Neural Tangent Kernel

Convergence and Generalization in Neural Networks

Presentation by Lukas Häuser

Neural Tangent Kernel

Neural Tangent Kernel:
Convergence and Generalization in Neural Networks

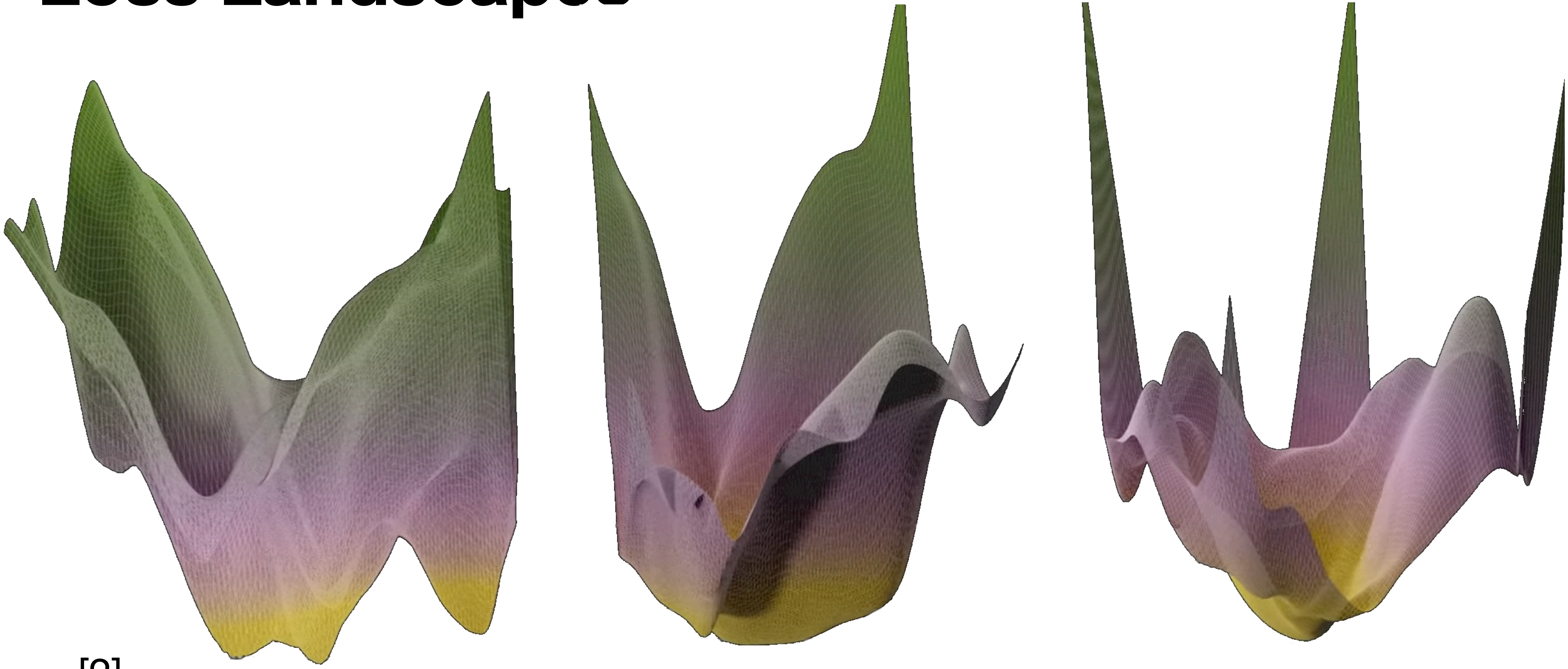
Arthur Jacot
École Polytechnique Fédérale de Lausanne
arthur.jacot@netopera.net

Franck Gabriel
Imperial College London and École Polytechnique Fédérale de Lausanne
franckrgabriel@gmail.com

Clément Hongler
École Polytechnique Fédérale de Lausanne
clement.hongler@gmail.com

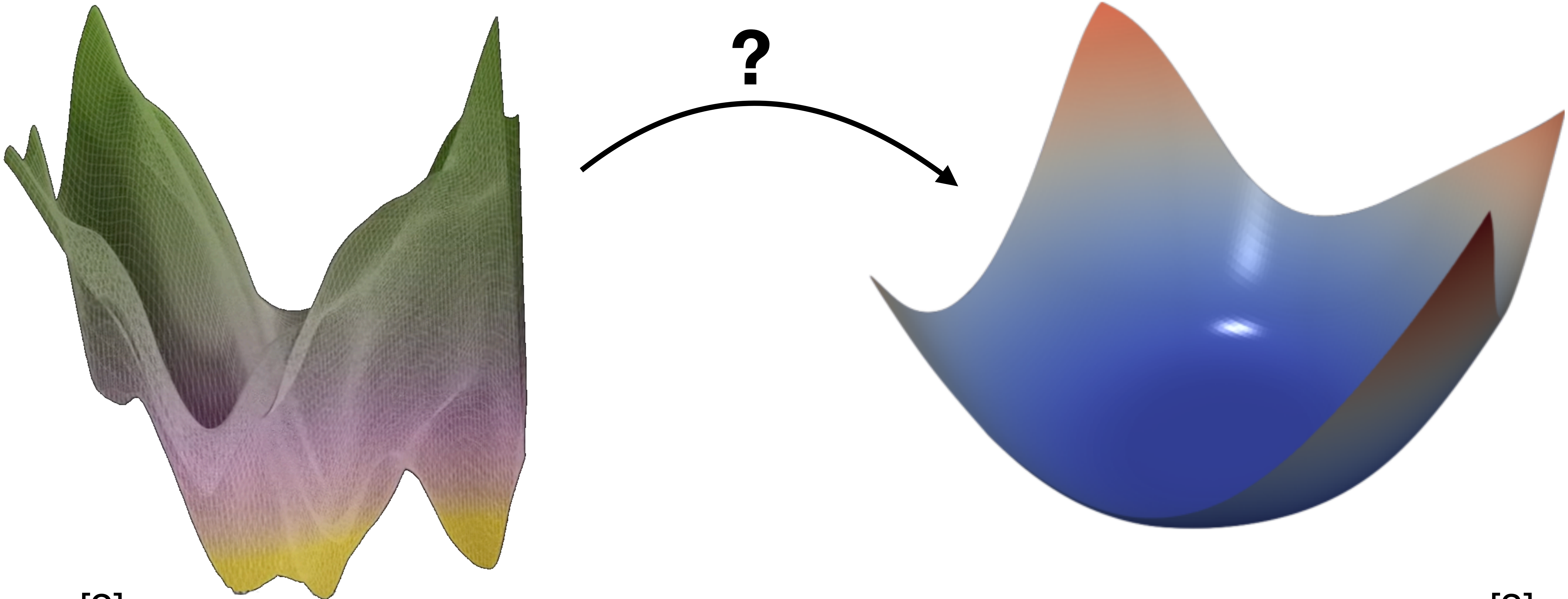
[1]

Loss Landscapes



[2]

Loss Landscapes

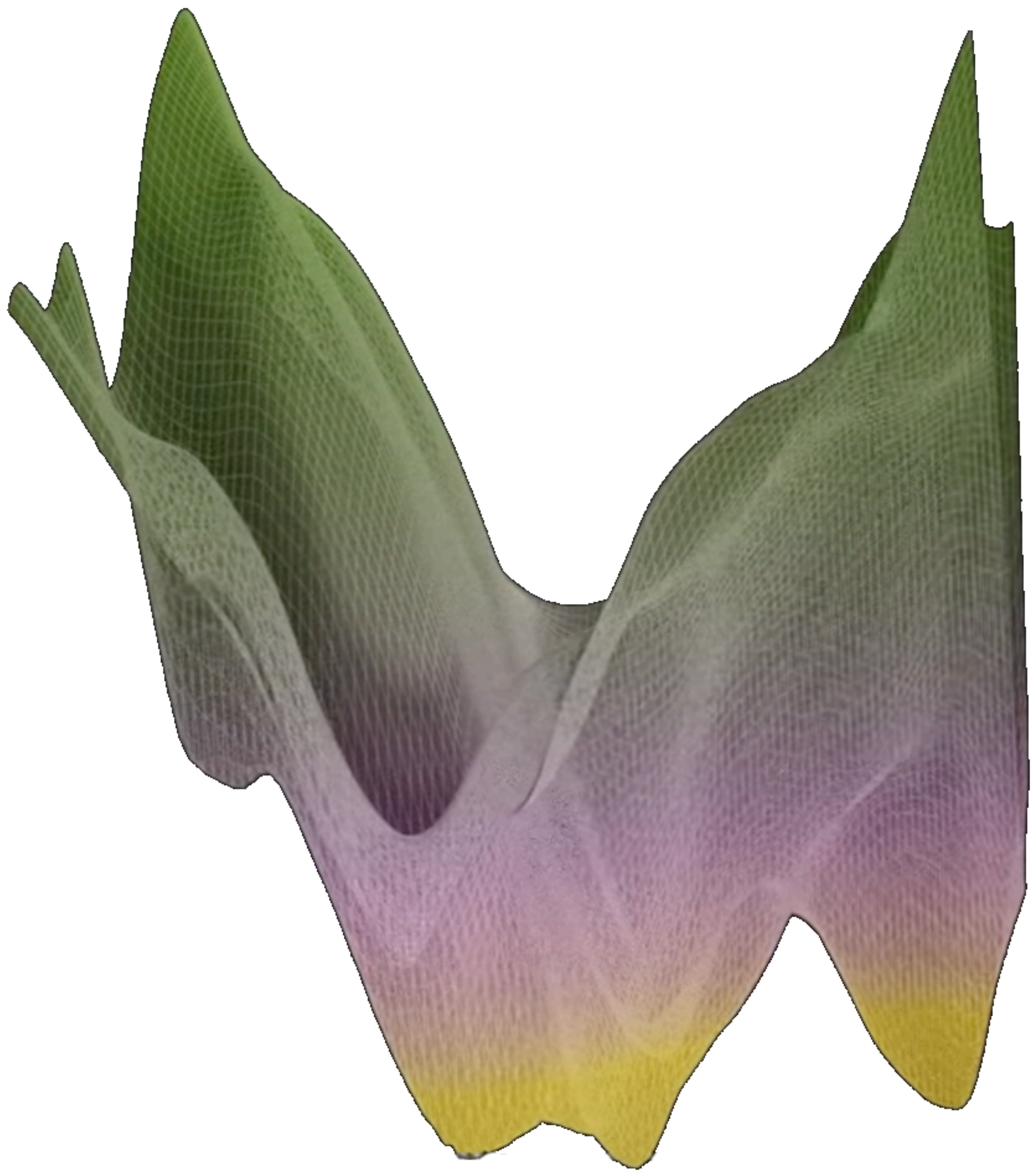


[2] [3]

Problem Statement

- Minimization of convex loss function L
- In parameter space, we minimize $L \circ f = L(f(x;\theta))$
- Instead, minimization of convex loss function L in function space

$\Longrightarrow$ Neural Tangent Kernel



[2]

Background: Neural Networks

- Network function: $f_\theta : \mathbb{R}^{n_0} \times \mathbb{R}^P \to \mathbb{R}^{n_L} \quad \text{with} \quad f_\theta(x;\theta) = \tilde{\alpha}^{(L)}(x;\theta)$
- Activation functions: $\alpha^{(l)} : \mathbb{R}^{n_0} \times \mathbb{R}^P \to \mathbb{R}^{n_l}$

$$\alpha^{(0)}(x;\theta) = x$$

$$\tilde{\alpha}^{(l+1)}(x;\theta) = \frac{1}{\sqrt{n_L}} W^{(l)} \tilde{\alpha}^{(l)}(x;\theta) + \beta b^{(l)}$$

$$\alpha^{(l)}(x;\theta) = \sigma(\tilde{\alpha}^{(l)}(x;\theta))$$

Background: Neural Networks

- Network function: $f_\theta : \mathbb{R}^{n_0} \times \mathbb{R}^P \to \mathbb{R}^{n_L}$ with $f_\theta(x;\theta) = \tilde{\alpha}^{(L)}(x;\theta)$
- Activation functions: $\alpha^{(l)} : \mathbb{R}^{n_0} \times \mathbb{R}^P \to \mathbb{R}^{n_l}$

$$\alpha^{(0)}(x;\theta) = x$$

$$\tilde{\alpha}^{(l+1)}(x;\theta) = \frac{1}{\sqrt{n_L}} W^{(l)} \tilde{\alpha}^{(l)}(x;\theta) + \beta b^{(l)}$$

$$\alpha^{(l)}(x;\theta) = \sigma(\tilde{\alpha}^{(l)}(x;\theta))$$

Background: Kernels

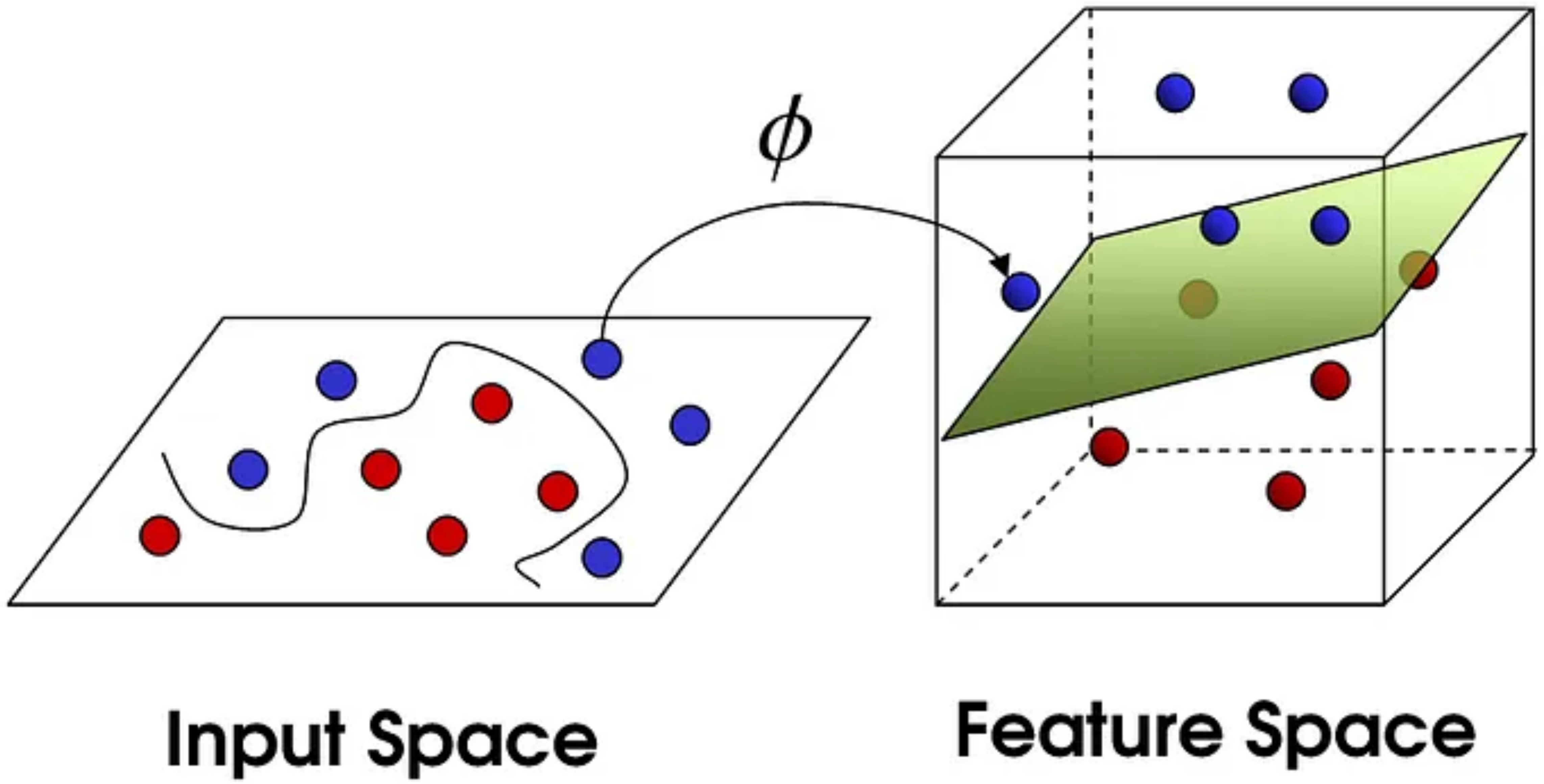
- Kernel: $K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$
- Feature map: $\phi : \mathscr{X} \to \mathscr{V}$

$$K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathscr{V}}$$

- Positive-definite kernel:

$$\langle x, x' \rangle_K = \langle x, K(x, x') x' \rangle$$

$$||x||_K^2 = \langle x, x \rangle_K \geq 0$$



[3]

Main Statement of Paper

- Neural Tangent Kernel for infinitely wide networks:

$$K_\infty(x, x') = \nabla_\theta f(x)^T \nabla_\theta f(x')$$

- Gradient Kernel Descent in infinite width limit:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = -K_\infty \nabla_{f(t)} L \implies \frac{\mathrm{d}L}{\mathrm{d}t} = -\|\nabla_{f(t)} L\|^2_{K_\infty}$$

  - Training dynamics along Neural Tangent Kernel in infinite width limit
  - Guaranteed convergence in asymptotics to global minimum under further conditions

Consequences of Paper

- Framework to understand the training process
- Series of papers [4]:
  - Calculation of Neural Tangent Kernel for various network architecture [5,6,7]
  - Explanation of phenomena during training [8,9,10]

Kernel Gradient Descent

Consider a model $f(x;\theta)$ of input data x and parameters θ with loss function $L(f(x;\theta), y)$ and training data $\{(x_i, y_i)\}$:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\nabla_\theta L$$

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \nabla_\theta f^T \frac{\mathrm{d}\theta}{\mathrm{d}t} = -\nabla_\theta f^T \nabla_\theta L = -\underbrace{\nabla_\theta f^T \nabla_\theta f}_{=K} \nabla_{f(t)} L = -K \nabla_{f(t)} L$$

$$K = \nabla_\theta f^T \nabla_\theta f = \phi(f)^T \phi(f) = \langle \phi(f), \phi(f) \rangle$$

$$\phi(f) = \nabla_\theta f$$

Dynamics of Loss

$$\frac{\mathrm{d}f}{\mathrm{d}t} = -K \nabla_{f(t)} L$$

$$\frac{\mathrm{d}L}{\mathrm{d}t} = \nabla_{f(t)} L \ \frac{\mathrm{d}f}{\mathrm{d}t} = -\ \nabla_{f(t)} L^T \ K \ \nabla_{f(t)} L$$

$$= -\langle \nabla_{f(t)} L, K \nabla_{f(t)} L \rangle = -\langle \nabla_{f(t)} L, \nabla_{f(t)} L \rangle_K$$

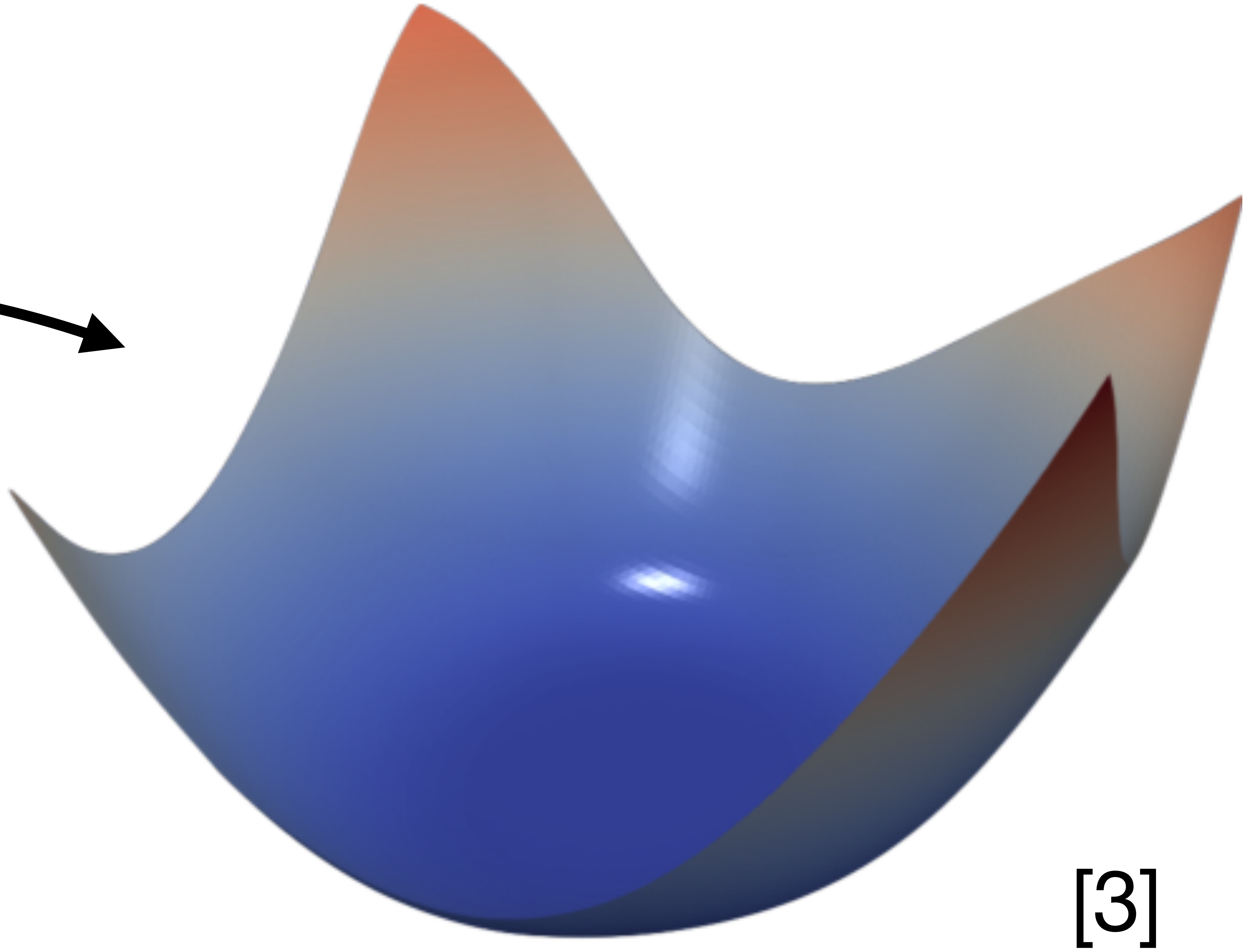
$$= -\| \nabla_{f(t)} L \|_K^2$$

Guaranteed Convergence

$$\frac{\mathrm{d}f}{\mathrm{d}t} = -K\nabla_{f(t)}L$$

$$\frac{\mathrm{d}L}{\mathrm{d}t} = -\|\nabla_{f(t)}L\|_K^2$$

$$K = \nabla_\theta f^T \nabla_\theta f$$



[3]

K constant over training and positive-definite

$\implies$ convergence to global minimum

Simple Linear Example

Random function approximation

- Linear combination f of P random basis functions $(f^{(1)}, \ldots, f^{(P)})$
- Calculation of Gradient Kernel:

$$f(x;\theta) \to \nabla_\theta f \to K(x, x'; \theta) = \nabla_\theta f^T(x;\theta)\,\nabla_\theta f(x';\theta)$$

$$f(x;\theta) = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \theta_p f^{(p)}(x)$$

Simple Linear Example

Random function approximation

- Linear combination f of P random basis functions $(f^{(1)}, \ldots, f^{(P)})$
- Calculation of Gradient Kernel:

$$f(x;\theta) \rightarrow \nabla_\theta f \rightarrow K(x, x'; \theta) = \nabla_\theta f^T(x;\theta)\, \nabla_\theta f(x';\theta)$$

$$\nabla_\theta f(x) = \nabla_\theta \left(\frac{1}{\sqrt{P}} \sum_{p=1}^{P} \theta_p f^{(p)}(x) \right) \underset{\uparrow}{=} \frac{1}{\sqrt{P}} \sum_{p=1}^{P} f^{(p)}(x)\, \mathrm{e}_p = \frac{1}{\sqrt{P}} (f^{(1)}(x), \ldots, f^{(P)}(x))^T$$

$$\nabla_\theta \theta_p = \mathrm{e}_p$$

Simple Linear Example

Random function approximation

- Linear combination f of P random basis functions $(f^{(1)}, \ldots, f^{(P)})$
- Calculation of Gradient Kernel:

$$f(x;\theta) \rightarrow \nabla_\theta f \rightarrow K(x,x';\theta) = \nabla_\theta f^T(x;\theta)\,\nabla_\theta f(x';\theta)$$

$$K(x,x') = \nabla_\theta f^T(x)\,\nabla_\theta f(x')$$

$$\nabla_\theta f(x) = \frac{1}{\sqrt{P}}(f^{(1)}(x), \ldots, f^{(P)}(x))^T$$

Simple Linear Example

Random function approximation

- Linear combination f of P random basis functions $(f^{(1)}, \ldots, f^{(P)})$
- Calculation of Gradient Kernel:

$$f(x;\theta) \rightarrow \nabla_\theta f \rightarrow K(x, x';\theta) = \nabla_\theta f^T(x;\theta)\,\nabla_\theta f(x';\theta)$$

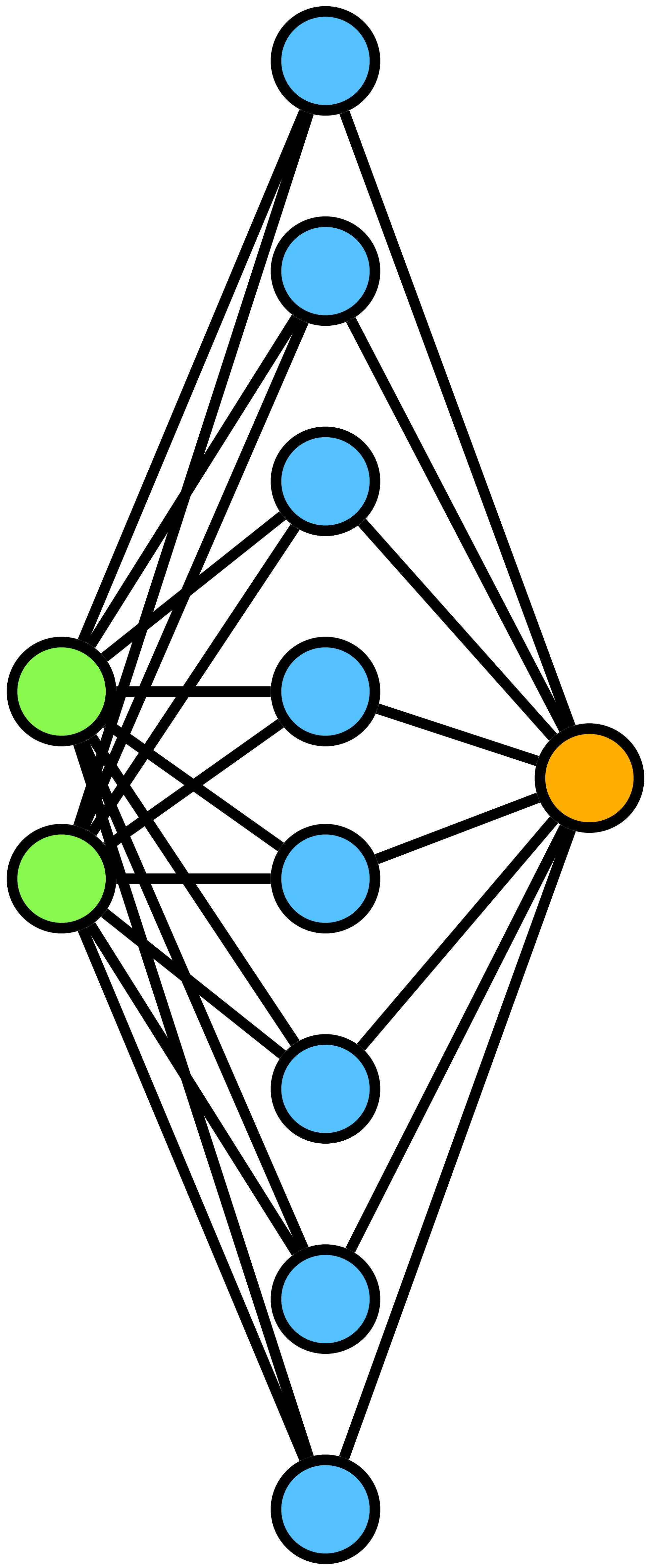
$$K(x, x') = \frac{1}{P}\,(f^{(1)}, \ldots, f^{(P)})\,(f^{(1)}, \ldots, f^{(P)})^T = \frac{1}{P}\begin{pmatrix} f^1(x)f^1(x') & \cdots & f^1(x)f^P(x') \\ \vdots & \ddots & \vdots \\ f^P(x)f^1(x') & \cdots & f^P(x)f^P(x') \end{pmatrix}$$

Neural Tangent Kernel

- Gradient Kernel for neural networks depends on θ:
  - Random at initialization
  - Kernel varies during training
- Linearize $f(x;\theta)$ w.r.t θ [10]:

$$f(\theta) \approx f^{\text{lin}}(\theta) = f(\theta_0) + \nabla_\theta f(\theta_0)(\theta(t) - \theta_0)$$

- In infinite width limit $f(x;\theta)$ becomes linear w.r.t θ [1]



Neural Tangent Kernel

Infinite Width Limit: Initialization

- At initialization with i.i.d. Gaussian distributed parameters θ, with Lipschitz nonlinearitiy σ, and in the infinite width limit as $n_1, \ldots, n_L \to \infty$ the network function f tends to a i.i.d. centered Gaussian process of covariance L:

$$\Sigma^{(1)}(x, x') = \frac{1}{n_0} x^T x' + \beta^2$$

$$\Sigma^{(l+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}(0, \Sigma^{(l)})}[\sigma(f(x))\sigma(f(x'))] + \beta^2$$

- Connection to Gaussian processes [11,12,13,14,15]

Neural Tangent Kernel

Infinite Width Limit: Kernel at Initialization

- Under same conditions, the NTK K converges in probability to a deterministic limiting kernel by the law of large numbers:

$$K^{(L)} \to K_{\infty}^{(L)} \otimes \mathrm{Id}_{n_L}$$

- The scalar kernel $K_{\infty}^{(L)}$ is given by

$$K_{\infty}^{(1)}(x, x') = \Sigma^{(1)}(x, x')$$

$$K_{\infty}^{(L+1)}(x, x') = K_{\infty}^{(l)}(x, x')\dot{\Sigma}^{(l+1)}(x, x') + \Sigma^{(l+1)}(x, x')$$

where $\dot{\Sigma}^{(l+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}\left(0, \Sigma^{(l)}\right)}[\dot{\sigma}(f(x))\dot{\sigma}(f(x'))]$

Neural Tangent Kernel

Infinite Width Limit: Kernel during Training

- For Lipschitz, twice differentiable nonlinearity σ with bound second derivative and infinite width limit $n_1, \ldots, n_L \to \infty$ the NKT K converges uniformly for $t \in [0,T]$:

$$K^{(L)}(t) \to K_\infty^{(L)} \otimes \mathrm{Id}_{n_L}$$

- Also, the dynamics follow the kernel gradient descent:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = -\, K_\infty^{(L)} \ \nabla_{f(t)} L$$

- $K_\infty^{(L)}$ is positive-definite on $\mathbb{S}^{n_0 - 1}$ if network depth $L \geq 2$ and nonlinearity σ is non-polynomial and Lipschitz.

$\Longrightarrow$ Guaranteed convergence to global minimum in asymptotic!

Neural Tangent Kernel

Infinite Width Limit: Choices for Limit

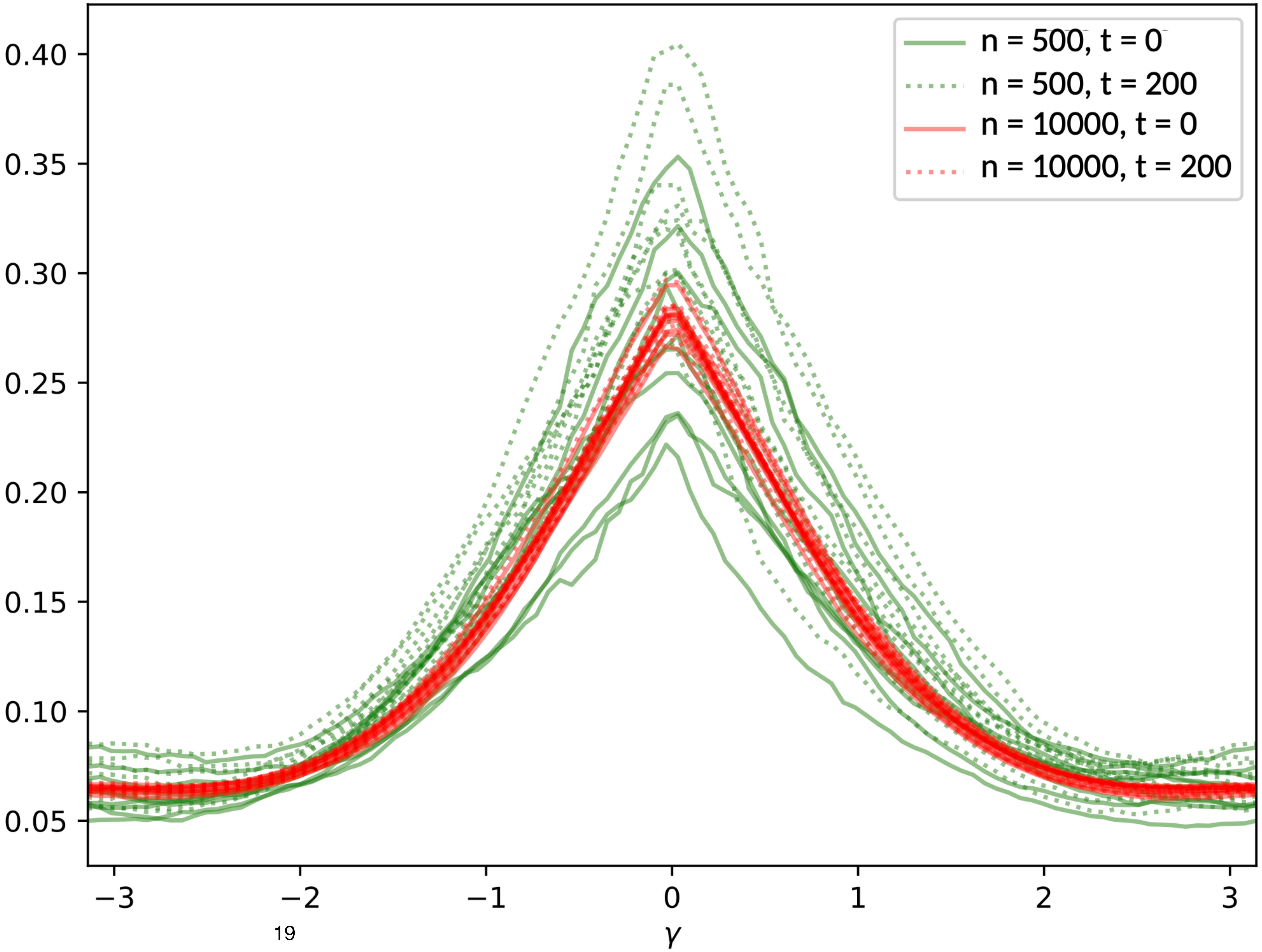
- Made choices for getting the limit $K^{(L)} \to K_\infty^{(L)} \otimes \mathrm{Id}_{n_L}$
  - Initialization: All parameters are initialized as i.i.d Gaussians with mean $\mu = 0$ and variance $\sigma = 1$.
  - Scaling:

$$\tilde{\alpha}^{(l+1)}(x;\theta) = \frac{1}{\sqrt{n_L}} W^{(l)} \tilde{\alpha}^{(l)}(x;\theta) + \beta b^{(l)}$$

- Different initializations and scalings yield different results

Convergence to NTK

- Convergence on unit circle
- $K_{\infty}^{(4)}(x_0, x)$ with $x_0 = (1,0)$
- Less variance for wider network



Least Square Regression

Approximate f^* with least square error with N data points from p^{in}:

$$L = \frac{1}{2}\mathbb{E}_{x\sim p^{\text{in}}}\left[\|f(x) - f^*(x)\|^2\right]$$

$$\frac{\mathrm{d}f}{\mathrm{d}t} = -K\nabla_{f(t)}L$$

$$f(t) = f^* + \mathrm{e}^{-t\Pi}(f_o - f^*), \quad \text{where} \quad \Pi(f)_k(x) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k'=1}^{n_L} f_{k'}(x_i)\, K_{kk'}(x_i, x)$$

$$= f^* + f^{(0)} + \sum_{i=1}^{Nn_L} \mathrm{e}^{-t\lambda_i} f^{(i)}, \quad \text{where} \quad f_0 - f^* = f^{(0)} + f^{(1)} + \ldots + f^{(Nn_L)}$$

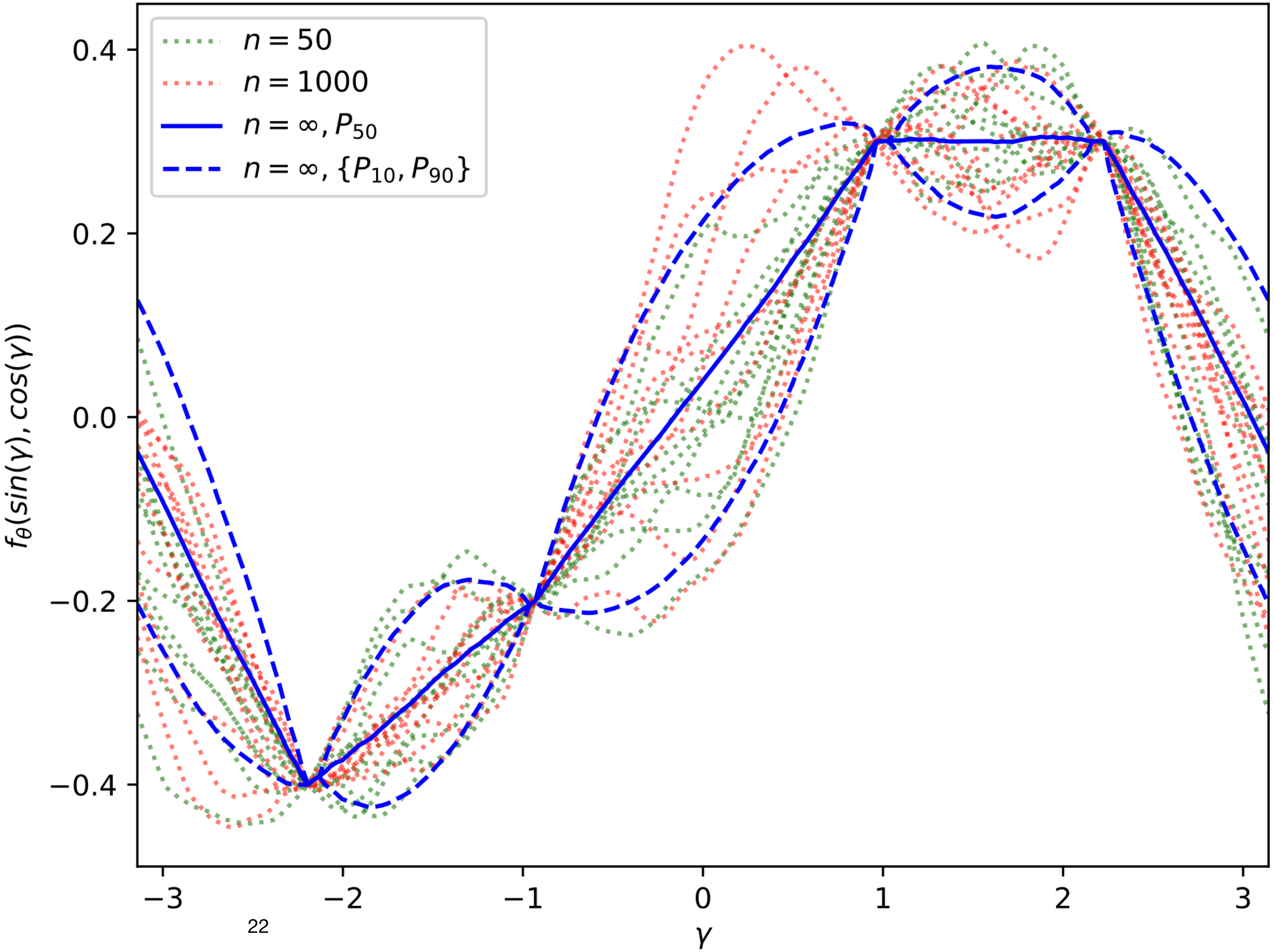
Least Square Regression

$$f(t) = f^* + f^{(0)} + \sum_{i=1}^{Nn_L} e^{-t\lambda_i} f^{(i)} \qquad \Pi(f)_k(x) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k'=1}^{n_L} f_{k'}(x_i)\, K_{kk'}(x_i, x)$$

- Eigenvalues of Π are decay constants λ_i
- Argument for early stopping

Kernel Regression

- Comparison of Gaussian distributions
- Approximation for $K_\infty^{(4)}$ and $\Sigma^{(4)}$
- For wider networks:
    - Mean closer to $K_\infty^{(4)}$
    - Lower variance

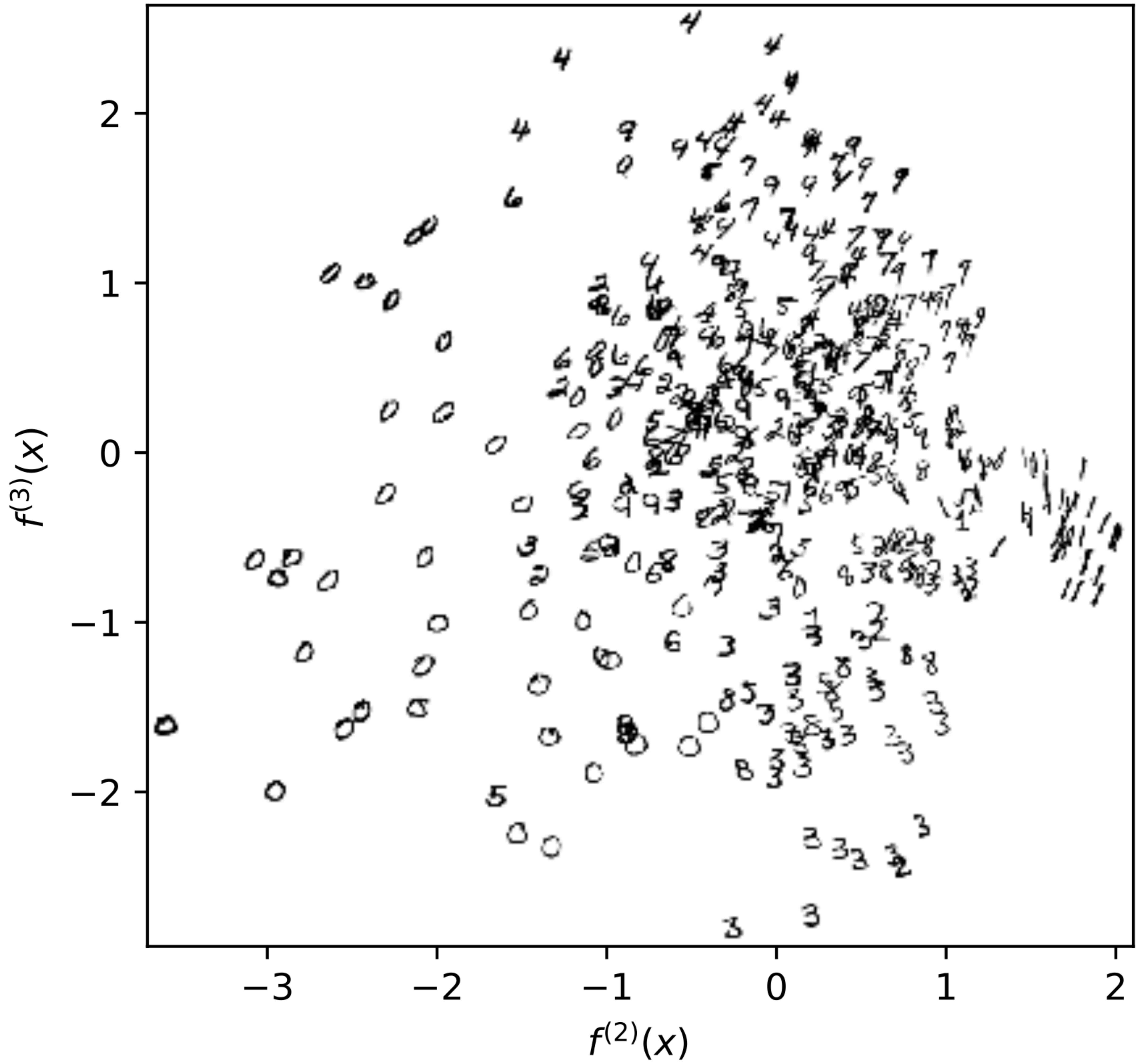


Convergence along principal components

- Decay of principal components:

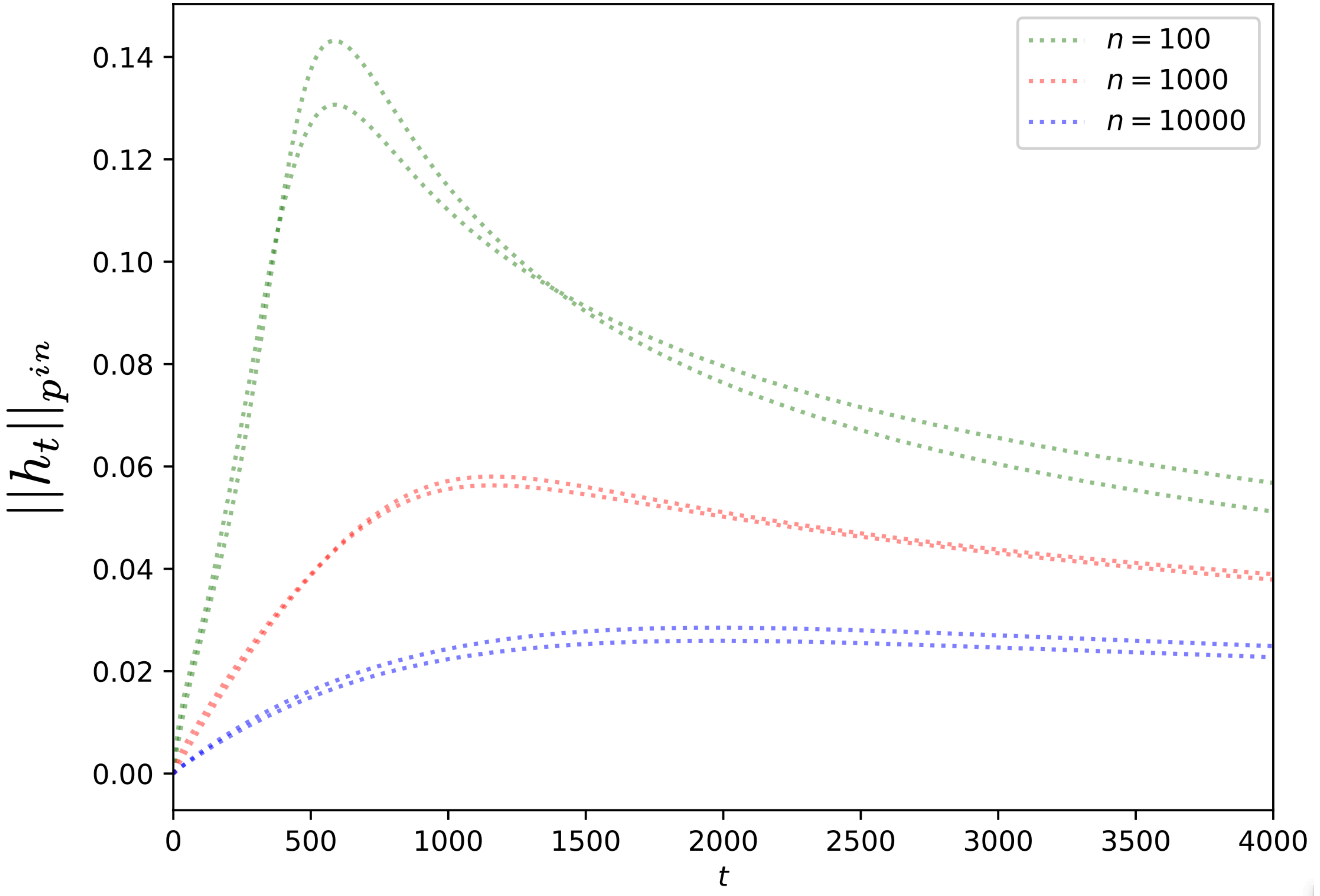
$$f_t = f^* + f^{(0)} + \sum_{i=1}^{Nn_L} \mathrm{e}^{-t\lambda_i} f^{(i)}$$

- Trained on MNIST dataset of handwritten digits



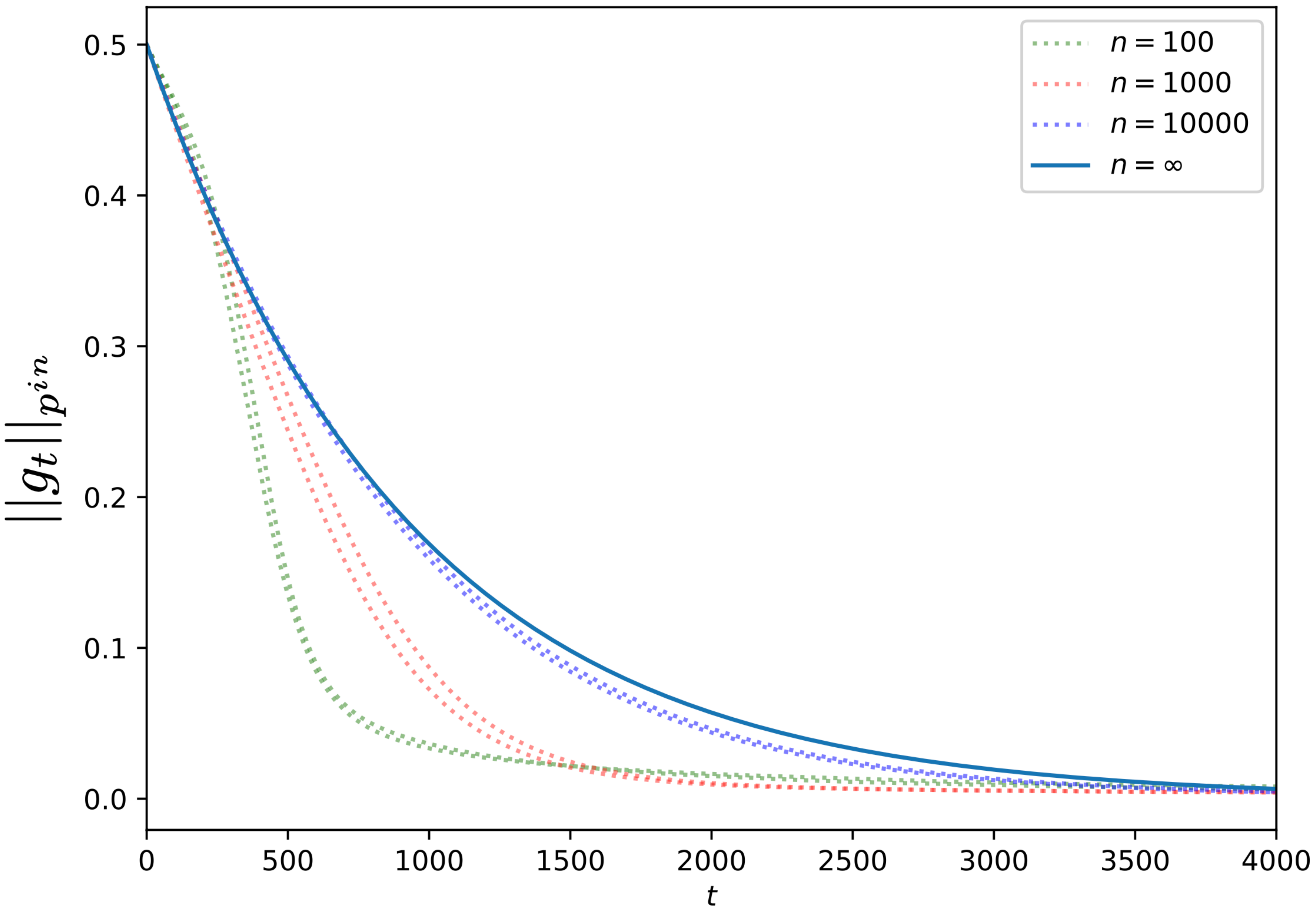
Convergence along principal components

- Deviation from linear dependency on θ
- Wider networks behave more linearly



Convergence along principal components

- Convergence along 2nd principal component
- Wider networks show exponential decay
- Narrower networks converge faster



Conclusion of Paper

- Gradient Kernel determines training dynamics and can guarantee convergence to global minimum
- Neural Tangent Kernel for infinite width networks
- Framework to understand training dynamics

Critique of Paper

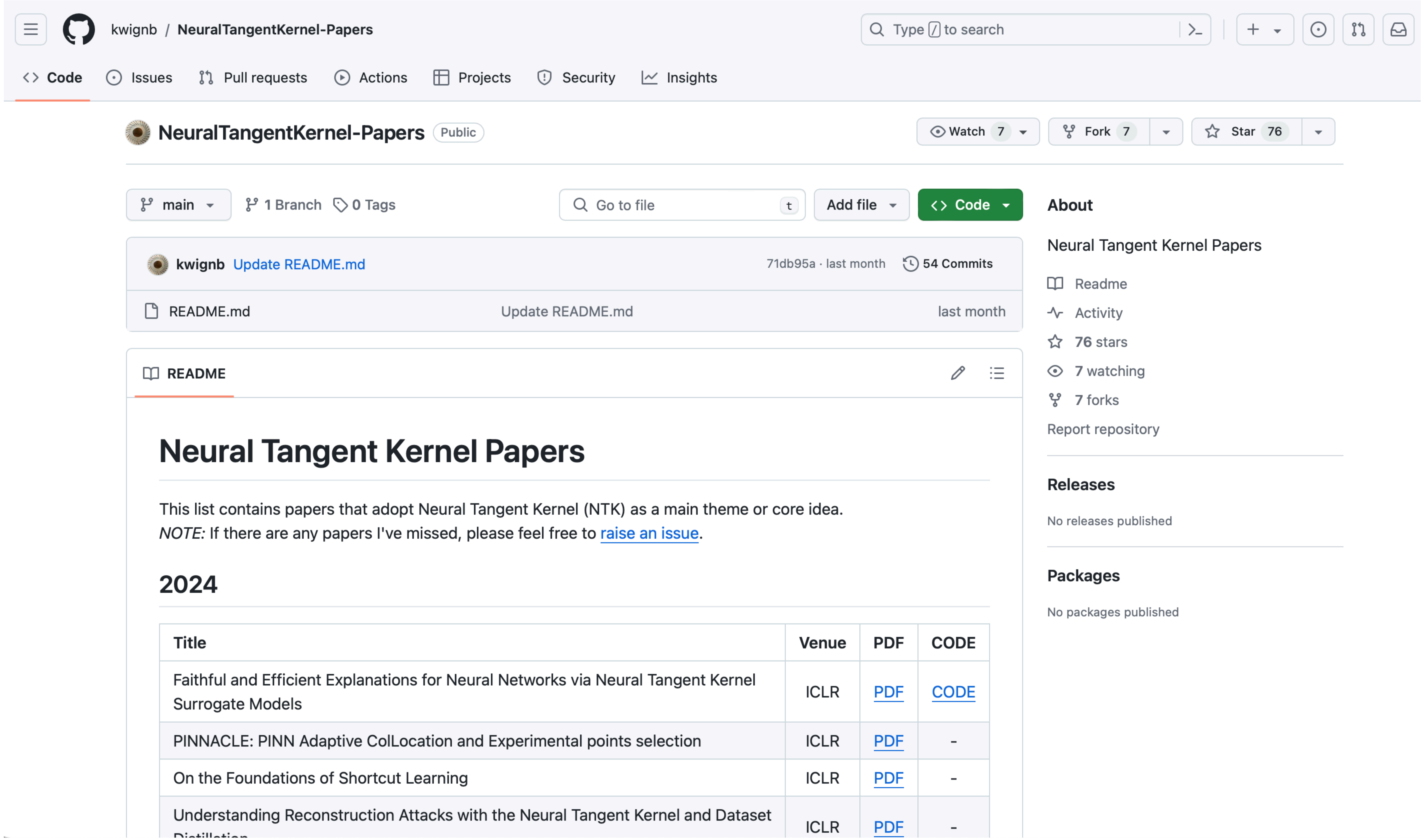
Pro

- Analytical understanding
- General approach
- Effects also empirical

Contra

- For wide networks, but deep are more interesting
- No bounds on width
- Only fully connected feed forward networks

Other Papers



kwignb / NeuralTangentKernel-Papers

Type / to search

Code Issues Pull requests Actions Projects Security Insights

NeuralTangentKernel-Papers Public

Watch 7 Fork 7 Star 76

main 1 Branch 0 Tags Go to file t Add file Code

kwignb Update README.md 71db95a · last month 54 Commits

README.md Update README.md last month

README

Neural Tangent Kernel Papers

This list contains papers that adopt Neural Tangent Kernel (NTK) as a main theme or core idea.
NOTE: If there are any papers I've missed, please feel free to raise an issue.

2024

Title	Venue	PDF	CODE
Faithful and Efficient Explanations for Neural Networks via Neural Tangent Kernel Surrogate Models	ICLR	PDF	CODE
PINNACLE: PINN Adaptive ColLocation and Experimental points selection	ICLR	PDF	-
On the Foundations of Shortcut Learning	ICLR	PDF	-
Understanding Reconstruction Attacks with the Neural Tangent Kernel and Dataset Distillation	ICLR	PDF	-

About

Neural Tangent Kernel Papers

Readme
Activity
76 stars
7 watching
7 forks
Report repository

Releases

No releases published

Packages

No packages published

[4]

Other Papers: NTK for other Architectures

[cs.LG] 15 Jun 2021

3 Nov 2019

Enhanced Convolutional Neural Tangent Kernels*

Zhiyuan Li† Ruosong Wang‡ Dingli Yu§ Simon S. Du¶ Wei Hu‖
Ruslan Salakhutdinov** Sanjeev Arora††

Abstract

Recent research shows that for training with ℓ_2 loss, convolutional neural networks (CNNs) whose width (number of channels in convolutional layers) goes to infinity correspond to regression with respect to the CNN Gaussian Process kernel (CNN-GP) if only the last layer is trained, and correspond to regression with respect to the Convolutional Neural Tangent Kernel (CNTK) if all layers are trained. An exact algorithm to compute CNTK [Arora et al., 2019] yielded the finding that classification accuracy of CNTK on CIFAR-10 is within 6-7% of that of the corre-

[7]

[6]

Other Papers: Explaining convergence

On Lazy Training in Differentiable Programming

Lénaïc Chizat
CNRS, Université Paris-Sud
Orsay, France
lenaic.chizat@u-psud.fr

Edouard Oyallon
CentraleSupelec, INRIA
Gif-sur-Yvette, France
edouard.oyallon@centralesupelec.fr

Francis Bach
INRIA, ENS, PSL Research University
Paris, France
francis.bach@inria.fr

[10] [9]

Summary

- Framework for describing training process via kernel gradient
- For neural networks in infinite width limit: constant Neural Tangent Kernel
- Guaranteed convergence to global minimum in infinite width limit under certain conditions

Sources

[1] Seleznova, Mariia, et al. "Neural (tangent kernel) collapse." *Advances in Neural Information Processing Systems* 36 (2024).

[2] https://losslandscape.com/videos/

[3] Li, Hao, Zheng Xu, Gavin Taylor and Tom Goldstein. “Visualizing the Loss Landscape of Neural Nets.” ArXiv abs/1712.09913 (2017): n. pag.

[4] https://github.com/kwignb/NeuralTangentKernel-Papers

[5] Alemohammad, Sina, et al. "The recurrent neural tangent kernel." *arXiv preprint arXiv:2006.10246* (2020).

[6] Du, Simon S., et al. "Graph neural tangent kernel: Fusing graph neural networks with graph kernels." *Advances in neural information processing systems* 32 (2019).

[7] Li, Zhiyuan, et al. "Enhanced convolutional neural tangent kernels." *arXiv preprint arXiv:1911.00809* (2019).

[8] Wang, Sifan, Xinling Yu, and Paris Perdikaris. "When and why PINNs fail to train: A neural tangent kernel perspective." *Journal of Computational Physics* 449 (2022): 110768.

[9] Seleznova, Mariia, et al. "Neural (tangent kernel) collapse." *Advances in Neural Information Processing Systems* 36 (2024).

[10] Chizat, Lenaic, Edouard Oyallon, and Francis Bach. "On lazy training in differentiable programming." *Advances in neural information processing systems* 32 (2019).

[11] A. Daniely, R. Frostig, and Y. Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 2253–2261. Curran Associates, Inc., 2016.

[12] A. G. de G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In International Conference on Learning Representations, 2018.

[13] A. G. de G. Matthews, J. Hron, R. E. Turner, and Z. Ghahramani. Sample-then-optimize posterior sampling for bayesian linear models. In NIPS workshop on Advances in Approximate Bayesian Inference, 2017.

[14] J. H. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. ICLR, 2018.