이학석사 학위논문

Input Distribution Analysis for Learning Two Layer Neural Network

(2층 신경망의 학습을 위한 입력 분포 분석)

2019년 2월

서울대학교 대학원

수 리 과 학 부

박 정 명

Input Distribution Analysis for Learning Two Layer Neural Network

(2층 신경망의 학습을 위한 입력 분포 분석)

지도교수 강 명 주

이 논문을 이학석사 학위논문으로 제출함

2018년 10월

서울대학교 대학원

수 리 과 학 부

박 정 명

박 정 명의 이학석사 학위논문을 인준함

2018년 12월

위 원 장 _____ (인)

부 위 원 장 _____ (인)

위 원 _____ (인)

Input Distribution Analysis for Learning Two Layer Neural Network

by

Jeongmyeong Park

A DISSERTATION

Submitted to the faculty of the Graduate School
in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mathematical Sciences
Seoul National University
February 2019

Abstract

Input Distribution Analysis for Learning Two Layer Neural Network

Jeongmyeong Park

Department of Mathematical Sciences
The Graduate School
Seoul National University

In recent years, Deep neural networks have achieved state-of-the-art performance in many tasks. Despite those empirical successes, it remains hard to explain why stochastic gradient descent can solve the highly non-convex optimization problem. In this paper, we analyze the convergence of stochastic gradient descent for weights in learning process. The model we used is two layer neural network with ReLU activation function. In particular, our theory used the notion of the smoothness of the input distribution so that we don't need any specific input distribution.

Key words: Optimization, Input distribution, Neural Network, Deep Learning
Student Number: 2016-20235

Contents

Chapter 1

Introduction

In recent years, Deep neural networks have achieved state-of-the-art performance in many tasks such as computer vision [6], natural language processing [2], speech recognition [4]. Despite those empirical successes, it remains difficult to explain the performance theoretically. One of the theoretical problems is the optimization. To optimize the neural network, We use stochastic gradient descent(SGD). SGD is a simple algorithm using first order which is effective and easy to modify. Even though there are many different optimizers used in real works, all of them are based on SGD. Why such a simple algorithm can be so successful? Because of the simplicity of SGD, one may think that it won't be hard to explain it. But it is wrong. Deep neural networks are the highly non-convex optimization problems. Not only because it is deep, people usually adopt ReLU function as their activation function. So it is almost impossible to analyze all of the aspect in deep network. Although it is a difficult problem, researchers have tried many theoretical attempts to analyze it. Recently, there have been many papers which try to explain the convergence of shallow networks. Zhang et al.[7] analyzed two layer network on certain types of activation functions, but generalization on ReLU function remained unclear. Brutzkus et al. [1] showed optimization of convolutional network with ReLU function and gaussian input can be proved. Janzamin et al. [5] guarantees the reexery of parameters on 2-layered network with tensor decomposition. However, most of those attempts relies on the certain input distribution, like gaussian distribution. Instead of the specific form of input

distribution, Du et al. [3] used the notion of smoothness of the input distribution. Their analysis shows that any kind of inputs can be optimized with some conditions. In this paper, we will follow the similar method. We build a shallow network and discuss about its convergence. More precisely, we are going to follow the Idea of Du et al. [3]. Their suggestion of smoothness will be used and there will be no assumption of input distribution except the smoothness. Since his idea is constrained on one layer convolutional network, we will adapt the notion of smoothness on two layer network with ReLU activation function. With the notion of smoothness, We are going to connect it to the convergence rate for weights recovery. The smoother input distribution is, the faster convergence happens. The content of this paper is organized as follows. We introduce our network and apply Du's idea in cases that only one weight vector is used in **Chapter 2**. We generalize the idea in cases when multiple weight vectors are used in **Chapter 3**. Conclusions are in **Chapter 4**.

Chapter 2

In case of n=1

Now, we introduce our two layer neural network. We will analyze the optimization on this architecture. We call $z \in \mathbb{R}^k$ as an input vector, $\mathbf{W} = (w_1, w_2, ...w_n) \in \mathbb{R}^{k \times n}$ and $w_i \in \mathbb{R}^k$ as weights. $\sigma(x) = max(x, 0)$ is the ReLU activation function. In this paper, we focus on the following two layer ReLU neural network.

$$f(\mathbf{W}, z) = \frac{1}{k} \sum_{i=1}^{k} \sigma\left(w_i^\top z\right) \tag{2.1}$$

After multiplication of input and weights, ReLU activation function follows. Lastly average pooling derives output. For Loss function,

$$\ell(w, z) = \frac{1}{2} \left(f(w, z) - f\left(w_*, z\right)\right)^2 \tag{2.2}$$

When we apply stochastic gradient descent in real experiment, it works a s follows

$$w_{t+1} = w^t - \eta^t \nabla \ell(w^t, z) \tag{2.3}$$

However, z is not a fixed input. It is a random variable of input distribution. So we use population gradient in analysis.

$$w_{t+1} = w^t - \eta^t \mathbb{E}[\nabla \ell(w^t, z)] \tag{2.4}$$

Before we start the discussion, we clarify some notations. $\lambda_{max}(R)$ is the largest singular value of the matrix R and $\lambda_{min}(R)$ is the smallest singular

value of the matrix R. $\|\cdot\|_{oper}$ denotes the operator norm of a matrix. For the convenience in the calculation, we will use vector $mathbfW_{vec}$ instead of matrix $\mathbf{W}$. The definition is as follows. $W_{vec} = [w_1^\top, w_2^\top, ...w_n^\top] \in \mathbb{R}^{kn}$ Also we make some assumptions. Assume that inputs, weights are bounded. As a result, gradient functions are uniformly bounded.

In our paper, activation function is ReLU function. ReLU function is hard to deal because of its vanishing properties. To solve the problem with minimum assumptions, we split regions so that we can verify the vanishment by ReLU function. Since our network is shallow network, it is affordable calculations. Also using following definitions, we will define the smoothness of input distribution.

Definition 2.1. *(in Du[3]) We define two events and matrices. w can be any w_i in W_{vec}*

$$\begin{aligned} R(w, w_*) &= \left\{z : w^\top z \geq 0, w_*^\top z \geq 0\right\} \\ R(w, -w_*) &= \left\{z : w^\top z \geq 0, -w_*^\top z \geq 0\right\} \\ \mathbf{R}_{w,w_*} &= \mathbb{E}\left[zz^\top \mathbb{I}\left\{R(w, w_*)\right\}\right] \\ \mathbf{R}_{w,-w_*} &= \mathbb{E}\left[zz^\top \mathbb{I}\left\{R(w, -w_*)\right\}\right] \end{aligned} \tag{2.5}$$

Now it's time to introduce the smoothness in the case of n=1. We are going to generalize it for arbitrary n in Chapter 3.

Definition 2.2. *(in Du[3]) For ϕ in [0, π],*

$$\begin{aligned} m_{min}^{w,w_*}(\phi) &= \min_{w:\angle w,w_*=\phi} \lambda_{\min}(\mathbf{R}_{w,w_*}) \\ m_{max}^{w,w_*}(\phi) &= \max_{w:\angle w,w_*=\phi} \lambda_{\max}(\mathbf{R}_{w,w_*}) \end{aligned} \tag{2.6}$$

Du defined the smoothness as the diffrenece of the largest and smallest eigenvalue values of the $\mathbf{R}_{w,w_*}$ If the difference of the two eigenvalue is large, then we can consider it as the sign of the biased probability mass. When the input distribution is gaussian distribution or the rotationally invariant, Two eigenvalue will match by definition. So, closer two eigenvalues are, more even the probability mass is. Therefore the definition can be called the smoothness. This is the definition of smoothness when n=1. Du generalized the smoothness to use on the convolutional network. Since our network is not convolutional

network, we tried a different way of generalization. Before we discuss the two layer network for arbitrary n, we will analyze the case of n=1.

Theorem 2.3. *(in Du[3]) Assume that there exists positive d s.t.* $m_{max}^{w,-w_*}(\phi) = \max_{w:\angle w,w_*=\phi} \lambda_{\max}\left(\mathbf{R}_{w,-w_*}\right) < d\phi$. *and* $\|w_0 - w_*\|_2 < \|w_*\|_2$ *for the initialization* w_0. *Let* $\phi^t = \arcsin\left(\|w_0 - w_*\|_2 / \|w_*\|_2\right)$. *If* $0 \leq \eta_t \leq \min_{0\leq\phi\leq\phi_t} \frac{m_{min}(\phi)}{2(m_{max}(\phi)+2\alpha)^2}$, *then*

$$\|w_{t+1} - w_*\|_2^2 \leq \left(1 - \frac{\eta_t m_{min}\left(\phi_t\right)}{2}\right) \|w_t - w_*\|_2^2 \tag{2.7}$$

Now, by this theorem, we can sure that network will converge in a proper time. This theorem show the convergence rate of the n=1 network. It suggests the relationship between smoothness and converging time. With small α, small m_{max} and Large m_{min}, we can choose large learning rate. Also the bigger m_{min} is, the faster convergence comes.

Chapter 3

Generalization for arbitrary n

It is time to generalize for our network. Our network has two layers. In the hidden layer, each w_i's multiplies z and pass through the ReLU function. So when we investigate the gradient of loss function, we have to face the intersection of ReLU functions. Therefore, we need more complex split of regions.

Definition 3.1.

$$\begin{aligned} b_{i,j}^{k,l,m} &= zz^\top R\left(w_i, (-1)^k w_{i,*}\right) R\left((-1)^l w_j, (-1)^m w_{j,*}\right) \\ \mathbf{B}_{k,l,m} &= \frac{1}{n^2}\mathbb{E}\begin{bmatrix} b_{1,1}^{k,l,m} & \cdots & b_{1,n}^{k,l,m} \\ \vdots & \ddots & \vdots \\ b_{n,1}^{k,l,m} & \cdots & b_{n,n}^{k,l,m} \end{bmatrix} \end{aligned} \tag{3.1}$$

Definition 3.2.

$$\begin{aligned} m_{max}(\phi, w_{i,*}) &= \max_{\forall i, \angle w_i, w_{i,*} = \phi} \lambda_{max}(\mathbf{B}_{0,0,0}) \\ m_{min}(\phi, w_{vec,*}) &= \min_{\forall i, \angle w_i, w_i = \phi} \lambda_{min}(\mathbf{B}_{0,0,0}) \end{aligned} \tag{3.2}$$

Assumption 3.3. *We assume that exists positive α and m_{max}^{ex} such that*

$$\begin{aligned} &\max_{(k,l,m)=(1,0,0),(0,0,1),(0,1,0),(1,1,0)} \left\{ \max_{\forall i, \angle w_i, w_{i,*} = \phi} \lambda_{max}(\mathbf{B}_{k,l,m}) \right\} \leqq m_{max}^{ex}\phi \\ &max_{\forall i, \angle w_i, w_{i,*} = \phi} \lambda_{max}(\mathbf{B}_{1,0,1}) \leq \alpha\phi \end{aligned} \tag{3.3}$$

By converging n weights in one vector, we derived the very similar detail with Du's generalization on Convolutional Neural Network. So the following theorems are adaptation of his work on two layer neural network. By following theorem, we show the convergence guarantee of the two neural network in the following theorem.

Theorem 3.4. *(adapted from Du[3]) Assume that* $\|\mathbf{W}_{vec}^0 - \mathbf{W}_{vec,*}\|_2 < \|\mathbf{W}_{vec,*}\|_2$ *for the initialization* $\mathbf{W}_{vec}^0$*. Let* $\phi^t = \arcsin\left(\|\mathbf{W}_{vec}^t - \mathbf{W}_{vec,*}\|_2 / \|\mathbf{W}_{vec,*}\|_2\right)$*. If* $\eta_t \leq \min_{0\leq\phi\leq\phi_t} \frac{m_{min}(\phi)-10m_{max}^{ex}}{2(L(\phi)+14L_{ex}+4\alpha)^2}$*, then*

$$\left\|\mathbf{W}_{vec}^{t+1} - \mathbf{W}_{vec,*}\right\|_2^2 \leq \left\|\mathbf{W}_{vec}^{t} - \mathbf{W}_{*,vec}\right\|_2^2 \left(1 - \frac{\eta_t(m_{min} - 10m_{max}^{ex})}{2}\right)$$

Proof. In Appendix □

Now, we have our result. Population gradient leads the network to the convergence. For the faster convergence, we have to get a smaller m_{max}^{ex}, larger m_{min}, smaller L. Then we can have larger learning rate and better convergence rate. But we haven't show whether $m_{min} - 10m_{max}^{ex}$ is positive or not. Therefore, we introduce the next theorem.

Theorem 3.5. *(adapted from Du[3]) Suppose Z has unit norm.* ϕ *is angle between* $w_i, w_{i,*}$ *Assume that there exists* β *such that for all i*

$$\mathbb{P}\left[R(w_i, -w_{i,*})\right], \mathbb{P}\left[R(-w_i, w_{i,*})\right] \leq \beta\phi$$

then we have $m_{max}^{ex} \leq \beta\phi$

Proof. Assume $[x_1, x_2, ...x_n]$ has unit norm.

$$
\begin{aligned}
&[x_1, x_2, ...x_n]\mathbf{B}_{0,0,1}[x_1, x_2, ...x_n]^\top \\
&= \frac{1}{n^2}\sum E\left[x_i^\top Z Z^\top x_j R(w_i, w_{i,*})R(w_j, -w_{j,*})\right] \\
&\leq \frac{1}{n^2}\sum \|x_i\|\,\|x_j\|\,E\left[R(w_i, w_{i,*})R(w_j, -w_{j,*})\right] \\
&\leq \frac{1}{n^2}\sum \frac{1}{2}\left(\|x_i\|^2 + \|x_j\|^2\right)E\left[R(w_i, w_{i,*})R(w_j, -w_{j,*})\right] \\
&\leq \frac{1}{n^2}\sum \frac{1}{2}\left(\|x_i\|^2 + \|x_j\|^2\right)E\left[R(w_j, -w_{j,*})\right] \\
&\leq \frac{1}{n^2}\sum \frac{1}{2}\left(\|x_i\|^2 + \|x_j\|^2\right)\beta\phi \\
&= \frac{\beta\phi}{n}
\end{aligned}
$$

For $\mathbf{B}_{1,0,0}, \mathbf{B}_{0,1,0}, \mathbf{B}_{1,1,0}$, the same as above. □

Now we have upper bound of m_{max}^{ex}. If input distribution has low probability in $R(w_i, -w_{i,*})$, $R(-w_i, w_{i,*})$, then the assumptions above can be fulfilled.

Chapter 4

Conclusion

In this paper, we have discussed about the conditions on the convergence of two layer neural network. First, we discussed the case of n=1. We defined the smoothness and established theorem for convergence.Secondly, We generalized the smoothness for arbitrary n. Using the notion,we provided the conditions to guarantee the convergence for two layer neural network. While input distribution satisfies the conditions, it doesn't heed guassian distribution. Also, with the convergence rate we calculated, we can check the two layer neural network can learn weights in a proper time. For the future works, we have to think about how to expand the smoothness for deeper network. Basically, our method is based on the dividing the regions. If a network has more layers, there are too many regions to consider. So it is not likely to work on a deeper network.

Bibliography

[1] A. Brutzkus and A. Globerson, *Globally optimal gradient descent for a convnet with gaussian inputs*, arXiv preprint arXiv:1702.07966, (2017).

[2] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, *Language modeling with gated convolutional networks*, arXiv preprint arXiv:1612.08083, (2016).

[3] S. S. Du, J. D. Lee, and Y. Tian, *When is a convolutional filter easy to learn?*, arXiv preprint arXiv:1709.06129, (2017).

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*, IEEE Signal processing magazine, 29 (2012), pp. 82–97.

[5] M. Janzamin, H. Sedghi, and A. Anandkumar, *Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods*, arXiv preprint arXiv:1506.08473, (2015).

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Advances in neural information processing systems, 2012, pp. 1097–1105.

[7] X. Zhang, Y. Yu, L. Wang, and Q. Gu, *Learning one-hidden-layer relu networks via gradient descent*, arXiv preprint arXiv:1806.07808, (2018).

Chapter 5

appendix

5.1 Proof of Theorem 3.4

Assume that $\|\mathbf{W}_{vec}^0 - \mathbf{W}_{vec,*}\|_2 < \|\mathbf{W}_{vec,*}\|_2$ for the initialization $\mathbf{W}_{vec}^0$. Let $\phi^t = \arcsin\left(\|\mathbf{W}_{vec}^t - \mathbf{W}_{vec,*}\|_2 / \|\mathbf{W}_{vec,*}\|_2\right)$. If $\eta_t \leq \min_{0\leq\phi\leq\phi_t} \frac{m_{min}(\phi)-10m_{max}^{ex}}{2(L(\phi)+14L_{ex}+4\alpha)^2}$, then

$$\left\|\mathbf{W}_{vec}^{t+1} - \mathbf{W}_{vec,*}\right\|_2^2 \leq \left\|\mathbf{W}_{vec}^{t} - \mathbf{W}_{*,vec}\right\|_2^2 \left(1 - \frac{\eta_t(m_{min} - 10m_{max}^{ex})}{2}\right)$$

Proof. For w_i without time mark, consider it as weights of time t.

$$\mathbb{E}[\nabla_{w_i}\ell(\mathbf{W}^t, z)] =$$
$$\frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[zz^\top\mathbb{I}\left\{R\left(w_i, w_{i,*}\right)R\left(w_j, w_{j,*}\right) + R\left(w_i, -w_{i,*}\right)R\left(w_j, w_{j,*}\right)\right\}\right]\left(w_j - w_{j,*}\right)$$
$$+\frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[zz^\top\mathbb{I}\left\{R\left(w_i, w_{i,*}\right)R\left(w_j, -w_{j,*}\right) + R\left(w_i, -w_{i,*}\right)R\left(w_j, -w_{j,*}\right)\right\}\right]w_j$$
$$+\frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}\left[zz^\top\mathbb{I}\left\{R\left(w_i, w_{i,*}\right)R\left(-w_j, w_{j,*}\right) + R\left(w_i, -w_{i,*}\right)R\left(-w_j, w_{j,*}\right)\right\}\right](-w_{j,*})$$

$$\sum_{i=1}^{n} \mathbb{E}[< \nabla_{w_i} \ell(\mathbf{W}^t, z), w_i - w_{i,*} >] =$$

$$\frac{1}{n^2} \sum_{(i,j)=(1,1)}^{(n,n)} (w_i - w_{i,*})^\top \mathbb{E}\left[zz^\top \mathbb{I}\left\{ R\left(w_i, w_{i,*}\right) R\left(w_j, w_{j,*}\right) + \right.\right.$$

$$R\left(w_i, -w_{i,*}\right) R\left(w_j, w_{j,*}\right) \left(w_j - w_{j,*}\right)$$

$$+ \frac{1}{n^2} \sum_{(i,j)=(1,1)}^{(n,n)} (w_i - w_{i,*})^\top \mathbb{E}\left[zz^\top \mathbb{I}\left\{ R\left(w_i, w_{i,*}\right) R\left(w_j, -w_{j,*}\right) + \right.\right.$$

$$R\left(w_i, -w_{i,*}\right) R\left(w_j, -w_{j,*}\right) w_j$$

$$+ \frac{1}{n^2} \sum_{(i,j)=(1,1)}^{(n,n)} (w_i - w_{i,*})^\top \mathbb{E}\left[zz^\top \mathbb{I}\left\{ R\left(w_i, w_{i,*}\right) R\left(-w_j, w_{j,*}\right) + \right.\right.$$

$$R\left(w_i, -w_{i,*}\right) R\left(-w_j, w_{j,*}\right) \left(-w_{j,*}\right)$$

$$\begin{aligned}
&= \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right)^\top \left(\mathbf{B}_{0,0,0} + \mathbf{B}_{1,0,0}\right) \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right) \\
&+ \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right)^\top \left(\mathbf{B}_{0,0,1} + \mathbf{B}_{1,0,1}\right) \left(\mathbf{W}_{vec}\right) \\
&+ \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right)^\top \left(\mathbf{B}_{0,1,0} + \mathbf{B}_{1,1,0}\right) \left(-\mathbf{W}_{*,vec}\right)
\end{aligned}$$

Use definiton so fm_{min} and remove positive terms

$$\begin{aligned}
&\geq \left\|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right\|_2^2 m_{min} \\
&+ \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right)^\top \left(\mathbf{B}_{1,0,0}\right) \left(-\mathbf{W}_{*,vec}\right) \\
&+ \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right)^\top \left(\mathbf{B}_{0,0,1}\right) \left(\mathbf{W}_{vec}\right) \\
&+ \left(\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right)^\top \left(\mathbf{B}_{0,1,0} + \mathbf{B}_{1,1,0}\right) \left(-\mathbf{W}_{*,vec}\right)
\end{aligned}$$

Now use norm.

$$
\begin{aligned}
&\geq \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, m_{min} \\
&- \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{1,0,0}\|_{oper} \\
&- \|\mathbf{W}_{vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{0,0,1}\|_{oper} \\
&- \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{0,1,0}\|_{oper} \\
&- \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{1,1,0}\|_{oper}
\end{aligned}
$$

With properly assigned initial weights, we can assume $\|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \leq \|\mathbf{W}_{*,vec}\|_2$ when t=0. If this theorem satisfies, then we can use it inductively for all t. Therefore, $\|\mathbf{W}_{vec}\|_2 \leq 2 \|\mathbf{W}_{*,vec}\|_2$

$$
\begin{aligned}
&\geq \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, m_{min} \\
&- \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{1,0,0}\|_{oper} \\
&- 2 \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{0,0,1}\|_{oper} \\
&- \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{0,1,0}\|_{oper} \\
&- \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, \|\mathbf{B}_{1,1,0}\|_{oper}
\end{aligned}
$$

By assumption, operation norm of $\mathbf{B}$ can be replaced.

$$
\geq \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, m_{min} - 5 \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, m_{max}^{ex} \phi
$$

Since $\phi \leq 2sin\phi$ for $0 \leq \phi \leq 2/\pi$, $sin\phi = \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\| \, / \, \|\mathbf{W}_{*,vec}\|_2$

$$
\begin{aligned}
&\geq \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, m_{min} - 10 \|\mathbf{W}_{*,vec}\|_2 \, \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2 \, m_{max}^{ex} sin(\phi) \\
&\geq \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, m_{min} - 10 \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, m_{max}^{ex} \\
&= \|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\|_2^2 \, (m_{min} - 10 m_{max}^{ex})
\end{aligned}
$$

In the same way,

$$
\begin{aligned}
&(\sum \left\|\mathbb{E}[\nabla_{w_i}\ell(\mathbf{W}^t, z)]\right\|_2^2)^{0.5} \\
&\leq \left\|\mathbf{B}_{0,0,0}\right\|_{oper} \left\|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right\|_2 + \left\|\mathbf{B}_{1,0,0}\right\|_{oper} \left\|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right\|_2 \\
&+ \left\|\mathbf{B}_{0,0,1}\right\|_{oper} \left\|\mathbf{W}_{vec}\right\|_2 + \left\|\mathbf{B}_{1,0,1}\right\|_{oper} \left\|\mathbf{W}_{vec}\right\|_2 \\
&+ \left\|\mathbf{B}_{0,1,0}\right\|_{oper} \left\|\mathbf{W}_{vec,*}\right\|_2 + \left\|\mathbf{B}_{1,1,0}\right\|_{oper} \left\|\mathbf{W}_{vec,*}\right\|_2 \\
&\leq L \left\|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right\|_2 + \left\|\mathbf{B}_{1,0,0}\right\|_{oper} 3 \left\|\mathbf{W}_{*,vec}\right\|_2 \\
&+ \left\|\mathbf{B}_{0,0,1}\right\|_{oper} 2 \left\|\mathbf{W}_{vec,*}\right\|_2 + \left\|\mathbf{B}_{1,0,1}\right\|_{oper} 2 \left\|\mathbf{W}_{vec,*}\right\|_2 \\
&+ \left\|\mathbf{B}_{0,1,0}\right\|_{oper} \left\|\mathbf{W}_{vec,*}\right\|_2 + \left\|\mathbf{B}_{1,1,0}\right\|_{oper} \left\|\mathbf{W}_{vec,*}\right\|_2 \\
&\leq \left\|\mathbf{W}_{vec} - \mathbf{W}_{*,vec}\right\|_2 (L + 4\alpha + 14 m_{max}^{ex})
\end{aligned}
$$

If η_t is small enough,

$$
\begin{aligned}
\left\|\mathbf{W}_{vec}^{t+1} - \mathbf{W}_{vec,*}\right\|_2^2 &\leq \left\|\mathbf{W}_{vec}^{t} - \mathbf{W}_{*,vec}\right\|_2^2 (1 - \eta_t(m_{min} - 10 m_{max}^{ex}) + \eta_t^2 (L + 4\alpha + 14 m_{max}^{ex}) \\
&\leq \left\|\mathbf{W}_{vec}^{t} - \mathbf{W}_{*,vec}\right\|_2^2 (1 - \frac{\eta_t(m_{min} - 10 m_{max}^{ex})}{2})
\end{aligned}
$$

□

국문초록

최근 몇년간 딥러닝은 여러 분야에서 최고의 성능을 보여줬다. 그러한 경험적인 성공에도 불구하고 확률적 경사하강법이 왜 최적화 문제에서 높은 성능을 보여주는지에 대해 이론적인 설명이 아직 충분하지 않다. 이 논문에서는 확률적 경사하강법이 각 파라미터들을 학습과정에서 수렴시키는 과정에 대하여 분석하였다. 이 논문에서 사용된 딥러닝 모델은 ReLU 함수를 사용한 2층 신경망이다. 특히 입력 분포의 smoothness 개념을 사용하여 특정 분포가 아닌 일반적인 입력분포에 대해 적용 가능한 분석을 이끌어낼 수 있었다.

주요어휘: 최적화, 입력분포, 신경망 , 딥러닝
학번: 2016-20235