# Implicit bias of gradient descent for mean squared error regression with wide neural networks

Presenter: Hui Jin

August 12, 2020

# Introduction

- The implicit bias of optimization plays a important role in the generalization performance of deep neural networks.
- Among all hypotheses that fit all training data, the optimization algorithm selects one which generalizes well.

# Introduction

In classification problems, gradient descent on deep linear networks converges to the $l_2$ maximum margin solution if training data are linearly separable. [Soudry et al., 2018]

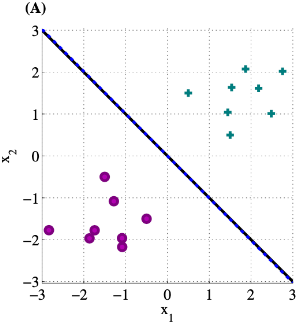

Figure: Results of training deep linear networks for classification problems

# Problem Setup

We consider shallow networks, with one input and a single hidden layer of $n$ ReLUs and one linear output:

$$f(x,\theta) = \sum_{i=1}^{n} W_i^{(2)}[W_i^{(1)}x + b_i^{(1)}]_+ + b^{(2)}. \tag{1}$$

These parameters are initialized by sampling independent random variables in following way:

$$\begin{cases} W_j^{(l)} \overset{d}{=} \sqrt{\frac{1}{n_l}}\mathcal{W}, \\ b_j^{(l)} \overset{d}{=} \sqrt{\frac{1}{n_l}}\mathcal{B}. \end{cases} \tag{2}$$

Here $\mathcal{W}$ and $\mathcal{B}$ are some pre-specified random variables. We assume that the joint distribution of $(\mathcal{W}, \mathcal{B})$ is sub-gaussian.

To be more specific,

$$\begin{cases} W_j^{(1)} \overset{d}{=} \mathcal{W}, \\ b_j^{(1)} \overset{d}{=} \mathcal{B}, \\ W_j^{(2)} \overset{d}{=} \sqrt{\frac{1}{n}}\mathcal{W}, \\ b^{(2)} \overset{d}{=} \sqrt{\frac{1}{n}}\mathcal{B}, \end{cases} \tag{3}$$

# Main Result

## Theorem (Implicit bias of gradient descent in wide ReLU networks)

*Consider a feedforward network with a single input unit, a hidden layer of $n$ rectified linear units, a single linear output unit, and weights and biases initialized from a sub-Gaussian distribution. For any finite data set $\{(x_i, y_i)\}_{i=1}^M$, there exist constant $u$ and $v$ so that optimization of the mean square error on the adjusted training data $\{(x_i, y_i - ux_i - v)\}_{i=1}^M$ by full-batch gradient descent with sufficiently small step size converges to a parameter $\theta^*$ for which $f(x, \theta^*)$ attains zero training error. Furthermore, for any prescribed bounded interval $[-L, L]$, we have $\inf_{x\in[-L,L]} \|f(x,\theta^*) - g^*(x)\|_2 = O(n^{-\frac{1}{2}})$ with high probability over the random initialization $\theta_0$, where $g^*$ solves following variational problem:*

$$\begin{aligned} \min_{g\in C^2(\mathrm{supp}(\zeta))} \quad & \int \frac{1}{\zeta(x)} (g''(x) - f''(x,\theta_0))^2 \, \mathrm{d}x \\ \text{subject to} \quad & g(x_i) = y_i - ux_i - v, \quad i = 1, \ldots, M. \end{aligned} \tag{4}$$

*Here, the reciprocal of the function $\zeta(x) = \int |W|^3 p_{\mathcal{W},\mathcal{B}}(W, -Wx) \, \mathrm{d}W$.*

# Anti-Symmetrical Initialization (ASI)

To simplify the presentation, we use a special initialization called Anti-Symmetrical Initialization (ASI) to make the initial output function to be zero. It is defined as follows:

$$f_{\mathrm{ASI}}(x,\vartheta_0)=\frac{\sqrt{2}}{2}f(x,\vartheta_0')-\frac{\sqrt{2}}{2}f(x,\vartheta_0''). \tag{5}$$

Here $\vartheta_0=(\vartheta_0',\vartheta_0'')$ is initialized with $\vartheta_0'=\vartheta_0''$, so that

$$f_{\mathrm{ASI}}(x,\vartheta_0)=\sum_{i=1}^{n}\frac{\sqrt{2}}{2}\overline{V}_i^{(2)}[\overline{V}_i^{(1)}x+\overline{a}_i^{(1)}]_{+}+\sum_{i=1}^{n}-\frac{\sqrt{2}}{2}\overline{V}_i^{(2)}[\overline{V}_i^{(1)}x+\overline{a}_i^{(1)}]_{+}\equiv 0. \tag{6}$$

The parameter vector is thus
$\vartheta_0=\mathrm{vec}(\overline{V}^{(1)},\overline{V}^{(1)},\overline{a}^{(1)},\overline{a}^{(1)},\frac{\sqrt{2}}{2}\overline{V}^{(2)},-\frac{\sqrt{2}}{2}\overline{V}^{(2)},\frac{\sqrt{2}}{2}\overline{a}^{(2)},-\frac{\sqrt{2}}{2}\overline{a}^{(2)})$.

# Experimental Result

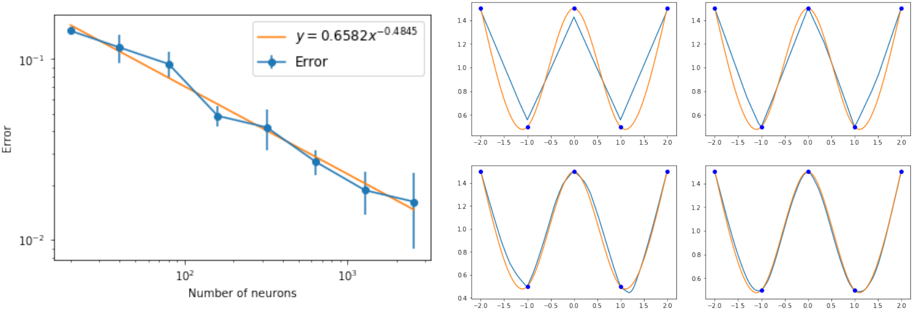

Figure: Error between the solution function obtained by gradient descent training a neural network and the solution to the variational problem, against the number of neurons. Here the parameters were initialized by $W \sim U(-1, 1)$, $B \sim U(-2, 2)$. In the right panel, blue lines are examples of output functions from trained networks with 10, 40, 160, and 640 neurons. Orange is the solution to the variational problem. In close agreement with our theoretical results, as the number $n$ of neurons increases, the blue lines approximate the orange line uniformly at a rate $O(1/\sqrt{n})$.

# Experimental Result

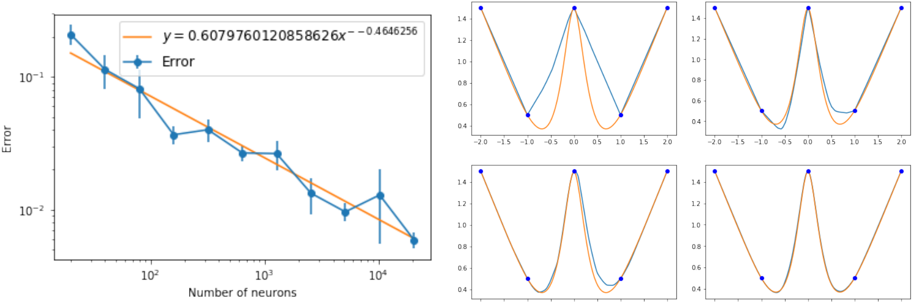

Figure: Error between the output of the neural network and solution of variational problem against number of neurons. Trained networks with 20, 80, 320, and 2560 neurons. Initialization $W \sim N(0,1)$, $B \sim N(0,0.1)$. Blue is the output of neural network at the end of training, and orange is the solution to the variational problem.

# Linearized Model

This is obtained by the first order Taylor expansion of the network function with respect to the parameter, at the initial parameter value,

$$f^{\mathrm{lin}}(x, \omega) = f(x, \theta_0) + \nabla_\theta f(x, \theta_0)(\omega - \theta_0). \tag{7}$$

We write $\omega$ for the parameter of the linearized model. According to Lee et al. [2019, Theorem H.1],

$$\sup_t \|f^{\mathrm{lin}}(x, \omega_t) - f(x, \theta_t)\|_2 = O(n^{-\frac{1}{2}})$$

with arbitrarily high probability.

# Training only the output layer approximates training all parameters

According to initialization, with probability arbitrarily close to 1, $\overline{W}_i^{(1)}, \bar{b}_i^{(1)} = O(1)$ and $\overline{W}_i^{(2)}, \bar{b}^{(2)} = O(n^{-\frac{1}{2}})$. Therefore, writing $H$ for the Heaviside function, we have

$$\begin{aligned} \nabla_{W_i^{(1)}} f(x, \theta_0) =& \overline{W}_i^{(2)} H(\overline{W}_i^{(1)} x + \bar{b}^{(1)}) \cdot x = O(n^{-\frac{1}{2}}), \quad (8) \\ \nabla_{b_i^{(1)}} f(x, \theta_0) =& \overline{W}_i^{(2)} H(\overline{W}_i^{(1)} x + \bar{b}_i^{(1)}) = O(n^{-\frac{1}{2}}), \quad (9) \end{aligned}$$

and

$$\begin{aligned} \nabla_{W_i^{(2)}} f(x, \theta_0) =& [\overline{W}_i^{(1)} x + \bar{b}_i^{(1)}]_+ = O(1), \quad (10) \\ \nabla_{b^{(2)}} f(x, \theta_0) =& 1 = O(1). \quad (11) \end{aligned}$$

# Training only the output layer approximates training all parameters

### Theorem (Training only output weights vs linearized network)

*Consider a finite data set* $\{(x_i, y_i)\}_{i=1}^{M}$. *Assume that (1) we use the MSE loss, i.e.* $\ell(\widehat{y}, y) = \frac{1}{2}\|\widehat{y} - y\|_2^2$; *(2)* $\inf_n \lambda_{\min}(\hat{\Theta}_n) > 0$. *Let* $\omega_t$ *denote the parameters of the linearized model at time* $t$ *when we train all parameters and let* $\widetilde{\omega}_t$ *denote the parameters at time* $t$ *when we only train weights of the output layer. If we use the same learning rate* $\eta$ *in these two training processes and* $\eta < \frac{2}{n\lambda_{\max}(\hat{\Theta}_n)}$, *then for any* $x \in \mathbb{R}$, *with probability arbitrarily close to* 1 *over random initialization,*

$$\sup_t |f^{\text{lin}}(x, \widetilde{\omega}_t) - f^{\text{lin}}(x, \omega_t)| = O(n^{-1}), \text{ as } n \to \infty. \tag{12}$$

# Implicit bias in parameter space of a linearized model

## Theorem (Parameters of the linearized model)

*Consider a convex loss function* $\ell$ *which is* $K$*-Lipschitz continuous, i.e.* $|\ell(\widehat{y}_1, y) - \ell(\widehat{y}_2, y)| \leq K|\widehat{y}_1 - \widehat{y}_2|$. *If* $\operatorname{rank}(\nabla_\theta f(\mathcal{X}, \theta_0)) = M$, *then the gradient descent iteration with learning rate* $\eta \leq \frac{1}{Kn\sqrt{M}\lambda_{\max}(\hat{\Theta}_n)}$ *converges to the unique solution of following constrained optimization problem:*

$$\min_{\omega} \|\omega - \theta_0\|_2 \quad s.t. \;\; f^{\text{lin}}(\mathcal{X}, \omega) = \mathcal{Y}. \tag{13}$$

# Optimization problem of the parameters

Gradient descent only on the output layer finds a solution of zero loss under the assumption of the about theorem, so that

$$\begin{aligned} f^{\text{lin}}(x_j, \widetilde{\omega}_\infty) - f^{\text{lin}}(x_j, \theta_0) &= \sum_{i=1}^{n} (\widetilde{W}_i^{(2)} - \overline{W}_i^{(2)})[\overline{W}_i^{(1)} x_j + \bar{b}_i]_+ \\ &= y_j - f(x_j, \theta_0), \quad j = 1, \ldots, M, \end{aligned} \tag{14}$$

and $\|\widetilde{W}^{(2)} - \overline{W}^{(2)}\|_2^2$ is minimized. Then gradient descent is actually solving the following problem:

$$\begin{aligned} \min_{W^{(2)}} \quad & \|W^{(2)} - \overline{W}^{(2)}\|_2^2 \\ \text{subject to} \quad & \sum_{i=1}^{n} (W_i^{(2)} - \overline{W}_i^{(2)})[W_i^{(1)} x_j + b_i]_+ = y_j, \quad j = 1, \ldots, M. \end{aligned} \tag{15}$$

Note that we let $f^{\text{lin}}(x, \theta_0) \equiv 0$ by using ASI trick (as before, this is not essential but simplifies the presentation)

# Infinite width limit

Let $\mu_n$ denote the empirical distribution of the samples $(W_i^{(1)}, b_i)_{i=1}^n$, so that $\mu_n(W^{(1)}, b) = 1/n$ for $(W^{(1)}, b) = (W_i^{(1)}, b_i)$, $i = 1, \dots, n$, and $\mu_n(W^{(1)}, b) = 0$ otherwise. We further consider a function $\alpha_n(W_i^{(1)}, b_i) = n(W_i^{(2)} - \overline{W}_i^{(2)})$. Then

$$\begin{aligned} \min_{W^{(2)}} \quad & \|W^{(2)} - \overline{W}^{(2)}\|_2^2 \\ \text{subject to} \quad & \sum_{i=1}^n (W_i^{(2)} - \overline{W}_i^{(2)})[W_i^{(1)} x_j + b_i]_+ = y_j, \quad j = 1, \dots, M. \end{aligned} \tag{16}$$

becomes

$$\begin{aligned} \min_{\alpha_n \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha_n^2(W^{(1)}, b) \, \mathrm{d}\mu_n(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha_n(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b) = y_j, \quad j = 1, \dots, M. \end{aligned} \tag{17}$$

# Infinite width limit

we can consider the infinite width neural network, i.e. the limit when $n \to \infty$. Let $\mu$ be the probability measure of $(\mathcal{W}, \mathcal{B})$. Then we can write a continuous version of problem (17):

$$\begin{aligned} \min_{\alpha \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b) \, \mathrm{d}\mu(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) = y_j, \quad j = 1, \ldots, M. \end{aligned} \tag{18}$$

## Theorem

*Assume that $\mathcal{W}$ and $\mathcal{B}$ have finite fourth moments. Let $(W_i^{(1)}, b_i)_{i=1}^n$ be i.i.d. samples drawn from the distribution of $(\mathcal{W}, \mathcal{B})$. Suppose $\mu_n$ in problem (17) is the empirical distribution of the $n$ samples $(W_i^{(1)}, b_i)_{i=1}^n$. Let $\overline{\alpha}_n(W^{(1)}, b)$ be the solution of (17) and $\overline{\alpha}(W^{(1)}, b)$ be the solution of (18). Then for any bounded interval $[-L, L]$, $\sup_{x \in [-L,L]} |g_n(x, \overline{\alpha}_n) - g(x, \overline{\alpha})| = O(n^{-1/2})$ with high probability.*

# Function space description

We write $c = -b/W^{(1)}$, which is the breakpoint of a ReLU with weight $W^{(1)}$ and bias $b$, and we define a corresponding random variable $\mathcal{C} = -\mathcal{B}/\mathcal{W}$. Here we assume that $\mathbb{P}(\mathcal{W} = 0) = 0$. This means that the random variable $\mathcal{C}$ is well defined. Let $\nu$ denote the probability measure of $(\mathcal{W}, \mathcal{C})$. Finally, let $\gamma(W^{(1)}, c) = \alpha(W^{(1)}, -cW^{(1)})$, which corresponds to $\alpha(W^{(1)}, b)$. Then problem (18) is equivalent to

$$\begin{aligned} \min_{\gamma \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c) \, \mathrm{d}\nu(W^{(1)}, c) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x_j - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c) = y_j, \quad j = 1, \cdots, M. \end{aligned} \tag{19}$$

# Function space description

Suppose $\nu_{\mathcal{C}}$ has a density function $p_{\mathcal{C}}(c)$. Let $g(x,\gamma) = \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x-c)]_+ \, \mathrm{d}\nu(W^{(1)}, c)$, which again corresponds to the function represented by the network. Then, writing $g''$ for the second derivative with respect to $x$,

$$\begin{aligned} g''(x,\gamma) &= \int_{\mathbb{R}^2} \gamma(W^{(1)}, c) \left|W^{(1)}\right| \delta(x-c) \, \mathrm{d}\nu(W^{(1)}, c) \\ &= \int_{\mathrm{supp}(\nu_{\mathcal{C}})} \left( \int_{\mathbb{R}} \gamma(W^{(1)}, c) \left|W^{(1)}\right| \, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)}) \right) \delta(x-c) \, \mathrm{d}\nu_{\mathcal{C}}(c) \\ &= \int_{\mathrm{supp}(\nu_{\mathcal{C}})} \left( \int_{\mathbb{R}} \gamma(W^{(1)}, c) \left|W^{(1)}\right| \, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)}) \right) \delta(x-c) p_{\mathcal{C}}(c) \mathrm{d}c \\ &= p_{\mathcal{C}}(x) \int_{\mathbb{R}} \gamma(W^{(1)}, x) \left|W^{(1)}\right| \, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}). \end{aligned} \tag{20}$$

# Function space description

From the above, we see that $\gamma(W^{(1)}, c)$ is closely related to $g''(x, \gamma)$. So we want to express problem (19) in terms of $g''(x, \gamma)$. However, $g''(x, \gamma)$ determines $g(x, \gamma)$ only up to linear functions. Therefore we consider the following problem:

$$\begin{aligned} \min_{\gamma \in C(\mathbb{R}^2), u \in \mathbb{R}, v \in \mathbb{R}} \quad & \int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c) \, \mathrm{d}\nu(W^{(1)}, c) \\ \text{subject to} \quad & u x_j + v + \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x_j - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c) = y_j \end{aligned} \tag{21}$$

# Function space description

## Theorem

*Suppose* $(\overline{\gamma}, \overline{u}, \overline{v})$ *is the solution of* (21), *and consider*

$$g(x, (\overline{\gamma}, \overline{u}, \overline{v})) = \overline{u}x + \overline{v} + \int_{\mathbb{R}^2} \overline{\gamma}(W^{(1)}, c)[W^{(1)}(x-c)]_+ \, \mathrm{d}\nu(W^{(1)}, c). \tag{22}$$

*Let* $\nu_{\mathcal{C}}$ *denote the marginal distribution of* $\mathcal{C}$ *and assume it has a density function* $p_{\mathcal{C}}$. *Let* $\mathbb{E}(\mathcal{W}^2|\mathcal{C})$ *denote the conditional expectation of* $\mathcal{W}^2$ *given* $\mathcal{C}$. *Consider the function*

$$\zeta(x) = p_{\mathcal{C}}(x)\mathbb{E}(\mathcal{W}^2|\mathcal{C} = x). \tag{23}$$

*Assume that training data* $x_i \in \operatorname{supp}(\zeta)$, $i = 1, \dots, m$. *Consider the set* $S = \operatorname{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$. *Then* $g(x, (\overline{\gamma}, \overline{u}, \overline{v}))$ *satisfies* $g''(x, (\overline{\gamma}, \overline{u}, \overline{v})) = 0$ *for* $x \notin S$ *and for* $x \in S$ *it is the solution of the following problem:*

$$\begin{aligned} \min_{h \in C^2(S)} \quad & \int_S \frac{(h''(x))^2}{\zeta(x)} \, \mathrm{d}x \\ \text{subject to} \quad & h(x_j) = y_j, \quad j = 1, \dots, m. \end{aligned} \tag{24}$$

# Function space description

## Proposition

1. *Gaussian initialization. Assume that* $\mathcal{W}$ *and* $\mathcal{B}$ *are independent,* $\mathcal{W} \sim \mathcal{N}(0, \sigma_w^2)$ *and* $\mathcal{B} \sim \mathcal{N}(0, \sigma_b^2)$*. Then* $\zeta$ *is given by* $\zeta(x) = \frac{2\sigma_w^3\sigma_b^3}{\pi(\sigma_b^2 + x^2\sigma_w^2)^2}$*.*
2. *Binary-uniform initialization. Assume that* $\mathcal{W}$ *and* $\mathcal{B}$ *are independent,* $\mathcal{W} \in \{-1, 1\}$ *and* $\mathcal{B} \sim \mathcal{U}(-a_b, a_b)$ *with* $a_b \geq L$*. Then* $\zeta$ *is constant on* $[-L, L]$*.*
3. *Uniform initialization. Assume that* $\mathcal{W}$ *and* $\mathcal{B}$ *are independent,* $\mathcal{W} \sim \mathcal{U}(-a_w, a_w)$ *and* $\mathcal{B} \sim \mathcal{U}(-a_b, a_b)$ *with* $\frac{a_b}{a_w} \geq L$*. Then* $\zeta$ *is constant on* $[-L, L]$*.*

# Experimental

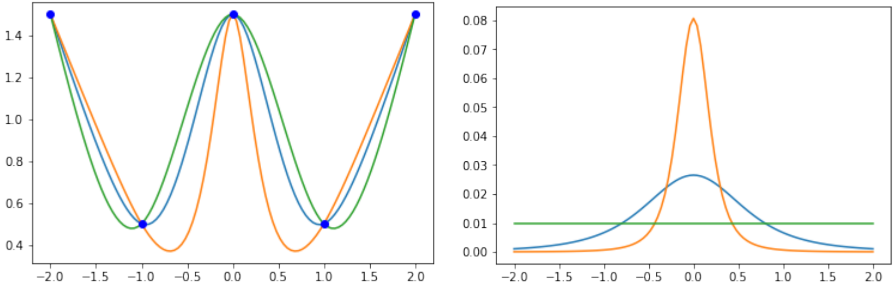

Figure: Effect of $\zeta$ on the shape of the solution function. The left panel shows the solution of the variational problem for various different $\zeta$ shown in the right panel. Green line is for $\zeta$ constant on $[-2, 2]$, which results from $W \sim U(-1, 1)$, $B \sim U(-2, 2)$; the blue line is for $\zeta(x) = 1/(1 + x^2)^2$, from $W \sim N(0, 1)$, $B \sim N(0, 1)$; and the orange line is for $\zeta(x) = 1/(0.1 + x^2)^2$, which results from $W \sim N(0, 1)$, $B \sim N(0, 0.1)$. We see that where $\zeta$ peaks strongly, the solution function can use a high curvature in order to fit the data.

# Future work

1. Multidimensional inputs.
2. Deep networks.
3. Other optimization method.

# References

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.