Linearly Convergent Algorithms for Learning Shallow Residual Networks

Gauri Jagatap and Chinmay Hegde

Electrical and Computer Engineering
Iowa State University

July 11, 2019

Introduction

Objective: To introduce and analyze algorithms for learning shallow ReLU based neural network mappings.

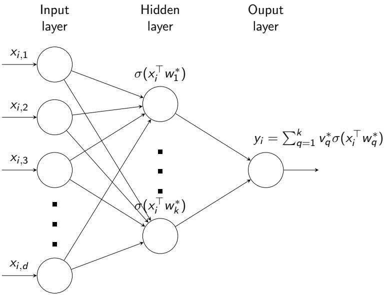
Main Challenges:

- Limited algorithmic guarantees for (stochastic) gradient descent.
- Gradient descent requires the learning rate to be tuned appropriately.
  - Small enough learning rate may guarantee local convergence but requires high running time.
- Problem is typically non-convex; global convergence is not guaranteed unless network is initialized appropriately.

Objective

We analyze the problem of learning the weights of a two-layer *teacher* network with:

- d-dimensional input samples x_i (n such), stacked in matrix X,



- forward model: $f^*(X) = \sum_{q=1}^{k} v_q^* \sigma(X w_q^*) = \sigma(XW^*)v^*$,
- layer 1 weights $W^* := [w_1^* \dots w_q^* \dots w_k^*] \in \mathbb{R}^{d \times k}$, k-hidden neurons,
- fixed weights in layer 2, $v^* = [v_1^* \dots v_q^* \dots v_k^*]^\top \in \mathbb{R}^k$, such that $v_q^* \in \{+1, -1\}$.

Our Formulation

Skipped connections

A special formulation of this problem is when there is a *skipped connection* between the network output and input.

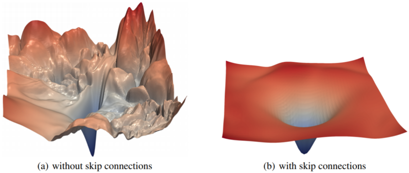


Figure: Li et. al. "Visualizing the Loss Landscape of Neural Nets."

- $W^* \in \mathbb{R}^{d \times d}$ is a square matrix with $k = d$ columns.
- The effective forward model: $f^*_{res}(X) = \sigma(X(W^* + \mathbf{I}))v^*$,
- Additionally, elements of X are assumed to be distributed as i.i.d Gaussian $\mathcal{N}(0, 1/n)$.
  Note: We also assume that a fresh batch of samples is drawn in each iteration of given training algorithm to simplify theoretical analysis.

Our Formulation

Observation: ReLU is a piece-wise linear transformation. One can introduce a "linearization" mapping as follows.

- let e_q represent the q^{th} column of identity matrix $\mathbf{I}_{d\times d}$
- diagonal matrix $\mathbb{P}_q = \mathrm{diag}(\mathbb{1}_{\{X(w_q+e_q)>0\}}), \forall q$ stores the state of q^{th} hidden neuron for all samples.

Then,

$$y = f^*_{res}(X) = [v^*_1\mathbb{P}^*_1X \dots v^*_d\mathbb{P}^*_dX]_{n\times d^2} \cdot \mathrm{vec}(W^* + \mathbf{I})_{d^2\times 1},$$
$$:= B^* \cdot \mathrm{vec}(W^* + \mathbf{I}).$$

Note: that the mapping is not truly linear in the weights $(W^* + \mathbf{I})$, as B^* depends on W^*.

The loss is:

$$\mathcal{L}(W^t) = \frac{1}{2n}\|y - B^t \cdot \mathrm{vec}(W^t + \mathbf{I})\|_2^2$$

where $B^t = [v^*_1\mathbb{P}^t_1X \dots v^*_d\mathbb{P}^t_qX \dots v^*_d\mathbb{P}^t_dX]$.

Prior Work

Table: $\mathcal{O}_\epsilon(\cdot)$ *hides polylogarithmic dependence on* $\frac{1}{\epsilon}$. *Alternating Minimization and (Stochastic) Gradient descent are denoted as AM and (S)GD respectively. "*" indicates re-sampling assumption.*

Alg.	Paper	Sample complexity	Convergence rate	Initialization	Type	Parameters
SGD	[1]	$\times$ (population loss)	$\mathcal{O}_\epsilon\left(\frac{1}{\epsilon}\right)$	Random	ReLU ResNets	step-size η
GD	[2]	$\times$ (population loss)	$\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$	Identity	Linear	step-size η
GD*	[3]	$\mathcal{O}_\epsilon\left(dk^2 \cdot \mathrm{poly}(\log d)\right)$	$\mathcal{O}_\epsilon\left(\log\frac{1}{\epsilon}\right)$	Tensor	Smooth (not ReLU)	step-size η
GD	[4]	$\mathcal{O}_\epsilon\left(dk^9 \cdot \mathrm{poly}(\log d)\right)$	$\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$	Tensor	ReLU	step-size η
GD*	(this paper)	$\mathcal{O}_\epsilon\left(dk^2 \cdot \mathrm{poly}(\log d)\right)$	$\mathcal{O}_\epsilon\left(\log\frac{1}{\epsilon}\right)$	Identity	ReLU ResNets	step-size η
AM*	(this paper)	$\mathcal{O}_\epsilon\left(dk^2 \cdot \mathrm{poly}(\log d)\right)$	$\mathcal{O}_\epsilon\left(\log\frac{1}{\epsilon}\right)$	Identity	ReLU ResNets	none

[1] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with relu activation," in *Advances in Neural Information Processing Systems*, pp. 597–607, 2017.

[2] P. Bartlett, D. Helmbold, and P. Long, "Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks," *arXiv preprint arXiv:1802.06093*, 2018.

[3] K. Zhong, Z. Song, P. Jain, P. Bartlett, and I. Dhillon, "Recovery guarantees for one-hidden-layer neural networks," in *International Conference on Machine Learning*, pp. 4140–4149, 2017.

[4] X. Zhang, Y. Yu, L. Wang, and Q. Gu, "Learning one-hidden-layer relu networks via gradient descent," *Proc. Int. Conf. Art. Intell. Stat. (AISTATS)*, 2018.

Gradient descent

Local linear convergence

Gradient of loss:

$$\nabla \mathcal{L}(W^t) = -\frac{1}{n} B^{t\top}(y - B^t \cdot \mathrm{vec}(W^t + \mathbf{I})).$$

The gradient descent update rule is as follows:

$$\begin{aligned} \mathrm{vec}(W^{t+1}) &= \mathrm{vec}(W^t) - \eta \nabla \mathcal{L}(\mathrm{vec}(W^t)) \\ &= \mathrm{vec}(W^t) + \frac{\eta}{n} B^{t\top}(y - B^t \, \mathrm{vec}(W^t + \mathbf{I})), \end{aligned} \tag{1}$$

where η is appropriately chosen step size and

Alternating minimization

Local linear convergence

Alternating minimization framework:

- linearize network by estimating $B^{t'}$,

$$B^{t'} = [v_1^* \mathrm{diag}(\mathbb{1}_{X(w_1^{t'}+e_1)})X \ldots v_d^* \mathrm{diag}(\mathbb{1}_{X(w_d^{t'}+e_d)})X], \quad (2)$$

- estimate weights $W^{t'+1}$ of linearized model,

$$\mathrm{vec}(W^{t'+1}) = \arg\min_{\mathrm{vec}(W)} \left\| B^{t'} \cdot \mathrm{vec}(W + \mathbf{I}) - y \right\|_2^2, \quad (3)$$

This paper:
Linear local convergence guarantees for both gradient descent (update rule (1)) and alternating minimization (update rule (3)).

Guarantees: Theorem 1

Given an initialization W^0 satisfying $\|W^0 - W^*\|_F \leq \delta \|W^* + \mathbf{I}\|_{\mathsf{F}}$, for $0 < \delta < 1$, if we have number of training samples $n > C \cdot d \cdot k^2 \cdot \text{poly}(\log k, \log d, t)$, then with high probability $1 - ce^{-\alpha n} - d^{-\beta t}$, where c, α, β are positive constants and $t \geq 1$, the iterates of Gradient Descent (1) satisfy:

$$\left\|W^{t+1} - W^*\right\|_{\mathsf{F}} \leq \rho_{GD} \left\|W^t - W^*\right\|_{\mathsf{F}}. \tag{4}$$

and the iterates of Alternating Minimization (3) satisfy:

$$\left\|W^{t+1} - W^*\right\|_{\mathsf{F}} \leq \rho_{AM} \left\|W^t - W^*\right\|_{\mathsf{F}}. \tag{5}$$

where and $0 < \rho_{AM} < \rho_{GD} < 1$.

- How do we ensure the initialization requirement?
  - (Assumption 1) the architecture satisfies $\|W^*\|_{\mathsf{F}} \leq \gamma \leq \frac{\delta\sqrt{d}}{1+\delta}$, then $W^0 = \mathbf{0}$ satisfies requirement (identity initialization).

Guarantees

Gradient descent

Using update rule (1) and taking the Frobenius normed difference between the learned weights and the weights of the teacher network,

$$\begin{aligned}
&\left\| W^{t+1} - W^* \right\|_F \\
&\le \left\| \mathbf{I} - \frac{\eta}{n}(B^{t\top} B^t) \right\|_2 \left\| W^t - W^* \right\|_F + \left\| \frac{B^{t\top}}{\sqrt{n}} \right\|_2 \left\| \frac{1}{\sqrt{n}}(B^* - B^t)\, \mathrm{vec}(W^* + \mathbf{I}) \right\|_2 , \\
&\le \frac{\sigma_{max}^2 - \sigma_{min}^2}{\sigma_{max}^2 + \sigma_{min}^2} \left\| W^t - W^* \right\|_F + \eta \sigma_{max} \sum_{q=1}^{k} \left\| E_q \right\|_2 , \\
&= \rho_4 \left\| W^t - W^* \right\|_F + \eta \sigma_{max} \rho_3 \left\| W^t - W^* \right\|_F = \rho_{GD} \left\| W^t - W^* \right\|_F , \\
&\quad (\textit{via Lemma } 1) \qquad\qquad (\textit{via Lemma } 2)
\end{aligned}$$

where $E_q := (B^t - B^*)\, \mathrm{vec}(W^* + \mathbf{I})/\sqrt{n}$ (error due to non-linearity of ReLU) and $\sigma_{min}, \sigma_{max}$ are the minimum and maximum singular values of $\frac{B^t}{\sqrt{n}}$.

$\implies \rho_{GD} = \frac{\kappa - 1}{\kappa + 1} + \frac{2\kappa\rho_3}{\sigma_{max} \cdot (\kappa + 1)}$, with $\kappa = \frac{\sigma_{max}^2}{\sigma_{min}^2}$.

Guarantees

Alternating minimization

Since the minimization in (3) can be solved exactly, we get:

$$\begin{aligned}
\mathrm{vec}(W^{t'+1} + \mathbf{I}) &= (B^{t\top} B^{t'})^{-1} B^{t'\top} y \\
&= (B^{t'\top} B^{t'})^{-1} B^{t'\top} B^{*}\, \mathrm{vec}(W^{*} + \mathbf{I}) \\
&= \mathrm{vec}(W^{*} + \mathbf{I}) + (B^{t'\top} B^{t'})^{-1} B^{t'\top} (B^{*} - B^{t'})\, \mathrm{vec}(W^{*} + \mathbf{I}).
\end{aligned}$$

Taking the Frobenius normed difference between the learned weights and the weights of the teacher network,

$$\begin{aligned}
\left\| W^{t+1} - W^{*} \right\|_{\mathsf{F}} &= \left\| (B^{\top} B)^{-1} B^{\top} (B^{*} - B^{t})\, \mathrm{vec}(W^{*} + \mathbf{I}) \right\|_{2}, \\
&\leq \left\| n (B^{\top} B)^{-1} \right\|_{2} \left\| \frac{B^{\top}}{\sqrt{n}} \right\|_{2} \left\| \frac{1}{\sqrt{n}} (B^{*} - B^{t})\, \mathrm{vec}(W^{*} + \mathbf{I}) \right\|_{2}, \\
&\leq \frac{\sigma_{max}}{\sigma_{min}^{2}} \cdot \rho_{3} \left\| W^{t} - W^{*} \right\|_{\mathsf{F}} < \rho_{AM} \left\| W^{t} - W^{*} \right\|_{\mathsf{F}} \\
&\quad \textit{(via Lemmas 1 and 2)}
\end{aligned}$$

$\implies \rho_{AM} = \frac{\kappa \rho_{3}}{\sigma_{max}}$, with $\kappa = \frac{\sigma_{max}^{2}}{\sigma_{min}^{2}}$.

Guarantees: Lemma 1 (borrowed from [4])

If singular values of $W^* + \mathbf{I}$, and the condition numbers κ_w and λ are defined as $\sigma_1 \geq \cdots \geq \sigma_k$, $\kappa_w = \frac{\sigma_1}{\sigma_k}$ and $\lambda = \prod_{q=1}^{k} \sigma_q / \sigma_k^k$,

$$\text{then,} \quad \Omega(1/(\kappa_w^2 \lambda)) \leq \frac{1}{n} \sigma_{min}^2(B) \leq \frac{1}{n} \sigma_{max}^2(B) \leq O(k),$$

as long as $\|W - W^*\|_2 \lessapprox \frac{1}{k^2 \kappa_w^5 \lambda^2} \|W^* + \mathbf{I}\|_2$ and $n \geq d \cdot k^2 \text{poly}(\log d, t, \lambda, \kappa_w)$, w.p. at least $1 - d^{-\Omega(t)}$.

Note: (Assumption 2) Lemma 1 requires fresh samples X be used in each iteration of the algorithm.

Guarantees: Lemma 2 (this paper)

As long as $\|W^0 - W^*\| \leq \delta_0 \|W^* + \mathbf{I}\|$, w.p. at least $1 - e^{-\Omega(n)}$, and $n > C \cdot d \cdot k^2 \cdot \log k$, the following holds:

$$\sum_{q=1}^{k} \|E_q\|_2^2 = \frac{1}{n} \sum_{i,q=1}^{n,k} \left(x_i^\top (w_q^* + e_q) \right)^2 \cdot \mathbf{1}_{\left\{ (x_i^\top (w_q^t + e_q))(x_i^\top (w_q^* + e_q)) \leq 0 \right\}}$$

$$\leq \rho_3^2 \|W^t - W^*\|_F^2,$$

Note: (Assumption 3) Lemma 2 requires balanced column norms of W^* : $c(\frac{\gamma^2}{d}) \leq \|w_q^*\|_2^2 \leq C(\frac{\gamma^2}{d})$ for positive constants c, C for all q. Lemma analysis borrows from techniques from phase retrieval literature.

Comparison

Theoretical:

From previous derivation, $\rho_{GD} = \frac{\kappa-1}{\kappa+1} + \frac{2\rho_{AM}}{\kappa+1}$.

- Alternating minimization exhibits faster convergence!

#Epochs T_{GD} and T_{AM} for ϵ-accuracy satisfy $\frac{T_{GD}}{T_{AM}} = \frac{\log(1/\rho_{AM})}{\log(1/\rho_{GD})}$.

Experimental:

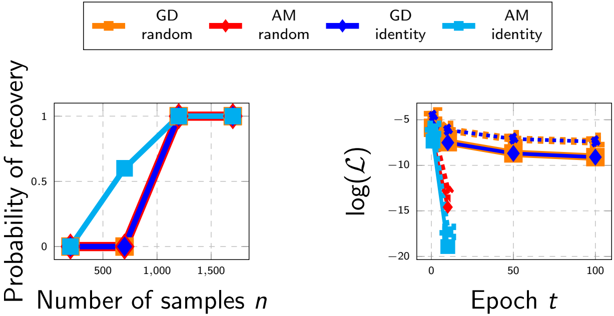


Figure: (left) Successful parameter recovery averaged on 10 trials for $d = 20$, with identity and random initializations; (right) training (solid) and testing (dotted) losses for fixed trial with $n = 1700$.

Conclusion and future directions

Conclusions:

- Introduced alternating minimization framework for training neural networks, which gives faster convergence.
- Local linear convergence analysis for gradient descent and alternating minimization.
- Performance comparison under specific assumptions on neural network architecture.

Future directions:

- Removing assumptions on data.
- Global convergence guarantees with random initialization.
- Extending alternating minimization approach to multiple layers.