Notes on "Neural Tangent Kernel: Convergence and Generalization in Neural Networks" by Jacot et al., 2018

Junghyun Lee (Dept. of Mathematical Sciences, School of Computing)

May 24, 2021

In this note, I will cover the first paper[JGH18] that introduced the neural tangent kernel (NTK). In my opinion, the paper is written in a very mathematically (and physically[1]) intricate manner, but with not so much intuition. Many blogs and articles [Dwa19, DH20, Hus20, Pé19] provide excellent expositions to the intuition behind NTKs, but not much materials are available for understanding the paper as it is with all the mathematical subtleties[2].

This note is meant to serve as a "complete" companion to reading [JGH18]; assuming only elementary undergrad mathematics (little bit of analysis, some linear algebra...etc.), this note will supplement the paper with missing calculation steps, missing definitions from advanced mathematics...etc. (Some of the elementary proofs will be left as exercises, although the solutions will be provided at the Appendix) Also, borrowing from the mentioned blogs and articles, I will give the intuition behind NTK and some of its implications in the connection between kernel theory and deep learning.

1 Basic Setup (ANN)

Here, we only consider fully-connected networks of depth L, as shown in Figure ?.

2 Deep Learning as Optimization over Function Space

2.1 Essential Functional Analysis

For each given parameter vector $\theta \in \mathbb{R}^P$, the corresponding neural network f_θ can be regarded as an element of the function space $\mathcal{F} = \{f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}\}$, which is basically the collection of all possible functions from $\mathbb{R}^{n_0}$ to $\mathbb{R}^{n_L}$.

In order to endow $\mathcal{F}$ with a certain topological structure, let us recall few definitions from linear algebra [**?**] and (real) functional analysis [**?**]:

Definition 1. L_p space

Definition 2. Hilbert space

Definition 3. Banach space

[1] as in physics

[2] The closest resource was [Lee]

Definition 4. Let X be a real vector space. Then a function $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{R}$ is a **bilinear form (on X)** if it is linear in both arguments i.e. for all $x, y, z \in X$ and $c \in \mathbb{R}$,

$$\langle cx + y, z \rangle = c\langle x, z \rangle + \langle y, z \rangle$$

and

$$\langle z, cx + y \rangle = c\langle z, x \rangle + \langle z, y \rangle$$

Definition 5. A bilinear form $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{R}$ is a **semi-inner product (on X)** if it is symmetric and positive semi-definite i.e. for all $x, y \in X$, $\langle x, y \rangle = \langle y, x \rangle$ and $\langle x, x \rangle \geq 0$.

Theorem 6 (Cauchy-Schwarz Inequality)**.** *Let $\langle \cdot, \cdot \rangle$ be a semi-inner product on X. Then for any $x, y \in X$,*

$$\langle x, y \rangle^2 \leq \langle x, x \rangle \langle y, y \rangle$$

Proof. Exercise to the reader! ■

Definition 7. A function $p : X \to \mathbb{R}$ is a **seminorm (on X)** if the following properties hold:

1. Triangle inequality (Subadditivity):

$$p(x + y) \leq p(x) + p(y) \quad \forall x, y \in X$$

2. Absolute homogeneity:

$$p(sx) = |s| p(x) \quad \forall x \in X, s \in \mathbb{F}$$

Proposition 8. *Given a semi-inner product $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{R}$, the function $\|\cdot\| : X \to \mathbb{R}$ defined as $\|x\| = \sqrt{\langle x, x \rangle}$ is a seminorm on X.*

Proof. Exercise to the reader! ■

Remark 1. For semi-inner product $\langle \cdot, \cdot \rangle$, $\langle x, x \rangle = 0 \Rightarrow x = 0$ does not hold. For seminorm p, $p(x) = 0 \Rightarrow x = 0$ does not hold. With the positive definiteness, we call them inner product and norm, respectively.

Let p^{in} be a fixed probability measure over the input space $\mathbb{R}^{n_0}$. Then we can endow $\mathcal{F}$ with a semi-inner product as follows:

Proposition 9. *The function $\langle \cdot, \cdot \rangle_{p^{in}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, defined as below, is a semi-inner product.*

$$\langle f, g \rangle_{p^{in}} = \mathbb{E}_{x \sim p^{in}} \left[f(x)^\intercal g(x) \right] \tag{1}$$

Proof. Bilinearity follows directly from the linearity of expectation. Symmetry and positive semi-definiteness are trivial. ■

Observe that this quantifies how **similar** two given functions $f, g \in \mathcal{F}$ are w.r.t. the given input data distribution!

From hereon, we assume that p^{in} is the empirical distribution on the given finite training set $\{x_i\}_{i=1}^N \subset \mathbb{R}^{n_0}$ i.e. $p = \frac{1}{N} \sum_{i=1}^N \delta_{x_i}$ where δ_x is the Dirac measure on $x \in \mathbb{R}^{n_0}$.

2.2 Overview

Before diving into the mathematical details, let us get a grip of the big picture, first. We usually think of deep learning as optimizing over $\mathbb{R}^P$ and finding some optimal θ^* with respect to some loss function C i.e.

$$\theta^* = \operatorname*{argmin}_{\theta \in \mathbb{R}^P} C(\theta) \tag{2}$$

where $C(\theta)$ is the loss of the neural network constructed using the parameter vector θ. However, the curse of dimensionality and non-convexity of the loss surface deter us from studying the learning dynamics of deep learning. Here, we change the viewpoint of optimizing over the parameter space to **optimizing over the function space** $\mathcal{F}$. In other words, we consider C to be a cost functional[3] defined *on the function space* i.e. $C : \mathcal{F} \to \mathbb{R}$ and reformulate the optimization as follows:

$$\theta^* = \operatorname*{argmin}_{\theta \in \mathbb{R}^P} C(f_\theta) = \operatorname*{argmin}_{\theta \in \mathbb{R}^P} C \circ F^{(L)}(\theta) \tag{3}$$

Here, $F^{(L)} : \mathbb{R}^P \to \mathcal{F}$ is the ANN realization function that maps the parameter vector θ to the corresponding neural network f_θ. In this perspective, we observe that our difficulties mainly arose from the complexity of $F^{(L)}$, and thus, we can *exploit the intrinsic structure of C and $\mathcal{F}$ to gain a better understanding of the training dynamics of ANNs.*

Such exploitation is done in a kernel trick-fashioned way. Intuitively, we consider **neural network as a feature map**, and we construct the corresponding kernel that can be calculated without knowing what the neural network looks like With that kernel, we impose a topological structure on $\mathcal{F}$ induced by the kernel (spoiler: semi-inner product space) such that the complex training dynamics of ANN can be described by a simple dynamics on $\mathcal{F}$ w.r.t. that topology.

2.3 Kernel Gradient

We start with the basic definition:

Definition 10. A function $K : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}$ is a **multi-dimensional kernel** if it is a symmetric tensor in $\mathcal{F} \otimes \mathcal{F}$ i.e. $K(x, x') = K(x', x)^\intercal$ for all $x, x' \in \mathbb{R}^{n_0}$.

Remark 2. The kernel method that we're familiar with (i.e. kernel PCA) deals with the case when $n_L = 1$, which is usually denoted with $k(x, x')$. The connection, however, may be deceiving and sometimes lead to serious confusion. For instance, since $K(x, x')$ is a matrix, one may try to relate it to the Gram matrix. However, Gram matrix is $[k(x_i, x_j)]_{ij}$, while $K(x, x')$ is more like $\phi(x)\phi(x')^\intercal$ where $\phi : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is the feature map associated with K. Gram matrix for K, thus, have to be defined in a different way, as we will see later on.

In this subsection, let K be a fixed multi-dimensional kernel. We can endow $\mathcal{F}$ with a different semi-inner product, w.r.t. K, as follows:

Proposition 11. *The function $\langle \cdot, \cdot \rangle_K : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, defined as below, is a semi-inner product.*

$$\langle f, g \rangle_K = \mathbb{E}_{x, x' \sim p^{in}} \left[f(x)^\intercal K(x, x') g(x') \right] \tag{4}$$

[3]Functional is just a terminology used to describe any mapping that maps a function to a real number.

Proof. Bilinearity follows directly from the linearity of expectation. Symmetry and positive semi-definiteness are trivial. ■

Definition 12. Fréchet derivative

3 Case Study: Least-squares regression

References

[DH20] Simon Du and Wei Hu. Ultra-Wide Deep Nets and the Neural Tangent Kernel (NTK), Jul 2020.

[Dwa19] Rajat Vadiraj Dwaraknath. Understanding the Neural Tangent Kernel, Nov 2019.

[Hus20] Ferenc Huszár. Some Intuition on the Neural Tangent Kernel, Nov 2020.

[JGH18] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Lee] Jin Woo Lee. (Review) Neural Tangent Kernel.

[Pé19] Guillermo Valle Pérez. Notes on Neural Tangent Kernel: Convergence and Generalization in Neural Networks, by Jacot et al. Aug 2019. for the Fall2019 NTK reading group in Oxford.