Stochastic Gradient Descent Optimizes Over-parameterized Deep ReLU Networks

Aishwarya Mehta (amehta9@buffalo.edu)
Ayush Utkarsh (ayushutk@buffalo.edu)

Introduction

In this presentation, we try and explain the findings of the ICML 2019 paper Stochastic Gradient Descent Optimizes Over-Parameterized Deep RELU Networks. The intent of this presentation is to focus on the key takeaways of this paper so that everyone can utilize the learnings from it.

Layout

- Overview
- Related Work
- Conclusive Findings
- Implementational takeaways from these findings
- Setup of experiment/concept
- Assumptions (loss function, inputs, etc)
- Exponential bounds
- References

Overview

- Why do Gradient Descent (GD) and Stochastic Gradient Descent (SGD) work for over-parameterized training deep neural networks with RELU activation?
- What's overparameterization?
- How overparameterization helps?
- How does random weight initialization impact model convergence?

Relevant implementational findings: Related Work

- SGD can recover underlying parameters of a 2-layer residual network in Polynomial time. [Li and Yuan (2017)]
- Deep linear residual networks have no spurious local minima [Hardt and Ma (2016)]
- Depth can accelerate the optimization of deep linear networks [Arora et al. (2018b)]
- Identity initialization and proper regularizer helps GD converge to the least square solution for deep linear network. [Arora et al. (2018a)]

Findings

- GD & SGD can find global minima of train loss for an over-parameterized deep RELU net under **mild** data assumption.
  - What Assumptions?
    - **Data separation assumption**
- Gaussian random initialization with (S)GD produces a sequence of iterations that stay in small perturbations around init weight.
- Empirical loss of deep RELU has nice local curvature properties ensuring global convergence of (S)GD.

Implementational Take Away

- Gaussian random initialization can achieve **zero training loss** with (S)GD within ***O(poly(n, ϕ-1,L))*** iterations if number of nodes per layer is **atleast *Ω(poly(n, ϕ-1,L))***
  - This finding gives us the requirement of over-parameterization.

  Assumptions:

  - Only one: Data separation

SETUP

- L-hidden layer neural network:

$$f_{\mathbf{W}}(\mathbf{x}) = \mathbf{v}^\top \sigma(\mathbf{W}_L^\top \sigma(\mathbf{W}_{L-1}^\top \cdots \sigma(\mathbf{W}_1^\top \mathbf{x}) \cdots))$$

- Empirical risk minimization problem:

$$\min_{\mathbf{W}} L_S(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i \widehat{y}_i)$$

Loss Function Assumptions

- Loss function *l(.)* is continuous and satisfies :

$$\ell'(x) \leqslant 0, \lim_{x\to\infty} \ell(x) = 0$$

$$\lim_{x\to\infty} \ell'(x) = 0.$$

- Loss function is ***λ*-smooth**

  λ-smooth?

- **Gaussian Initialization:** each column of **W** is independently gaussian:
  - ***N(0, 2/m (Eye))***

Input Assumptions

- $\|\mathbf{x}_i\|_2 = 1$ and $(\mathbf{x}_i)_d = \mu$ for all $i \in \{1, \ldots, n\}$, where $\mu \in (0, 1)$ is a constant

- For all $i, i' \in \{1, \ldots, n\}$, if $y_i \neq y_{i'}$, then $\|\mathbf{x}_i - \mathbf{x}_{i'}\|_2 \geqslant \phi$ for some $\phi > 0$.

Gaussian Initialization Assumptions

- The following assumptions were taken to hold true under gaussian initialization

(i) $\big|\|\mathbf{x}_{l,i}\|_2 - 1\big| \leqslant \overline{C}' L\sqrt{\log(nL/\delta)/m}$, $\|\mathbf{W}_l\|_2 \leqslant \overline{C}'$ for all $l = 1 \ldots, L$ and $i = 1, \ldots, n$.

(ii) $\big\|\|\mathbf{x}_{l,i}\|_2^{-1}\mathbf{x}_{l,i} - \|\mathbf{x}_{l,i'}\|_2^{-1}\mathbf{x}_{l,i'}\big\|_2 \geqslant \phi/2$ for all $l = 1, \ldots, L$ and $i, i' \in \{1, \ldots, n\}$ such that $y_i \neq y_{i'}$.

(iii) $|\widehat{y}_i| \leqslant \overline{C}'\sqrt{\log(n/\delta)}$ for all $i = 1, \ldots, n$.

(iv) $\big|\{j \in [m_l] : |\langle \mathbf{w}_{l,j}, \mathbf{x}_{l-1,i}\rangle| \leqslant \beta\}\big| \leqslant 2m_l^{3/2}\beta$ for all $l = 1, \ldots, L$ and $i = 1, \ldots, n$.

(v) $\big\|\mathbf{W}_{l_2}^\top \big(\prod_{r=l_1}^{l_2-1} \mathbf{\Sigma}_{r,i}\mathbf{W}_r^\top\big)\big\|_2 \leqslant \overline{C}' L$ for all $1 \leqslant l_1 < l_2 \leqslant L$ and $i = 1, \ldots, n$.

(vi) $\mathbf{v}^\top \big(\prod_{r=l}^{L} \mathbf{\Sigma}_{r,i}\mathbf{W}_r^\top\big)\mathbf{a} \leqslant \overline{C}' L\sqrt{s\log(M)}$ for all $l = 1, \ldots, L$, $i = 1, \ldots, n$ and all $\mathbf{a} \in S^{m_{l-1}-1}$ with $\|\mathbf{a}\|_0 \leqslant s$.

(vii) $\mathbf{b}^\top \mathbf{W}_{l_2}^\top \big(\prod_{r=l_1}^{l_2-1} \mathbf{\Sigma}_{r,i}\mathbf{W}_r^\top\big)\mathbf{a} \leqslant \overline{C}' L\sqrt{s\log(M)/m}$ for all $l = 1, \ldots, L$, $i = 1, \ldots, n$ and all $\mathbf{a} \in S^{m_{l_1-1}-1}$, $\mathbf{b} \in S^{m_{l_2}-1}$ with $\|\mathbf{a}\|_0, \|\mathbf{b}\|_0 \leqslant s$.

(viii) For any $\mathbf{a} = (a_1, \ldots, a_n)^\top \in \mathbb{R}_+^n$, there exist at least $\underline{C}' m_L \phi/n$ nodes satisfying

$$\left\|\frac{1}{n}\sum_{i=1}^{n} a_i \sigma'(\langle \mathbf{w}_{L,j}, \mathbf{x}_{L-1,i}\rangle)\mathbf{x}_{L-1,i}\right\|_2 \geqslant \underline{C}''\|\mathbf{a}\|_\infty/n.$$

Perturbation Assumptions

- Given the gaussian initialization follows the above assumptions, the authors showed that the perturbations created would be bounded by the following rules:

(i) $\|\widetilde{\mathbf{W}}_l\|_2 \leqslant \overline{C}$ for all $l \in [L]$.

(ii) $\|\hat{\mathbf{x}}_{l,i} - \tilde{\mathbf{x}}_{l,i}\|_2 \leqslant \overline{C}L \cdot \sum_{r=1}^{l} \|\widehat{\mathbf{W}}_r - \widetilde{\mathbf{W}}_r\|_2$ for all $l \in [L]$ and $i \in [n]$.

(iii) $\|\widehat{\mathbf{\Sigma}}_{l,i} - \widetilde{\mathbf{\Sigma}}_{l,i}\|_0 \leqslant \overline{C}L^{4/3}\tau^{2/3}m_l$ for all $l \in [L]$ and $i \in [n]$.

(iv) $\left|\{j \in [m_L] : \text{there exists } i \in [n] \text{ such that } (\widetilde{\mathbf{\Sigma}}_{L,i} - \mathbf{\Sigma}_{L,i})_{jj} \neq 0\}\right| \leqslant \overline{C}nL^{4/3}\tau^{2/3}m_L$.

(v) $\left\|\prod_{r=l_1}^{l_2} \widetilde{\mathbf{\Sigma}}_{r,i}\widetilde{\mathbf{W}}_r^\top\right\|_2 \leqslant \overline{C}L$ for all $1 \leqslant l_1 < l_2 \leqslant L$.

(vi) $\mathbf{v}^\top\left(\prod_{r=l}^{L} \widetilde{\mathbf{\Sigma}}_{r,i}\widetilde{\mathbf{W}}_r^\top\right)\mathbf{a} \leqslant \overline{C}' L^{5/3}\tau^{1/3}\sqrt{M\log(M)}$ for all $\mathbf{a} \in \mathbb{R}^{m_{l-1}}$ satisfying $\|\mathbf{a}\|_2 = 1$, $\|\mathbf{a}\|_0 \leqslant \overline{C}L^{4/3}\tau^{2/3}m_l$ and any $1 \leqslant l \leqslant L$.

Findings: asymptotic bounds

- $\|\,.\,\|_2$- perturbations on Gaussian initialization within a radius t has good local curvature properties.

$$\|\nabla_{\mathbf{W}_L}[L_S(\widetilde{\mathbf{W}})]\|_F^2 \geqslant \underline{C}'\frac{m_L\phi}{n^5}\left(\sum_{i=1}^{n}\ell'(y_i\widetilde{y}_i)\right)^2.$$

  - This gradient lower bound gives that within perturbation region, empirical loss of deep NN has good local curvature properties.
- Assumption that all perturbations are within t radius from init gives a condition on iterations *k* * step-size *η* for convergence guarantee.

$$\|\nabla_{\mathbf{W}_l}[L_S(\widetilde{\mathbf{W}})]\|_2 \leqslant -\frac{\overline{C}L^2M^{1/2}}{n}\sum_{i=1}^{n}\ell'(y_i\widetilde{y}_i) \text{ and } \|\tilde{\mathbf{G}}_l\|_2 \leqslant -\frac{\overline{C}L^2M^{1/2}}{B}\sum_{i\in\mathcal{B}}\ell'(y_i\widetilde{y}_i),$$

  - This gradient upper bound quantifies how much weights would change during (S)GD. This guarantees that weights won't escape from the perturbation region during training.

Findings: asymptotic bounds

- While *k* * *η* < ***T*** (constant), gradient descent with *k* iterations remains in pert. region around Gauss. Initialization:

$$T = O(L^{-4}n^{-3}\tau^2\phi) = O(L^{-38}n^{-21}\phi^7)$$

- Lower bound on hidden nodes per layer:

$$m = \begin{cases} \widetilde{\Omega}\big(n^{26}L^{38}/\phi^8\big) & 0 \leqslant p < \frac{1}{2} \\ \widetilde{\Omega}\big(n^{26}L^{38}/\phi^8\big) + \widetilde{\Omega}\big(n^{25}L^{38}/\phi^8\big) \cdot \Omega\big(\log(1/\epsilon)\big) & p = \frac{1}{2} \\ \widetilde{\Omega}\big(n^{26-2p}L^{38}/\phi^8\big) + \widetilde{\Omega}\big(n^{26}L^{38}/\phi^8\big) \cdot \Omega(\epsilon^{1-2p}) & \frac{1}{2} < p \leqslant 1. \end{cases}$$

Where ***p*** is an exponential factor on loss such that $-\ell'(x) \geqslant \min\{\alpha_0, \alpha_1 \ell^p(x)\}$

Findings: asymptotic bounds

- Similarly, we get a upper bound on the maximum number of iterations to be:

$$K = \begin{cases} \widetilde{O}\big(n^{12-2p}B^{-2}L^9\phi^{-2}\big) & 0 \leqslant p < \frac{1}{2} \\ \widetilde{O}\big(n^{11}B^{-2}L^9\phi^{-2}\big) + \widetilde{O}\big(n^{10}B^{-2}L^9\phi^{-2}\big) \cdot O\big(\log(1/\epsilon)\big) & p = \frac{1}{2} \\ \widetilde{O}\big(n^{12-2p}B^{-2}L^9\phi^{-2}\big) + \widetilde{O}\big(n^{12-4p}B^{-2}L^9\phi^{-2}\big) \cdot O\big(\epsilon^{1-2p}\big) & \frac{1}{2} < p \leqslant 1 \end{cases}$$

Findings: Stochastic Gradient Descent

- In case of stochastic gradient descent, we have number of hidden nodes per layer as:

$$m = \begin{cases} \widetilde{\Omega}\big(\mathrm{poly}(n, \phi^{-1}, L)\big) & 0 \leqslant p < \frac{1}{2} \\ \widetilde{\Omega}\big(\mathrm{poly}(n, \phi^{-1}, L)\big) \cdot \Omega\big(\log^2(1/\epsilon)\big) & p = \frac{1}{2} \\ \widetilde{\Omega}\big(\mathrm{poly}(n, \phi^{-1}, L)\big) \cdot \Omega(\epsilon^{2-4p}) & \frac{1}{2} < p \leqslant 1, \end{cases}$$

Findings: Stochastic Gradient Descent

- Number of iterations have an asymptotic upper limit of

$$K = \begin{cases} \tilde{O}\big(\mathrm{poly}(n, \phi^{-1}, L)\big) & 0 \leqslant p < \frac{1}{2} \\ \tilde{O}\big(\mathrm{poly}(n, \phi^{-1}, L)\big) \cdot O\big(\log(1/\epsilon)\big) & p = \frac{1}{2} \\ \tilde{O}\big(\mathrm{poly}(n, \phi^{-1}, L)\big) \cdot O(\epsilon^{1-2p}) & \frac{1}{2} < p \leqslant 1, \end{cases}$$

Conclusion

- This paper studied training deep neural networks by gradient descent and stochastic gradient descent.
- The authors proved that both gradient descent and stochastic gradient descent can achieve global minima of over-parameterized deep ReLU networks with random initialization.
- This holds for a general class of loss functions, with only mild assumption on training data

Reference

- Stochastic Gradient Descent Optimizes Over-parameterized Deep ReLU Networks (ICML 2019) (1811.08888.pdf (arxiv.org))
- Gradient Descent Provably Optimizes Over-parameterized Neural Networks (Gradient Descent Provably Optimizes Over-parameterized Neural Networks | OpenReview)

Q & A ?

Thank you