Neural Tangent Kernel 概説

作成日 2020.2.24
最終更新 2024.4.3
渡部　海斗[*1]

1　本稿作成の目的

Neural Tangent Kernel (NTK) (Jacot et al., 2018) については既に多くのわかり易い解説 (Rajat, 2019; 甘利俊一, 2019; 鈴木大慈, 2019; 大阪大学医学部 Python 会, 2020) があります．本稿は自分自身の理解を深める，またその過程の中で NTK について様々な方に共有することができればと思い，作成しております．なるべく数学的な厳密性を保った上で作成を進めていくつもりではありますが，何か間違い等ありましたらissuesにあげていただく，あるいは Twitter(@kwignb) までご一報いただけますと非常に助かります．

2　記法とニューラルネットワークの定義式

$\mathcal{D} \subseteq \mathbb{R}^{n_0} \times \mathbb{R}^k$ をデータセットの集合とし，$\mathcal{X} = \{\boldsymbol{x} \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}\}$ と $\mathcal{Y} = \{\boldsymbol{y} \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}\}$ をそれぞれ入力データとラベルとする．中間層が L 層，各層の幅を n_l $(l = 1, \ldots, L)$ とし，出力層の幅 (クラス数) を $n_{L+1} = k$ とする．入力 $\boldsymbol{x} \in \mathbb{R}^{n_0}$ に対し，$h^l(\boldsymbol{x}), x^l(\boldsymbol{x}) \in \mathbb{R}^{n_l}$ を pre-activation function，post-activation function とする．このとき，ニューラルネットワークの再帰関係の定義式を，

$$\begin{cases} h^{l+1} = x^l \boldsymbol{W}^{l+1} + \boldsymbol{b}^{l+1} \\ x^{l+1} = \varphi(h^{l+1}) \end{cases} \tag{1}$$

$$\begin{cases} W_{ij}^l = \dfrac{\sigma_\omega}{\sqrt{n_l}} \omega_{ij}^l \\ b_j^l = \sigma_b \beta_j^l \end{cases} \tag{2}$$

とする．ここで，$\varphi : \mathbb{R} \to \mathbb{R}$ はリプシッツ連続 (Lipschitz continuous) かつ 2 回連続微分可能な要素毎の活性化関数 (element-wise activation function) であり，$\boldsymbol{W}^{l+1} \in \mathbb{R}^{n_l \times n_{l+1}}$ と $\boldsymbol{b}^{l+1} \in \mathbb{R}^{n_{l+1}}$ は重み行列とバイアスベクトルを表し，ω_{ij}^l, β_j^l は標準ガウス分布 $\mathcal{N}(0, 1)$ に従って初期化される．$\sigma_\omega^2, \sigma_b^2$ は重みとバイアスの分散である．この定義は標準的なニューラルネットワークの再帰関係の式

$$\begin{cases} h^{l+1} = x^l \boldsymbol{W}^{l+1} + \boldsymbol{b}^{l+1} \\ x^{l+1} = \varphi(h^{l+1}) \\ W_{ij}^l, b_j^l \sim \mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right), \quad k = \dfrac{6}{n_l + n_{l-1}} \end{cases} \tag{3}$$

とは異なり，NTK parametrization と呼ばれる (ここでは重みとバイアスの初期化に Glorot の一様分布 (Glorot and Bengio, 2010) を用いている)．NTK parametrization はネットワークのフォワードダイナミクスのみでなく，誤差逆伝播 (backpropagation) 時のダイナミクスも正規化している．

また，各層 l 毎のパラメータベクトル $\boldsymbol{\theta}^l \in \mathbb{R}^{(n_{l-1}+1)n_l}$ と全ネットワークのパラメータベクトル $\boldsymbol{\theta} \in \mathbb{R}^P$ $(P =$

[*1] watanabe.kaito.xu@alumni.tsukuba.ac.jp

$\sum_{l=0}^{L-1}(n_l+1)n_{l+1})$ を以下のように定義する．

$$\boldsymbol{\theta}^l \equiv \mathrm{vec}\left(\{\boldsymbol{W}^l, \boldsymbol{b}^l\}\right), \quad \boldsymbol{\theta} = \mathrm{vec}\left(\bigcup_{l=1}^{L} \boldsymbol{\theta}^l\right) \tag{4}$$

ここで $\mathrm{vec}(\cdot)$ は，行列の各列を縦に並べ，1 つの列ベクトルの形にするベクトル化を表す．連続時間 $t \in \mathbb{R}_0^+$ ($\mathbb{R}_0^+$ は 0 を含む正の実数の集合) におけるパラメータの時間依存を $\boldsymbol{\theta}_t$，その初期値を $\boldsymbol{\theta}_0$ とし，ニューラルネットワークの出力を $f_t(\boldsymbol{x}) \equiv h_t^{L+1}(\boldsymbol{x}) \in \mathbb{R}^k$ とする．$\hat{\boldsymbol{y}}$ をニューラルネットワークによる予測値とすると，損失関数 (loss function) は $\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ となる．教師あり学習 (supervised learning) では，以下に記述する経験損失 (empirical loss) $\mathcal{L}$ を最小化する $\boldsymbol{\theta}$ の学習を行う．

$$\mathcal{L}_t = \frac{1}{k|\mathcal{D}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \ell(f_t(\boldsymbol{x}, \boldsymbol{\theta}_t), \boldsymbol{y}) \tag{5}$$

3　学習の定義と NTK

学習は損失関数を減らしていくよう，勾配の逆方向にパラメータを変動させていく．バッチ学習を考えるものとして，以下にパラメータ $\boldsymbol{\theta}_t$ の学習を記述する．学習率を η と置く．このとき，

$$\begin{aligned} \dot{\boldsymbol{\theta}}_t &= -\eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} && (6) \\ &= -\eta \frac{\partial f_t(\mathcal{X})^{\mathrm{T}}}{\partial \boldsymbol{\theta}} \frac{\partial \mathcal{L}}{\partial f_t(\mathcal{X})} && (7) \\ &= -\eta \nabla_{\boldsymbol{\theta}} f_t(\mathcal{X})^{\mathrm{T}} \nabla_{f_t(\mathcal{X})} \mathcal{L} && (8) \end{aligned}$$

となる．ここで，$\dot{\boldsymbol{\theta}}$ は $\boldsymbol{\theta}$ の時間微分であり，$\partial/\partial x = \nabla_x$ である．また，$\nabla_{f_t(\mathcal{X})}\mathcal{L}$ はネットワークの出力 $f_t(\mathcal{X})$ に関する損失の勾配であり，$f_t(\mathcal{X}) = \mathrm{vec}([f_t(\boldsymbol{x})]_{\boldsymbol{x}\in\mathcal{X}}) \in \mathbb{R}^{k|\mathcal{D}|}$ である．

次に，学習によってネットワークが得る出力を表す関数 $f_t(\mathcal{X})$ がどのように変化していくかを確認する．パラメータ $\boldsymbol{\theta}_t$ の学習と同様，時間微分を考えると，

$$\begin{aligned} \dot{f}_t(\mathcal{X}) &= \frac{\partial f_t(\mathcal{X})}{\partial t} && (9) \\ &= \frac{\partial f_t(\mathcal{X})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial t} && (10) \\ &= \nabla_{\boldsymbol{\theta}} f_t(\mathcal{X}) \dot{\boldsymbol{\theta}}_t && (11) \end{aligned}$$

と書くことができる．さらに，式 (8) を用いれば，

$$\begin{aligned} \dot{f}_t(\mathcal{X}) &= -\eta \nabla_{\boldsymbol{\theta}} f_t(\mathcal{X}) \nabla_{\boldsymbol{\theta}} f_t(\mathcal{X})^{\mathrm{T}} \nabla_{f_t(\mathcal{X})} \mathcal{L} && (12) \\ &= -\eta \hat{\Theta}_t(\mathcal{X}, \mathcal{X}) \nabla_{f_t(\mathcal{X})} \mathcal{L} && (13) \end{aligned}$$

と書くことができる．$\hat{\Theta}_t = \hat{\Theta}_t(\mathcal{X}, \mathcal{X}) \in \mathbb{R}^{k|\mathcal{D}|\times k|\mathcal{D}|}$ は時間 t における Neural Tangent Kernel (NTK) であり，以下のように定義される．

$$\hat{\Theta}_t = \nabla_{\boldsymbol{\theta}} f_t(\mathcal{X}) \nabla_{\boldsymbol{\theta}} f_t(\mathcal{X})^{\mathrm{T}} = \sum_{l=1}^{L+1} \nabla_{\boldsymbol{\theta}^l} f_t(\mathcal{X}) \nabla_{\boldsymbol{\theta}^l} f_t(\mathcal{X})^{\mathrm{T}} \tag{14}$$

また，$\mathcal{X}$ 以外の入力 $\boldsymbol{x}' \in \mathbb{R}^{n_0}$ に対する NTK は $\hat{\Theta}_t(\boldsymbol{x}', \mathcal{X})$ と定義できる．この NTK を用いて，学習方程式を関数空間で考える．

4　中間層が無限幅のネットワーク

3 節では NTK の定義を与えたが，一般には関数空間内での学習を考える式 (13) の計算は難しい．何故なら，NTK $\hat{\Theta}_t$ は時間毎に変化するためである．しかし，損失関数を平均二乗誤差 (mean square error; MSE)，$\lambda_{\min/\max}(\Theta)$ を NTK Θ の最小/最大固有値，$\eta_{\text{critical}} := 2(\lambda_{\min}(\Theta) + \lambda_{\max}(\Theta))^{-1}$ としたとき，以下の定理が証明されている．

定理 4.1. (**Lee et al., 2019**)

$n_1 = \cdots = n_L = n$ とし，$\lambda_{\min}(\Theta) > 0$ を仮定する．学習率 η が η_{critical} より小さく，任意の入力 $\boldsymbol{x} \in \mathbb{R}^{n_0}$ が $\|\boldsymbol{x}\|_2 \le 1$ を満たすとき，以下が成立する．

$$\sup_{t\ge0} \|f_t(\boldsymbol{x}) - f_t^{lin}(\boldsymbol{x})\|_2,\ \sup_{t\ge0} \frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2}{\sqrt{n}},\ \sup_{t\ge0} \left\|\hat{\Theta}_t - \hat{\Theta}_0\right\|_F = \mathcal{O}\left(n^{-\frac{1}{2}}\right),\ \text{as}\quad n \to \infty$$

Proof. □

$f_t^{lin}(\boldsymbol{x})$ について，詳細は式 (15) で示すが NTK $\hat{\Theta}_0$ を用いて記述されたモデルの出力である．この定理は中間層の幅を無限にしたとき，t を大きく取ったときにも学習の最適なパラメータ $\boldsymbol{\theta}_t$ は $\boldsymbol{\theta}_0$ の近傍にあることを主張しており，さらに Θ_t を Θ_0 で近似できることも主張している．この定理を用いて，NTK を用いたネットワークの出力ダイナミクスの記述を行う．

5　NTK を用いたネットワークの出力ダイナミクス

本節ではニューラルネットワークの出力を 1 次のテイラー展開 (Taylor expansion) で置き換えることにより，線形化されたニューラルネットワーク (linearized neural network) の学習のダイナミクスを記述する．時間 t 時点における線形化されたネットワークの出力を $f_t^{lin}(\boldsymbol{x})$ とすると，1 次のテイラー展開は，

$$f_t^{lin}(\boldsymbol{x}) \equiv f_0(\boldsymbol{x}) + \nabla_{\boldsymbol{\theta}} f_0(\boldsymbol{x})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}\boldsymbol{\omega}_t \tag{15}$$

と書ける．ここで，$\boldsymbol{\omega}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$ である．f_t^{lin} は第一項がネットワークの初期値であり，第二項が学習中の初期値からの値の変化を示す．$\boldsymbol{\omega}_t$ の時間微分は

$$\dot{\boldsymbol{\omega}}_t = -\eta \nabla_{\boldsymbol{\theta}} f_0(\mathcal{X})^{\mathrm{T}} \nabla_{f_t^{lin}(\mathcal{X})}\mathcal{L} \tag{16}$$

となる．$\nabla_{\boldsymbol{\theta}} f_0(\boldsymbol{x})$ は学習を通して一定の値を取るので，損失関数に MSE，すなわち $\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}|\hat{\boldsymbol{y}} - \boldsymbol{y}|_2^2$ を使用したとき，式 (16) について常微分方程式を解けば，

$$\boldsymbol{\omega}_t = -\nabla_{\boldsymbol{\theta}} f_0(\mathcal{X})^{\mathrm{T}} \hat{\Theta}_0^{-1}\left(I - e^{-\eta\hat{\Theta}_0 t/k|\mathcal{D}|}\right)(f_0(\mathcal{X}) - \mathcal{Y}) \tag{17}$$

が得られる．つまり，$f_t^{lin}(\mathcal{X})$ は，

$$\begin{aligned} f_t^{lin}(\mathcal{X}) &= f_0(\mathcal{X}) - \nabla_{\boldsymbol{\theta}} f_0(\mathcal{X}) \nabla_{\boldsymbol{\theta}} f_0(\mathcal{X})^{\mathrm{T}} \hat{\Theta}_0^{-1}\left(I - e^{-\eta\hat{\Theta}_0 t/k|\mathcal{D}|}\right)(f_0(\mathcal{X}) - \mathcal{Y}) && (18)\\ &= f_0(\mathcal{X}) - \hat{\Theta}_0\hat{\Theta}_0^{-1}\left(I - e^{-\eta\hat{\Theta}_0 t/k|\mathcal{D}|}\right)(f_0(\mathcal{X}) - \mathcal{Y}) && (19)\\ &= f_0(\mathcal{X}) - \left(I - e^{-\eta\hat{\Theta}_0 t/k|\mathcal{D}|}\right)(f_0(\mathcal{X}) - \mathcal{Y}) && (20)\\ &= \left(I - e^{-\eta\hat{\Theta}_0 t/k|\mathcal{D}|}\right)\mathcal{Y} + e^{-\eta\hat{\Theta}_0 t/k|\mathcal{D}|} f_0(\mathcal{X}) && (21) \end{aligned}$$

と書ける．$\mathcal{X}$ 以外の入力 $\boldsymbol{x}'$ に対する出力 $f_t^{lin}(\boldsymbol{x}')$ は，

$$f_t^{lin}(\boldsymbol{x}') = f_0(\boldsymbol{x}') - \nabla_{\boldsymbol{\theta}} f_0(\boldsymbol{x}') \nabla_{\boldsymbol{\theta}} f_0(\mathcal{X})^{\mathrm{T}} \hat{\Theta}_0^{-1} \left(I - e^{-\eta \hat{\Theta}_0 t / k|\mathcal{D}|} \right) (f_0(\mathcal{X}) - \mathcal{Y}) \tag{22}$$

$$= f_0(\boldsymbol{x}') - \hat{\Theta}_0(\boldsymbol{x}', \mathcal{X}) \hat{\Theta}_0^{-1} \left(I - e^{-\eta \hat{\Theta}_0 t / k|\mathcal{D}|} \right) (f_0(\mathcal{X}) - \mathcal{Y}) \tag{23}$$

と書ける．したがって，初期化時のネットワークの出力 $f_0(\mathcal{X})$，$f_0(\boldsymbol{x}')$ と NTK $\hat{\Theta}_0$，$\hat{\Theta}_0(\boldsymbol{x}', \mathcal{X})$ を計算すれば，勾配降下法を実行することなく，線形化されたニューラルネットワークの各時間における学習結果を計算できるということである．通常であれば，このようにネットワークの出力を構成したとしても，各時間における NTK $\hat{\Theta}_t$ を計算する必要があり，何もメリットがない．しかし，4.3 節で述べた定理4.1により，中間層の幅を十分に大きく取れば，任意の学習時間における NTK $\hat{\Theta}_t$ を $\hat{\Theta}_0$ で近似することができるため，関数空間内においてネットワークを線形化することができる．

6 NTK の関連研究

6.1 NTK の関連研究の概要

NTK はJacot et al. (2018) によって提案されたニューラルネットワークの大域収束性の保証に貢献する理論である．近年，Cho and Saul (2009) により深層ニューラルネットワークの学習に対応するカーネル関数が導出され，Neal (2012); Lee et al. (2018) がそのカーネル関数をガウス過程 (gaussian process; GP) の共分散関数として使用することで，中間層の幅を無限に近づけたニューラルネットワークを表現することができることを確認した．NTK は，ニューラルネットワークの各層の幅を無限に近づけることで，NTK というグラム行列が決定的 (deterministic) になるということを証明し，特定の条件下においてその行列の正定値性を示すことで，大域的最適解への線形収束を保証した．Lee et al. (2019) はある有限の学習率の設定のもと，モデルの出力のダイナミクスが NTK によって決定されるダイナミクスに従うことを示した．NTK の性質に関する解析も進んでおり，入力の違いに対する安定性と近似能力が優れていることが示されている (Bietti and Mairal, 2019)．Amari (2020) はJacot et al. (2018) が与えている NTK の数学的に複雑な内容について，任意の目的関数がランダムに生成される目的関数のごく近傍に存在することを幾何的に示した．また，NTK は層の幅を十分に大きくすることを前提としているが，NTK における層の深さの役割に関する研究も進んでいる．Yang and Salman (2019) は識別したい関数の複雑さに応じて最適な深さが存在することを示している．Hanin and Nica (2020) は層の幅と深さの比率を固定して十分に大きくとったときの NTK の平均と分散を研究している．

また，Jacot et al. (2018) が提案した NTK が適用される構造は通常のニューラルネットワークのみであり，畳み込みニューラルネットワーク (convolutional neural network; CNN) についてはArora et al. (2019) が Convolutional NTK (CNTK) を，自己符号化器 (autoencoder; AE) についてはNguyen et al. (2019) が対応する NTK，グラフニューラルネットワーク (graph neural network; GNN) についてはDu et al. (2019) が Graph NTK (GNTK) を提案している．

6.2 Fast Finite Width Neural Tangent Kernel

参考文献

S. Amari. Any Target Function Exists in a Neighborhood of Any Sufficiently Wide Random Network: A Geometrical Perspective. *arXiv preprint arXiv:2001.06931*, 2020.

S. Arora, S. S. Du, H. Wei, Z. Li, R. Salakhutdinov, and R. Wang. On Exact Computation with an Infinitely Wide Neural Net. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A. Bietti and J. Mairal. On the Inductive Bias of Neural Tangent Kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Y. Cho and L. K. Saul. Kernel Methods for Deep Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 342–350, 2009.

S. S. Du, K. Hou, R. R. Salakhutdinov, B. Poczos, R. Wang, and K. Xu. Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5724–5734, 2019.

X. Glorot and Y. Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Artificial Intelligence and Statistics (AISTATS)*, pages 249–256, 2010.

B. Hanin and M. Nica. Finite Depth and Width Corrections to the Neural Tangent Kernel. In *International Conference on Learning Representations (ICLR)*, 2020.

A. Jacot, F. Gabriel, and C. Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. In *International Conference on Learning Representations (ICLR)*, 2018.

J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, J. Sohl-Dickstein, and J. Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

R. M. Neal. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media, 2012.

T. V. Nguyen, R. K. Wong, and C. Hegde. Benefits of Jointly Training Autoencoders: An Improved Neural Tangent Kernel Analysis. *arXiv preprint arXiv:1911.11983*, 2019.

V. D. Rajat. Understanding the Neural Tangent Kernel, 2019.

G. Yang and H. Salman. A Fine-Grained Spectral Perspective on Neural Networks. *arXiv preprint arXiv:1907.10599*, 2019.

大阪大学医学部 Python 会. Neural Tangents による無限幅深層ニューラルネットワークの構築とベイズ推論, 2020.

甘利俊一. 新版　情報幾何学の新展開. サイエンス社, 2019.

鈴木大慈. 大阪大学集中講義　深層学習の数理, 2019.