Paper Review: Neural Tangent Kernel

Beomsu Kim*

July 4, 2021

Paper Information.

- Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel.
  In Neural Information Processing Systems, 2018.

1 Introduction

2 Neural Networks

- We consider fully-connected ANNs with layers numbered from 0 (input) to L (output).
- n_l : number of neurons in layer l.
- $\sigma : \mathbb{R} \to \mathbb{R}$: Lipschitz, twice differentiable nonlinearity function, with bounded second derivative.
- θ : weights $W^{(l)} \in \mathbb{R}^{n_l \times n_{l+1}}$ and bias vectors $b^{(l)} \in \mathbb{R}^{n_l+1}$. Initialized as i.i.d. Gaussians $\mathcal{N}(0,1)$.
- $P = \sum_{l=0}^{L-1}(n_l+1)n_{l+1}$: number of parameters.
- $\mathcal{F} = \{f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}\}$: space of functions.
- $F^{(L)} : \mathbb{R}^P \to \mathcal{F}$: ANN realization function, mapping parameters to the functions $f_\theta \in \mathcal{F}$.
- $p^{in} = \sum_{i=1}^{N} \delta_{x_i}$: input distribution.
- $\langle f, g \rangle_{p^{in}} = \mathbb{E}_{x \sim p^{in}}[f(x)^\top g(x)]$: bilinear form defined on p^{in}.
- $\|f\|_{p^{in}} = \langle f, f \rangle_{p^{in}}$: seminorm defined on p^{in}.
- Define the functions

$$
\begin{aligned}
\alpha^{(0)}(x;\theta) &= x, \\
\tilde{\alpha}^{(l+1)}(x;\theta) &= \frac{1}{\sqrt{n_l}} W^{(l)} \alpha^{(l)}(x;\theta) + \beta b^{(l)}, \\
\alpha^{(l)}(x;\theta) &= \sigma(\tilde{\alpha}^{(l)}(x;\theta)), \\
f_\theta(x) &= \tilde{\alpha}^{(L)}(x;\theta).
\end{aligned}
$$

*Department of Mathematical Sciences, KAIST. Email beomsu.kim@kaist.ac.kr

3 Kernel Gradient

- $C : \mathcal{F} \to \mathbb{R}$: functional cost.
- $K : \mathbb{R}^{n_0 \times n_0} \to \mathbb{R}^{n_L \times n_L}$: multi-dimensional kernel which satisfies $K(x, x') = K(x', x)^\top$.
- $\langle f, g \rangle_K = \mathbb{E}_{x,x' \sim p^{in}}[f(x)^\top K(x, x') g(x')]$: inner product w.r.t. kernel K.
- The kernel K is positive definite w.r.t. $\| \cdot \|_{p^{in}}$ if $\|f\|_{p^{in}} > 0 \implies \|f\|_K > 0$.
- $\mathcal{F}^* = \{\mu = \langle d, \cdot \rangle_{p^{in}} : d \in \mathcal{F}\}$: the dual space of $\mathcal{F}$.
- $\Phi_K : \mathcal{F}^* \to \mathcal{F}$ is defined such that

$$\Phi_K : \langle d, \cdot \rangle_{p^{in}} \mapsto \frac{1}{N} \sum_{i=1}^{N} K(\cdot, x_i) d(x_i).$$

  Φ_K can be interpreted as a map which interpolates d using the kernel K.
- $\partial_f^{in} C|_{f_0} = \langle d|_{f_0}, \cdot \rangle_{p^{in}}$: functional derivative of C at a point $f_0 \in \mathcal{F}$.
- $\nabla_K C|_{f_0} = \Phi_K(\partial_f^{in} C|_{f_0})$: kernel gradient.
- In contrast to $\partial_f^{in} C$ which is only defined on the dataset, the kernel gradient generalizes to values x outside the dataset thanks to the kernel K.
- A time-dependent function $f(t)$ follows the kernel gradient descent w.r.t. K if it satisfies

$$\partial_t f(t) = -\nabla_K C|_{f(t)} = -\Phi_K(\partial_f^{in} C|_{f(t)}) = -\frac{1}{N} \sum_{i=1}^{N} K(\cdot, x_i) d|_{f(t)}(x_i).$$

- During kernel gradient descent, the cost $C(f(t))$ evolves as

$$\begin{aligned}
\partial_t C|_{f(t)} = \partial_t C(f(t)) &= \partial_f^{in} C|_{f(t)}(\partial_t f(t)) \\
&= \langle d|_{f(t)}, \partial_t f(t) \rangle_{p^{in}} \\
&= \left\langle d|_{f(t)}, -\frac{1}{N} \sum_{i=1}^{N} K(\cdot, x_i) d|_{f(t)}(x_i) \right\rangle_{p^{in}} \\
&= \frac{1}{N} \sum_{j=1}^{N} d|_{f(t)}(x_j)^\top \left(-\frac{1}{N} \sum_{i=1}^{N} K(x_j, x_i) d|_{f(t)}(x_i) \right) \\
&= -\frac{1}{N^2} \sum_{j=1}^{N} \sum_{i=1}^{N} d|_{f(t)}(x_j)^\top K(x_j, x_i) d|_{f(t)}(x_i) \\
&= -\mathbb{E}_{x,x' \sim p^{in}}[d|_{f(t)}(x)^\top K(x, x') d|_{f(t)}(x')] \\
&= -\|d|_{f(t)}\|_K^2.
\end{aligned}$$

  Convergence to a critical point of C is hence guaranteed if the kernel K is positive definite with respect to $\| \cdot \|_{p^{in}}$: the cost is then strictly decreasing except at points such that $\|d|_{f(t)}\|_{p^{in}} = 0$. If the cost is convex and bounded from below, the function $f(t)$ therefore converges to a global minimum as $t \to \infty$.

3.1 Random Functions Approximation

- A kernel K can be approximated by a choice of P random functions $f^{(p)}$ sampled independently from any distribution on $\mathcal{F}$ whose (non-centered) covariance is given by the kernel K:

$$\mathbb{E}[f^{(p)}(x)f^{(p)}(x')^\top] = K(x,x')$$

  or equivalently,

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

- These functions define a random linear parametrization

$$F^{lin} : \mathbb{R}^P \to \mathcal{F} : \theta \mapsto f_\theta^{lin} = \frac{1}{\sqrt{P}}\sum_{p=1}^{P} \theta_p f^{(p)}.$$

- The partial derivatives of the parametrization are given by (e_p is the p-th standard basis vector)

$$\partial_{\theta_p} F^{lin}(\theta) = \lim_{h\to 0} \frac{F^{lin}(\theta + he_p) - F^{lin}(\theta)}{h} = \frac{1}{\sqrt{P}} f^{(p)}.$$

- Optimizing the cost $C \circ F^{lin}$ through gradient descent, the parameters follow the ODE

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

  The first equality holds since we are performing gradient descent, i.e., the instantaneous change of θ_p at time t must equal the gradient of θ_p w.r.t. the cost at time t.

- As a result, the function $f^{lin}_{\theta(t)}$ evolves according to

$$\begin{aligned}\partial_t f^{lin}_{\theta(t)} = \partial_t\left(\frac{1}{\sqrt{P}}\sum_{p=1}^{P}\theta_p(t)f^{(p)}\right) &= \frac{1}{\sqrt{P}}\sum_{p=1}^{P}\partial_t\theta_p(t)f^{(p)} \\ &= -\frac{1}{P}\sum_{p=1}^{P}\left\langle d|_{f^{lin}_{\theta(t)}}, f^{(p)}\right\rangle_{p^{in}} f^{(p)} \\ &= -\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N}\sum_{i=1}^{N} d|_{f^{lin}_{\theta(t)}}(x_i)^\top f^{(p)}(x_i)f^{(p)}(\cdot) \\ &= -\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N}\sum_{i=1}^{N}(f^{(p)}\otimes f^{(p)})(\cdot,x_i)d|_{f^{lin}_{\theta(t)}}(x_i) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{P}\sum_{p=1}^{P} f^{(p)}\otimes f^{(p)}\right)(\cdot,x_i)d|_{f^{lin}_{\theta(t)}}(x_i) \\ &= -\Phi_{\tilde{K}}(\partial_f^{in} C|_{f^{lin}_{\theta(t)}}) \\ &= -\nabla_{\tilde{K}} C|_{f^{lin}_{\theta(t)}}\end{aligned}$$

where

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

- This is a random n_L-dimensional kernel with values

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

- Performing gradient descent on the cost $C \circ F^{lin}$ is therefore equivalent to performing kernel gradient descent with the tangent kernel $\tilde{K}$ in the function space.
- With $P \to \infty$, by the law of large numbers, the random kernel $\tilde{K}$ tends to the fixed kernel K.

$$\lim_{P \to \infty} \tilde{K}_{ii'}(x, x') = \lim_{P \to \infty} \frac{1}{P} \sum_{p=1}^{P} f_i^{(p)}(x) f_{i'}^{(p)}(x') = \mathbb{E}[f_i^{(p)}(x) f_{i'}^{(p)}(x')] = K_{ii'}(x, x').$$

Hence, this method approximates kernel gradient descent with respect to the limiting kernel K.

4 Neural Tangent Kernel

- During training, the network function f_θ evolves along the negative kernel gradient

$$\partial_t f_{\theta(t)} = -\nabla_{\Theta^{(L)}} C|_{f_{\theta(t)}}$$

with respect to the neural tangent kernel (NTK)

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

This can be derived by following the steps in Section 3.1 with $F^{(L)}$ in place of F^{lin}.

- However, in contrast to F^{lin}, the realization function $F^{(L)}$ of ANNs is not linear.
- As a consequence, the derivatives $\partial_{\theta_p} F^{(L)}(\theta)$ and the NTK depend on the parameters θ.

4.1 Initialization

- The first key result is that in the limit $n_1, \ldots, n_{L-1} \to \infty$, the NTK converges in probability to a deterministic limiting kernel.

4.2 Training

- The second key result is that the NTK stays asymptotically constant during training.
- In general, the parameters can be updated according to a training direction $d_t \in \mathcal{F}$.

$$\partial_t \theta_p(t) = \left\langle \partial_{\theta_p} F^{(L)}(\theta(t)), d_t \right\rangle_{p^{in}}$$

- In the case of gradient descent,

$$\begin{aligned}
\partial_t \theta_p(t) = -\partial_{\theta_p}(C \circ F^{(L)})(\theta(t)) &= -\partial_{\theta_p} C(f_{\theta(t)}) \\
&= -\partial_f^{in} C|_{f_{\theta(t)}}(\partial_{\theta_p} f_{\theta(t)}) \\
&= \left\langle \partial_{\theta_p} f_{\theta(t)}, -d|_{f_{\theta(t)}} \right\rangle_{p^{in}} \\
&= \left\langle \partial_{\theta_p} F^{(L)}(\theta(t)), -d|_{f_{\theta(t)}} \right\rangle_{p^{in}}
\end{aligned}$$

and so

$$d_t = -d|_{f_{\theta(t)}}.$$