High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation

Jimmy Ba[1], **Murat A. Erdogdu**[1], **Taiji Suzuki**[2], **Zhichao Wang**[3], **Denny Wu**[1], **Greg Yang**[4]

[1]University of Toronto and Vector Institute, [2]University of Tokyo and RIKEN AIP,
[3]University of California, San Diego, [4]Microsoft Research AI

{jba,erdogdu,dennywu}@cs.toronto.edu, taiji@mist.i.u-tokyo.ac.jp,
zhw036@ucsd.edu, gregyang@microsoft.com

Abstract

We study the first gradient descent step on the first-layer parameters $\boldsymbol{W}$ in a two-layer neural network: $f(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\boldsymbol{a}^\top\sigma(\boldsymbol{W}^\top\boldsymbol{x})$, where $\boldsymbol{W} \in \mathbb{R}^{d\times N}, \boldsymbol{a} \in \mathbb{R}^N$ are randomly initialized, and the training objective is the empirical MSE loss: $\frac{1}{n}\sum_{i=1}^n(f(\boldsymbol{x}_i) - y_i)^2$. In the proportional asymptotic limit where $n, d, N \to \infty$ at the same rate, and an idealized student-teacher setting where the teacher f^* is a single-index model, we compute the prediction risk of ridge regression on the conjugate kernel after one gradient step on $\boldsymbol{W}$ with learning rate η. We consider two scalings of the first step learning rate η. For small η, we establish a Gaussian equivalence property for the trained feature map, and prove that the learned kernel improves upon the initial random feature model, but cannot defeat the best linear model on the input. Whereas for sufficiently large η, we prove that for certain f^*, the same ridge estimator on trained features can go beyond this "linear regime" and outperform a wide range of (fixed) kernels. Our results demonstrate that even one gradient step can lead to a considerable advantage over random features, and highlight the role of learning rate scaling in the initial phase of training.

1 Introduction

We consider the training of a fully-connected two-layer neural network (NN) with N neurons,

$$f_{\text{NN}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sum_{i=1}^N a_i\sigma(\langle\boldsymbol{x}, \boldsymbol{w}_i\rangle) = \frac{1}{\sqrt{N}}\boldsymbol{a}^\top\sigma(\boldsymbol{W}^\top\boldsymbol{x}), \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{W} \in \mathbb{R}^{d\times N}, \boldsymbol{a} \in \mathbb{R}^N$, σ is the nonlinear activation function applied entry-wise, and the training objective is to minimize the empirical risk. Our analysis will be made in the *proportional asymptotic limit*, i.e., the number of training data n, the input dimensionality d, and the number of neurons N jointly tend to infinity. Intuitively, this regime reflects the setting where the network width and data size are comparable, which is consistent with practical choices of model scaling.

When the first layer $\boldsymbol{W}$ is fixed and the second layer $\boldsymbol{a}$ is optimized, we arrive at a kernel model, where the kernel defined by features $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}^\top\boldsymbol{x})$ (often called the *hidden representation*) is referred to as the *conjugate kernel* (CK) [Nea95]. When $\boldsymbol{W}$ is randomly initialized, this model is an example of the *random features* (RF) model [RR08], the training and test performance of which has been extensively studied in the proportional limit [LLC18, MM22]. These precise characterizations reveal interesting phenomena also present in practical deep learning [BHMM19].

However, RF models do not fully explain the empirical success of NNs: one crucial advantage of deep learning is the *ability to learn useful features* [GDDM14, DCLT18] that "adapt" to the learning problem [Suz18]. In fact, recent works have shown that such adaptivity enables NNs optimized by

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-2749

gradient descent to outperform a wide range of linear/kernel estimators [AZL19, GMMM19]. While many explanations of this separation have been proposed, our starting point is the empirical finding that "non-kernel" behavior often occurs in the *early phase* of NN optimization, especially under large learning rates [JSF$^+$20, FDP$^+$20]. The goal of this work is to answer the following question:

Can we precisely capture the emergence of feature learning in the early phase of gradient descent, and demonstrate its improvement over the initial (fixed) kernel in the proportional limit?

1.1 Contributions

Motivated by the above observations, we investigate a simplified scenario of the "early phase" of learning: how *the first gradient step* on the first-layer parameters $\boldsymbol{W}$ impacts the representation of the two-layer NN (1.1). Specifically, we consider regression with the squared loss (MSE), and a student-teacher setting in the proportional asymptotic limit; we aim to characterize the prediction risk of the kernel ridge regression estimator on top of the first-layer CK feature $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}^\top \boldsymbol{x})$, before and after one gradient descent step on the empirical risk (starting from Gaussian initialization).

Following prior works on the precise asymptotics of RF regression [GLK$^+$20, DL20], we focus on the setting where the input $\boldsymbol{x}$ is Gaussian and the teacher f^* is a single-index model. In this case, the prediction risk of a large class of RF/kernel ridge regression estimators is lower-bounded by the L^2-norm of the "nonlinear" component of the teacher $\|\mathsf{P}_{>1} f^*\|_{L^2}^2$, i.e., they only learn *linear* functions on the input. After one gradient step on $\boldsymbol{W}$, we compute the CK ridge estimator using separate training data, and compare its prediction risk against this linear lower bound. Our analysis will be made under two choices of learning rate scalings:

Figure 1: Prediction risk of ridge regression on trained CK features (erf) after one feature learning step. Markers represent empirical simulations and solid curves are predicted asymptotic values; red line indicates $\Theta(d/n)$ rate.

- **Small lr:** $\eta = \Theta(1)$. In Section 4, we extend the *Gaussian Equivalence Theorem* (GET) in [HL20] to the updated feature map after one gradient descent step on $\boldsymbol{W}$ with learning rate $\eta = \Theta(1)$; this allows us to precisely characterize the prediction risk using random matrix theoretical tools. We prove that after one gradient step, the ridge regression estimator on the learned CK features already exhibits nontrivial improvement over the initial RF ridge regression model (see pink curve in Figure 1), but it remains in the "linear regime" and cannot outperform the best linear estimator on the input (black dashed line).
- **Large lr:** $\eta = \Theta(\sqrt{N})$. In Section 5, we analyze a larger learning rate that coincides with the *maximal update parameterization* in [YH20]. For certain target functions f^*, we prove that kernel ridge regression after one feature learning step can achieve lower risk than the lower bound $\|\mathsf{P}_{>1} f^*\|_{L^2}^2$; thus, it outperforms a wide range of kernel estimators (see purple curve in Figure 1).

1.2 Related works

Asymptotics of kernel regression. Recent works provided precise analysis of RF and kernel models in the proportional limit [GLK$^+$20, DL20, LCM20, AP20, MM22]. These results typically build upon analyses of the spectrum of kernel matrices, a key ingredient in which is the "linearization" of nonlinear random matrices via Taylor expansion [EK10] or orthogonal polynomials [CS13, PW17].

Consequently, a large class of kernel models are essentially linear in the proportional asymptotic limit [LR20, BMR21]. In the case of RF models, a similar property is captured by the Gaussian Equivalence Theorem [GMKZ20, HL20, GLR$^+$21], which roughly states that RF estimators achieve the same prediction risk as a (noisy) linear model. For inputs with unit norm, [GMMM21, MMM21] showed that sample size $n = \Omega(d^2)$ is required to go beyond this "linear" regime. As we will see in certain settings, such a limitation can also be overcome (in the $n \asymp d$ scaling) by training the feature map for one gradient step with a sufficiently large learning rate.

Advantage of NNs over fixed kernels. It is well-known that under a specific initialization, the learning dynamics of overparameterized NNs can be described by the neural tangent kernel (NTK) [JGH18]. However, the NTK description essentially "freezes" the model around its initialization [COB19], and thus does not explain the presence of *feature learning* in NNs [YH20].

In fact, various works have shown that deep learning is more powerful than kernel methods in terms of approximation and estimation ability [Bac17, Suz18, IF19, SH20, GMMM20]. Moreover, in some specialized settings, NNs optimized with gradient-based methods can outperform the NTK (or more generally any kernel estimators) in terms of generalization error [AZL19, WLLM19, GMMM19, LMZ20, DM20, SA20, AZL20, RGKZ21, KWLS21, ABAB$^+$21] (see [MKAS21, Table 2] for a survey). These results often require a careful analysis of the landscape (e.g., properties of global optimum) or optimization dynamics; in contrast, our goal is to precisely characterize the first gradient step and demonstrate a similar separation.

Early phase of NN optimization. Recent empirical studies suggest that properties of the final trained model is strongly influenced by the early stages of optimization [GAS19, LM20, PPVF21], and the NTK evolves most rapidly in the first few epochs [FDP$^+$20]. Large learning rate in the initial steps can impact the conditioning of loss surface [JSF$^+$20, CKL$^+$21] and potentially improve the generalization performance [LWM19, LBD$^+$20]. Under structural assumptions on the data, it has been proved that one gradient step with sufficiently large learning rate can drastically decrease the training loss [CLB21], extract task-relevant features [DM20, FCB22], or escape the trivial stationary point at initialization [HCG21]. While these works also highlight the benefit of one feature learning step, to our knowledge this advantage has not been *precisely* characterized in the proportional regime (where the performance of RF models has been extensively studied).

2 Problem setup and assumptions

Notations. Throughout this paper, $\|\cdot\|$ denotes the ℓ_2-norm for vectors and the $\ell_2 \to \ell_2$ operator norm for matrices, and $\|\cdot\|_F$ is the Frobenius norm. For matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, $\mathrm{tr}(\boldsymbol{M}) = \frac{1}{n}\mathrm{Tr}(\boldsymbol{M})$ is the normalized trace. $\mathcal{O}_d(\cdot)$ and $o_d(\cdot)$ stand for the standard big-O and little-o notations, where the subscript highlights the asymptotic variable; we write $\tilde{\mathcal{O}}(\cdot)$ when the (poly-)logarithmic factors are ignored. $\mathcal{O}_{d,\mathbb{P}}(\cdot)$ (resp. $o_{d,\mathbb{P}}(\cdot)$) represents big-O (resp. little-o) in probability as $d \to \infty$. $\Omega(\cdot), \Theta(\cdot)$ are defined analogously. Γ is the standard Gaussian distribution in $\mathbb{R}^d$. Given $f : \mathbb{R}^d \to \mathbb{R}$, we denote its L^p-norm w.r.t. Γ as $\|f\|_{L^p(\mathbb{R}^d,\Gamma)}$, which we abbreviate as $\|f\|_{L^p}$ when the context is clear.

2.1 Training procedure

Gradient descent on the 1st layer. Given training examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, we learn the two-layer NN (1.1) by minimizing the empirical risk: $\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^n \ell(f(\boldsymbol{x}_i), y_i)$, where ℓ is the squared loss $\ell(x, y) = \frac{1}{2}(x-y)^2$. As previously remarked, fixing the first layer $\boldsymbol{W}$ at random initialization and learning the second layer $\boldsymbol{a}$ yields an RF model, which is a convex problem with closed-form solution. In contrast, we are interested in *learning the feature map (representation)*; hence we first fix $\boldsymbol{a}$ (at initialization) and perform gradient descent on $\boldsymbol{W}$. We write the initialized first-layer as $\boldsymbol{W}_0$, and the weights after one gradient step as $\boldsymbol{W}_1$. The gradient update, which we refer to as the *feature learning step*, with learning rate η is given as: $\boldsymbol{W}_1 = \boldsymbol{W}_0 + \eta\sqrt{N}\cdot\boldsymbol{G}_0$ where

$$\boldsymbol{G}_0 := \frac{1}{n}\boldsymbol{X}^\top\left[\left(\frac{1}{\sqrt{N}}\left(\boldsymbol{y} - \frac{1}{\sqrt{N}}\sigma(\boldsymbol{X}\boldsymbol{W}_0)\boldsymbol{a}\right)\boldsymbol{a}^\top\right)\odot\sigma'(\boldsymbol{X}\boldsymbol{W}_0)\right], \tag{2.1}$$

in which $\odot$ is the Hadamard product, σ' is the derivative of σ (acting entry-wise), and we denoted the input feature matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, and the corresponding label vector $\boldsymbol{y} \in \mathbb{R}^n$. We remark that the $\sqrt{N}$-scaling in front of η accounts for the $\frac{1}{\sqrt{N}}$-prefactor in our definition of two-layer NN (1.1).

Ridge regression for the 2nd layer. After obtaining the updated weights $\boldsymbol{W}_1$, we evaluate the quality of the new CK features by computing the prediction risk of the *kernel ridge regression* estimator on top of the first-layer representation. Note that if ridge regression is performed on the same data $\boldsymbol{X}$, then after one feature learning step, $\boldsymbol{W}_1$ is no longer independent of $\boldsymbol{X}$, which significantly complicates the analysis. To circumvent this difficulty, we estimate the regression coefficients $\hat{\boldsymbol{a}}$ using *a new set of training data* $\{\tilde{\boldsymbol{x}}_i, \tilde{y}_i\}_{i=1}^n$, which for simplicity we assume to have the same size as the original dataset. This can be interpreted as the representation being "pretrained" on separate data before the ridge regression estimator is learned.

Denoting the feature matrix on the fresh training set $\{\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{y}}\}$ as $\boldsymbol{\Phi} := \frac{1}{\sqrt{N}}\sigma(\tilde{\boldsymbol{X}}\boldsymbol{W}_1) \in \mathbb{R}^{n\times N}$, the CK ridge regression estimator can be obtained by solving $\hat{\boldsymbol{a}} = \mathrm{argmin}_{\boldsymbol{a}}\left\{\frac{1}{n}\|\tilde{\boldsymbol{y}} - \boldsymbol{\Phi}\boldsymbol{a}\|^2 + \frac{\lambda}{N}\|\boldsymbol{a}\|^2\right\}$.

2.2 Student-teacher setting and main assumptions

Given a target function (teacher model) f^* and a learned model $\hat{f}$, we evaluate the model performance using the prediction risk: $\mathcal{R}(\hat{f}) = \mathbb{E}_{\boldsymbol{x}}(\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 = \|\hat{f} - f^*\|_{L^2}^2$, where the expectation is taken over the test data from the same training distribution.

We utilize the orthogonal decomposition of the activation function σ. Define the coefficients

$$\mu_0 = \mathbb{E}[\sigma(z)], \quad \mu_1 = \mathbb{E}[z\sigma(z)], \quad \mu_2 = \sqrt{\mathbb{E}[\sigma(z)^2] - \mu_0^2 - \mu_1^2}, \quad \text{where } z \sim \mathcal{N}(0,1). \tag{2.2}$$

This implies $\sigma(z) = \mu_0 + \mu_1 z + \sigma_\perp(z)$, where $\mathbb{E}[\sigma_\perp(z)] = \mathbb{E}[z\sigma_\perp(z)] = 0$, and $\mathbb{E}[\sigma_\perp(z)^2] = \mu_2^2$.

Similarly, for square integrable target function f^*, we have the orthogonal decomposition

$$f^*(\boldsymbol{x}) = \mu_0^* + \mu_1^*\langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle + \mathsf{P}_{>1} f^*(\boldsymbol{x}), \;\; \mu_1^* \boldsymbol{\beta}_* = \mathbb{E}[\boldsymbol{x} f^*(\boldsymbol{x})], \tag{2.3}$$

where $\mathsf{P}_{>1}$ is the projector orthogonal to constant and linear functions in $L^2(\mathbb{R}^d, \Gamma)$, which implies that $\mathbb{E}[\mathsf{P}_{>1} f^*(\boldsymbol{x})] = 0, \mathbb{E}[\boldsymbol{x}\mathsf{P}_{>1} f^*(\boldsymbol{x})] = \boldsymbol{0}$. As $d \to \infty$, quantities defined in (2.3) satisfy $\|\boldsymbol{\beta}_*\| = 1$, $\|\mathsf{P}_{>1} f^*\|_{L^2} \to \mu_2^*$, where $\mu_0^*, \mu_1^*, \mu_2^*$ are bounded constants. Intuitively, μ_0^*, μ_1^*, and μ_2^* can be interpreted as the "magnitude" of the constant, linear, and nonlinear components of f^*, respectively.

Assumption 1.

1. ***Proportional limit.*** *$n, d, N \to \infty$, $n/d \to \psi_1$, $N/d \to \psi_2$, where $\psi_1, \psi_2 \in (0, \infty)$.*
2. ***Gaussian initialization.*** *$\sqrt{d} \cdot [\boldsymbol{W}_0]_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $\sqrt{N} \cdot [\boldsymbol{a}]_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, for $i \in [d], j \in [N]$.*
3. ***Normalized activation.*** *The activation function σ has λ_σ-bounded first three derivatives almost surely. In addition, σ satisfies $\mu_0 = 0$ and $\mu_1, \mu_2 \neq 0$ defined in (2.2).*
4. ***Single-index teacher.*** *Labels are generated as $y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$, where $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \boldsymbol{I})$, and ε_i is i.i.d. sub-Gaussian noise with mean 0 and variance σ_ε^2. The teacher $f^*(\boldsymbol{x}) = \sigma^*(\langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle)$, where $\boldsymbol{\beta}_* \in \mathbb{R}^d$ with $\|\boldsymbol{\beta}_*\| = 1$, and σ^* is Lipschitz with $\mu_0^* = 0$, $\mu_1^* \neq 0$ as defined in (2.3).*

Remark. *We make the following comments on the above assumptions.*

- *Following [HL20], we assume smooth centered activation to simplify the computation; empirical evidence suggests that similar result holds beyond this condition (e.g. [LGC+21]). We also expect the Gaussian input assumption may be replaced by weaker orthogonality conditions as in [FW20].*
- *The single-index setting has been extensively studied in the proportional regime [GLK+20, DL20, HL20]. However, prior works only considered training the coefficients $\boldsymbol{a}$ on top of fixed feature map, and such RF models cannot efficiently learn a single-index f^* in high dimensions [YS19].*

Under Assumption 1, a relatively large sample size corresponds to larger ψ_1, and a relatively large network width corresponds to larger ψ_2. The proportional scaling of n, d, N implies that the model width is not significantly larger than the training set size, in contrast to the polynomial overparameterization often required in NTK analyses [DZPS19], which may be less realistic in practical settings.

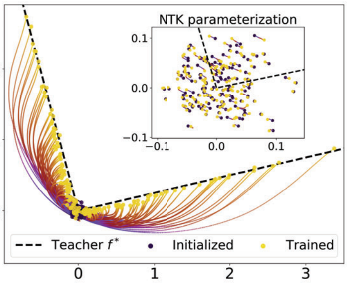


Figure 2: 2D visualization of optimization trajectory under mean-field (main) and NTK (subfigure) parameterizations. f^* consists of two ReLU neurons and the student is a two-layer ReLU neural network. Darker color indicates earlier in training, and vice versa. We set $d = 512$, $\psi_1 = \psi_2 = 10$; both models are optimized until training losses are below 10^{-3}.

Importantly, the initialization of our two-layer NN (1.1) resembles the *mean-field* parameterization [MMN18, CB18]: the second layer is divided by an additional $\sqrt{N}$-factor compared to the kernel (NTK) scaling — this ensures that $f_{\text{NN}}(\boldsymbol{x}) = o_{d,\mathbb{P}}(1)$ at initialization and enables feature learning (see [YH20, Corollary 3.10]). As an illustrative example in Figure 2, we plot the gradient descent trajectory of the first-layer parameters $\boldsymbol{W}$ in two coordinates. Observe that under the mean-field parameterization (main figure), the neurons travel away from the initialization and align with the target function (black dashed lines), whereas in the NTK parameterization (subfigure, which omits the $\frac{1}{\sqrt{N}}$-prefactor), the parameters remain close to their initialization and hence do not learn useful features.

https://doi.org/10.52202/068431-2749

3 Preliminary results

3.1 Lower bound for kernel ridge regression

To illustrate the benefits of *feature learning*, we compare the prediction risk of ridge regression on the trained CK (after one gradient step) against that on the initial RF and fixed kernels. Specifically, given training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, we consider the following classes of kernel models for comparison.

- **Random features model.** We introduce two RF kernels associated with (1.1) at initialization: the conjugate kernel (CK) defined by features $\boldsymbol{\phi}_{\text{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}_0^\top \boldsymbol{x}) \in \mathbb{R}^N$, and the neural tangent kernel (NTK) [JGH18] defined by features $\boldsymbol{\phi}_{\text{NTK}}(\boldsymbol{x}) = \frac{1}{\sqrt{Nd}}\text{Vec}\big(\sigma'(\boldsymbol{W}_0^\top \boldsymbol{x})\boldsymbol{x}^\top\big) \in \mathbb{R}^{Nd}$. Given a feature map RF $\in$ {CK,NTK}, the RF ridge regression estimator can be written as

$$\hat{f}_{\text{RF}}(\boldsymbol{x}) = \langle \boldsymbol{\phi}_{\text{RF}}(\boldsymbol{x}), \hat{\boldsymbol{a}}\rangle, \ \ \hat{\boldsymbol{a}} = \underset{\boldsymbol{a}\in\mathbb{R}^N}{\text{argmin}} \left\{\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{\phi}_{\text{RF}}(\boldsymbol{x}_i), \boldsymbol{a}\rangle)^2 + \frac{\lambda}{N}\|\boldsymbol{a}\|^2\right\}. \tag{3.1}$$

- **Rotation invariant kernel model.** Consider the inner-product kernel: $k(\boldsymbol{x},\boldsymbol{y}) = g\Big(\frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle}{d}\Big)$, and the Euclidean distance kernel: $k(\boldsymbol{x},\boldsymbol{y}) = g\Big(\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{d}\Big)$, where g satisfies the smoothness conditions in [EK10]. Denoting the associated RKHS with $\mathcal{H}$, the kernel ridge estimator is given by

$$\hat{f}_{\text{ker}} = \underset{f\in\mathcal{H}}{\text{argmin}} \left\{\frac{1}{n}\sum_{i=1}^n (y_i - f(\boldsymbol{x}_i))^2 + \lambda\|f\|_{\mathcal{H}}^2\right\} \Rightarrow \hat{f}_{\text{ker}}(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{X})^\top(\boldsymbol{K}+\lambda\boldsymbol{I})^{-1}\boldsymbol{y}. \tag{3.2}$$

We write the prediction risk of the above kernel estimators as $\mathcal{R}_{\text{CK}}(\lambda), \mathcal{R}_{\text{NTK}}(\lambda), \mathcal{R}_{\text{ker}}(\lambda)$, respectively. The following lower bound on the prediction risk is a simple combination of existing results.

Proposition 1 ([HL20, MZ20, BMR21])**.** *Under Assumption 1, we have*

$$\inf_{\lambda>0} \min\{\mathcal{R}_{\text{CK}}(\lambda), \mathcal{R}_{\text{NTK}}(\lambda), \mathcal{R}_{\text{ker}}(\lambda)\} \geq \|\mathsf{P}_{>1}f^*\|_{L^2}^2 + o_{d,\mathbb{P}}(1), \tag{3.3}$$

where $\mathsf{P}_{>1}$ denotes the projector orthogonal to constant and linear functions in $L^2(\mathbb{R}^d, \Gamma)$.

This proposition implies that in the proportional limit, ridge regression on the RF or rotationally invariant kernels defined above does not outperform the best linear estimator on the input – it cannot achieve vanishing risk unless the target function is linear ($\|\mathsf{P}_{>1}f^*\|_{L^2} = 0$). In the following, we compare the prediction risk of the ridge estimator on trained features against this lower bound.

3.2 Almost rank-1 property of the gradient matrix

Before we analyze the prediction risk of the ridge regression estimator on the trained CK, we first need to understand the gradient matrix $\boldsymbol{G}_0$ in (2.1). The following proposition shows that the first gradient step on $\boldsymbol{W}$ can be approximated in operator norm by a rank-1 matrix under Assumption 1.

Proposition 2. *Define $\boldsymbol{G}_0 := \frac{1}{\eta\sqrt{N}}(\boldsymbol{W}_1 - \boldsymbol{W}_0)$ and a rank-1 matrix $\boldsymbol{A} := \frac{\mu_1}{n\sqrt{N}}\boldsymbol{X}^\top\boldsymbol{y}\boldsymbol{a}^\top$. Given Assumption 1, there exist some constants $c, C > 0$ such that for all large n, N, and d, we have*

$$\|\boldsymbol{G}_0 - \boldsymbol{A}\| \leq \frac{C\log^2 n}{\sqrt{n}} \cdot \|\boldsymbol{G}_0\|,$$

with probability at least $1 - ne^{-c\log^2 n}$.

Scaling of learning rate η. Based on the above proposition, we can now specify an appropriate learning rate η such that the change in the first-layer weights after one gradient descent step is neither insignificant nor unreasonably large. Assumption 1 implies that, for proportional n, d, N, the initial weight matrix satisfies $\|\boldsymbol{W}_0\| = \Theta_{d,\mathbb{P}}(1)$, $\|\boldsymbol{W}_0\|_F = \Theta_{d,\mathbb{P}}(\sqrt{d})$, and due to Proposition 2, the first gradient step satisfies $\sqrt{N}\|\boldsymbol{G}_0\| = \Theta_{d,\mathbb{P}}(1)$, $\sqrt{N}\|\boldsymbol{G}_0\|_F = \Theta_{d,\mathbb{P}}(1)$.

In light of the above scaling, if we write $\eta = \Theta(N^\alpha)$, then $\alpha \geq 0$ is required so that the change in the weight matrix is non-negligible (one may verify that for $\eta = o_d(1)$, the test performance of kernel

https://doi.org/10.52202/068431-2749

ridge regression remains unchanged after one GD step). On the other hand, when $\alpha > 1/2$, the gradient update "overwhelms" the initialized parameters $\boldsymbol{W}_0$, and the preactivation feature $\langle \boldsymbol{x}, \boldsymbol{w}_i \rangle$ in the NN (1.1) becomes unbounded as $N \to \infty$. This motivates us to consider the following two regimes of learning rate scaling.

$$\textbf{Small lr}: \eta = \Theta(1) \ \Rightarrow\ \|\boldsymbol{W}_1 - \boldsymbol{W}_0\| \asymp \|\boldsymbol{W}_0\|$$
$$\textbf{Large lr}: \eta = \Theta(\sqrt{N}) \ \Rightarrow\ \|\boldsymbol{W}_1 - \boldsymbol{W}_0\|_F \asymp \|\boldsymbol{W}_0\|_F$$

In Section 4, we consider small step size $\eta = \Theta(1)$, which is parallel to common practice in NN optimization[1]. Whereas in Section 5, we analyze the larger step size $\eta = \Theta(\sqrt{N})$, which resembles the learning rate scaling in the maximal update parameterization in [YH20]; in particular, from Lemma 10 in Appendix B.1, one can easily verify that given data point $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})$, the change in each coordinate of the feature vector is roughly of the same order as its initialized magnitude, that is, for $i \in [N]$, $\left|\sigma(\boldsymbol{W}_1^\top \boldsymbol{x}) - \sigma(\boldsymbol{W}_0^\top \boldsymbol{x})\right|_i \asymp \left|\sigma(\boldsymbol{W}_0^\top \boldsymbol{x})\right|_i = \tilde{\Theta}(1)$ with probability 1 as $N \to \infty$.

4 $\eta = \Theta(1)$: improvement over the initial CK

From Proposition 2, we observe that the dominant rank-1 direction in the first-step gradient matrix $\boldsymbol{G}_0$ contains information of the teacher model f^* (through label vector $\boldsymbol{y}$). Intuitively, this indicates that the learned feature map after one GD step $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$ can "adapt" to f^*, and hence we may expect the ridge regression estimator on the trained CK to achieve better performance. In this section, we precisely characterize the CK prediction risk under the small learning rate $\eta = \Theta(1)$. We first introduce the Gaussian equivalence property which will be useful in the risk computation.

4.1 The Gaussian equivalence property

The Gaussian Equivalence Theorem (GET) states that the performance of a nonlinear kernel model is the same as that of a noisy linear model. Specifically, for the ridge regression estimator, define

$$\mathcal{R}_{\mathrm{F}}(\lambda) = \mathbb{E}_{\boldsymbol{x}}\big(\langle \boldsymbol{\phi}_{\mathrm{F}}(\boldsymbol{x}), \hat{\boldsymbol{a}}_\lambda \rangle - f^*(\boldsymbol{x})\big)^2,$$
$$\hat{\boldsymbol{a}}_\lambda = \mathrm{argmin}_{\boldsymbol{a}} \Big\{ \frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{\phi}_{\mathrm{F}}(\boldsymbol{x}_i), \boldsymbol{a} \rangle)^2 + \frac{\lambda}{N}\|\boldsymbol{a}\|^2 \Big\}, \tag{4.1}$$

where $\mathrm{F} \in \{\mathrm{CK}, \mathrm{GE}\}$ indicates the choice of feature map, which can be either the nonlinear CK feature $\boldsymbol{\phi}_{\mathrm{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}^\top \boldsymbol{x})$, or the linear Gaussian equivalent (GE) feature $\boldsymbol{\phi}_{\mathrm{GE}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\big(\mu_1 \boldsymbol{W}^\top \boldsymbol{x} + \mu_2 \boldsymbol{z}\big)$ where $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ is independent of $\boldsymbol{x}, \boldsymbol{W}$. In the following, for both $\boldsymbol{\phi}_{\mathrm{CK}}$ and $\boldsymbol{\phi}_{\mathrm{GE}}$, we take $\boldsymbol{W}$ to be the updated weight matrix $\boldsymbol{W}_1$ after one GD step.

The Gaussian equivalence refers to the universality phenomenon $\mathcal{R}_{\mathrm{CK}}(\lambda) \approx \mathcal{R}_{\mathrm{GE}}(\lambda)$. For RF models (3.1), the GET has been rigorously proved in [HL20, MS22, MM22]. Furthermore, [GLR+21, LGC+21] provided empirical evidence that such equivalence holds for more general feature maps, including the representation of certain pretrained NNs (e.g., see [LGC+21, Figure 4]). Since our setting goes beyond RF models and cannot be covered by the prior results, we establish the GET for our *trained* feature map under small learning rate.

Theorem 3. *Suppose that Assumption 1 holds and the activation σ is an odd function. If the learning of $\boldsymbol{W}_1$ in (2.1) and estimation of $\hat{\boldsymbol{a}}_\lambda$ in (4.1) are performed on independent training data $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$, respectively, then the GET holds after the first-layer weight is trained for one gradient step with learning rate $\eta = \Theta(1)$; that is, for the CK feature $\boldsymbol{\phi}_{\mathrm{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$, and $\lambda > 0$,*

$$|\mathcal{R}_{\mathrm{CK}}(\lambda) - \mathcal{R}_{\mathrm{GE}}(\lambda)| = o_{d,\mathbb{P}}(1). \tag{4.2}$$

This is to say, for learning rate $\eta = \Theta(1)$, the Gaussian equivalent model provides an accurate description of the prediction risk of CK ridge regression after one feature learning step. The important observation is that even though the trained parameters in $\boldsymbol{W}_1$ are no longer i.i.d., the Gaussian equivalence property can still hold when $\boldsymbol{W}_1 - \boldsymbol{W}_0$ remains "small" (in some norm, see (C.3) in Appendix C.1 for details), which entails that the neurons remain nearly orthogonal to one another.

[1] Heuristically, the updated NN under $\eta = \Theta(1)$ remains close to the "kernel regime" in the sense that each neuron is close to initialization, i.e., as $N \to \infty$, $\left|[\boldsymbol{W}_1 - \boldsymbol{W}_0]_{ij}\right| \ll \left|[\boldsymbol{W}_0]_{ij}\right|$ with high probability.

https://doi.org/10.52202/068431-2749

Implications of Gaussian equivalence. Under the GET, we can alternatively compute $\mathcal{R}_{\rm GE}(\lambda)$, the prediction risk of ridge regression on noisy Gaussian features $\boldsymbol{\phi}_{\rm GE}$, which is much easier to analyze. Theorem 3 is empirically validated in Figure 3(a)(b), where we observe an agreement between the experimental values and the analytic predictions[2] from Section 4.2. On the other hand, the GET also implies that the kernel estimator is essentially "linear" in high dimensions. For the squared loss, it is straightforward to verify that the Gaussian equivalent model cannot learn the nonlinear component of the target function $\mathsf{P}_{>1}f^*$ as follows.

Fact 4. *Under the same assumptions as Theorem 3, $\mathcal{R}_{\rm GE}(\lambda) \geq \|\mathsf{P}_{>1}f^*\|_{L^2}^2$ for any $\psi_1, \psi_2, \lambda > 0$.*

Hence when $\eta = \Theta(1)$, even though training the first-layer $\boldsymbol{W}$ for one step can lead to non-trivial improvement over the initial RF model (which we precisely quantify in Section 4.2), the learned CK cannot outperform the best linear model on the input features. In other words, to (possibly) learn a nonlinear f^*, the trained feature map needs to violate the GET. In the case of one gradient step on $\boldsymbol{W}$, this amounts to using a sufficiently large step size, which we analyze in Section 5.

4.2 Precise asymptotics of CK ridge regression

Having established the Gaussian equivalence property for the CK ridge estimator after one gradient step with $\eta = \Theta(1)$, we can now compute the asymptotic prediction risk for the trained kernel and compare with the initialized RF. To quantify the discrepancy in the prediction risk (4.1), we write $\mathcal{R}_0(\lambda)$ as the prediction risk of the initialized RF ridge regression estimator (on the feature map $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}_0^\top \boldsymbol{x})$), and $\mathcal{R}_1(\lambda)$ as the prediction risk of the ridge estimator on the trained feature map after one feature learning step $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$.

Importantly, because of the dependency between the trained weights $\boldsymbol{W}_1$ and the teacher model f^* (due to the gradient update (2.1)), we cannot simply apply a rotation invariance argument (e.g., [MM22, Lemma 9.2]) to remove the dependency on the true parameters $\boldsymbol{\beta}_*$ and reduce the prediction risk to the trace of certain rational functions of the kernel matrix. In other words, knowing the spectrum (or the Stieltjes transform) of the CK is not sufficient for these purposes. Instead, we utilize the GET and the almost rank-1 property of $\boldsymbol{G}_0$ in Proposition 2, which, in combination with techniques from operator-valued free probability theory [MS17, AP20], enables us to obtain the asymptotic expression of the difference in the prediction risk before and after one gradient step.

Theorem 5. *Under the same assumptions as Theorem 3 and $\eta = \Theta(1)$, we have*

$$\mathcal{R}_0(\lambda) - \mathcal{R}_1(\lambda) \xrightarrow{\mathbb{P}} \delta(\eta, \lambda, \psi_1, \psi_2) \geq 0,$$

where $\delta(\eta, \lambda, \psi_1, \psi_2)$ is defined by (C.19) in Appendix C.3. Here, δ is a non-negative function of $\eta, \lambda, \psi_1, \psi_2 \in (0, +\infty)$ with parameters μ_1^*, μ_1, μ_2, and it vanishes if and only if (at least) one of μ_1^*, μ_1 and η is equal to zero.

Remark. *Performance of the initial RF ridge estimator $\mathcal{R}_0(\lambda)$ has been characterized by the prior works [GLK+20, MM22]; hence, the precise asymptotics of δ provided in Theorem 5 allows us to explicitly compute the asymptotic prediction risk of the CK model after one gradient step, i.e. $\mathcal{R}_1(\lambda)$.*

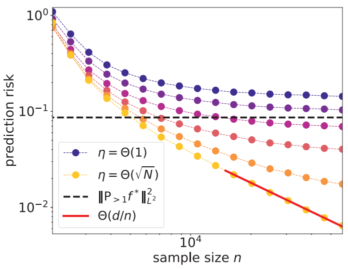
Theorem 5 confirms our intuition that training the first-layer parameters improves the CK model, as shown in Figure 3(a)(b). Remarkably, this improvement (when $\delta > 0$) holds for any $\psi_1, \psi_2 \in (0, \infty)$, that is, taking one gradient step (with learning rate $\eta = \Theta(1)$) is *always* beneficial, even when the training set size n is small. Moreover, we do not require the student and teacher models to have the same nonlinearity — a non-vanishing decrease in the prediction risk is present as long as $\mu_1, \mu_1^* \neq 0$. On the other hand, the GET also implies an upper bound on the possible improvement: $\delta \leq \mathcal{R}_0(\lambda) - \mu_2^{*2}$ as $n, d, N \to \infty$; this is to say, the trained CK remains in the "linear" regime.

Fore the details of Theorem 5, see Appendix C.3.2. Additionally, from inspecting the asymptotic risk formulae (C.19), we can arrive at the following characterization of two special cases of interest.

- **Large sample regime ($\psi_1 \to \infty$):** δ is increasing with respect to the learning rate η; that is, taking a larger step results in greater decrease in the prediction risk, as shown in Figure 3(a).
- **Large width regime ($\psi_2 \to \infty$):** In this case $\delta \to 0$; thus, the benefit of one-step feature learning (with $\eta = \Theta(1)$) becomes less significant as the width increases, as shown in Figure 3(b).

[2] We note that when the first-layer weights are trained for more gradient steps, the GET (Theorem 3) will likely fail eventually, as empirically demonstrated in Appendix A.1.

https://doi.org/10.52202/068431-2749



(a) $\eta = \Theta(1)$ (risk vs. sample size). (b) $\eta = \Theta(1)$ (risk vs. width). (c) $\eta = N^\alpha$ (risk vs. sample size).

Figure 3: Prediction risk of CK ridge regression on trained features: dots represent empirical simulations ($d = 512$, averaged over 50 runs) and solid curves are asymptotic predictions; dashed black line corresponds to the kernel lower bound (3.3). (a) $\eta = \Theta(1)$, $\sigma = \tanh, \sigma^* = \text{SoftPlus}$; we set $\psi_2 = 2$, $\lambda = 10^{-4}$, $\sigma_\varepsilon = 0.25$. (b) $\eta = \Theta(1)$, $\sigma = \tanh, \sigma^* = \text{ReLU}$; we set $\psi_1 = 5$, $\lambda = 10^{-2}$, $\sigma_\varepsilon = 0.1$. (c) $\eta = N^\alpha$ for $\alpha \in [0, 1/2]$; brighter color represents larger step size. We choose $\sigma = \sigma^* = \text{erf}$, $\psi_2 = 2$, $\lambda = 10^{-3}$, and $\sigma_\varepsilon = 0.1$.

5 $\eta = \Theta(\sqrt{N})$: improvement over the kernel lower bound

In this section, we consider a gradient step with large learning rate $\eta = \Theta(\sqrt{N})$, which matches the asymptotic order of the Frobenius norm of the gradient $\boldsymbol{G}_0$ and that of the initialized weight matrix $\boldsymbol{W}_0$. Note that after absorbing the prefactors, this learning rate scaling is analogous to the maximal update parameterization [YH20], which admits a feature learning limit. More specifically, the change in each coordinate of the feature vector $[\sigma(\boldsymbol{W}^\top \boldsymbol{x})]_i$ is $\tilde{\Theta}_{d,\mathbb{P}}(1)$, which has roughly the same order of magnitude as its value at initialization.

Due to the large step size, columns of the updated weight matrix $\boldsymbol{W}_1$ are no longer near-orthogonal, which is an important property in existing analyses of the Gaussian equivalence (e.g., see Proposition 13 in Appendix C.1 or [HL20, Equation (66)]). Indeed, we will see that in this regime, the ridge regression estimator on the trained CK features is no longer "linear" and can potentially outperform the kernel lower bound (3.3) in the proportional limit. However, in the absence of GET, it is difficult to derive the precise asymptotics of the CK model. As an alternative, we establish an *upper bound* on the prediction risk $\mathcal{R}_1(\lambda)$, which we then compare against the kernel ridge lower bound.

Existence of a "good" solution. Given the trained first-layer weights $\boldsymbol{W}_1$, we first construct a second-layer $\tilde{\boldsymbol{a}}$ for which the prediction risk can be upper-bounded. For a pair of nonlinearities (σ, σ^*), we introduce a scalar τ^* which is the optimum of the following minimization problem:

$$\tau^* := \inf_{\kappa \in \mathbb{R}} \mathbb{E}_{\xi_1}\left[\left(\sigma^*(\xi_1) - \mathbb{E}_{\xi_2}\sigma(\kappa\xi_1 + \xi_2)\right)^2\right], \tag{5.1}$$

where $\xi_1, \xi_2 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We write κ^* as an optimal value at which τ^* is attained (when τ^* is not achieved by a finite κ, the same argument holds by introducing a small tolerance factor $\epsilon > 0$ in τ^*; see Appendix D.2). Roughly speaking, τ^* approximates the prediction risk of a specific student model which takes the form of an average over a *subset* of neurons (after one feature learning step). In particular, the first term on the RHS of (5.1) containing σ^* corresponds to the teacher f^*, and the second term $\mathbb{E}_{\xi_2}$ represents the constructed student model. The following lemma shows that we can find some $\tilde{\boldsymbol{a}}$ on the trained CK features whose prediction risk is approximately τ^*, under the additional assumption that the activation function σ is bounded. For more details, see Appendix D.

Lemma 6 (Informal)**.** *Suppose that Assumption 1 holds and σ is bounded. Then, after one gradient step on $\boldsymbol{W}$ with $\eta = \Theta(\sqrt{N})$, there exist some second-layer coefficients $\tilde{\boldsymbol{a}}$ such that the constructed student model $\tilde{f}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\tilde{\boldsymbol{a}}^\top \sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$ achieves a prediction risk which is "close" to τ^*.*

It is worth noting that the definition of τ^* does not involve the specific value of the learning rate η. This is because for any choice of $\eta = \Theta(\sqrt{N})$, due to the Gaussian initialization of a_i, we can find a subset of weights that receive a "good" learning rate (with high probability) such that the corresponding neurons are useful for learning the teacher model. In addition, observe that τ^* is a simple Gaussian integral which can be numerically or analytically computed (see Appendix D.2 for more examples). For instance, when $\sigma = \sigma^* = \text{erf}$, one can easily verify that $\kappa^* = \sqrt{3}$ and $\tau^* = 0$.

https://doi.org/10.52202/068431-2749

Prediction risk of ridge regression. Since we have established the existence of a "good" student model $\tilde{f}$ that can achieve a prediction risk close to τ^* (as defined in (5.1)), in what follows, we prove an upper bound for the prediction risk of the ridge regression estimator on the trained CK features $\mathcal{R}_1(\lambda)$ in terms of the scalar τ^*. The proof of the following result is shown in Appendix D.3.

Theorem 7. *Under the same assumptions as Lemma 6, after one gradient step on $\boldsymbol{W}$ with $\eta = \Theta(\sqrt{N})$, there exist constants $C, \psi_1^* > 0$ such that for any $n/d > \psi_1^*$, the ridge regression estimator (4.1) with regularization parameter $n^{\varepsilon-1} < N^{-1}\lambda < n^{-\varepsilon}$ for some small $\varepsilon > 0$ satisfies*

$$\mathcal{R}_1(\lambda) \le 10\tau^* + C\left(\sqrt{\tau^*} \cdot \sqrt{\tfrac{d}{n}} + \tfrac{d}{n}\right),$$

with probability 1 as $n, d, N \to \infty$ proportionally.

While Theorem 7 does not provide exact expression of the prediction risk, the upper bound still allows us to compare the prediction risk of the CK ridge regression before and after one large gradient step. In particular, if $\|\mathsf{P}_{>1} f^*\|_{L^2}^2 \ge 10\tau^*$ (the constant 10 is not optimized), we know that the trained CK can outperform the kernel lower bound (3.3) (and also the initialized CK) in the proportional limit, when the ratio $\psi_1 = n/d$ is sufficiently large. The following corollary provides two examples of this separation (see Figure 3(c)).

Corollary 8. *Under the same conditions as Theorem 7, there exists a constant ψ_1^* such that for any $\psi_1 > \psi_1^*$, the following holds with probability 1 when $n, d, N \to \infty$ proportionally:*
• For $\sigma = \sigma^* = \mathrm{erf}$, we have $\mathcal{R}_1(\lambda) = \mathcal{O}(d/n)$. • For $\sigma = \sigma^* = \tanh$, we have $\mathcal{R}_1(\lambda) < \|\mathsf{P}_{>1} f^*\|_{L^2}^2$.

In the two examples outlined above, training the features by taking one large gradient step on the first-layer parameters can lead to substantial improvement in the performance of the CK model. In fact, the new ridge regression estimator may outperform a wide range of kernel models as described in Section 3.1, and as shown in Figure 3(c). However, we emphasize that this separation is only present in specific pairs of (σ, σ^*) for which the scalar τ^* is sufficiently small. In general settings, learning a good representation would likely require a training procedure that takes more than one gradient step (even if f^* is as simple as a single-index model, see Figure 4(c) in Appendix A.1).

6 Conclusion

We investigated how the conjugate kernel of a two-layer neural network (1.1) benefits from feature learning in an idealized student-teacher setting, where the first-layer parameters $\boldsymbol{W}$ are updated by one gradient descent step on the empirical risk. Based on the approximate low-rank property of the gradient matrix, we quantified the improvement in the prediction risk of conjugate kernel ridge regression under two different scalings of first-step learning rate η. To the best of our knowledge, this is the first work that rigorously characterizes the precise asymptotics of kernel models (defined by neural networks) in the presence of feature learning.

We outline a few limitations of our current analysis as well as future directions.

- **Dependence between $\boldsymbol{W}_1$ and $\boldsymbol{X}$.** One crucial assumption that we make is that the trained weight matrix $\boldsymbol{W}_1$ is *independent* of the data $\tilde{\boldsymbol{X}}$ on which the CK is computed. While this does not cover the important scenario where feature learning and kernel evaluation are performed on the same data, our setting is very natural in the analysis of pretrained models or transfer learning, which would be an interesting extension.
- **Scaling of learning rate.** Our findings illustrate that different learning rate scalings such as $\eta = \Theta(1)$ and $\eta = \Theta(\sqrt{N})$ result in drastically different behavior. One natural question to ask is whether there exists a "phase transition" in between the two regimes that dictates whether the GET holds. Interestingly, [RGKZ21] showed that instead of breaking the near-orthogonality of the weights $\boldsymbol{W}$ (via large gradient step), one can also introduce sufficiently large low-rank shifts to the input $\boldsymbol{X}$ to enable the initial RF model to fit a nonlinear f^*. Intuitively, this may be due to the "dual" relation of the inputs $\boldsymbol{X}$ and the weights $\boldsymbol{W}$ in the CK model.

https://doi.org/10.52202/068431-2749

Acknowledgement

The authors would like to thank (in alphabetical order) Konstantin Donhauser, Zhou Fan, Hong Hu, Masaaki Imaizumi, Ryo Karakida, Bruno Loureiro, Yue M. Lu, Atsushi Nitanda, Sejun Park, Ji Xu, Yiqiao Zhong for discussions and feedback on the manuscript.

JB was supported by NSERC Grant [2020-06904], CIFAR AI Chairs program, Google Research Scholar Program and Amazon Research Award. MAE was supported by NSERC Grant [2019-06167], Connaught New Researcher Award, CIFAR AI Chairs program, and CIFAR AI Catalyst grant. TS was partially supported by JSPS KAKENHI (20H00576) and JST CREST. ZW was supported by NSF Grant DMS-2055340. DW was partially supported by a Borealis AI Fellowship. Part of this work was completed when DW interned at Microsoft Research (hosted by GY).

References

[ABAB$^+$21] Emmanuel Abbe, Enric Boix Adsera, Matthew Brennan, Guy Bresler, and Dheeraj Nagaraj, *The staircase property: How hierarchical structure can guide deep learning*, Advances in Neural Information Processing Systems **34** (2021).

[ABAM22] Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz, *The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks*, arXiv preprint arXiv:2202.08658 (2022).

[Ada15] Radoslaw Adamczak, *A note on the hanson-wright inequality for random vectors with dependencies*, Electronic Communications in Probability **20** (2015), 1–13.

[ADH$^+$19] 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).

[AP20] Ben Adlam and Jeffrey Pennington, *The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization*, International Conference on Machine Learning, PMLR, 2020, pp. 74–84.

[AZL19] Zeyuan Allen-Zhu and Yuanzhi Li, *What can resnet learn efficiently, going beyond kernels?*, Advances in Neural Information Processing Systems **32** (2019).

[AZL20] ______, *Backward feature correction: How deep learning performs deep learning*, arXiv preprint arXiv:2001.04413 (2020).

[AZLL19] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang, *Learning and generalization in overparameterized neural networks, going beyond two layers*, Advances in neural information processing systems **32** (2019).

[Bac17] Francis Bach, *Breaking the curse of dimensionality with convex neural networks*, The Journal of Machine Learning Research **18** (2017), no. 1, 629–681.

[Bac23] ______, *Learning theory from first principles*, MIT Press, 2023.

[BES$^+$22] Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang, *High-dimensional asymptotics of feature learning in the early phase of neural network training*, In Preparation (2022).

[BHMM19] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal, *Reconciling modern machine-learning practice and the classical bias–variance trade-off*, Proceedings of the National Academy of Sciences **116** (2019), no. 32, 15849–15854.

[BL20] Yu Bai and Jason D. Lee, *Beyond linearization: On quadratic and higher-order approximation of wide neural networks*, International Conference on Learning Representations, 2020.

[BM21] Antoine Bodin and Nicolas Macris, *Model, sample, and epoch-wise descents: exact solution of gradient flow in the random feature model*, Advances in Neural Information Processing Systems **34** (2021).

[BMR21] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin, *Deep learning: a statistical viewpoint*, Acta numerica **30** (2021), 87–201.

https://doi.org/10.52202/068431-2749

[BP21] Lucas Benigni and Sandrine Péché, *Eigenvalue distribution of some nonlinear models of random matrices*, Electronic Journal of Probability **26** (2021), 1–37.

[BP22] ______, *Largest eigenvalues of the conjugate kernel of single-layered neural networks*, arXiv preprint arXiv:2201.04753 (2022).

[BS98] Zhi-Dong Bai and Jack W Silverstein, *No eigenvalues outside the support of the limiting spectral distribution of large-dimensional sample covariance matrices*, The Annals of Probability **26** (1998), no. 1, 316–345.

[BS10] Zhidong Bai and Jack W Silverstein, *Spectral analysis of large dimensional random matrices*, vol. 20, Springer, 2010.

[CB18] Lenaic Chizat and Francis Bach, *On the global convergence of gradient descent for over-parameterized models using optimal transport*, Advances in neural information processing systems, 2018, pp. 3036–3046.

[CB20] ______, *Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss*, Conference on Learning Theory, PMLR, 2020, pp. 1305–1338.

[Chi22] Lénaïc Chizat, *Mean-field langevin dynamics: Exponential convergence and annealing*, arXiv preprint arXiv:2202.01009 (2022).

[CKL$^+$21] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar, *Gradient descent on neural networks typically occurs at the edge of stability*, International Conference on Learning Representations, 2021.

[CLB21] Niladri S Chatterji, Philip M Long, and Peter L Bartlett, *When does gradient descent with logistic loss find interpolating two-layer networks?*, Journal of Machine Learning Research **22** (2021), no. 159, 1–48.

[COB19] Lenaic Chizat, Edouard Oyallon, and Francis Bach, *On lazy training in differentiable programming*, Advances in Neural Information Processing Systems **32** (2019).

[CS13] Xiuyuan Cheng and Amit Singer, *The spectrum of random inner-product kernel matrices*, Random Matrices: Theory and Applications **2** (2013), no. 04, 1350010.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805 (2018).

[DGA20] Ethan Dyer and Guy Gur-Ari, *Asymptotics of wide networks from feynman diagrams*, International Conference on Learning Representations, 2020.

[DL20] Oussama Dhifallah and Yue M Lu, *A precise performance analysis of learning with random features*, arXiv preprint arXiv:2008.11904 (2020).

[DM20] Amit Daniely and Eran Malach, *Learning parities with neural networks*, Advances in Neural Information Processing Systems **33** (2020), 20356–20365.

[DV13] Yen Do and Van Vu, *The spectrum of random kernel matrices: universality results for rough and varying kernels*, Random Matrices: Theory and Applications **2** (2013), no. 03, 1350005.

[DW18] Edgar Dobriban and Stefan Wager, *High-dimensional asymptotics of prediction: Ridge regression and classification*, The Annals of Statistics **46** (2018), no. 1, 247–279.

[DWY21] Konstantin Donhauser, Mingqi Wu, and Fanny Yang, *How rotational invariance of common kernels prevents generalization in high dimensions*, International Conference on Machine Learning, PMLR, 2021, pp. 2804–2814.

[DZPS19] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh, *Gradient descent provably optimizes over-parameterized neural networks*, International Conference on Learning Representations, 2019.

[EK10] Noureddine El Karoui, *The spectrum of kernel random matrices*, The Annals of Statistics **38** (2010), no. 1, 1–50.

[EK18] ______, *On the impact of predictor geometry on the performance on high-dimensional ridge-regularized generalized robust regression estimators*, Probability Theory and Related Fields **170** (2018), no. 1, 95–175.

https://doi.org/10.52202/068431-2749

[FCB22] Spencer Frei, Niladri S Chatterji, and Peter L Bartlett, *Random feature amplification: Feature learning and generalization in neural networks*, arXiv preprint arXiv:2202.07626 (2022).

[FDP$^+$20] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli, *Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel*, Advances in Neural Information Processing Systems **33** (2020), 5850–5861.

[FM19] Zhou Fan and Andrea Montanari, *The spectral norm of random inner-product kernel matrices*, Probability Theory and Related Fields **173** (2019), no. 1-2, 27–85.

[FOBS06] Reza Rashidi Far, Tamer Oraby, Wlodzimierz Bryc, and Roland Speicher, *Spectra of large block matrices*, arXiv preprint cs/0610045 (2006).

[FW20] Zhou Fan and Zhichao Wang, *Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks*, Advances in neural information processing systems **33** (2020), 7710–7721.

[GAS19] Aditya Sharad Golatkar, Alessandro Achille, and Stefano Soatto, *Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence*, Advances in Neural Information Processing Systems **32** (2019).

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[GLK$^+$20] Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mézard, and Lenka Zdeborová, *Generalisation error in learning with random features and the hidden manifold model*, International Conference on Machine Learning, PMLR, 2020, pp. 3452–3462.

[GLR$^+$21] Sebastian Goldt, Bruno Loureiro, Galen Reeves, Florent Krzakala, Marc Mézard, and Lenka Zdeborová, *The gaussian equivalence of generative models for learning with shallow neural networks*, Proceedings of Machine Learning Research vol **145** (2021), 1–46.

[GMKZ20] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová, *Modeling the influence of data structure on learning in neural networks: The hidden manifold model*, Physical Review X **10** (2020), no. 4, 041044.

[GMMM19] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari, *Limitations of lazy training of two-layers neural network*, Advances in Neural Information Processing Systems **32** (2019).

[GMMM20] ______, *When do neural networks outperform kernel methods?*, Advances in Neural Information Processing Systems **33** (2020), 14820–14830.

[GMMM21] ______, *Linearized two-layers neural networks in high dimension*, The Annals of Statistics **49** (2021), no. 2, 1029–1054.

[GSJW20] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart, *Disentangling feature and lazy training in deep neural networks*, Journal of Statistical Mechanics: Theory and Experiment **2020** (2020), no. 11, 113301.

[HCG21] Karl Hajjar, Lénaïc Chizat, and Christophe Giraud, *Training integrable parameterizations of deep neural networks in the infinite-width limit*, arXiv preprint arXiv:2110.15596 (2021).

[HFS07] J William Helton, Reza Rashidi Far, and Roland Speicher, *Operator-valued semicircular elements: solving a quadratic matrix equation with positivity constraints*, International Mathematics Research Notices **2007** (2007), no. 9, rnm086–rnm086.

[HL20] Hong Hu and Yue M Lu, *Universality laws for high-dimensional learning with random features*, arXiv preprint arXiv:2009.07669 (2020).

[HMS18] J William Helton, Tobias Mai, and Roland Speicher, *Applications of realizations (aka linearizations) to free probability*, Journal of Functional Analysis **274** (2018), no. 1, 1–79.

https://doi.org/10.52202/068431-2749

[HY20] Jiaoyang Huang and Horng-Tzer Yau, *Dynamics of deep neural networks and neural tangent hierarchy*, International conference on machine learning, PMLR, 2020, pp. 4542–4551.

[IF19] Masaaki Imaizumi and Kenji Fukumizu, *Deep neural networks learn non-smooth functions effectively*, The 22nd international conference on artificial intelligence and statistics, PMLR, 2019, pp. 869–878.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, Advances in neural information processing systems, 2018, pp. 8571–8580.

[JSF$^+$20] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras, *The break-even point on optimization trajectories of deep neural networks*, International Conference on Learning Representations, 2020.

[JT20] Ziwei Ji and Matus Telgarsky, *Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks*, International Conference on Learning Representations, 2020.

[KWLS21] Stefani Karp, Ezra Winston, Yuanzhi Li, and Aarti Singh, *Local signal adaptivity: Provable feature learning in neural networks beyond kernels*, Advances in Neural Information Processing Systems **34** (2021).

[LBD$^+$20] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari, *The large learning rate phase of deep learning: the catapult mechanism*, arXiv preprint arXiv:2003.02218 (2020).

[LCM20] Zhenyu Liao, Romain Couillet, and Michael W Mahoney, *A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent*, Advances in Neural Information Processing Systems **33** (2020), 13939–13950.

[LGC$^+$21] Bruno Loureiro, Cedric Gerbelot, Hugo Cui, Sebastian Goldt, Florent Krzakala, Marc Mezard, and Lenka Zdeborová, *Learning curves of generic features maps for realistic datasets with a teacher-student model*, Advances in Neural Information Processing Systems **34** (2021).

[LLC18] Cosme Louart, Zhenyu Liao, and Romain Couillet, *A random matrix approach to neural networks*, The Annals of Applied Probability **28** (2018), no. 2, 1190–1248.

[LM20] Guillaume Leclerc and Aleksander Madry, *The two regimes of deep network training*, arXiv preprint arXiv:2002.10376 (2020).

[LMZ20] Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang, *Learning over-parametrized two-layer neural networks beyond ntk*, Conference on learning theory, PMLR, 2020, pp. 2613–2682.

[LR20] Tengyuan Liang and Alexander Rakhlin, *Just interpolate: Kernel "ridgeless" regression can generalize*, The Annals of Statistics **48** (2020), no. 3, 1329–1347.

[LWM19] Yuanzhi Li, Colin Wei, and Tengyu Ma, *Towards explaining the regularization effect of initial large learning rate in training neural networks*, Advances in Neural Information Processing Systems, 2019, pp. 11674–11685.

[MKAS21] Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro, *Quantifying the benefit of using differentiable learning over tangent kernels*, International Conference on Machine Learning, PMLR, 2021, pp. 7379–7389.

[MM22] Song Mei and Andrea Montanari, *The generalization error of random features regression: Precise asymptotics and the double descent curve*, Communications on Pure and Applied Mathematics **75** (2022), no. 4, 667–766.

[MMM21] Song Mei, Theodor Misiakiewicz, and Andrea Montanari, *Generalization error of random feature and kernel methods: hypercontractivity and kernel matrix concentration*, Applied and Computational Harmonic Analysis (2021).

[MMN18] Song Mei, Andrea Montanari, and Phan-Minh Nguyen, *A mean field view of the landscape of two-layer neural networks*, Proceedings of the National Academy of Sciences **115** (2018), no. 33, E7665–E7671.

[MS17] James A Mingo and Roland Speicher, *Free probability and random matrices*, vol. 35, Springer, 2017.

[MS22] Andrea Montanari and Basil N Saeed, *Universality of empirical risk minimization*, Conference on Learning Theory, PMLR, 2022, pp. 4310–4312.

[MZ20] Andrea Montanari and Yiqiao Zhong, *The interpolation phase transition in neural networks: Memorization and generalization under lazy training*, arXiv preprint arXiv:2007.12826v1 (2020).

[Nea95] Radford M Neal, *Bayesian learning for neural networks*, vol. 118, Springer Science & Business Media, 1995.

[Ngu21] Phan-Minh Nguyen, *Analysis of feature learning in weight-tied autoencoders via the mean field lens*, arXiv preprint arXiv:2102.08373 (2021).

[NS17] Atsushi Nitanda and Taiji Suzuki, *Stochastic particle gradient descent for infinite ensembles*, arXiv preprint arXiv:1712.05438 (2017).

[NWS22] Atsushi Nitanda, Denny Wu, and Taiji Suzuki, *Convex analysis of the mean field langevin dynamics*, arXiv preprint arXiv:2201.10469 (2022).

[Péc19] S Péché, *A note on the pennington-worah distribution*, Electronic Communications in Probability **24** (2019), 1–7.

[PPVF21] Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion, *Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity*, Advances in Neural Information Processing Systems **34** (2021).

[PW17] Jeffrey Pennington and Pratik Worah, *Nonlinear random matrix theory for deep learning*, Advances in Neural Information Processing Systems, 2017, pp. 2637–2646.

[RGKZ21] Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová, *Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed*, International Conference on Machine Learning, PMLR, 2021, pp. 8936–8947.

[RR08] Ali Rahimi and Benjamin Recht, *Random features for large-scale kernel machines*, Advances in neural information processing systems, 2008, pp. 1177–1184.

[SA20] Taiji Suzuki and Shunta Akiyama, *Benefit of deep learning with non-convex noisy gradient descent: Provable excess risk bound and superiority to kernel methods*, arXiv preprint arXiv:2012.03224 (2020).

[SH20] Johannes Schmidt-Hieber, *Nonparametric regression using deep neural networks with relu activation function*, The Annals of Statistics **48** (2020), no. 4, 1875–1897.

[Suz18] Taiji Suzuki, *Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality*, arXiv preprint arXiv:1810.08033 (2018).

[TAP21] Nilesh Tripuraneni, Ben Adlam, and Jeffrey Pennington, *Covariate shift in high-dimensional random feature regression*, arXiv preprint arXiv:2111.08234 (2021).

[Ver18] Roman Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47, Cambridge university press, 2018.

[VSL$^+$22] Rodrigo Veiga, Ludovic Stephan, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová, *Phase diagram of stochastic gradient descent in high-dimensional two-layer neural networks*, arXiv preprint arXiv:2202.00293 (2022).

[WGL$^+$20] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro, *Kernel and rich regimes in overparametrized models*, Conference on Learning Theory, PMLR, 2020, pp. 3635–3673.

[WLLM19] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma, *Regularization matters: Generalization and optimization of neural nets vs their induced kernel*, Advances in Neural Information Processing Systems, 2019, pp. 9712–9724.

[WX20] Denny Wu and Ji Xu, *On the optimal weighted ℓ_2 regularization in overparameterized linear regression*, Advances in Neural Information Processing Systems **33** (2020), 10112–10123.

https://doi.org/10.52202/068431-2749

[WZ21] Zhichao Wang and Yizhe Zhu, *Deformed semicircle law and concentration of nonlinear random matrices for ultra-wide neural networks*, arXiv preprint arXiv:2109.09304 (2021).

[Yan20] Greg Yang, *Tensor programs iii: Neural matrix laws*, arXiv preprint arXiv:2009.10685 (2020).

[YH20] Greg Yang and Edward J Hu, *Feature learning in infinite-width neural networks*, arXiv preprint arXiv:2011.14522 (2020).

[YS19] Gilad Yehudai and Ohad Shamir, *On the power and limitations of random features for understanding neural networks*, Advances in Neural Information Processing Systems **32** (2019).

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-2749