STAT991 Topics in Deep Learning
Neural Tangent Kernels (NTK)

Presented by

Jiayao Zhang

University of Pennsylvania

jiayaozhang@acm.org

Sept. 12, 2019

Contents

Overview

- Consider linearizing the network function:

$$f_{\theta}(\boldsymbol{x}) = f_{\theta}(\boldsymbol{x}) + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})^{\top}(\boldsymbol{\theta} - \boldsymbol{\theta}) + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}\|), \tag{1}$$

where $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$ defines a feature map inducing the neural tangent kernel (NTK)

$$K\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathbb{E}_{\boldsymbol{\theta}}\left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{\theta}} \right\rangle. \tag{2}$$

- GD Training $\approx$ kernel gradient descent wrt NTK.
- The NTK will converge to a determinist kernel in the infinte size limit, and stays approximately constant during training.

Notations

- Data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \subset \mathbb{R}^d \times \mathbb{R}^{d_h}\}_{i=1}^n$, $\boldsymbol{x} \sim p^{in}$ $(= 1/n \sum_x \delta_x)$.
- MLP with L layers, d_h neurons at the h-th layer, $d_0 = d$.
- Preactivation $f^h(\boldsymbol{x}) = \boldsymbol{W}^h g^{(h-1)}(\boldsymbol{x}) \in \mathbb{R}^{d_h}$.
- Postactivation $g^h(\boldsymbol{x}) = \sqrt{\frac{c_\sigma}{d_h}} \sigma(f^h(\boldsymbol{x})) \in \mathbb{R}^{d_l}$, where $\sigma(\cdot)$ is elementwise activation, $c_\sigma^{-1} = \mathbb{E}_{z \sim \mathcal{N}(0,1)} \sigma^2(z)$ is a normalizing constant.
- $P = \sum_{l=1}^L n_l n_{l-1}$ = total # of parameters.
- $\boldsymbol{\theta} \in \mathbb{R}^P$ is the parameter.
- Network function $f_{\boldsymbol{\theta}} \in \mathcal{F} := \{\mathbb{R}^d \to \mathbb{R}^{d_L}\}$.
- Cost functional $C \in \mathcal{F}^* := \{\mathcal{F} \to \mathbb{R}\}$.
- Network realization function $F : \mathbb{R}^P \to \mathcal{F}$.

Contents

Contents

More Notations (1/2)

- **Multi-dimensional kernel** $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d_L \times d_L}$.
- $\langle f, g \rangle_{p^{in}} = \mathbb{E}_{x \sim p^{in}}[f(x)^\top g(x)]$.
- $\langle f, g \rangle_K = \mathbb{E}_{x, x' \sim p^{in}}[f(x)^\top K(x, x') g(x')]$.
- Partial applications of K: $K(\boldsymbol{x}, \cdot) : \mathbb{R}^d \to \mathbb{R}^{d_L \times d_L}$, $K_{i,\cdot}(\boldsymbol{x}, \cdot) : \mathbb{R}^d \to \mathbb{R}^{d_L}$.
- Given $\mu : \mathcal{F} \to \mathbb{R} \in \mathcal{F}^*$, for some $d \in \mathcal{F}$, $\mu = \langle d, \cdot \rangle_{p^{in}}$.
- Given K, define $\Phi_K : \mathcal{F}^* \to \mathcal{F} : \mu = \langle d, \cdot \rangle \mapsto f_\mu : \mathbb{R}^d \to \mathbb{R}^{d_L} : x \mapsto [\langle d, K_{i,\cdot}(\boldsymbol{x}, \cdot) \rangle]_i$.
- **Functional derivative** of C at $f_0 \in \mathcal{F}$: $\partial_f^{in} C|_{f_0} = \langle d|_{f_0}, \cdot \rangle$ for some $d|_{f_0} \in \mathcal{F}$.
- **Kernel gradient** of C wrt K: $\nabla_K C|_{f_0} = \Phi_K(\partial_f^{in} C|_{f_0}) = \frac{1}{N} \sum_{j=1}^N K(x, x_j) d|_{f_0}(x_j)$.
- $f(t)$ follows the **kernel gradient** wrt K if $\partial_t f(t) = -\nabla_K C|_{f(t)}$.
- $C(f(t))$ evolves as $\partial_t C|_{f(t)} = -\langle d|f(t), \nabla_K C|_{f(t)} \rangle_{p^{in}} = -\|d|_{f(t)}\|_K^2$.

More Notations (2/2)

- *Derivation of kernel gradient.*

$$\begin{aligned}\partial_t C|_{f(t)} &= \partial_f C|_{f(t)} \partial_t f(t) \\ &= -\langle d|_{f(t)}, \nabla_K C|_{f(t)} \rangle_{p^{in}},\end{aligned} \tag{3}$$

where we recall

$$\partial_f^{in} C|_{f(t)} = \langle d|_{f(t)}, \cdot \rangle_{p^{in}}. \tag{4}$$

Linear Approximation via Random Functions (1/6)

- A kernel can be approximated by P random functions $f^{(p)}$ from any distribution on $\mathcal{F}$ with covariance given by K, i.e.,

$$\mathbb{E}_{x,x'}[f_k^p(x)f_{k'}^p(x')] = K_{kk'}(x,x'). \tag{5}$$

- These functions define a random linear parameterization $F^{lin} : \mathbb{R}^p \to \mathcal{F}$:

$$\boldsymbol{\theta} \mapsto f_{\boldsymbol{\theta}}^{lin} = \frac{1}{\sqrt{P}} \sum_{p\in[P]} \theta f^{(p)}. \tag{6}$$

The partial derivatives are given by

$$\partial_{\theta_p} F^{lin}(\boldsymbol{\theta}) = \frac{1}{\sqrt{P}} f^{(p)}. \tag{7}$$

When $\boldsymbol{\theta}(t)$ varies with time, F^{lin} depends on t only through $\boldsymbol{\theta}$.

Linear Approximation via Random Functions (2/6)

- Optimizing $C \circ F^{lin} : \mathbb{R}^P \to \mathbb{R}$ (from parameters to scalar costs) using GD, the parameters follow

$$\begin{aligned} \partial_t \theta p(t) &= -\partial_{\theta_p}(C \circ F^{lin})(\theta(t)) = -\frac{1}{\sqrt{P}} \partial_f^{in} C|_{f^{lin}_{\theta(t)}} f^{(p)} \\ &= -\frac{1}{\sqrt{P}} \langle d|_{f^{lin}_{\theta(t)}}, f^{(p)} \rangle_{p^{in}}, \end{aligned} \tag{8}$$

since

$$\begin{aligned} \partial_f^{in} C|_{F^{lin}(\theta(t))} &= \langle d|_{f^{lin}_{\theta(t)}}, \cdot \rangle_{p^{in}}, \\ \partial_{\theta_p} F^{lin}(\theta) &= \frac{1}{\sqrt{P}} f^{(p)}. \end{aligned} \tag{9}$$

Linear Approximation via Random Functions (3/6)

- $\partial_t \theta p(t) = -\frac{1}{\sqrt{P}} \langle d|_{f^{lin}_{\theta(t)}}, f^{(p)} \rangle_{p^{in}}$.
- The function $f^{lin}_{\theta(t)}$ follows

$$\begin{aligned} \partial_t f^{lin}_{\theta(t)} &= \frac{1}{\sqrt{P}} \sum_{p \in [P]} \partial_t \theta_p(t) f^{(p)} \\ &= -\frac{1}{P} \sum_{p \in [P]} \langle d|_{f^{lin}_{\theta(t)}}, f^{(p)} \rangle_{p^{in}} f^{(p)}, \end{aligned} \tag{10}$$

and we want to interpret the RHS as the negative kernel gradient of some kernel.

Linear Approximation via Random Functions (4/6)

- $\partial_t f^{lin}_{\theta(t)} = -\frac{1}{P} \sum_{p \in [P]} \langle d|_{f^{lin}_{\theta(t)}}, f^{(p)} \rangle_{p^{in}} f^{(p)}$.
- The last term in Equation (10) is the kernel gradient $\nabla_{\tilde{K}} C$ wrt the **tangent kernel**

$$\tilde{K} = \sum_{p \in [P]} \partial_{\theta_p} F^{lin}(\theta) \otimes \partial_{\theta_p} F^{lin}(\theta) = \frac{1}{P} \sum_{p \in [P]} f^{(p)} \otimes f^{(p)}. \tag{11}$$

Linear Approximation via Random Functions (5/6)

- *Proof.* $\forall x \in \mathbb{R}^d$,

$$\partial_t f^{lin}_{\theta(t)}(x) = -\frac{1}{P} \sum_{p\in[P]} \langle d|_{f^{lin}_{\theta(t)}}, f^{(p)} \rangle_{p^{in}} f^{(p)}(x), \tag{12}$$

but

$$\begin{aligned} -\nabla_{\tilde{K}} C|_{f^{lin}_{\theta(t)}}(x) &= -\Phi_{\tilde{K}}(\partial^{in}_f C|f^{lin}_{\theta(t)}) \\ &= -\left[\langle d|f^{lin}_{\theta(t)}, \tilde{K}_{i,\cdot}(x,\cdot)\rangle_{p^{in}}\right]_i \in \mathbb{R}^{d_L} \\ &= -\frac{1}{P} \sum_{p\in[P]} \langle d|f^{lin}_{\theta(t)}, f^{(p)} \rangle_{p^{in}} f^{(p)}(x). \end{aligned} \tag{13}$$

Linear Approximation via Random Functions (6/6)

- This is a random kernel with

$$\tilde{K}_{ii'}(x, x') = \frac{1}{P} \sum_{p=1}^{P} f_i^{(p)}(x) f_{i'}^{(p)}(x'). \tag{14}$$

- Hence GD on the parameters amounts to kernel GD with the tangent kernel in the function space.
- In the limit $P \to \infty$, by LLN, the random $\tilde{K}$ tends to a fixed kernel K.

Neural Tangent Kernel

- For NNs trained using GD on the composition $C \circ F$, during training, the network function f_θ evolves along the (negative) kernel gradient

$$\partial_t f_{\theta(t)} = -\nabla_{\boldsymbol{\theta}} C|_{f_{\theta(t)}} \tag{15}$$

with respect to the NTK

$$\Theta(\theta) = \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta) \otimes \partial_{\theta_p} F^{(L)}(\theta). \tag{16}$$

Convergence of NTK

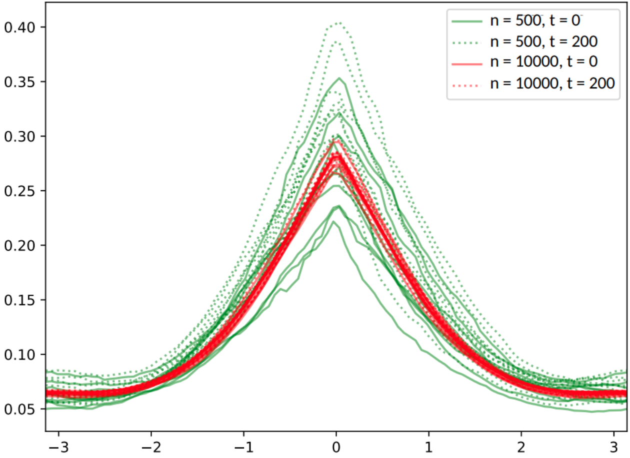


Figure 1: Convergence of the NTK to a fixed limit for two widths n and times t [JGH18].

Contents

Recall

- NTK can be given by the kernel function

$$K\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathbb{E}_{\boldsymbol{\theta}} \left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{\theta}} \right\rangle, \tag{17}$$

- where the gradient $\frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}$ appears from the gradient descent.

Infinite Width Limit of the MLP (1/6)

- Now suppose $d_L = 1$, i.e., $f(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}$. Consider training the neural network by minimizing the squared loss over training data:

$$\ell(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{n} \left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i \right)^2 . \tag{18}$$

Infinite Width Limit of the MLP (2/6)

- The last layer of the neural network is

$$
\begin{aligned}
f(\boldsymbol{\theta}, \boldsymbol{x}) =& f^{(L+1)}(\boldsymbol{x}) = \boldsymbol{W}^{(L+1)} \cdot \boldsymbol{g}^{(L)}(\boldsymbol{x}) \\
=& \boldsymbol{W}^{(L+1)} \cdot \sqrt{\frac{c_\sigma}{d_L}} \sigma \left(\boldsymbol{W}^{(L)} \cdot \sqrt{\frac{c_\sigma}{d_{L-1}}} \right. \\
& \left. \times \sigma \left(\boldsymbol{W}^{(L-1)} \cdots \sqrt{\frac{c_\sigma}{d_1}} \sigma \left(\boldsymbol{W}^{(1)} \boldsymbol{x} \right) \right) \right),
\end{aligned}
\tag{19}
$$

where $\boldsymbol{W}^{(L+1)} \in \mathbb{R}^{1 \times d_L}$ is the weights in the final layer, and $\boldsymbol{\theta} = \left(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(L+1)} \right)$ represents all the parameters in the network.

Infinite Width Limit of the MLP (3/6)

- All the weights are initialized as i.i.d. $\mathcal{N}(0,1)$. In the limit of $d_1, d_2, \ldots, d_L \to \infty$, the scaling factor $\sqrt{c_\sigma / d_h}$ in Equation (19) ensures that the norm of $\boldsymbol{g}^{(h)}(\boldsymbol{x})$ for each $h \in [L]$ is approximately preserved at initialization [DLL+18].
- In particular, for ReLU activation, we have $\mathbb{E}\left[\|\boldsymbol{g}^{(h)}(\boldsymbol{x})\|^2\right] = \|\boldsymbol{x}\|^2$ $(\forall h \in [L])$.

Infinite Width Limit of the MLP (4/6)

- From [LBN+17], one has the preactivations of each layer $h \in [L]$ have their each coordinates tending to Gaussian process of covariance $\Sigma^{(h-1)} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$:

$$
\begin{aligned}
\Sigma^{(0)}(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{x}^\top \boldsymbol{x}', \\
\boldsymbol{\Lambda}^{(h)}(\boldsymbol{x}, \boldsymbol{x}') &= \begin{pmatrix} \Sigma^{(h-1)}(\boldsymbol{x}, \boldsymbol{x}) & \Sigma^{(h-1)}(\boldsymbol{x}, \boldsymbol{x}') \\ \Sigma^{(h-1)}(\boldsymbol{x}', \boldsymbol{x}) & \Sigma^{(h-1)}(\boldsymbol{x}', \boldsymbol{x}') \end{pmatrix} \in \mathbb{R}^{2\times 2}, \qquad (20) \\
\Sigma^{(h)}(\boldsymbol{x}, \boldsymbol{x}') &= c_\sigma \mathbb{E}_{(u,v)\sim\mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^{(h)}\right)} \left[\sigma\left(u\right)\sigma\left(v\right)\right],
\end{aligned}
$$

Infinite Width Limit of the MLP (5/6)

- The intuition is that

$$\left[\boldsymbol{f}^{(h+1)}(\boldsymbol{x})\right]_i = \sum_{j=1}^{d_h} \left[\boldsymbol{W}^{(h+1)}\right]_{i,j} \left[\boldsymbol{g}^{(h)}(\boldsymbol{x})\right]_j \tag{21}$$

is a centered Gaussian process conditioned on $\boldsymbol{f}^{(h)}$ ($\forall i \in [d_{h+1}]$), with covariance

$$\mathbb{E}\left[\left[\boldsymbol{f}^{(h+1)}(\boldsymbol{x})\right]_i \cdot \left[\boldsymbol{f}^{(h+1)}(\boldsymbol{x}')\right]_i \middle| \boldsymbol{f}^{(h)}\right]. \tag{22}$$

Infinite Width Limit of the MLP (6/6)

- We further have

$$
\begin{aligned}
&\mathbb{E}\left[\left[\boldsymbol{f}^{(h+1)}(\boldsymbol{x})\right]_i \cdot \left[\boldsymbol{f}^{(h+1)}(\boldsymbol{x}')\right]_i \middle| \boldsymbol{f}^{(h)}\right] \\
=&\mathbb{E}\left[\sum_{j,k\in[d_h]} \boldsymbol{W}_{ij}^{(h+1)} \boldsymbol{W}_{ik}^{(h+1)} \boldsymbol{g}^{(h)}(\boldsymbol{x})_j \boldsymbol{g}^{(h)}(\boldsymbol{x}')_k \middle| \boldsymbol{f}^{(h)}\right] \\
=&\sum_{j,k\in[d_h]} \delta_{jk} \boldsymbol{g}^{(h)}(\boldsymbol{x})_j \boldsymbol{g}^{(h)}(\boldsymbol{x}')_k = \langle \boldsymbol{g}^{(h)}(\boldsymbol{x}), \boldsymbol{g}^{(h)}(\boldsymbol{x}') \rangle \\
=&\frac{c_\sigma}{d_h} \langle \sigma\left(f^{(h)}(\boldsymbol{x})\right), \sigma\left(f^{(h)}(\boldsymbol{x}')\right) \rangle \\
=&\frac{c_\sigma}{d_h} \sum_{j\in[d_h]} \sigma\left(f^{(h)}(\boldsymbol{x})\right)_j \cdot \sigma\left(f^{(h)}(\boldsymbol{x}')\right)_j \\
&\xrightarrow[\text{a.s.}]{\text{LLN}} c_\sigma \mathbb{E}_{(u,v)\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{\Lambda}^{(h)})}\left[\sigma(u)\,\sigma(v)\right] \equiv \Sigma^{(h)}(\boldsymbol{x}, \boldsymbol{x}'),
\end{aligned}
\tag{23}
$$

as $d_h \to \infty$ given that each $\boldsymbol{f}_j^{(h)}$ is a centered Gaussian process with covariance $\Sigma^{(h-1)}$.

Derivation of the NTK (1/8)

- To obtain the NTK, one computes the value that

$$\left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{\theta}} \right\rangle \tag{24}$$

converges to at random initialization in the infinite width limit.

Derivation of the NTK (2/8)

- We can write the partial derivative with respect to a particular weight matrix $\boldsymbol{W}^{(h)}$ in a compact form:

$$\frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{W}^{(h)}} = \mathbf{b}^{(h)}(\boldsymbol{x}) \cdot \left(\boldsymbol{g}^{(h-1)}(\boldsymbol{x})\right)^{\top}, \qquad h = 1, 2, \ldots, L+1, \tag{25}$$

where

$$\mathbf{b}^{(h)}(\boldsymbol{x}) = \begin{cases} 1 \in \mathbb{R}, & \text{for } h = L+1, \\ \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}) \left(\boldsymbol{W}^{(h+1)}\right)^{\top} \mathbf{b}^{h+1}(\boldsymbol{x}) \in \mathbb{R}^{d_h}, & \text{for } h \in [L]. \end{cases} \tag{26}$$

$$\boldsymbol{D}^{(h)}(\boldsymbol{x}) = \mathrm{diag}\left(\dot{\sigma}\left(\boldsymbol{f}^{(h)}(\boldsymbol{x})\right)\right) \in \mathbb{R}^{d_h \times d_h}, \qquad h \in [L]. \tag{27}$$

Derivation of the NTK (3/8)

- Then, for any $h \in [L+1]$,

$$\left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{W}^{(h)}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{W}^{(h)}} \right\rangle = \left\langle \mathbf{b}^{(h)}(\boldsymbol{x}) \cdot \left(\boldsymbol{g}^{(h-1)}(\boldsymbol{x})\right)^{\top}, \mathbf{b}^{(h)}(\boldsymbol{x}') \cdot \left(\boldsymbol{g}^{(h-1)}(\boldsymbol{x}')\right)^{\top} \right\rangle = \left\langle \boldsymbol{g}^{(h-1)}(\boldsymbol{x}), \boldsymbol{g}^{(h-1)}(\boldsymbol{x}') \right\rangle \times \left\langle \mathbf{b}^{(h)}(\boldsymbol{x}), \mathbf{b}^{(h)}(\boldsymbol{x}') \right\rangle. \tag{28}$$

- Note that we have established in Equation (22) that

$$\left\langle \boldsymbol{g}^{(h-1)}(\boldsymbol{x}), \boldsymbol{g}^{(h-1)}(\boldsymbol{x}') \right\rangle \to \Sigma^{(h-1)}\left(\boldsymbol{x}, \boldsymbol{x}'\right). \tag{29}$$

Derivation of the NTK (4/8)

- For the other factor $\left\langle \mathbf{b}^{(h)}(\boldsymbol{x}), \mathbf{b}^{(h)}(\boldsymbol{x}') \right\rangle$, by definition (26),

$$\begin{aligned}
&\left\langle \mathbf{b}^{(h)}(\boldsymbol{x}), \mathbf{b}^{(h)}(\boldsymbol{x}') \right\rangle \\
&= \left\langle \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}) \left(\boldsymbol{W}^{(h+1)}\right)^\top \mathbf{b}^{h+1}(\boldsymbol{x}), \right. \\
&\qquad \left. \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}') \left(\boldsymbol{W}^{(h+1)}\right)^\top \mathbf{b}^{h+1}(\boldsymbol{x}') \right\rangle .
\end{aligned} \tag{30}$$

- In Eq. (11) of [ADH+19], there is another factor $\frac{c_\sigma^{L-h}}{d_{h+1}\cdots d_L}$ preceeding the RHS of Eq. (30), which is likely to be a typo.

Derivation of the NTK (5/8)

- Although $\boldsymbol{W}^{(h+1)}$ and $\mathbf{b}^{h+1}(\boldsymbol{x})$ are dependent, the Gaussian initialization of $\boldsymbol{W}^{(h+1)}$ allows us to replace $\boldsymbol{W}^{(h+1)}$ with a fresh new sample $\widetilde{\boldsymbol{W}}^{(h+1)}$ without changing its limit.

$$
\begin{aligned}
&\left\langle \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}) \left(\boldsymbol{W}^{(h+1)}\right)^\top \mathbf{b}^{h+1}(\boldsymbol{x}), \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}') \left(\boldsymbol{W}^{(h+1)}\right)^\top \mathbf{b}_{h+1}(\boldsymbol{x}') \right\rangle \\
\approx &\left\langle \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}) \left(\widetilde{\boldsymbol{W}}^{(h+1)}\right)^\top \mathbf{b}_{h+1}(\boldsymbol{x}), \sqrt{\frac{c_\sigma}{d_h}} \boldsymbol{D}^{(h)}(\boldsymbol{x}') \left(\widetilde{\boldsymbol{W}}^{(h+1)}\right)^\top \mathbf{b}^{h+1}(\boldsymbol{x}') \right\rangle \\
\to &\frac{c_\sigma}{d_h} \operatorname{tr}\left(\boldsymbol{D}^{(h)}(\boldsymbol{x}) \boldsymbol{D}^{(h)}(\boldsymbol{x}')\right) \left\langle \mathbf{b}^{(h+1)}(\boldsymbol{x}), \mathbf{b}^{(h+1)}(\boldsymbol{x}')\right\rangle \\
\to &\dot{\Sigma}^{(h)}\left(\boldsymbol{x}, \boldsymbol{x}'\right) \left\langle \mathbf{b}^{(h+1)}(\boldsymbol{x}), \mathbf{b}^{(h+1)}(\boldsymbol{x}')\right\rangle,
\end{aligned} \tag{31}
$$

where

$$
\dot{\Sigma}^{(h)}(\boldsymbol{x}, \boldsymbol{x}') = c_\sigma \mathbb{E}_{(u,v)\sim\mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^{(h)}\right)} \left[\dot{\sigma}(u)\dot{\sigma}(v)\right]. \tag{32}
$$

Derivation of the NTK (6/8)

- Applying this approximation inductively in Equation (30), we get

$$\left\langle \mathbf{b}^{(h)}(\boldsymbol{x}), \mathbf{b}^{(h)}(\boldsymbol{x}') \right\rangle \to \prod_{h'=h}^{L} \dot{\Sigma}^{(h')}(\boldsymbol{x}, \boldsymbol{x}'). \tag{33}$$

Derivation of the NTK (7/8)

- Finally, since

$$\left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{\theta}} \right\rangle = \sum_{h=1}^{L+1} \left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{W}^{(h)}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{W}^{(h)}} \right\rangle, \tag{34}$$

Derivation of the NTK (8/8)

we have

$$\Theta^{(L)}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{h=1}^{L+1} \left(\Sigma^{(h-1)}(\boldsymbol{x}, \boldsymbol{x}') \cdot \prod_{h'=h}^{L+1} \dot{\Sigma}^{(h')}(\boldsymbol{x}, \boldsymbol{x}') \right), \tag{35}$$

where we write $\dot{\Sigma}^{(L+1)}(\boldsymbol{x}, \boldsymbol{x}') = 1$.

Contents

Convergence of NTK of MLP (1/2)

Theorem 3.1 [ADH+19]

Theorem 3.1 (Convergence to the NTK at initialization)

Fix $\epsilon > 0$ *and* $\delta \in (0,1)$*. Suppose* $\sigma(z) = \max(0, z)$ $(z \in \mathbb{R})$, $\min_{h\in[L]} d_h \geq \mathrm{poly}(L, 1/\epsilon) \cdot \log(L/\delta)$*, and* $[\boldsymbol{W}^{(h)}]_{i,j} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0,1)$ $\forall h \in [L+1], i \in [d_h], j \in [d_{h-1}]$*. Then for any inputs* $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{d_0}$ *such that* $\|\boldsymbol{x}\| \leq 1, \|\boldsymbol{x}'\| \leq 1$*, with probability at least* $1-\delta$ *we have:*

$$\left| \left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{\theta}} \right\rangle - \Theta^{(L)}(\boldsymbol{x}, \boldsymbol{x}') \right| \leq \epsilon. \tag{36}$$

Convergence of NTK of MLP (2/2)

Theorem 3.1 [ADH+19]

- Compared with [JGH18] and [Yan19], Theorem 3.1 of [ADH+19] is non-asymptotic.
- [JGH18] requires $d_0, \ldots, d_L \to \infty$ sequentially; [Yan19] requires $d_0, \ldots, d_L \to \infty$ at the same rate. But [ADH+19] requires $\min_h d_h \to \infty$.

Contents

Equivalence Between NTK and NN (1/1)

Theorem 3.2 [ADH+19]

Theorem 3.2 (Main theorem)

Suppose $\sigma(z) = \max(0, z)$ $(z \in \mathbb{R})$, $1/\kappa = \mathrm{poly}(1/\epsilon, \log(n/\delta))$ *and* $d_1 = d_2 = \cdots = d_L = m$ *with* $m \geq \mathrm{poly}(1/\kappa, L, 1/\lambda_0, n, \log(1/\delta))$. *Then for any* $\boldsymbol{x}_{te} \in \mathbb{R}^d$ *with* $\|\boldsymbol{x}_{te}\| = 1$, *with probability at least* $1 - \delta$ *over the random initialization, we have*

$$|f_{nn}(\boldsymbol{x}_{te}) - f_{ntk}(\boldsymbol{x}_{te})| \leq \epsilon. \tag{37}$$

Contents

Extensions and Recap

- NTK for MLPs ([JGH18, ADH+19]); NTK for convolutional nets (CNTK, [ADH+19]); NTK for graph nets (GNTK, [DHP+19]); finite approximation of NTK via MC methods [LXS+19].
- GD Training $\approx$ kernel gradient descent wrt NTK.
- The NTK will converge to a determinist kernel in the infinte size limit, and stays approximately constant during training.
- NTK provides a good tool for *theoretical analysis*, nonetheless it is debatable as to how well it captures reality.

GNTK: Experiments I

- Test accuracy is correlated with the dataset and architecture.

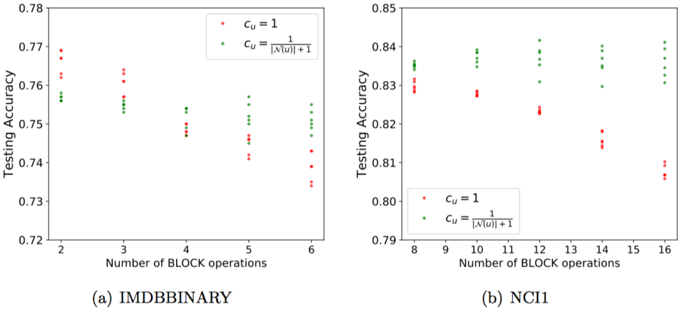
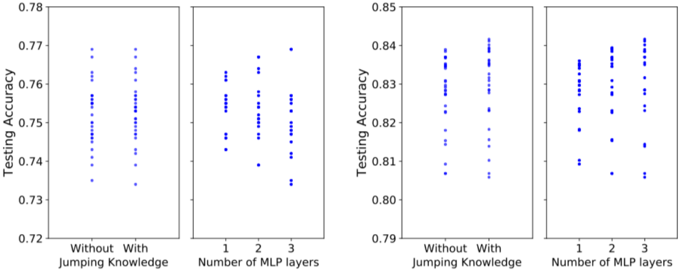


Figure 2: Effect of number of BLOCK operations, figure copied from [DHP+19].

GNTK: Experiments II

- Jump knowledge is expected to improve performance.
- The authors of [DHP+19] observed a 0.8% improvement in accuracy.



(a) IMDBBINARY (b) NCI1

Figure 3: Effect of jump knowledge, figure copied from [DHP+19].

References (1/2)

- Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang, *On exact computation with an infinitely wide neural net*, arXiv preprint arXiv:1904.11955 (2019).
- Stéphane Boucheron, Gábor Lugosi, and Pascal Massart, *Concentration inequalities: A nonasymptotic theory of independence*, Oxford university press, 2013.
- Peter L Bartlett and Shahar Mendelson, *Rademacher and gaussian complexities: Risk bounds and structural results*, Journal of Machine Learning Research **3** (2002), no. Nov, 463–482.
- Amit Daniely, Roy Frostig, and Yoram Singer, *Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity*, Advances In Neural Information Processing Systems, 2016, pp. 2253–2261.
- Simon S. Du, Kangcheng Hou, Barnabás Póczos, Ruslan Salakhutdinov, Ruosong Wang, and Keyulu Xu, *Graph neural tangent kernel: Fusing graph neural networks with graph kernels*, CoRR **abs/1905.13192** (2019).

References (2/2)

- Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai, *Gradient descent finds global minima of deep neural networks*, arXiv preprint arXiv:1811.03804 (2018).
- Arthur Jacot, Franck Gabriel, and Clément Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, Advances in neural information processing systems, 2018, pp. 8571–8580.
- Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein, *Deep neural networks as gaussian processes*, arXiv preprint arXiv:1711.00165 (2017).
- 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).
- Greg Yang, *Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation*, arXiv preprint arXiv:1902.04760 (2019).