Learnability in the Context of Neural Tangent Kernels

Anonymous Author(s)
Affiliation
Address
email

Abstract

Understanding the prioritization of certain samples over others during neural network training is a fundamental challenge in deep learning. This prioritization is intrinsically linked to the network's inductive bias—the inherent assumptions that enable generalization from training data to unseen data. In this study, we investigate the role of the diagonal elements of the Neural Tangent Kernel (NTK), $k(x, x)$, in determining sample learnability. Through theoretical analysis, we demonstrate that higher values of $k(x, x)$ correlate with faster convergence rates of individual sample errors during training, indicating that such samples are learned more rapidly and accurately. Conversely, lower $k(x, x)$ values are associated with slower learning dynamics, classifying these samples as harder to learn. Empirical evaluations conducted on standard datasets using convolutional neural networks (CNNs), validate our theoretical predictions. We observe that samples with higher $k(x, x)$ values consistently achieve higher accuracy in fewer training epochs compared to those with lower values. Visual inspections further reveal that high-$k(x, x)$ samples are typically clear and prototypical, whereas low-$k(x, x)$ samples often exhibit noise or atypical characteristics.

1 Introduction

Understanding why neural networks prioritize learning certain samples over others is a fundamental question in deep learning. This prioritization is closely tied to the network's inductive bias—the set of assumptions a model makes to generalize from training data to unseen data. The Neural Tangent Kernel (NTK) has emerged as a powerful theoretical tool to analyze the training dynamics of overparameterized neural networks [1, 2]. In this work, we focus on the diagonal elements of the NTK, $k(x, x)$, and their relationship with sample learnability. Our empirical observations show that

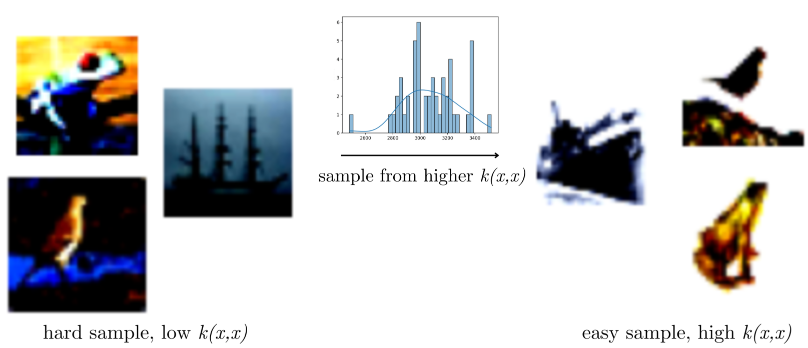

Figure 1: *Hard* (***left***) and *easy* (***right***) to learn CIFAR-10 samples for a CNN, picked by our method.

Submitted to Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

samples with higher $k(x, x)$ values tend to be learned faster and more accurately during training. These samples often correspond to "easy" examples with less complexity or ambiguity. Conversely, samples with lower $k(x, x)$ values are learned more slowly and are generally harder examples. We aim to theoretically justify this phenomenon by deriving convergence rates based on $k(x, x)$ and explaining how $k(x, x)$ affects the optimization dynamics of neural networks.

2 Theoretical Analysis

The NTK arises in the study of infinitely wide neural networks trained using gradient descent. For a neural network $f(\mathbf{x}; \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$, the NTK is defined as [2][3]:

$$K(\mathbf{x}, \mathbf{x}') = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})^\top \nabla_{\boldsymbol{\theta}} f(\mathbf{x}'; \boldsymbol{\theta}). \tag{1}$$

The diagonal elements $k(x, x)$ represent the inner product of the gradient of the network's output with respect to its parameters at input $\mathbf{x}$:

$$k(x, x) = \|\nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})\|^2. \tag{2}$$

This quantity measures the sensitivity of the output at $\mathbf{x}$ to changes in the parameters and thus can be interpreted as the influence of sample $\mathbf{x}$ on the training dynamics. High $k(x, x)$ indicates that small parameter updates can significantly affect the output for $\mathbf{x}$, potentially leading to faster learning for that sample.

Consider training a neural network with mean squared error (MSE) loss on dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The network output at time t is $f_t(\mathbf{x})$. The dynamics of $f_t(\mathbf{x})$ during gradient descent are governed by:

$$\frac{df_t(\mathbf{x})}{dt} = -\eta \sum_{i=1}^N K(\mathbf{x}, \mathbf{x}_i)(f_t(\mathbf{x}_i) - y_i), \tag{3}$$

where η is the learning rate. This differential equation characterizes how the network's output evolves over time under gradient descent. In the infinite-width limit, $K(\mathbf{x}, \mathbf{x}')$ remains constant[2][1], and the solution can be expressed as:

$$f_t = f_0 - (I - e^{-\eta K t})(f_0 - y), \tag{4}$$

where f_0 is the initial network output, y is the vector of target values, and K is the NTK matrix.

2.1 Convergence Rate for Individual Samples

To understand how the error for each individual sample evolves during training, we derive the Ordinary Differential Equation (ODE) governing the error dynamics within the Neural Tangent Kernel (NTK) framework. This derivation is foundational for analyzing the convergence rates and establishing the relationship between the NTK's diagonal elements and sample learnability. Consider training a neural network $f(\mathbf{x}; \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$ on a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ using gradient descent to minimize the mean squared error (MSE) loss:

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^N \left(f(\mathbf{x}_i; \boldsymbol{\theta}) - y_i\right)^2. \tag{5}$$

The continuous-time gradient descent (i.e, gradient flow) update rule is given by:

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -\eta \nabla_{\boldsymbol{\theta}} \mathcal{L}, \tag{6}$$

where η is the learning rate. The time derivative of the network's output for sample $\mathbf{x}$ is expressed as:

$$\frac{df(\mathbf{x}; \boldsymbol{\theta}(t))}{dt} = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta}(t))^\top \frac{d\boldsymbol{\theta}(t)}{dt}. \tag{7}$$

Substituting the gradient descent update rule into this equation yields,

$$\begin{aligned} \frac{df(\mathbf{x}; \boldsymbol{\theta}(t))}{dt} &= -\eta \sum_{i=1}^N \left(f(\mathbf{x}_i; \boldsymbol{\theta}(t)) - y_i\right) \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta}(t))^\top \nabla_{\boldsymbol{\theta}} f(\mathbf{x}_i; \boldsymbol{\theta}(t)) & (8) \\ &= -\eta \sum_{i=1}^N K(\mathbf{x}, \mathbf{x}_i) \left(f(\mathbf{x}_i; \boldsymbol{\theta}(t)) - y_i\right), & (9) \end{aligned}$$

where the Neural Tangent Kernel (NTK) is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})^\top \nabla_{\boldsymbol{\theta}} f(\mathbf{x}'; \boldsymbol{\theta}). \tag{10}$$

On defining the following vector quantities:

$$\mathbf{f}(t) = \begin{bmatrix} f(\mathbf{x}_1; \boldsymbol{\theta}(t)) \\ \vdots \\ f(\mathbf{x}_N; \boldsymbol{\theta}(t)) \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \qquad \mathbf{e}(t) = \mathbf{f}(t) - \mathbf{y}, \tag{11}$$

We may rewrite the evolution equation more compactly, as:

$$\frac{d\mathbf{f}(t)}{dt} = -\eta K \mathbf{e}(t), \tag{12}$$

where K is the NTK matrix with elements $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. Substituting $\mathbf{e}(t)$ into the evolution equation, we obtain a simple ordinary differential equation governing the error dynamics:

$$\frac{d\mathbf{e}(t)}{dt} = -\eta K \mathbf{e}(t). \tag{13}$$

Assuming that the NTK matrix K is *diagonal* or *approximately diagonal*—a reasonable assumption in the infinite-width limit where off-diagonal elements become negligible [4, 2, 1]—the system decouples into independent ODEs for each sample:

$$\frac{de_t(\mathbf{x}_i)}{dt} = -\eta k(x_i, x_i) e_t(\mathbf{x}_i), \implies e_t(\mathbf{x}_i) = e_0(\mathbf{x}_i) e^{-\eta k(x_i, x_i) t}, \tag{14}$$

where $e_0(\mathbf{x}_i) = e_{t=0}(\mathbf{x}_i)$ is the initial error for sample $\mathbf{x}_i$. The solution $e_t(\mathbf{x}_i) = e_0(\mathbf{x}_i) e^{-\eta k(x_i, x_i) t}$ indicates that the error for each sample decays *exponentially* over time. The *rate of convergence* is governed by the product $\eta k(x_i, x_i)$.

2.2 Bounds on Convergence Rates

We can formalize the convergence rates by deriving bounds on the error $e_t(\mathbf{x}_i)$:

$$e^{-\eta k_{\max} t} e_0(\mathbf{x}_i) \leq e_t(\mathbf{x}_i) \leq e^{-\eta k_{\min} t} e_0(\mathbf{x}_i), \tag{15}$$

These bounds indicate that the error decay rate is bounded by the minimum and maximum diagonal elements of the NTK. However, these bounds are loose as they do not capture the individual variability of each $k(x_i, x_i)$. A more precise estimate considers each $k(x_i, x_i)$ individually:

$$e_t(\mathbf{x}_i) = e^{-\eta k(x_i, x_i) t} e_0(\mathbf{x}_i). \tag{16}$$

This relationship implies that for **higher** $k(x_i, x_i)$**, the error** $e_t(\mathbf{x}_i)$ **decreases** ***faster***, leading to quicker learning for sample $\mathbf{x}_i$. **Lower** $k(x_i, x_i)$ **leads to the error** $e_t(\mathbf{x}_i)$ **decreasing** ***slower***, indicating that sample $\mathbf{x}_i$ is learned more gradually. Focusing on the error $e_t(\mathbf{x}) = f_t(\mathbf{x}) - y(\mathbf{x})$, we analyze the convergence rate for each sample $\mathbf{x}$. Assuming K is positive definite, the error dynamics for the i-th sample are:

$$e_t(\mathbf{x}_i) = e^{-\eta K t} e_0(\mathbf{x}_i). \tag{17}$$

If K is diagonal or approximately diagonal, the convergence rate for each sample simplifies to:

$$e_t(\mathbf{x}_i) \approx e^{-\eta k(x_i, x_i) t} e_0(\mathbf{x}_i). \tag{18}$$

This shows that the error for sample $\mathbf{x}_i$ decreases exponentially with a rate proportional to its $k(x_i, x_i)$. Thus, samples with higher $k(x_i, x_i)$ converge faster during training.

3 Empirical Evaluation

We conducted experiments to validate the theoretical findings using standard deep learning datasets. Specifically, we utilized a finite-width NTK [5][2][6] CNN trained on binary and multi-class classification and probed learnability scores (i.e, the diagonal values on the NTK matrix; for the multi-class case, we took the mean of the values across classes). Our findings on both MNIST and CIFAR-10 datasets were equivalent. We also looked at how samples with high and low learnability behave during the training process, for which the finite-width approximation was necessary.

We divided the dataset into distinct groups based on their $k(x, x)$ values to analyze learning dynamics:

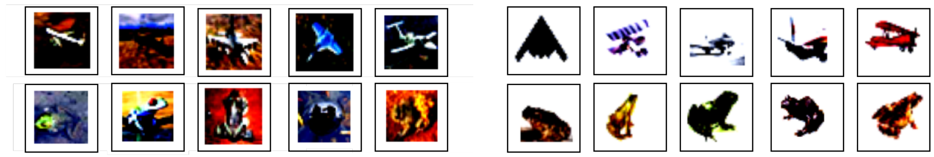

Figure 2: ***Left*:** Images sampled from CIFAR-10 with *low* learnability. ***Right:*** Images sampled from CIFAR-10 with *high* learnability.Sampling on the basis of a 3-layer CNN trained on binary classification to distinguish between frogs and planes.

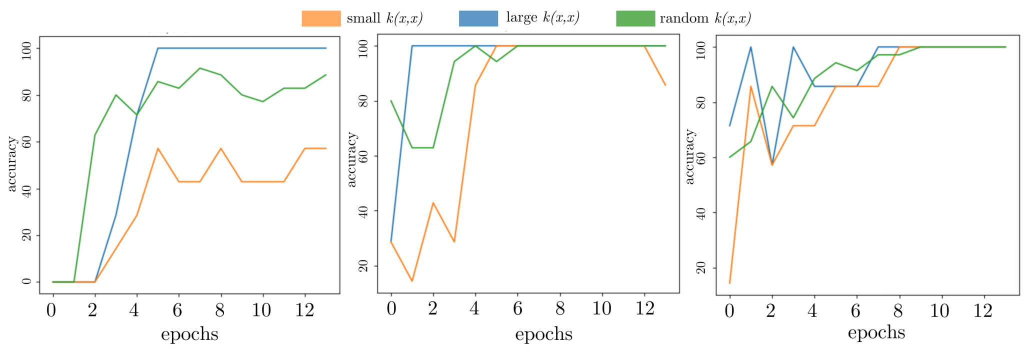

Figure 3: From *left* to *right*, accuracy vs. epochs for **ship | frog**, **horse | plane**, **bird | plane**.

- **High** $k(x, x)$ **Samples**: Top 10% of samples with the highest $k(x, x)$ values.
- **Low** $k(x, x)$ **Samples**: Bottom 10% of samples with the lowest $k(x, x)$ values.
- **Medium** $k(x, x)$ **Samples**: Remaining 80% of samples.

We tracked the training accuracy of each sample group over epochs. Figure 2 illustrates that the high $k(x, x)$ group achieves near-perfect accuracy within fewer epochs compared to the low $k(x, x)$ group. The medium group exhibits intermediate behavior, often *sheathed* by the easy and hard examples.

Visual inspection of samples from different groups reveals distinct characteristics. Figure 3 shows that high $k(x, x)$ samples are clear and prototypical, whereas low $k(x, x)$ samples often contain noise, distortions, or are atypical representations of their classes.

4 Related Works and Discussion

The Neural Tangent Kernel (NTK), introduced by Jacot et al. (2018) [1], has become a fundamental tool for analyzing the training dynamics of overparameterized neural networks, offering insights into how models behave in the infinite-width limit. Early works focused on the connection between NTK and the generalization properties of deep networks, including Arora et al. (2019) [2], who demonstrated that the NTK matrix governs learning dynamics in function space, significantly influencing how networks fit data over time. Several studies, such as those by Novak et al. (2019) [5] and Yang et al. (2020) [7], have shown how NTK can be used to analyze the convergence behavior of neural networks across different architectures, linking specific NTK properties to model generalization and performance. More recent efforts by Du et al. (2018) [3] have investigated how NTK can predict training outcomes in different learning environments, especially for classification tasks, by leveraging its kernel structure to estimate the sample complexity. Robust learning techniques, such as Jacobian regularization explored by Hoffman et al. (2019) [8], have also utilized NTK concepts. Finally, Ilyas et al. (2019) [9] demonstrated how NTK theory can help understand adversarial examples, showing that adversarial attacks exploit features that NTK-based networks consider salient. Our work builds on these foundational studies, focusing on the empirical evaluation and theoretical analysis of individual samples and establishing a direct relationship between NTK diagonal values and the learnability of training data. This novel perspective provides new insights into optimizing training strategies by identifying "hard" and "easy" samples based on NTK properties, contributing to the broader understanding of neural network training dynamics.

References

[1] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

[3] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[4] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

[5] Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A Alemi, Jascha Sohl-Dickstein, and Samuel S Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. *arXiv preprint arXiv:1912.02803*, 2019.

[6] Roman Novak, Jascha Sohl-Dickstein, and Samuel S Schoenholz. Fast finite width neural tangent kernel. In *International Conference on Machine Learning*, pages 17018–17044. PMLR, 2022.

[7] Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[8] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 5(6):7, 2019.

[9] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.