Why Do Deep Residual Networks Generalize Better than Deep Feedforward Networks? — A Neural Tangent Kernel Perspective

Kaixuan Huang[†◇], Yuqing Wang[†*], Molei Tao[*], Tuo Zhao[*]
[†]Equal contribution [*]Georgia Tech [◇]Princeton University

Motivation

Feedforward networks (FFNets)

- Usually layers < 30 (e.g., VGG, layers ≈ 20);
- Deeper FFNets yield worse generalization behavior.

Residual networks (ResNets)

- Can have hundreds of layers;
- Deep ResNets have the same or even better generalization performance.

Question: Why do deep residual networks generalize better than deep feedforward networks?

Neural tangent kernel (NTK) [3]

- Consider fully-connected neural networks trained by gradient descent;
- When width $\to \infty$, we have NTK as follows:

$$\Omega_L(x,y) = \sum_{i=1}^{n} \langle \nabla_{\theta_i} f(x), \nabla_{\theta_i} f(y) \rangle,$$

where θ_i is the parameters, n is the number of parameters, f is an $L-$layer network.

Dual activation and dual kernel

Let $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be a kernel function. Denote

$$\Sigma(x,\widetilde{x}) = \begin{pmatrix} K(x,x) & K(x,\widetilde{x}) \\ K(\widetilde{x},x) & K(\widetilde{x},\widetilde{x}) \end{pmatrix} \text{ and } N_\rho = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Definition 1 (Dual activation). *Given an activation function $\phi : \mathbb{R} \to \mathbb{R}$, its dual activation function $\widehat{\phi} : [-1,1] \to [-1,1]$ is defined to be $\widehat{\phi}(\rho) = \mathbb{E}_{(X,\widetilde{X})\sim\mathcal{N}(0,N_\rho)} \phi(X)\phi(\widetilde{X})$.*

Definition 2 (Dual kernel). *We say that $\Gamma_\phi(K) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the dual kernel of K with respect to the activation ϕ, if we have $\Gamma_\phi(K)(x,\widetilde{x}) = \mathbb{E}_{(X,\widetilde{X})\sim\mathcal{N}(0,\Sigma(x,\widetilde{x}))} \phi(X)\phi(\widetilde{X})$.*

Definition 3 (Normalized kernel). *For a general kernel K, its normalized kernel is $\overline{K}(x,\widetilde{x}) = \frac{K(x,\widetilde{x})}{\sqrt{K(x,x)K(\widetilde{x},\widetilde{x})}}$.*

Normalized ReLU $\sigma(z) = \sqrt{2}\max(0,z)$ [2]

$$\widehat{\sigma}(\rho) = \frac{\sqrt{1-\rho^2} + \left(\pi - \cos^{-1}(\rho)\right)\rho}{\pi},$$

$$\Gamma_\sigma(K)(x,\widetilde{x}) = \sqrt{K(x,x)K(\widetilde{x},\widetilde{x})}\;\widehat{\sigma}(\overline{K}(x,\widetilde{x})).$$

Derivative of normalized ReLU $\sigma'(z) = \sqrt{2}\,\mathbb{1}\, z \geq 0$ [2]

$$\widehat{\sigma'}(\rho) = \frac{\pi - \cos^{-1}(\rho)}{\pi}, \qquad \Gamma_{\sigma'}(K)(x,\widetilde{x}) = \widehat{\sigma'}(\overline{K}(x,\widetilde{x})).$$

Network setup

Consider the following network structures where all but the last layers are trained and ReLU activation σ_0.

Feedforward Networks

$$x_0 = x; \quad f(x) = v^\top x_L$$

$$x_\ell = \sqrt{\frac{2}{m}}\sigma_0(W_\ell x_{\ell-1}),$$

where $\ell = 1,\cdots,L$, $W_1 \in \mathbb{R}^{m\times D}$ and $W_2,\cdots,W_L \in \mathbb{R}^{m\times m}$ are weight matrices.

Residual Networks

$$x_0 = \sqrt{\frac{1}{m}}Ax; \quad f(x) = v^\top x_L$$

$$x_\ell = x_{\ell-1} + \alpha\sqrt{\frac{1}{m}}V_\ell\sigma_0\Big(\sqrt{\frac{2}{m}}W_\ell x_{\ell-1}\Big),$$

where $\ell = 1,\cdots,L$, $W_\ell, V_\ell \in \mathbb{R}^{m\times m}$ for $\ell = 1,\cdots,L$ are weight matrices, and $\alpha = L^{-\gamma}$.

The NTK of residual network

NTK is computed via Gaussian process kernel (GP kernel) where only the last layer is trained in the network.

GP kernel of the ResNet

$$K_0(x,\widetilde{x}) = x^\top\widetilde{x};$$

$$K_\ell(x,\widetilde{x}) = K_{\ell-1}(x,\widetilde{x}) + \alpha^2\Gamma_\sigma(K_{\ell-1})(x,\widetilde{x}),$$

where $\ell = 1,\cdots,L$, and $\alpha = L^{-\gamma}$ for $0.5 \leq \gamma \leq 1$.

The NTK of the ResNet

$$\Omega_L(x,\widetilde{x}) = \alpha^2\sum_{\ell=1}^{L}\big[B_{\ell+1}(x,\widetilde{x})\Gamma_\sigma(K_{\ell-1})(x,\widetilde{x}) + K_{\ell-1}(x,\widetilde{x})B_{\ell+1}(x,\widetilde{x})\Gamma_{\sigma'}(K_{\ell-1})(x,\widetilde{x})\big],$$

where $B_{L+1}(x,\widetilde{x}) = 1$, and for $\ell = 1,\cdots,L-1$, B_ℓ's are

$$B_{\ell+1}(x,\widetilde{x}) = B_{\ell+2}(x,\widetilde{x}) + \alpha^2 B_{\ell+2}(x,\widetilde{x})\Gamma_{\sigma'}(K_\ell)(x,\widetilde{x}).$$

Theorem 4. *For the ResNet, given two inputs $x,\widetilde{x} \in \mathbb{S}^{D-1}$, $\epsilon < 0.5$, and*

$$m \geq C\epsilon^{-4}L^{2-2\gamma}\big(\log(320(L^2+1)/\delta) + 1\big),$$

where C is a constant, with probability at least $1-\delta$ over the randomness of the initialization, we have

$$\big|\langle\nabla_\theta f, \nabla_\theta \widetilde{f}\rangle - \Omega_L(x,\widetilde{x})\big| \leq 2L\alpha^2\epsilon,$$

where $\alpha = L^{-\gamma}$ with $\gamma \in [0.5,1]$.

The NTK of feedforward network

The following results for FFNets are from [1–3]

GP kernel of the FFNet

$$K_0(x,\widetilde{x}) = x^\top\widetilde{x};$$

$$K_\ell(x,\widetilde{x}) = \Gamma_\sigma(K_{\ell-1})(x,\widetilde{x}), \ \ell = 1,\cdots,L.$$

The NTK of the FFNet

$$\Omega_L(x,\widetilde{x}) = \sum_{\ell=1}^{L}\Big[K_{\ell-1}(x,\widetilde{x})\prod_{i=\ell}^{L}\Gamma_{\sigma'}(K_{i-1})(x,\widetilde{x})\Big].$$

Theorem 5 ([1]). *For the FFNet, when width $m \geq CL^6\epsilon^{-4}\log(L/\delta)$, where C is a constant, with probability at least $1-\delta$ over the initialization, for input $x,\widetilde{x} \in \mathbb{S}^{D-1}$*

$$\big|\langle\nabla_\theta f, \nabla_\theta\widetilde{f}\rangle - \Omega_L(x,\widetilde{x})\big| \leq L\epsilon.$$

The limiting NTK: depth$\to\infty$

Consider normalized kernels. The NTK of FFNet degenerates as depth increases while the NTK of ResNet is non-degenerate.

The limiting NTK of FFNet

$$\overline{\Omega}_L(x,\widetilde{x}) = \frac{1}{L}\Omega_L(x,\widetilde{x}).$$

Theorem 6. *For the NTK of the FFNet, as $L \to \infty$, given $x,\widetilde{x} \in \mathbb{S}^{D-1}$ and $|1 - x^\top\widetilde{x}| \geq \delta > 0$, where δ is a constant independent of L, we have*

$$\Big|\overline{\Omega}_L(x,\widetilde{x}) - 1/4\Big| = \mathcal{O}(\mathrm{polylog}(L)/L).$$

When $x = \widetilde{x}$, we have $\overline{\Omega}_L(x,\widetilde{x}) = 1, \forall L$.

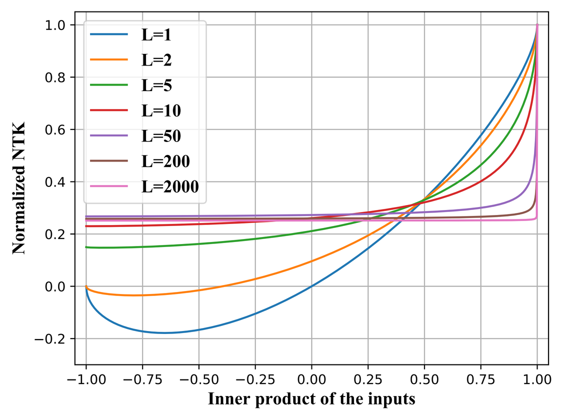
The limiting NTK of ResNet

$$\overline{\Omega}_L(x,\widetilde{x}) = \frac{1}{2L\alpha^2(1+\alpha^2)^{L-1}}\Omega_L(x,\widetilde{x}).$$

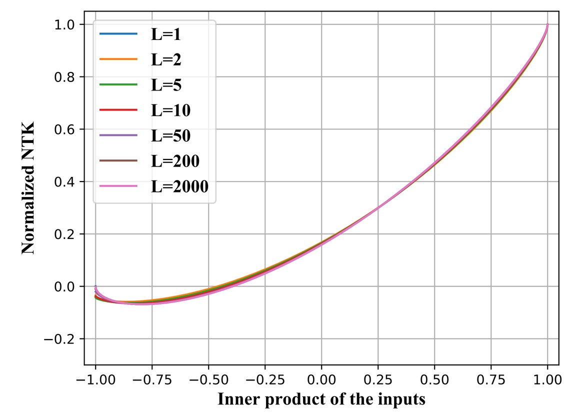
Theorem 7. *For the NTK of the ResNet, as $L \to \infty$, given $\alpha = \frac{1}{L}$ and $x,\widetilde{x} \in \mathbb{S}^{D-1}$ such that $|1 - x^\top\widetilde{x}| \geq \delta > 0$, where δ is a constant independent of L, we have*

$$\big|\overline{\Omega}_L(x,\widetilde{x}) - \overline{\Omega}_1(x,\widetilde{x})\big| = \mathcal{O}(1/L),$$

where $\overline{\Omega}_1(x,\widetilde{x}) = \frac{1}{2}\Big(\widehat{\sigma}(x^\top\widetilde{x}) + x^\top\widetilde{x}\cdot\widehat{\sigma'}(x^\top\widetilde{x})\Big)$.



NTKs of FFNet



NTKs of ResNet

Degeneracy of the NTK of FFNet

We will explain the degeneracy of the NTK of FFNet via the kernel regression problem.

Denote $\widetilde{\Omega}$ as the limiting NTK of the FFNets where

$$\widetilde{\Omega}(x,\widetilde{x}) = \lim_{L\to\infty}\overline{\Omega}_L(x,\widetilde{x}) = \begin{cases} 1/4, & x \neq \widetilde{x} \\ 1, & x = \widetilde{x} \end{cases}.$$

Kernel regression problem

For n independent observations $\{(x_i,y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^D$ is the feature vector, and $y_i \in \mathbb{R}$ is the response, assume $x_i \neq x_j$ for $i \neq j$, and $\sum_{i=1}^n y_i = 0$.

$$\text{Representer theorem} \Rightarrow f(\cdot) = \sum_{i=1}^{n}\beta_i\widetilde{\Omega}(x_i,\cdot).$$

Consider minimizing the regularized empirical risk,

$$\widehat{\beta} = \min_{\beta}\|y - \widetilde{\Omega}\beta\|^2 + \lambda\beta^\top\widetilde{\Omega}\beta,$$

where $\beta = (\beta_1,...\beta_n)^\top \in \mathbb{R}^n$, $y = (y_1,...,y_n)^\top \in \mathbb{R}^n$, and λ is the regularization parameter and usually very small for large n. This problem has a closed form solution

$$\widehat{\beta} = (\widetilde{\Omega} + \lambda I_n)^{-1}y = \frac{1}{\lambda + 3/4}\Big(I_n - \frac{1}{n+4\lambda+3}J_n\Big)y.$$

Then $f(x_j) = \sum_{i=1}^n\widehat{\beta}_i\widetilde{\Omega}(x_i,x_j) = \frac{3}{4\lambda+3}y_j$. Hence

- $f(x_j) \approx y_j$ for sufficiently large n and small λ.
- But for $x^* \neq x_1,...,x_n$,

$$f(x^*) = \sum_{i=1}^{n}\widehat{\beta}_i\widetilde{\Omega}(x_i,x^*) = \frac{1}{4}\sum_{i=1}^{n}\widehat{\beta}_i = 0.$$

This indicates that the function class induced by the limiting NTK of the FFNets $\widetilde{\Omega}$ is not learnable.

Reference

[1] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019.

[2] A. Daniely, R. Frostig, and Y. Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

[3] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.

Acknowledgement

Molei Tao was partially supported by NSF DMS-1847802 and ECCS-1936776 and Yuqing Wang was partially supported by NSF DMS-1847802.