Bounding the Width of Neural Networks via Coupled Initialization - A Worst Case Analysis

joint work with A. Munteanu (TU Dortmund), Z. Song (Adobe Research) and D. Woodruff (CMU)

Simon Omlor | ICML 2022

Motivation

- Neural networks have been a popular topic in recent research;
- Even though they perform well in practice little is known about theoretical bounds

Our goal:

Analyze worst case behavior of ’simple’ neural networks.

Two layer ReLU network

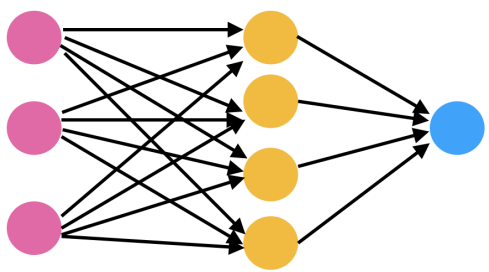
Assume that our data points are points in $\mathbb{R}^d$. Then a two layer ReLU network is given by:

- weights of the first layer, i.e. $w_1 \dots w_m \in \mathbb{R}^d$;
- weight vector $a \in \{-1, 1\}^m$ for the second layer;

Prediction for $x \in \mathbb{R}^d$:

$$f(W, x, a) := \sum_{j=1}^{m} a_j \mathrm{ReLU}(\langle w_j, x \rangle),$$

where $\mathrm{ReLU}(r) = \max\{r, 0\}$.



Train a two layer ReLU network

Assume that we are given a data set consisting of points $x_1 \ldots, x_n \in \mathbb{R}^d$ together with labels $y_1, \ldots, y_n \in \mathbb{R}$. In order to train the neural network to give good predictions, one tries to optimize

$$R(W) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(W, x_i, a), y_i)$$

where $\ell : \mathbb{R}^2 \to \mathbb{R}_+$ is an appropriate loss function.

Train a two layer ReLU network

Assume that we are given a data set consisting of points $x_1 \ldots, x_n \in \mathbb{R}^d$ together with labels $y_1, \ldots, y_n \in \mathbb{R}$. In order to train the neural network to give good predictions, one tries to optimize

$$R(W) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(W, x_i, a), y_i)$$

where $\ell : \mathbb{R}^2 \to \mathbb{R}_+$ is an appropriate loss function. Our loss functions:

$$\ell_1(f(W, x_i, a), y_i) = \ln(1 + \exp(y_i \cdot f(W, x_i, a))) \quad \text{logistic loss; } y_i \in \{-1, 1\}$$

$$\ell_2(f(W, x_i, a), y_i) = (f(W, x_i, a) - y_i)^2 \quad \text{squared loss; } y_i \in \mathbb{R}$$

Our goal

Get training error $R(W) \leq \epsilon$ using gradient descent.

- minimize the number m of inner nodes (also called the width) needed;
- minimize the number T of iterations needed.

Our results

References	Width m	Iterations T	Loss function
[Du et at. 2019]	$O(\lambda^{-4}n^6)$	$O(\lambda^{-2}n^2\log(1/\varepsilon))$	squared loss
[Song, Yang 2019]	$O(\lambda^{-4}n^4)$	$O(\lambda^{-2}n^2\log(1/\varepsilon))$	squared loss
Our work	$O(\lambda^{-2}n^2)$	$O(\lambda^{-2}n^2\log(1/\varepsilon))$	squared loss
[Ji, Telgarsky 2020]	$O(\gamma^{-8}\log n)$	$\tilde{O}(\varepsilon^{-1}\gamma^{-2})$	logistic loss
Our work	$O(\gamma^{-2}\log n)$	$\tilde{O}(\varepsilon^{-1}\gamma^{-2})$	logistic loss
[Ji, Telgarsky 2020]	$\Omega(\gamma^{-1/2})$	N/A	logistic loss
Our work	$\Omega(\gamma^{-1}\log n)$	N/A	logistic loss

Summary of previous work and our work. The improvements are mainly in the dependence on the parameters λ, γ, n affecting the width m.

Initalization scheme

Coupled initialization (introduced before by [Daniely 2020]):

- For each $r = 2i - 1$, we choose w_r to be a random Gaussian vector drawn from $\mathcal{N}(0, I)$.
- For each $r = 2i - 1$, we choose $a_r = 1$.
- For each $r = 2i$, we choose $w_r = w_{r-1}$.
- For each $r = 2i$, we choose $a_r = -1$.

$\rightarrow$ Allows us to scale the vectors w_r arbitrarily as we always have

$$f(W, x_i, a) = 0$$

for all $i \in [n]$ at initialization.

Gradient descent/NTK analysis

Update step:

$$W(t+1) = W(t) - \eta \frac{\partial L(W(t))}{\partial W(t)}.$$

Idea of the analysis:

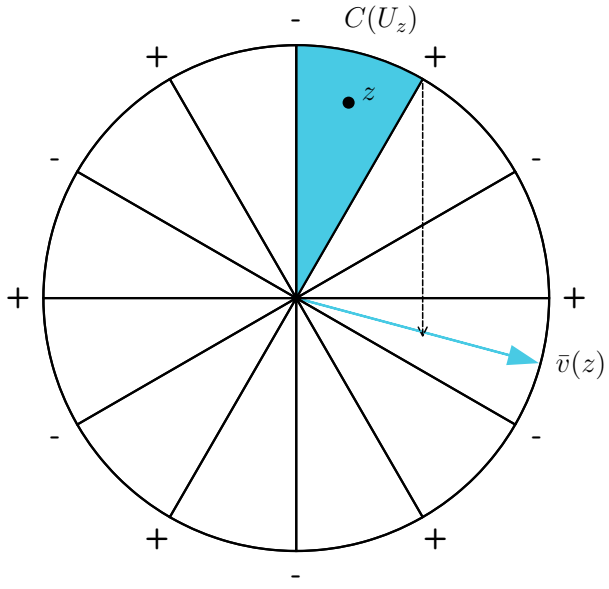
$$\frac{\partial f(W, x, a)}{\partial w_r} = a_r x \mathbf{1}_{w_r^\top x \geq 0}$$

does not change with high probability for any r.

In previous papers[Song, Yang 2019; Ji, Telgarsky 2020]: If m is large enough then the term above does not change for almost any r.
$\rightarrow$ need bounds on m only for the initialization.

Lower bound

Following example is used to show the lower bounds:



- Any two-layer ReLU neural networkwith width $m = o(\gamma^{-1})$ misclassifies at least $\Omega(\gamma^{-1})$ points.
- There is a natural choice for some parameter matrix U such that if $m = o(\gamma^{-2} \log n)$, then with constant probability there exists an $i \in [n]$ such that $y_i \langle \nabla f_i(W_0), \bar{U} \rangle \leq 0$.

Outlook/Future work

Can we further close the gaps between quadratic and linear bounds?

- What is the worst case bound on m for logistic loss: $\tilde{O}(\gamma^{-1})$ or $\tilde{O}(\gamma^{-2})$?
- What is the worst case bound on m for squared loss: $\tilde{O}(n)$ or $\tilde{O}(n^2)$?