# Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks

Yuan Cao and Quanquan Gu

Presented by Antonio Orvieto
4 Gregor's beautiful yet a bit crazy reading group

July 2020

# Outline

- Motivation + contributions
- Setup and notation
- Generalization bound 1
- Generalization bound 2 (using NTK)

Note 1: I'll be quick ;)

Note 2: here I present just the results of this paper. The work tho provides interesting comparisons with previous literature on generalization – I'll leave that for you to explore ;)

## Motivation

over-parameterized NN trained with SGD can still give small test error and do not overfit (Zhang et al. 2017)

(a) if random labels, an overparameterized NN fit the training data. However, does not generalize.

(b) If same NN trained with real labels, not only achieves small training loss, but also generalizes well.

**overparametrization : network width is much larger than the number of training data points*

## Literature

(a) is understood, yet existing generalization bound can't explain (b).

*It is essential to quantify the "classifiability" of the underlying data distribution, i.e., how difficult it can be classified.*

*Classifiability* considered by some recent works, but for 2-3 layer networks, or with assumptions (e.g. lin. separable data).

## Contributions

- Bound on the expected 0-1 error of deep ReLU networks trained by SGD with random initialization of the form

$$\widetilde{\mathcal{O}}(n^{-1/2}) + const,$$

  where the constant is determined by the performance on the *neural tangent random feature model* (NTRF).
  **This is in turn related to the dataset structure.**

- Connection of performance on NTRF with quantities typical of the NTK literature.

## Neural network

$$f_{\mathbf{W}}(\mathbf{x}) = \sqrt{m} \cdot \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\mathbf{W}_{L-2} \cdots \sigma(\mathbf{W}_1 \mathbf{x}) \cdots))$$

- Input $\mathbf{x} \in \mathbb{R}^d$
- Layers $L \geqslant 2$ (i.e. $f$ nonlinear)
- $\mathbf{W}_1 \in \mathbb{R}^{m \times d}$, $\mathbf{W}_l \in \mathbb{R}^{m \times m}$, $\mathbf{W}_L \in \mathbb{R}^{1 \times m}$
- $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_L)$
- ReLU activation: $\sigma(z) = \max\{0, z\}$

*Can be extended to other activations to different sizes in each layer.
*$\sqrt{m}$ at beginning because of limit theorems as $m \to \infty$.

## Data

- Data points $(\mathbf{x}, y)$ sampled from $\mathcal{D}$.
- $\|\mathbf{x}\|_2 = 1$ for all $(\mathbf{x}, y)$.

## Optimization

$$\min_{\mathbf{W}} L_{\mathcal{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} L_{(\mathbf{x},y)}(\mathbf{W}),$$

where $L_{(\mathbf{x},y)}(\mathbf{W}) = \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$ and $\ell(z) = \log[1 + \exp(-z)]$.

**Algorithm 1** SGD for DNNs starting at Gaussian initialization

Generate each entry of $\mathbf{W}_l^{(1)}$ indep. from $N(0, 2/m)$, $l \in [L-1]$.
Generate each entry of $\mathbf{W}_L^{(1)}$ indep. from $N(0, 1/m)$.
**for** $i = 1, 2, \ldots, n$ **do**
  Draw $(\mathbf{x}_i, y_i)$ from $\mathcal{D}$.
  Update $\mathbf{W}^{(i+1)} = \mathbf{W}^{(i)} - \eta \cdot \nabla_{\mathbf{W}} L_{(\mathbf{x}_i, y_i)}(\mathbf{W}^{(i)})$.
**end for**
**Output:** Randomly choose $\widehat{\mathbf{W}}$ uniformly from $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(n)}\}$.

- initialization s.t. expected length of the output vector in each hidden layer is equal to the length of the input (He initialization).
- last layer variance $1/m$ instead of $2/m$ since no ReLU.

**Ideal result.** SGD in $n$ iterations is s.t. $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} L_{\mathcal{D}}(\widehat{\mathbf{W}})] \leqslant \mathcal{O}(1/\sqrt{n}) + C$, where $C$ depends on the properties of the dataset (e.g. "*is it random?*")

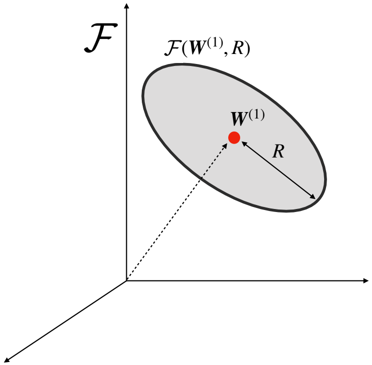
## Main object: Neural Tangent Random Feature (NTRF).

Let $\mathbf{W}^{(1)}$ be our inizialization. NTRF function class defined as

$$\mathcal{F}(\mathbf{W}^{(1)}, R) = \{f = f_{\mathbf{W}^{(1)}} + \langle \nabla_{\mathbf{W}} f_{\mathbf{W}^{(1)}}, \mathbf{W} \rangle : \mathbf{W} \in \mathcal{B}(\mathbf{0}, R \cdot m^{-1/2})\},$$

where $R > 0$ measures the size of the function class and

$$\mathcal{B}(\mathbf{W}, \omega) := \{\mathbf{W}' \in \mathcal{W} : \|\mathbf{W}'_l - \mathbf{W}_l\|_F \leqslant \omega, l \in [L]\}.$$



Define $L_{\mathcal{D}}^{0-1}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbb{1}\{y \cdot f_{\mathbf{W}}(\mathbf{x}) < 0\}]$.

**Generalization bound.** For any $\delta \in (0, e^{-1}]$ and $R > 0$, there exists

$$m^*(\delta, R, L, n) = \widetilde{\mathcal{O}}\big(\mathrm{poly}(R, L)\big) \cdot n^7 \cdot \log(1/\delta)$$

such that if $m \geqslant m^*(\delta, R, L, n)$, then with probability at least $1-\delta$ over the randomness of $\mathbf{W}^{(1)}$, the output of SGD with step size $\eta = \kappa \cdot R/(m\sqrt{n})$ for some small enough absolute constant $\kappa$ satisfies

$$\mathbb{E}\big[L_{\mathcal{D}}^{0-1}(\widehat{\mathbf{W}})\big] \leqslant \inf_{f\in\mathcal{F}(\mathbf{W}^{(1)},R)} \left\{\frac{4}{n}\sum_{i=1}^{n} \ell[y_i \cdot f(\mathbf{x}_i)]\right\} + \mathcal{O}\left[\frac{LR}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right],$$

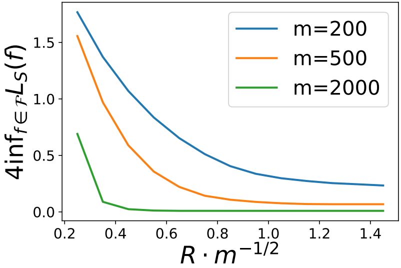
where expectation over the uniform draw of $\widehat{\mathbf{W}}$ from $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(n)}\}$.

- if the data can be classified in $\mathcal{F}(\mathbf{W}^{(1)}, \widetilde{\mathcal{O}}(1))$ with a small training error, the over-parameterized ReLU network learnt by SGD will have a small generalization error.
- second term independent of network width.
- a trade-off in the bound: $R$ is small, the corresponding NTRF class is small, making the first term large, and the second term small. When $R$ is large, first term in (3.1) is small, and second term will be large.

$$\mathbb{E}\big[L_{\mathcal{D}}^{0-1}(\widehat{\mathbf{W}})\big] \leqslant \inf_{f\in\mathcal{F}(\mathbf{W}^{(1)},R)} \left\{ \frac{4}{n}\sum_{i=1}^{n} \ell[y_i \cdot f(\mathbf{x}_i)] \right\} + \mathcal{O}\left[ \frac{LR}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}} \right]$$

## Experiment

- Five-layer fully connected NN on MNIST dataset (3 versus 8)
- Plotted is $\inf_{f\in\mathcal{F}(\mathbf{W}^{(1)},R)}\{(4/n)\cdot\sum_{i=1}^{n}\ell[y_i\cdot f(\mathbf{x}_i)]\}$




- larger the size of reference function class (i.e., $R$), smaller the inf.
- The wider the NN, the shorter SGD needs to travel to fit the training data.

## NTK matrix (review)

For any $i, j \in [n]$, define

$$\widetilde{\mathbf{\Theta}}_{i,j}^{(1)} = \mathbf{\Sigma}_{i,j}^{(1)} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad \mathbf{A}_{ij}^{(l)} = \begin{pmatrix} \mathbf{\Sigma}_{i,i}^{(l)} & \mathbf{\Sigma}_{i,j}^{(l)} \\ \mathbf{\Sigma}_{i,j}^{(l)} & \mathbf{\Sigma}_{j,j}^{(l)} \end{pmatrix},$$

$$\mathbf{\Sigma}_{i,j}^{(l+1)} = 2 \cdot \mathbb{E}_{(u,v)\sim N\left(\mathbf{0}, \mathbf{A}_{ij}^{(l)}\right)}[\sigma(u)\sigma(v)],$$

$$\widetilde{\mathbf{\Theta}}_{i,j}^{(l+1)} = \widetilde{\mathbf{\Theta}}_{i,j}^{(l)} \cdot 2 \cdot \mathbb{E}_{(u,v)\sim N\left(\mathbf{0}, \mathbf{A}_{ij}^{(l)}\right)}[\sigma'(u)\sigma'(v)] + \mathbf{\Sigma}_{i,j}^{(l+1)}.$$

Then we call $\mathbf{\Theta}^{(L)} = [(\widetilde{\mathbf{\Theta}}_{i,j}^{(L)} + \mathbf{\Sigma}_{i,j}^{(L)})/2]_{n\times n}$ the neural tangent kernel matrix of an $L$-layer ReLU network on training inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

**Theorem from Jacot et al.** For an $L$ layer ReLU network with parameter set $\mathbf{W}^{(1)}$ initialized as before, as the network width $m \to \infty$, it holds that

$$m^{-1}\langle \nabla_{\mathbf{W}} f_{\mathbf{W}^{(1)}}(\mathbf{x}_i), \nabla_{\mathbf{W}} f_{\mathbf{W}^{(1)}}(\mathbf{x}_j)\rangle \xrightarrow{\mathbb{P}} \mathbf{\Theta}_{i,j}^{(L)}.$$

**Generalization bound 2.** Let $\mathbf{y} = (y_1, \ldots, y_n)^\top$ and $\lambda_0 = \lambda_{\min}(\boldsymbol{\Theta}^{(L)})$. For any $\delta \in (0, e^{-1}]$, there exists $\widetilde{m}^*(\delta, L, n, \lambda_0)$ such that if $m \geqslant \widetilde{m}^*(\delta, L, n, \lambda_0)$, then with probability at least $1 - \delta$ over the randomness of $\mathbf{W}^{(1)}$, the output of SGD with small step size satisfies

$$\mathbb{E}\big[L_{\mathcal{D}}^{0-1}(\widehat{\mathbf{W}})\big] \leqslant \widetilde{\mathcal{O}}\left[L \cdot \inf_{\widetilde{y}_i y_i \geqslant 1} \sqrt{\frac{\widetilde{\mathbf{y}}^\top (\boldsymbol{\Theta}^{(L)})^{-1} \widetilde{\mathbf{y}}}{n}}\right] + \mathcal{O}\left[\sqrt{\frac{\log(1/\delta)}{n}}\right],$$

where expectation taken over uniform draw $\widehat{\mathbf{W}}$ from $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(n)}\}$.

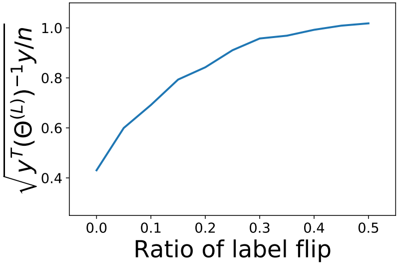
– demonstrates that the generalization bound does not increase with network width $m$, as long as $m$ is large enough.

– clear characterization of the classifiability of data. In fact, the $\sqrt{\widetilde{\mathbf{y}}^\top (\boldsymbol{\Theta}^{(L)})^{-1} \widetilde{\mathbf{y}}}$ is exactly the NTK-induced RKHS norm of the kernel regression classifier.

$$\mathbb{E}\big[L_{\mathcal{D}}^{0-1}(\widehat{\mathbf{W}})\big] \leqslant \tilde{\mathcal{O}}\left[L \cdot \inf_{\tilde{y}_i y_i \geqslant 1} \sqrt{\frac{\tilde{\mathbf{y}}^\top(\mathbf{\Theta}^{(L)})^{-1}\tilde{\mathbf{y}}}{n}}\right] + \mathcal{O}\left[\sqrt{\frac{\log(1/\delta)}{n}}\right],$$

**Experiment**

- Five-layer fully connected NN on MNIST dataset (3 versus 8)
- plotted value of $\sqrt{\mathbf{y}^\top(\mathbf{\Theta}^{(L)})^{-1}\mathbf{y}/n}$, where $\mathbf{y}$ is the true label vector with random flips.




- when most of the labels are true labels, bound can predict good test error.
- when the labels are purely random, bound can be larger than one.