(w)$ be a β-smooth, μ-PL function (i.e., $\| \nabla_w L(w) \|^2 \geq \mu(L(w) - L^*)$.

Then, GD with with step-size $\gamma = \dfrac{1}{L}$ satisfies

$$f(w_k) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)^k \left(f(w_0) - f^*\right)$$

$$f(w_{k+1}) - f(w_k) \leq \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\leq -\gamma\|\nabla f(w_k)\|^2 + \frac{\beta}{2\beta^2}\|\nabla f(w_k)\|^2$$

much faster rate, when is PL satisfied?

$$\implies f(w_{k+1}) - f(w_k) - f^* \leq -\frac{1}{\beta}\|\nabla f(w_k)\|^2 \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - f^*$$

$$f(w_{k+1}) - f^* \leq -\frac{\mu}{\beta}(f(w_k) - f^*) - (f^* - f(w_k))$$

$$f(w_{k+1}) - f^* \leq \left(1 - \frac{\mu}{\beta}\right)(f(w_k) - f^*)$$

Loss landscapes and optimization in over-parameterized non-linear systems and neural networks

Chaoyue Liu[a], Libin Zhu[b,c], and Mikhail Belkin[c]

[a]*Department of Computer Science and Engineering, The Ohio State University*
[b]*Department of Computer Science and Engineering, University of California, San Diego*
[c]*Halıcıoğlu Data Science Institute, University of California, San Diego*

May 28, 2021

Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path?

Samet Oymak* and Mahdi Soltanolkotabi†

A Convergence Theory for Deep Learning via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu

Yuanzhi Li
yuanzhil@stanford.edu

Zhao Song
zhaos@utexas.e
UT-Austin
ity of Washington

On the Convergence Rate of Training Recurrent Neural Networks

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

October 28, 2018

No bad local minima: Data independent training error guarantees for multilayer neural networks

Daniel Soudry
Department of Statistics
Columbia University
New York, NY 10027, USA
daniel.soudry@gmail.com

Yair Carmon
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA
yairc@stanford.edu

Gradient Descent Finds Global Minima of Deep Neural Networks

Simon S. Du [*1] **Jason D. Lee** [*2] **Haochuan Li** [*3 4] **Liwei Wang** [*5 4] **Xiyu Zhai** [*6]

Loss landscapes and optimization in over-parameterized non-linear systems and neural networks

Chaoyue Liu[a], Libin Zhu[b,c], and Mikhail Belkin[c]

[a] Department of Computer Science and Engineering, The Ohio State University
[b] Department of Computer Science and Engineering, University of California, San Diego
[c] Halicioğlu Data Science Institute, University of California, San Diego

May 28, 2021

Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path?

Samet Oymak* and Mahdi Soltanolkotabi†

A Convergence Theory for Deep Learning via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu

Yuanzhi Li
yuanzhil@stanford.edu

Zhao Song
zhaos@utexas.e
UT-Austin
ity of Washington

On the Convergence Rate of Training Recurrent Neural Networks

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

October 28, 2018

PL-like conditions hold in neighborhoods around initialization/optima.

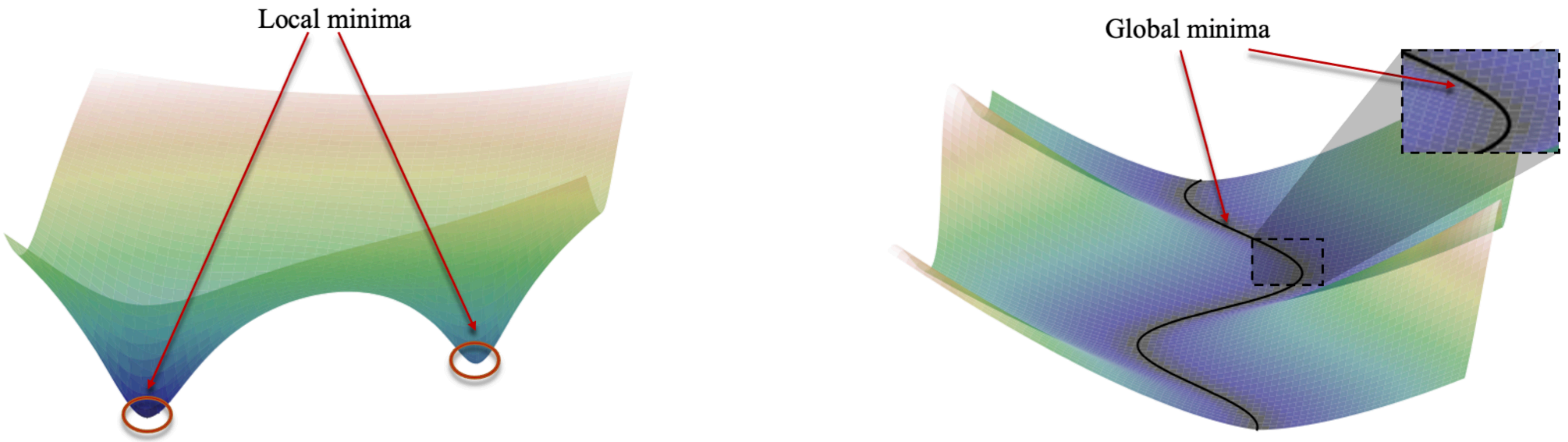
No bad local minima: Data independent training error guarantees for multilayer neural networks

Daniel Soudry
Department of Statistics
Columbia University
New York, NY 10027, USA
daniel.soudry@gmail.com

Yair Carmon
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA
yairc@stanford.edu

Gradient Descent Finds Global Minima of Deep Neural Networks

Simon S. Du *1 Jason D. Lee *2 Haochuan Li *3 4 Liwei Wang *5 4 Xiyu Zhai *6



(a) Loss landscape of under-parameterized models

(b) Loss landscape of over-parameterized models

Figure 1: Panel (a): Loss landscape is locally convex at local minima. Panel (b): Loss landscape incompatible with local convexity as the set of global minima is not locally linear.

PL-like conditions hold in neighborhoods around initialization/optima.

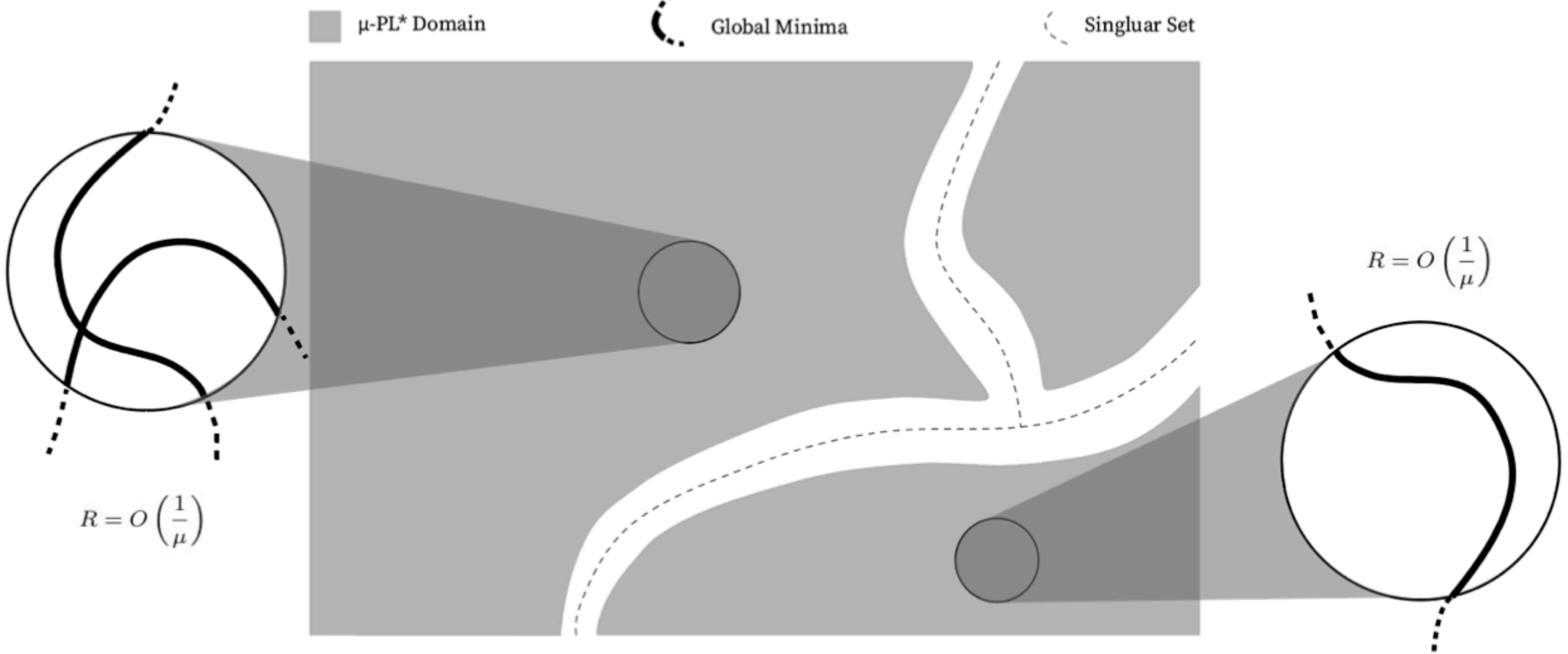
A Convergence Theory for Deep Learning via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

October 28, 2018




via Over-Parameterization

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research AI

Yuanzhi Li
yuanzhil@stanford.edu
Stanford University
Princeton University

Zhao Song
zhaos@utexas.edu
UT-Austin
University of Washington
Harvard University

PL-like conditions hold in neighborhoods around initialization/optima.

PL in Least Squares

GD on linear least squares

- Let's say we are trying to solve $\min_w \|X^T w - y\|^2$ with GD.

GD on linear least squares

- Let's say we are trying to solve $\min_w \|X^T w - y\|^2$ with GD.
- The gradient of the loss is equal to $\nabla_w \|X^T w - y\|^2 = X(X^T w - y)$

GD on linear least squares

- Let's say we are trying to solve $\min_w \|X^T w - y\|^2$ with GD.

- The gradient of the loss is equal to $\nabla_w \|X^T w - y\|^2 = X(X^T w - y)$
- Note that

$$\left\| \nabla_w L(w) \right\|^2 = \left\| 2X(X^T w - y) \right\|^2 \geq 4\lambda_{\min}(XX^T) \left\| X^T w - y \right\|^2 = 4\lambda_{\min}(XX^T) \cdot L(w)$$

GD on linear least squares

- Let's say we are trying to solve $\min_w \|X^T w - y\|^2$ with GD.

- The gradient of the loss is equal to $\nabla_w \|X^T w - y\|^2 = X(X^T w - y)$
- Note that

$$\| \nabla_w L(w) \|^2 = \| 2X(X^T w - y) \|^2 \geq 4\lambda_{\min}(XX^T) \| X^T w - y \|^2 = 4\lambda_{\min}(XX^T) \cdot L(w)$$

- Ha! that is the PL condition assuming $L^* = 0$, which is true when data mat = full rank

GD on linear least squares

- Let's say we are trying to solve $\min_w \|X^T w - y\|^2$ with GD.

- The gradient of the loss is equal to $\nabla_w \|X^T w - y\|^2 = X(X^T w - y)$
- Note that

$$\| \nabla_w L(w) \|^2 = \| 2X(X^T w - y) \|^2 \geq 4\lambda_{\min}(XX^T) \| X^T w - y \|^2 = 4\lambda_{\min}(XX^T) \cdot L(w)$$

- Ha! that is the PL condition assuming $L^* = 0$, which is true when data mat = full rank

Lemma:

Linear least squares where $\mathrm{rank}(X) = n$ is PL

GD on nonlinear least squares

- Let's say we are trying to solve $\min_{w} \|h(X; w) - y\|^2$ with GD.

GD on nonlinear least squares

- Let's say we are trying to solve $\min_w \|h(X; w) - y\|^2$ with GD.
- The gradient of the loss is equal to $\nabla_w \|h(X; w) - y\|^2 = [\nabla_w h(X; w)](h(X; w) - y)$

GD on nonlinear least squares

- Let's say we are trying to solve $\min_w \|h(X; w) - y\|^2$ with GD.
- The gradient of the loss is equal to $\nabla_w \|h(X; w) - y\|^2 = [\nabla_w h(X; w)](h(X; w) - y)$
- Let us refer to $J(w) = \nabla_w h(X; w) \in \mathbb{R}^{d \times n}$ as the Jacobian of the predictions

GD on nonlinear least squares

- Let's say we are trying to solve $\min_w \|h(X; w) - y\|^2$ with GD.
- The gradient of the loss is equal to $\nabla_w \|h(X; w) - y\|^2 = [\nabla_w h(X; w)](h(X; w) - y)$
- Let us refer to $J(w) = \nabla_w h(X; w) \in \mathbb{R}^{d \times n}$ as the Jacobian of the predictions
- Note that again

$$\left\| \nabla_w L(w) \right\|^2 = \left\| J(w)(h(X; w) - y) \right\|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \left\| h(X; w) - y \right\|^2$$

GD on nonlinear least squares

- Let's say we are trying to solve $\min_w \|h(X;w) - y\|^2$ with GD.
- The gradient of the loss is equal to $\nabla_w \|h(X;w) - y\|^2 = [\nabla_w h(X;w)](h(X;w) - y)$
- Let us refer to $J(w) = \nabla_w h(X;w) \in \mathbb{R}^{d\times n}$ as the Jacobian of the predictions
- Note that again

$$\left\| \nabla_w L(w) \right\|^2 = \left\| J(w)(h(X;w) - y) \right\|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \left\| h(X;w) - y \right\|^2$$

- Ha! that is the again PL condition (assuming $L^* = 0$) with $\mu = 4\lambda_{\min}(X^T X)$

GD on nonlinear least squares

- Let's say we are trying to solve $\min_w \|h(X;w) - y\|^2$ with GD.
- The gradient of the loss is equal to $\nabla_w \|h(X;w) - y\|^2 = [\nabla_w h(X;w)](h(X;w) - y)$
- Let us refer to $J(w) = \nabla_w h(X;w) \in \mathbb{R}^{d \times n}$ as the Jacobian of the predictions
- Note that again
$$\| \nabla_w L(w) \|^2 = \| J(w)(h(X;w) - y) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X;w) - y \|^2$$
- Ha! that is the again PL condition (assuming $L^* = 0$) with $\mu = 4\lambda_{\min}(X^T X)$

Lemma:

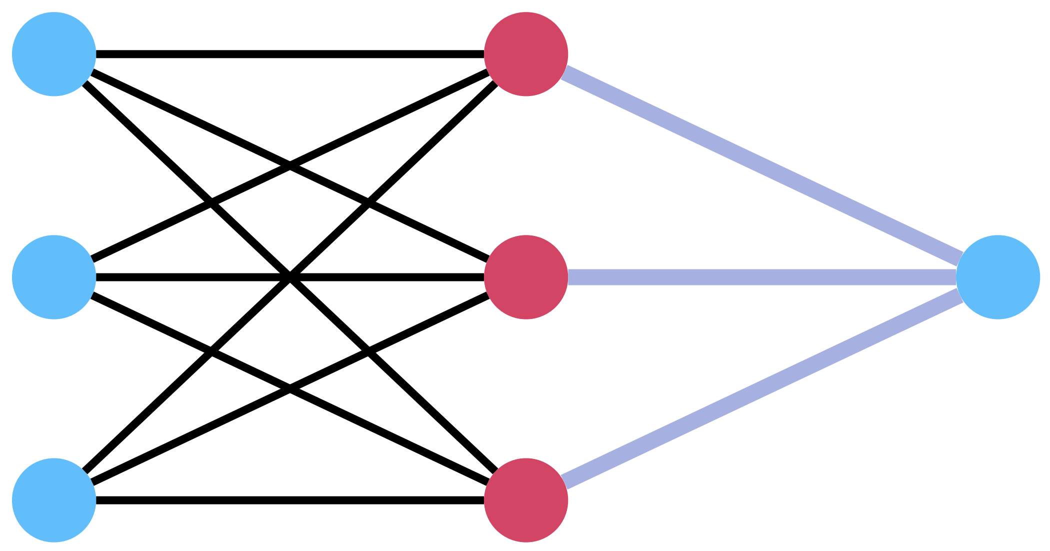
Non-linear least squares where $\min_{w \in \mathscr{W}} \text{rank}(J(w(w)) = n$ are PL in $\mathscr{W}$

Some examples of NNLS

1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W; x) = \langle v, Wx \rangle$

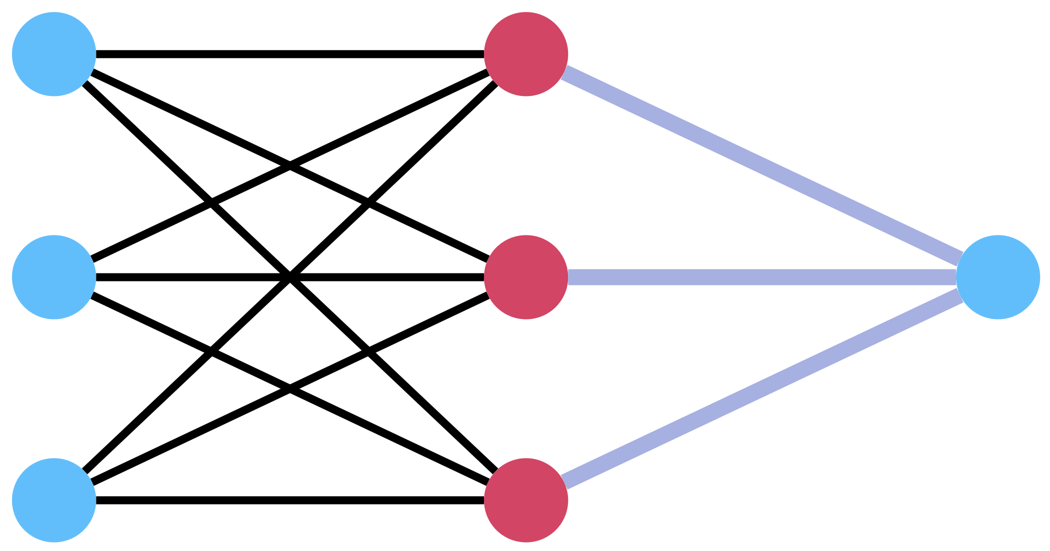
$\sigma(x) = x$



1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W; x) = \langle v, Wx \rangle$

$\sigma(x) = x$



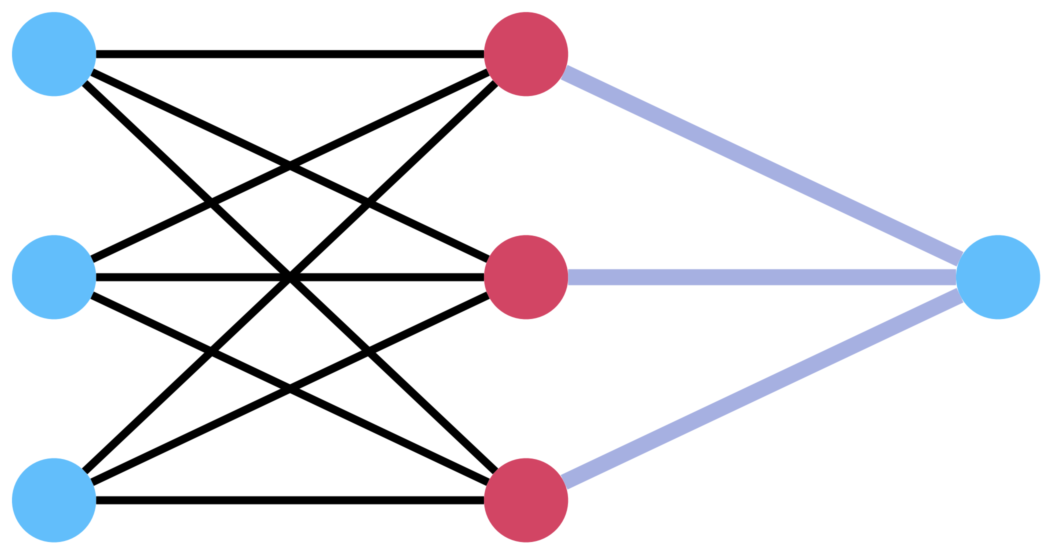
assume output edges are all fixed

A bit silly since
$h(W; x) = \langle v, Wx \rangle = \langle w', x \rangle$

1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W; x) = \langle v, Wx \rangle$

$\sigma(x) = x$



assume output edges are all fixed

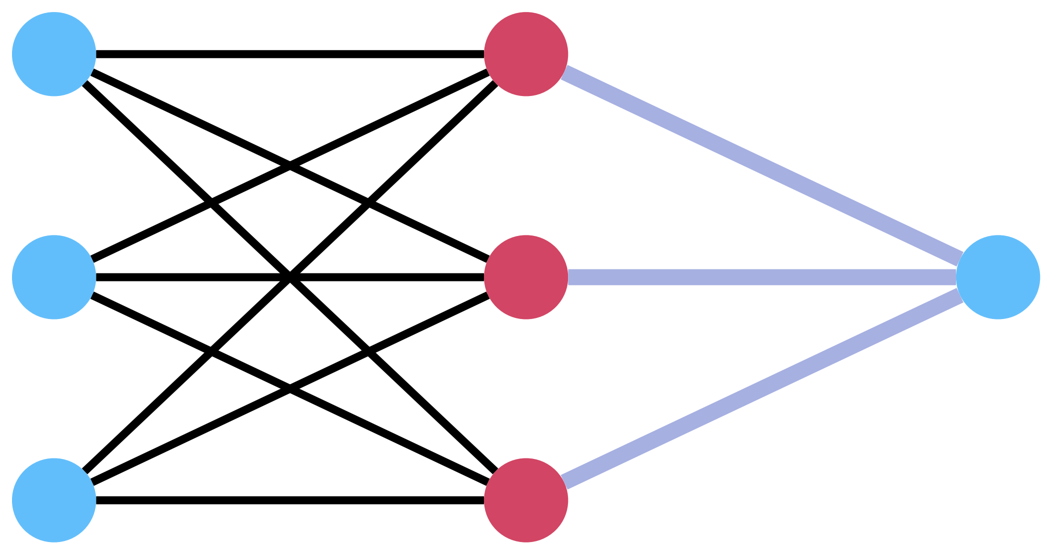
reminder:

$$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X; w) - y \|$$

1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W; x) = \langle v, Wx \rangle$

$\sigma(x) = x$



assume output edges are all fixed

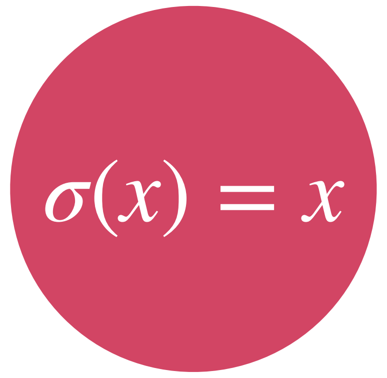
reminder:

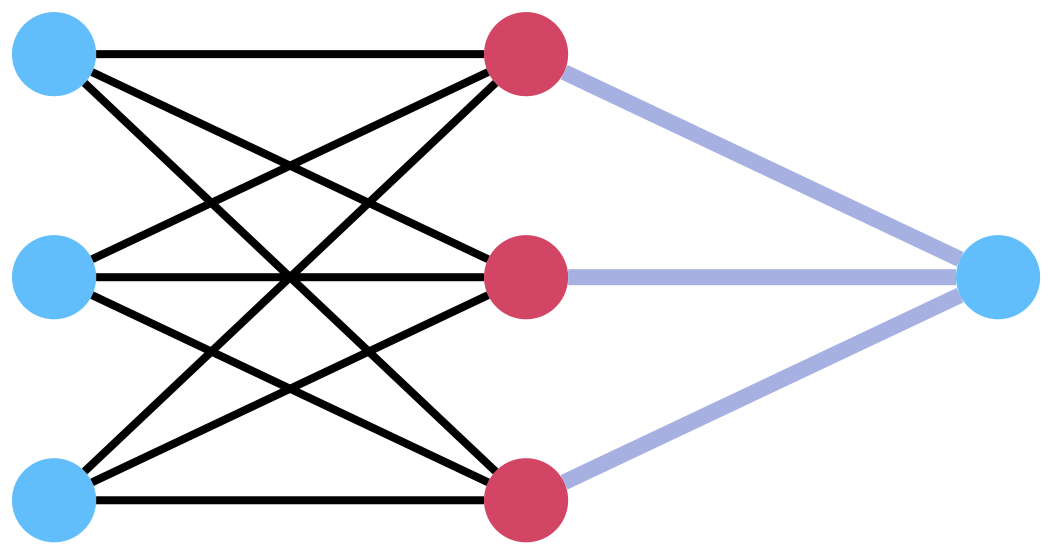
$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X; w) - y \|$

- The Jacobian is equal to $J(w) = \nabla_w h(W, x) = \begin{bmatrix} v_1 x_1 & v_1 x_2 & \dots & v_1 x_n \\ \vdots & \vdots & \dots & \vdots \\ v_k x_1 & v_k x_2 & \dots & v_k x_n \end{bmatrix} \in \mathbb{R}^{kd \times n}$

1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W;x) = \langle v, Wx \rangle$





assume output edges are all fixed

reminder:
$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X;w) - y \|$

- The Jacobian is equal to $J(w) = \nabla_w h(W,x) = \begin{bmatrix} v_1x_1 & v_1x_2 & \dots & v_1x_n \\ \vdots & \vdots & \dots & \vdots \\ v_kx_1 & v_kx_2 & \dots & v_kx_n \end{bmatrix} \in \mathbb{R}^{kd \times n}$

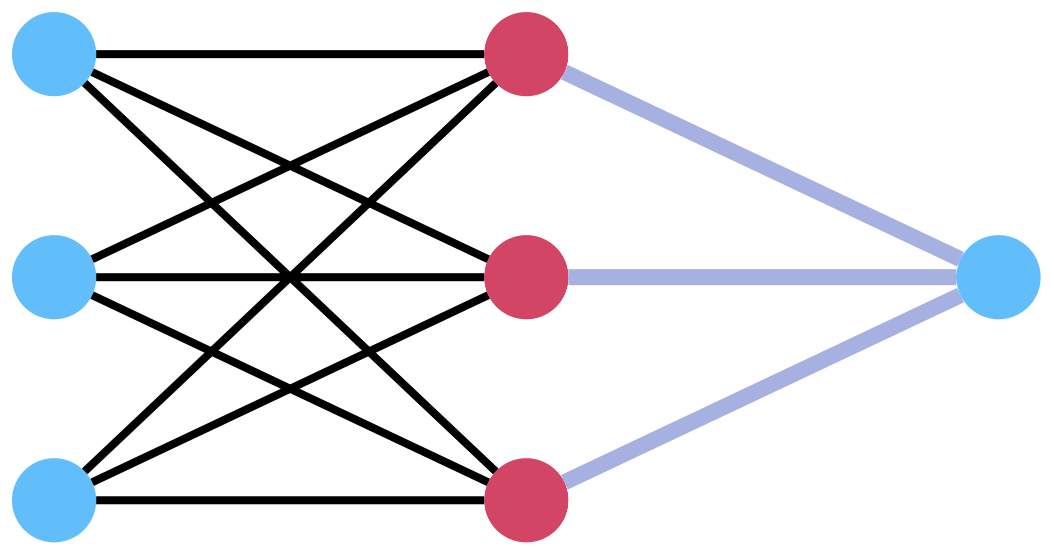
- Note that $J(w) = v \otimes X$ and we know that $\text{rank}(J(w)) = \text{rank}(v) \cdot \text{rank}(X) = \text{rank}(X)$

-

1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W; x) = \langle v, Wx \rangle$

$\sigma(x) = x$



assume output edges are all fixed

reminder:
$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X; w) - y \|$

- The Jacobian is equal to $J(w) = \nabla_w h(W, x) = \begin{bmatrix} v_1 x_1 & v_1 x_2 & \dots & v_1 x_n \\ \vdots & \vdots & \dots & \vdots \\ v_k x_1 & v_k x_2 & \dots & v_k x_n \end{bmatrix} \in \mathbb{R}^{kd \times n}$

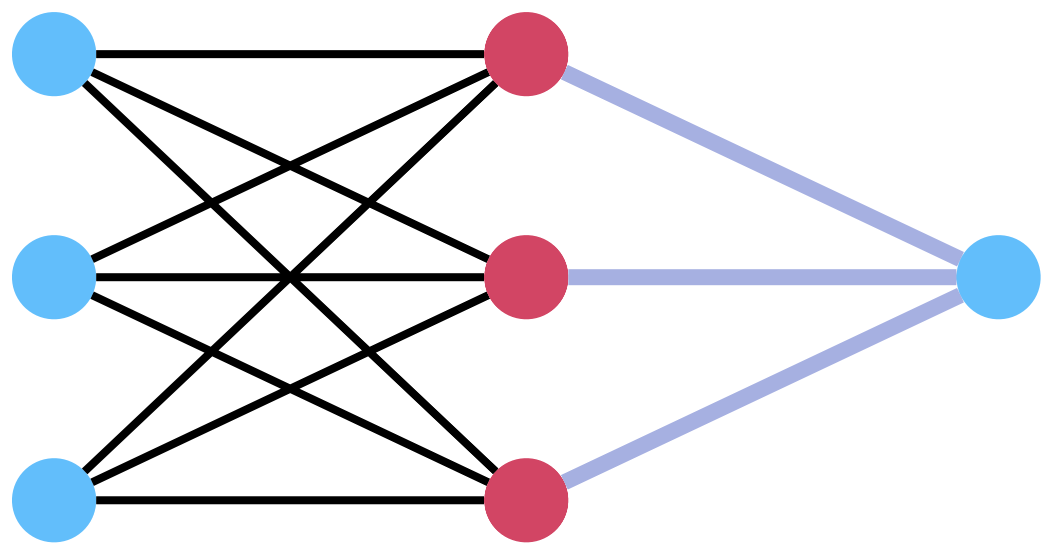
- Note that $J(w) = v \otimes X$ and we know that $\text{rank}(J(w)) = \text{rank}(v) \cdot \text{rank}(X) = \text{rank}(X)$

- Hence, again if the matrix of data points is full rank n, then the cost function is PL.

1-layer linear Neural Networks

- Let us assume we have a 1-layer linear network.
- The prediction of this network is given as $h(W;x) = \langle v, Wx \rangle$

$\sigma(x) = x$



assume output edges are all fixed

reminder:

$$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \, \| h(X;w) - y \|$$

- The Jacobian is equal to $J(w) = \nabla_w h(W,x) = \begin{bmatrix} v_1 x_1 & v_1 x_2 & \dots & v_1 x_n \\ \vdots & \vdots & \dots & \vdots \\ v_k x_1 & v_k x_2 & \dots & v_k x_n \end{bmatrix} \in \mathbb{R}^{kd \times n}$

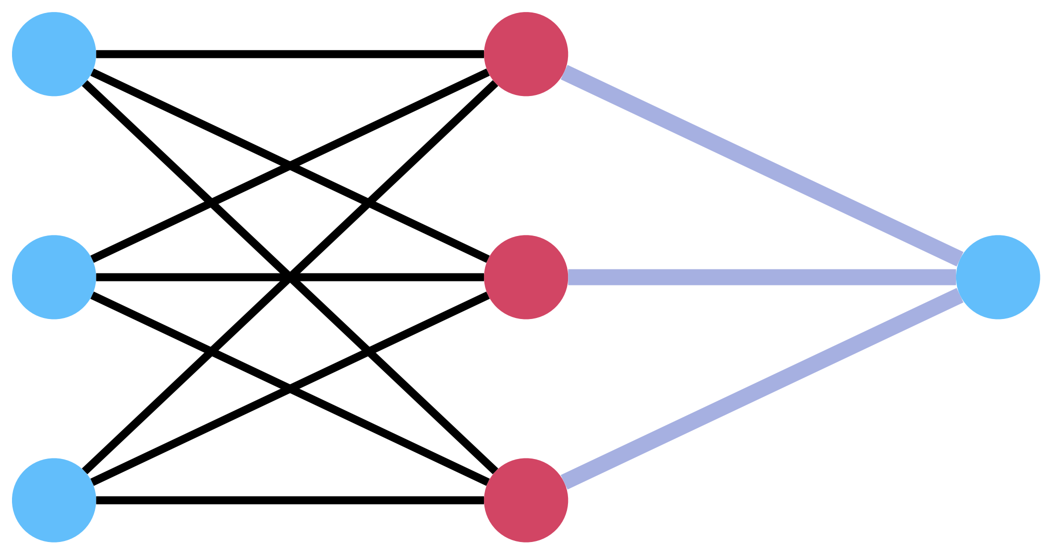
- Note that $J(w) = v \otimes X$ and we know that $\text{rank}(J(w)) = \text{rank}(v) \cdot \text{rank}(X) = \text{rank}(X)$

- Hence, again if the matrix of data points is full rank n, then the cost function is PL.

1-layer leaky ReLU Neural Networks

- Let us assume we have a 1-layer linear network
- The prediction of this network is given as $h(W; x) = \langle v, \sigma(Wx) \rangle$

$\sigma(x) = \mathbf{1}_{x \geq 0} + \epsilon \mathbf{1}_{x \leq 0}$

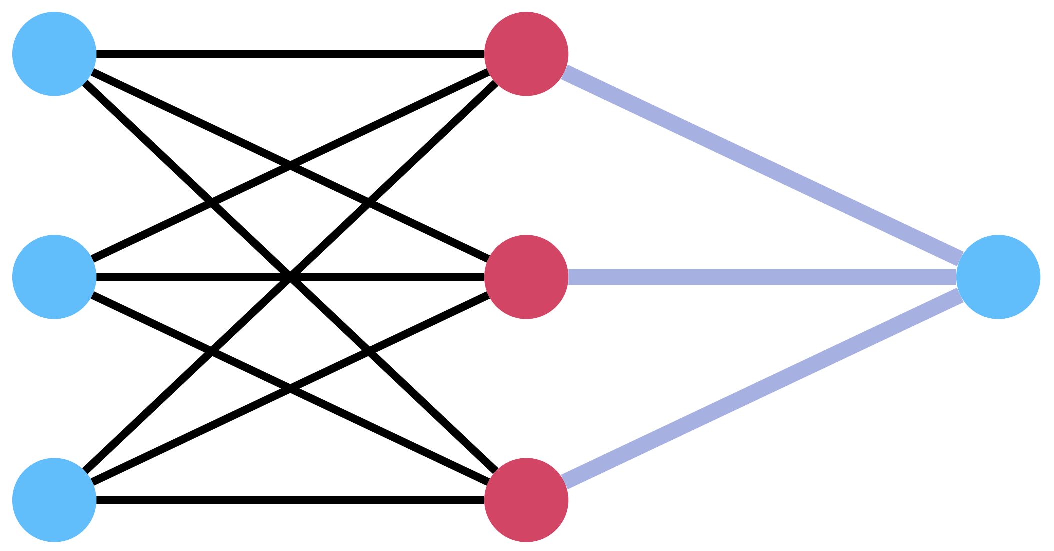


assume output edges
are all fixed

1-layer leaky ReLU Neural Networks

- Let us assume we have a 1-layer linear network
- The prediction of this network is given as $h(W;x) = \langle v, \sigma(Wx) \rangle$

$\sigma(x) = \mathbf{1}_{x\geq 0} + \epsilon \mathbf{1}_{x\leq 0}$



assume output edges are all fixed

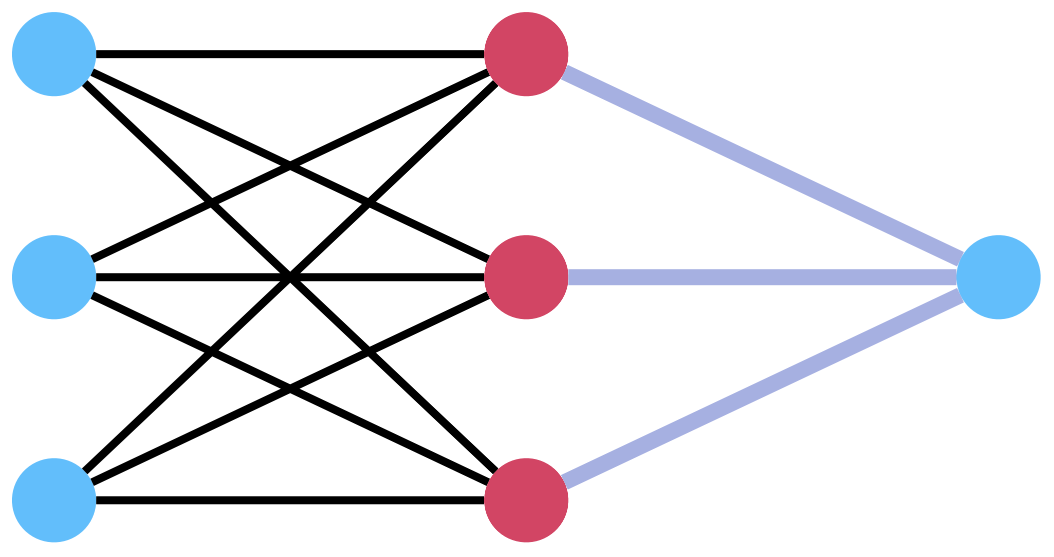
reminder:

$$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X;w) - y \|$$

1-layer leaky ReLU Neural Networks

- Let us assume we have a 1-layer linear network
- The prediction of this network is given as $h(W; x) = \langle v, \sigma(Wx) \rangle$

$\sigma(x) = \mathbf{1}_{x \geq 0} + \epsilon \mathbf{1}_{x \leq 0}$



assume output edges are all fixed

reminder:

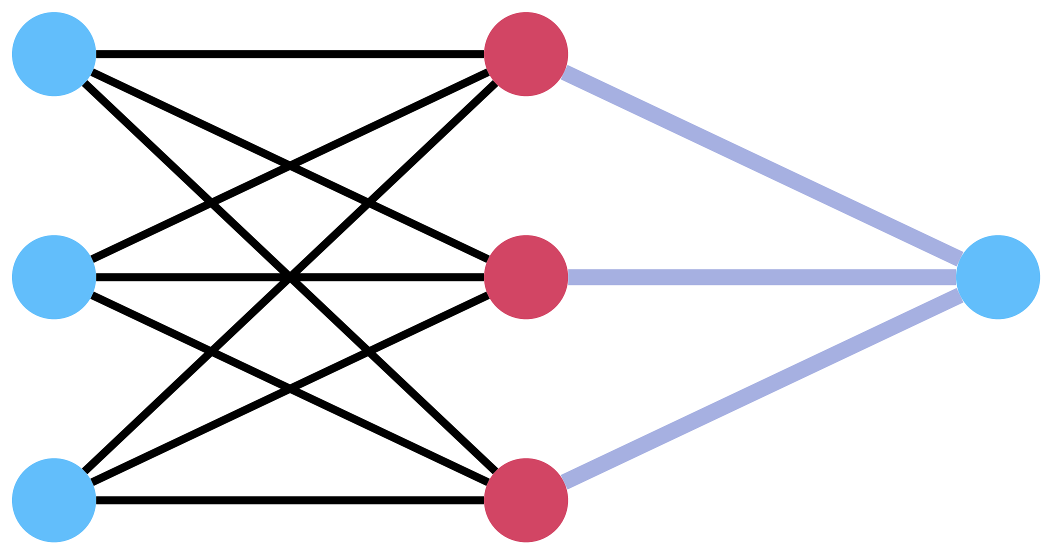
$$\| \nabla_w L(w) \|^2 \geq 4 \lambda_{\min}(J(w)^T J(w)) \| h(X; w) - y \|$$

The Jacobian is equal to $J(w) = \mathrm{diag}(v_1 I_d, \ldots, v_k I_d) \begin{bmatrix} \sigma'(\langle w_1, x_1 \rangle) \cdot x_1 & \sigma'(\langle w_1, x_2 \rangle) \cdot x_2 & \ldots & \sigma'(\langle w_1, x_n \rangle) \cdot x_n \\ \vdots & \vdots & \ldots & \vdots \\ \sigma'(\langle w_k, x_1 \rangle) \cdot x_1 & \sigma'(\langle w_k, x_2 \rangle) \cdot x_2 & \ldots & \sigma'(\langle w_k, x_n \rangle) \cdot x_n \end{bmatrix}$

1-layer leaky ReLU Neural Networks

- Let us assume we have a 1-layer linear network
- The prediction of this network is given as $h(W; x) = \langle v, \sigma(Wx) \rangle$

$\sigma(x) = \mathbf{1}_{x \geq 0} + \epsilon \mathbf{1}_{x \leq 0}$



assume output edges are all fixed

reminder:
$$\| \nabla_w L(w) \|^2 \geq 4\lambda_{\min}(J(w)^T J(w)) \| h(X; w) - y \|$$

The Jacobian is equal to $J(w) = \mathrm{diag}(v_1 I_d, \ldots, v_k I_d) \begin{bmatrix} \sigma'(\langle w_1, x_1 \rangle) \cdot x_1 & \sigma'(\langle w_1, x_2 \rangle) \cdot x_2 & \ldots & \sigma'(\langle w_1, x_n \rangle) \cdot x_n \\ \vdots & \vdots & \ldots & \vdots \\ \sigma'(\langle w_k, x_1 \rangle) \cdot x_1 & \sigma'(\langle w_k, x_2 \rangle) \cdot x_2 & \ldots & \sigma'(\langle w_k, x_n \rangle) \cdot x_n \end{bmatrix}$

Note $\mathrm{rank}(J(w)) = n$ if the rank of the data matrix is n and also at least one activation has nonzero derivate for all models.

Next time: More result on NNs/NTK/ Overparameterization

reading list

Karimi, H., Nutini, J. and Schmidt, M., 2016, September. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases
http://www.optimization-online.org/DB_FILE/2016/08/5590.pdf

Soudry, D. and Carmon, Y., 2016. No bad local minima: Data independent training error guarantees for multilayer neural networks. arXiv preprint arXiv:1605.08361.
https://arxiv.org/pdf/1605.08361

Du, S.S., Zhai, X., Poczos, B. and Singh, A., 2018. Gradient descent provably optimizes over-parameterized neural networks. ICLR 2019
https://arxiv.org/pdf/1810.02054

Allen-Zhu, Z., Li, Y. and Song, Z., 2019, May. A convergence theory for deep learning via over-parameterization. In International Conference on Machine Learning (pp. 242-252). PMLR.
http://proceedings.mlr.press/v97/allen-zhu19a/allen-zhu19a.pdf

Du, S., Lee, J., Li, H., Wang, L. and Zhai, X., 2019, May. Gradient descent finds global minima of deep neural networks. In International conference on machine learning (pp. 1675-1685). PMLR.
http://proceedings.mlr.press/v97/du19c/du19c.pdf

Liu, C., Zhu, L. and Belkin, M., 2022. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. Applied and Computational Harmonic Analysis.
Vancouver
https://arxiv.org/abs/2003.00307