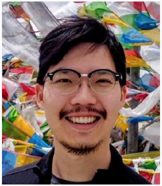
High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation

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

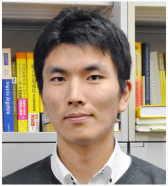
Introduction

- [BES+22] Ba, Erdogdu, Suzuki, Wang, Wu, Yang. "*High-dimensional asymptotics of feature learning: how one gradient step improves the representation*".

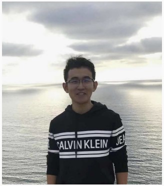


Jimmy Ba

Murat A. Erdogdu



Taiji Suzuki



Zhichao Wang

Denny Wu

Greg Yang

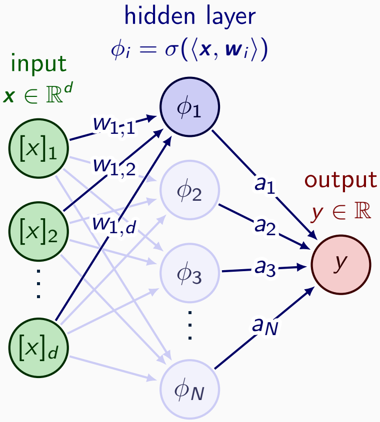
Width-N Two-layer NN

$$f_{\mathrm{NN}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\boldsymbol{x}^\top \boldsymbol{w}_i) = \frac{1}{\sqrt{N}} \boldsymbol{a}^\top \sigma(\boldsymbol{W}^\top \boldsymbol{x}).$$

- Input data: $\boldsymbol{x} \in \mathbb{R}^d$.
- Trainable parameters: $\boldsymbol{W} \in \mathbb{R}^{d \times N}, \boldsymbol{a} \in \mathbb{R}^N$.
- Element-wise nonlinearity: $\sigma : \mathbb{R} \to \mathbb{R}$.

Optimization: given a convex loss ℓ,

- Optimizing $\boldsymbol{a}$ under fixed $\boldsymbol{W}$ is *convex*.
- Optimizing $\boldsymbol{W}$ under fixed $\boldsymbol{a}$ is *non-convex*.



Our Goal: precise characterization of the performance of the trained NN.

Introduction: Training and Test Setting

- **Training.** Empirical risk minimization (potentially ℓ_2-regularized):

$$\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y_i)^2, \quad y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i,$$

  where f^* is the target function (teacher model), and ε is i.i.d. label noise.

- **Test.** Prediction risk: $\mathcal{R}(f) = \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2] = \|f - f^*\|_{L^2(P_x)}^2$.

Regime of Interest – **Proportional asymptotic limit**: $n, d, N \to \infty$, $n/d \to \psi_1$, $N/d \to \psi_2$, where $\psi_1, \psi_2 \in (0, \infty)$.

Why is this an interesting regime to analyze?

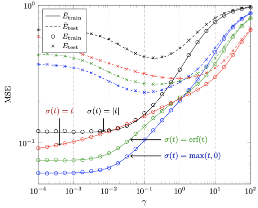
- It corresponds to the setting where the network width and data size are comparable, which is consistent with practical choices of model scaling.
- It might be possible to derive the *precise* prediction risk in this limit.

Kernel Models Related to NN

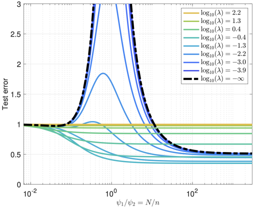
Two widely-studied kernels derived from two-layer NN:

- **Conjugate Kernel (CK)** with features: $\phi_{\mathrm{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}^\top \boldsymbol{x}) \in \mathbb{R}^N$.
  Regression on the CK corresponds to fixing $\boldsymbol{W}$ and only learning the 2nd layer $\boldsymbol{a}$.
- **Tangent Kernel (NTK)** with features: $\phi_{\mathrm{NT}}(\boldsymbol{x}) = \frac{1}{\sqrt{Nd}}\mathrm{Vec}\big(\sigma'(\boldsymbol{W}^\top \boldsymbol{x})\boldsymbol{x}^\top\big) \in \mathbb{R}^{Nd}$.
  This kernel arises from gradient descent on certain wide neural networks.

When $\boldsymbol{W}$ is randomly initialized, we arrive at a **random features (RF)** model, the precise asymptotics of which has been extensively studied in the proportional limit.



[Louart, Liao, and Couillet, 2018].



[Mei and Montanari, 2019].

Limitation of Kernel Ridge Regression

Can these RF models fully capture the effectiveness of NNs? *Not quite...*

Consider the *ridge regression estimator* for RF $\in \{\text{CK}, \text{NT}\}$:

$$f_{\text{RF}}^{\lambda}(\boldsymbol{x}) = \langle \phi_{\text{RF}}(\boldsymbol{x}), \hat{\boldsymbol{a}}_\lambda \rangle, \quad \hat{\boldsymbol{a}}_\lambda = \text{argmin}_{\boldsymbol{a}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \phi_{\text{RF}}(\boldsymbol{x}_i), \boldsymbol{a} \rangle)^2 + \frac{\lambda}{N} \|\boldsymbol{a}\|_2^2 \right\}.$$

Theorem (Ghorbani et al. 19, Hu and Lu 20, Bartlett et al. 21, ...)

[Informal] *Denote $P_{>1}$ as the projector orthogonal to constants and linear functions in $L^2(P_X)$. Then under certain concentration conditions on the input $\boldsymbol{x}$, we have*[1]

$$\inf_{\lambda>0} \min\left\{\mathcal{R}\left(f_{\text{CK}}^{\lambda}\right), \mathcal{R}\left(f_{\text{NT}}^{\lambda}\right)\right\} \geq \|P_{>1} f^*\|_{L^2}^2 + o_{d,\mathbb{P}}(1),$$

- In the proportional limit, RF models can only learn **linear functions**.
- NNs are clearly more powerful than linear models on the input...

[1] Similar lower bound also holds for certain rotationally invariant kernels studied in [El Karoui 10].

Feature Learning in Two-layer NN

Where does this gap come from? **Feature Learning!**

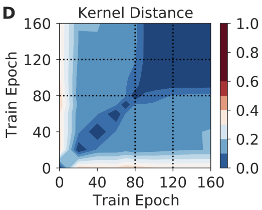
- When we optimize the first-layer parameters $\boldsymbol{W}$, we expect the model to "adapt" to the data and learn useful representations.
- In RF models, $\boldsymbol{W}$ is fixed, so there is no "representation learning".

Motivation: Can we precisely capture the presence of *feature learning* in the proportional limit, when the first-layer $\boldsymbol{W}$ is optimized via *gradient descent*?

Empirical Observation:

- Neural network features often change most rapidly in the **early phase** of gradient descent (GD) training.

We consider the most simplified setting of the "early phase": one gradient step on $\boldsymbol{W}$, and analyze how the learned CK adapts to the learning problem.



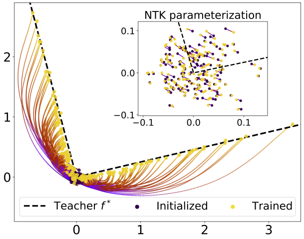
[Fort et al. 2020].

Problem Setting: Basic Assumptions

1. **Proportional Limit.** $n, d, N \to \infty$, $n/d \to \psi_1$, $N/d \to \psi_2$, $\psi_1, \psi_2 \in (0, \infty)$.
2. **Student-teacher Setup.** $y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$, where $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \boldsymbol{I})$, ε_i is i.i.d. noise with variance σ_ε^2, and f^* is Lipschitz with $\|f^*\|_{L^2} = \Theta_d(1)$.
3. **Normalized Activation.** σ has bounded first three derivatives, and is normalized such that $\mathbb{E}[\sigma(z)] = 0$, $\mathbb{E}[z\sigma(z)] = \mu_1 \neq 0$, for $z \sim \mathcal{N}(0, 1)$.
4. **Gaussian Initialization.** $[\boldsymbol{W}_0]_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/d)$, $[\boldsymbol{a}]_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/N)$.

Note: we use the mean-field parameterization[2], which admits a *feature learning limit* (i.e., the weights do not "freeze" around the initialization).

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{x}, \boldsymbol{w}_i \rangle) = \underbrace{\frac{1}{\sqrt{N}} \boldsymbol{a}^\top}_{\approx 1/N} \sigma(\boldsymbol{W}^\top \boldsymbol{x}).$$



NNs trained till $\mathcal{L}(f) < 10^{-3}$.

[2] The NTK scaling corresponds to dropping the $\frac{1}{\sqrt{N}}$-prefactor.

Problem Setting: One-step Gradient Descent

- **One-step GD on 1st Layer**. We take **one gradient step**[3] on the empirical MSE loss $\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y_i)^2$, that is, $\boldsymbol{W}_1 = \boldsymbol{W}_0 + \eta\sqrt{N}\cdot\boldsymbol{G}_0$, where

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

- **Ridge Regression for 2nd Layer**. After learning the features for one step, we perform ridge regression on the trained CK using **a fresh set of data** $\{\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{y}}\}$:

$$\hat{\boldsymbol{a}}_\lambda = \operatorname{argmin}_{\boldsymbol{a}}\left\{\frac{1}{n}\|\tilde{\boldsymbol{y}} - \boldsymbol{\Phi}\boldsymbol{a}\|^2 + \frac{\lambda}{N}\|\boldsymbol{a}\|^2\right\}, \quad \boldsymbol{\Phi} := \frac{1}{\sqrt{N}}\sigma(\tilde{\boldsymbol{X}}\boldsymbol{W}_1) \in \mathbb{R}^{n\times N}.$$

Denote $f_{\mathrm{GD}}^{\lambda}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\hat{\boldsymbol{a}}_\lambda^\top\sigma\left({\boldsymbol{W}_1}^\top\boldsymbol{x}\right)$, and prediction risk: $\mathcal{R}_{\mathrm{GD}}(\lambda) = \mathcal{R}(f_{\mathrm{GD}}^{\lambda})$.

This Work: We aim to compute $\mathcal{R}_{\mathrm{GD}}(\lambda)$, and show its *improvement* over the initialized RF, and potentially over the *lower bound* $\|\mathrm{P}_{>1}f^*\|_{L^2}^2$.

Challenge: cannot directly use *random matrix theory*, as $\boldsymbol{W}_1$ is no longer "random".

[3] Some of our results also apply to multiple gradient steps on $\boldsymbol{W}$.

Properties of the Gradient Matrix G_0

Can we exploit certain structure of the first GD step to simplify the calculation?

Orthogonal Decomposition of σ:

$\sigma(z) = \mu_1 z + \sigma_\perp(z)$, where $\mu_1 = \mathbb{E}[\sigma'(z)] \quad \Rightarrow \quad \mathbb{E}[\sigma_\perp(z)] = \mathbb{E}[z\sigma_\perp(z)] = 0.$

Proposition (BES+22)

Recall $\boldsymbol{G}_0 = \frac{1}{\eta\sqrt{N}}(\boldsymbol{W}_1 - \boldsymbol{W}_0)$. *Define rank-1 matrix* $\boldsymbol{A} := \frac{\mu_1}{n\sqrt{N}} \boldsymbol{X}^\top \boldsymbol{y} \boldsymbol{a}^\top$. *Then*

$$\sqrt{N} \cdot \|\boldsymbol{G}_0 - \boldsymbol{A}\| \lesssim \|\boldsymbol{G}_0\|, \text{ w.h.p.}$$

Intuition: Many commonly-used activations are monotone, so σ' is not centered:

$$n\sqrt{N} \cdot \boldsymbol{G}_0 = \mu_1 \boldsymbol{X}^\top (\boldsymbol{y} - f_0(\boldsymbol{X}))\boldsymbol{a}^\top + \boldsymbol{X}^\top \left((\boldsymbol{y} - f_0(\boldsymbol{X}))\boldsymbol{a}^\top \odot \sigma'_\perp(\boldsymbol{X}\boldsymbol{W}_0)\right)$$

Hence $\boldsymbol{G}_0$ contains: $\quad \|\boldsymbol{A}\|_F \asymp \|\boldsymbol{B}\|_F$, but $\|\boldsymbol{A}\| \gg \|\boldsymbol{B}\|$

- A rank-1 "spike" $\boldsymbol{A}$
- A "residual" with smaller operator norm (but not Frobenius norm) $\boldsymbol{B}$

Selection of Learning Rate η

Based on the decomposition of $\boldsymbol{G}_0$, we focus on the following choices[4] of η:

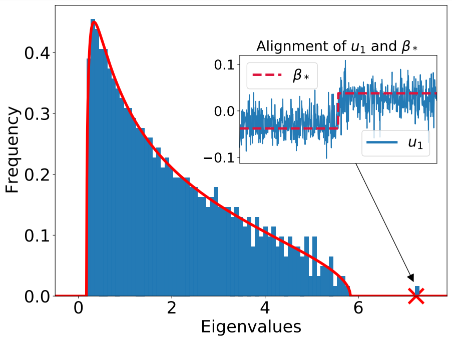
- Small lr: $\eta = \Theta(1) \;\Rightarrow\; \|\boldsymbol{W}_1 - \boldsymbol{W}_0\| \asymp \|\boldsymbol{W}_0\|$.
- Large lr: $\eta = \Theta(\sqrt{N}) \;\Rightarrow\; \|\boldsymbol{W}_1 - \boldsymbol{W}_0\|_F \asymp \|\boldsymbol{W}_0\|_F$.

Remarks:

- Under $\eta = \Theta(1)$, the NN after one GD step remains close to the **kernel regime**: each neuron (or parameter) does not travel far away from the initialization, i.e., $\left|[\boldsymbol{W}_1 - \boldsymbol{W}_0]_{ij}\right| \ll \left|[\boldsymbol{W}_0]_{ij}\right|$ for all i, j with high probability.
- $\eta = \Theta(\sqrt{N})$ mirrors the **maximal update parameterization** [Yang and Hu 2020]: for $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})$, the change in each coordinate of the feature vector is significant, i.e., $\left|\sigma(\boldsymbol{W}_1^\top \boldsymbol{x}) - \sigma(\boldsymbol{W}_0^\top \boldsymbol{x})\right|_i \asymp \left|\sigma(\boldsymbol{W}_0^\top \boldsymbol{x})\right|_i = \tilde{\Theta}(1)$ for all i with high probability.

[4] For smaller $\eta = o(1)$, one can easily verify that change in the prediction risk is negligible.

A Spiked Model for W_1



Blue: empirical simulation.
Red: analytic prediction.
(next slide)

- $\sigma = \tanh$, $f^*(\boldsymbol{x}) = \mathrm{ReLU}(\langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle)$.
- Teacher vector $\boldsymbol{\beta}_* \propto [-1_{d/2}; 1_{d/2}]$.
- $\psi_1 = n/d = 4$, $\psi_2 = N/d = 2$.
- $\eta = 2$.

Observation: after one gradient step with learning rate $\eta = \Theta(1)$:

- The **bulk** of the spectrum of $\boldsymbol{W}$ remains unchanged [5].
- A **spike** (×) appears in $\boldsymbol{W}_1$, which aligns with linear component of f^*.

[5] The spectrum of the initialized $\boldsymbol{W}_0$ is characterized by the Marchenko–Pastur law.

Spiked Model for W_1 (continued)

Orthogonal Decomposition: $f^*(\boldsymbol{x}) = \mu_0^* + \mu_1^* \langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle + \mathrm{P}_{>1} f^*(\boldsymbol{x})$,

- *Linear part:* $\|\boldsymbol{\beta}_*\| = 1$, $\mu_1^* \boldsymbol{\beta}_* = \mathbb{E}[\boldsymbol{x} f^*(\boldsymbol{x})]$;
- *Noninear part:* $\|\mathrm{P}_{>1} f^*\|_{L^2} = \mu_2^*$.

Theorem (BES+22)

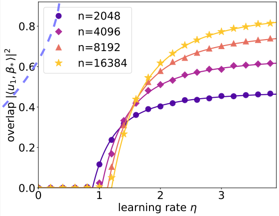
For $\eta = \Theta(1)$, define $\theta_1 := \sqrt{\|f^*\|_{L^2}^2 \psi_1^{-1} + \mu_1^{*2}} \cdot \mu_1 \eta$, $\theta_2 := \mu_1 \mu_1^* \eta$. The leading singular value $s_1(\boldsymbol{W}_1)$ and the corresponding singular vector $\boldsymbol{u}_1$ satisfy

$$s_1(\boldsymbol{W}_1) \to \sqrt{\frac{(1+\theta_1^2)(\psi_2+\theta_1^2)}{\theta_1^2}}, \qquad |\langle \boldsymbol{u}_1, \boldsymbol{\beta}_* \rangle|^2 \to \frac{\theta_2^2}{\theta_1^2}\left(1 - \frac{\psi_2+\theta_1^2}{\theta_1^2(\theta_1^2+1)}\right),$$

for $\theta_1 > \psi_2^{1/4}$; otherwise, $s_1(\boldsymbol{W}_1) \to 1 + \sqrt{\psi_2}$, $|\langle \boldsymbol{u}_1, \boldsymbol{\beta}_* \rangle| \to 0$.

When η exceeds some threshold, a "spike" appears:

- Increase step size η $\Rightarrow$ larger spike $s_1(\boldsymbol{W}_1)$.
- Increase sample size ψ_1 $\Rightarrow$ greater alignment.



A Spiked Model for CK?

Question: How does the spike in $\boldsymbol{W}_1$ affect the *kernel (CK) matrix*?

For $\eta = \Theta(1)$, and *odd activation* σ, the expected CK matrix $\boldsymbol{\Sigma}_\Phi$ satisfies

$$\left\|\boldsymbol{\Sigma}_\Phi - \overline{\boldsymbol{\Sigma}}_\Phi\right\| \xrightarrow{\mathbb{P}} 0, \text{ where } \boldsymbol{\Sigma}_\Phi = \mathbb{E}_{\boldsymbol{x}}\left[\sigma(\boldsymbol{W}_1^\top \boldsymbol{x})\sigma(\boldsymbol{x}^\top \boldsymbol{W}_1)\right], \ \overline{\boldsymbol{\Sigma}}_\Phi = \mu_1^2 \boldsymbol{W}_1^\top \boldsymbol{W}_1 + \mu_2^2 \boldsymbol{I}.$$

- Intuitively, we expect a spike to appear in the (empirical) CK matrix.
- How do we predict properties of the CK spike?

Gaussian Equivalence

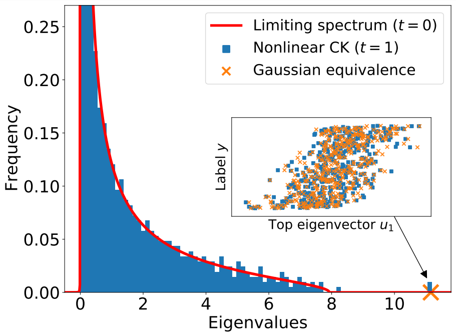
Nonlinear CK : $\boldsymbol{\Phi} = \frac{1}{\sqrt{N}}\sigma(\tilde{\boldsymbol{X}}\boldsymbol{W}_1)$, "Linearized" CK : $\bar{\boldsymbol{\Phi}} = \frac{1}{\sqrt{N}}\left(\mu_1 \tilde{\boldsymbol{X}}\boldsymbol{W}_1 + \mu_2 \boldsymbol{Z}\right)$.

Conjecture (Gaussian Equivalence of CK Spike)

For odd activation σ and $\eta = \Theta(1)$, given i.i.d. training data $\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{y}}$ (independent to $\boldsymbol{W}_1$). Denote the left singular vectors of $\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}$ as $\boldsymbol{u}_1, \bar{\boldsymbol{u}}_1$, we conjecture

$$\left|s_i(\boldsymbol{\Phi}) - s_i(\bar{\boldsymbol{\Phi}})\right| = o_{d,\mathbb{P}}(1), \ \forall i \in [n]; \quad |\langle \boldsymbol{u}_1, \tilde{\boldsymbol{y}}/\|\tilde{\boldsymbol{y}}\|\rangle|^2 = |\langle \bar{\boldsymbol{u}}_1, \tilde{\boldsymbol{y}}/\|\tilde{\boldsymbol{y}}\|\rangle|^2 + o_{d,\mathbb{P}}(1).$$

Spiked Model for CK (continued)



Blue: empirical simulation.
Red: analytic prediction (initial CK).
Orange: Gaussian equivalence.

- $\sigma = \text{SoftPlus}$.
- $f^*(\boldsymbol{x}) = \tanh(\langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle)$.
- $\psi_1 = n/d = 3/2$, $\psi_2 = N/d = 5/4$.
- $\eta = 2$.

- The **bulk** of the CK spectrum remains unchanged [6].
- A **spike** (×) appears in the learned CK, predicted by Gaussian equivalence.
- The corresponding eigenvector $\boldsymbol{u}_1$ aligns with training labels $\tilde{\boldsymbol{y}}$.

[6] The spectrum of the initialized CK_0 is characterized in [Fan and Wang 2020].

Prediction Risk of CK Ridge Regression

Question: does this alignment improve the performance of the kernel model?

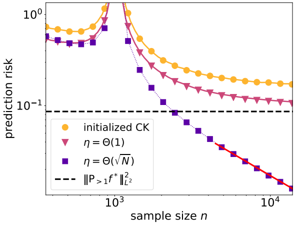
Case Study: Single-index target[7]. $f^*(\boldsymbol{x}) = \sigma^*(\langle \boldsymbol{x}, \boldsymbol{\beta}^* \rangle)$.
where $\|\boldsymbol{\beta}_*\| = 1$, and σ^* is Lipschitz with $\mu_0^* = 0$, $\mu_1^* \neq 0$.

Goal: compute the prediction risk $\mathcal{R}_{\mathrm{GD}}(\lambda)$ of the ridge estimator

$$f_{\mathrm{GD}}^{\lambda}(\boldsymbol{x}) = \frac{1}{\sqrt{N}} \hat{\boldsymbol{a}}_{\lambda}^{\top} \sigma\left(\boldsymbol{W}_1{}^{\top} \boldsymbol{x} \right), \ \hat{\boldsymbol{a}}_{\lambda} = \mathrm{argmin}_{\boldsymbol{a}} \left\{ \frac{1}{n} \left\| \tilde{\boldsymbol{y}} - \frac{1}{\sqrt{N}} \sigma(\tilde{\boldsymbol{X}} \boldsymbol{W}_1) \boldsymbol{a} \right\|^2 + \frac{\lambda}{N} \|\boldsymbol{a}\|^2 \right\}.$$

We consider the following learning rate scalings:

- Small lr $\eta = \Theta(1)$: trained CK always improve upon the initial CK ridge estimator ($\mathcal{R}_0(\lambda)$).
- Large lr $\eta = \Theta(\sqrt{d})$: for some f^*, trained CK may outperform the lower bound $\|\mathrm{P}_{>1} f^*\|_{L^2}$.



[7] This setting is often studied in RF regression (e.g. [Gerace et al. 20],[Dhifallah and Lu 20]).

The Gaussian Equivalence Property

Consider the prediction risk of ridge regression on features $\mathrm{F} \in \{\mathrm{CK}, \mathrm{GE}\}$:

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

- CK (nonlinear): $\phi_{\mathrm{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}^\top \boldsymbol{x})$.
- GE (linear): $\phi_{\mathrm{GE}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\left(\mu_1 \boldsymbol{W}^\top \boldsymbol{x} + \mu_2 \boldsymbol{z}\right)$, $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$.

  where $\mu_1 = \mathbb{E}[z\sigma(z)]$, $\mu_2 = \sqrt{\mathbb{E}[\sigma(z)^2] - \mu_1^2}$

The **Gaussian Equivalence Property** refers to: $\mathcal{R}_{\mathrm{CK}}(\lambda) \approx \mathcal{R}_{\mathrm{GE}}(\lambda)$.

Previously, the Gaussian equivalence theorem (GET) has been shown for certain RF models [Hu and Lu 2020], but not for the trained features.

Implications of the Gaussian Equivalence:

- We can equivalently compute $\mathcal{R}_{\mathrm{GE}}$, which can be handled via RMT tools ☺
- The nonlinear CK model achieves the same performance as a linear model ☹

Gaussian Equivalence for Trained Features

Theorem (BES+22)

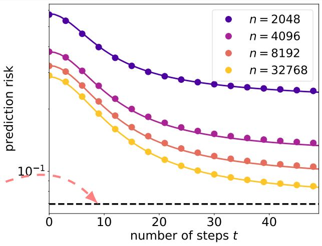
*Assume σ is **odd** in addition to the previous assumptions, then for fixed $t \in \mathbb{N}$, after the first-layer **W** is trained for t gradient steps with $\eta = \Theta(1)$,*

$$|\mathcal{R}_{\mathrm{CK}}(\lambda) - \mathcal{R}_{\mathrm{GE}}(\lambda)| = o_{d,\mathbb{P}}(1), \text{ for } \lambda > 0.$$

Intuition: GET holds when $\boldsymbol{W}_t$ is not far away from the random initialization $\boldsymbol{W}_0$.

Figure: dots represent empirical values, solid curves are asymptotics predicted by CGMT.

- For learning rate $\eta = \Theta(1)$, GET remains accurate in the early phase of training
- Prediction risk $\mathcal{R}_{\mathrm{GD}}(\lambda)$ can improve, but is still lower-bounded by $\|\mathrm{P}_{>1} f^*\|_{L^2}^2$



$\sigma = \mathrm{ReLU}, \sigma^* = \tanh.$

Gaussian Equivalence Theorem (continued)

Proof Sketch. We extend the argument in [Hu and Lu 2020] outline below.

1. **Lindeberg exchange**. Let $\hat{\boldsymbol{g}}_k$ be the solution of the optimization problem:

$$L_k \triangleq \min_{\boldsymbol{g}\in\mathbb{R}^N} \left\{ \sum_{i=1}^{k} \ell(y_i, \langle \boldsymbol{g}, \phi_{\mathrm{GE}}(\boldsymbol{x}_i)\rangle) + \sum_{j=k+1}^{n} \ell(y_j, \langle \boldsymbol{g}, \phi_{\mathrm{CK}}(\boldsymbol{x}_j)\rangle) + \frac{n}{N}(\lambda\|\boldsymbol{g}\|_2^2 + Q(\boldsymbol{g})) \right\}$$

As there are N *total swaps*, it suffices to show that for bounded test function ζ,

$$\left|\mathbb{E}\zeta\left(\tfrac{1}{N}L_k\right) - \mathbb{E}\zeta\left(\tfrac{1}{N}L_{k-1}\right)\right| = \mathcal{O}\left(\frac{\mathrm{polylog} N}{N^{3/2}}\right). \qquad \text{(A)}$$

2. **Central limit theorem**. A crucial step in establishing (A) is the following CLT:

$$\left|\mathbb{E}\varphi(\langle\phi_{\mathrm{GE}}, \boldsymbol{g}\rangle) - \mathbb{E}\varphi(\langle\phi_{\mathrm{CK}}, \boldsymbol{g}\rangle)\right| = \mathcal{O}\left(\frac{\mathrm{polylog} N}{\sqrt{N}} \cdot \left(1 + \|\boldsymbol{g}\|_\infty^2\right)\right).$$

This is shown using *Stein's method*, when $\boldsymbol{W}$ has *near-orthogonal* columns.

3. **ℓ_∞-norm control**. Finally, we show that entries of $\hat{\boldsymbol{g}}_k$ are "evenly distributed"[8]:

$$\mathbb{P}\left(\|\hat{\boldsymbol{g}}_k\|_\infty \geq \mathrm{polylog} N\right) \leq \exp\left(-c\log^2 N\right), \text{ for all } k \in [N].$$

[8] In this part of the analysis, [Hu and Lu 2020] required $\boldsymbol{W}_{ij}$ to be i.i.d. Gaussian.

Analysis of Small Learning Rate ($\eta = \Theta(1)$)

Goal: can we rigorously show that one feature learning step always *decreases* the prediction risk of the CK ridge regression estimator?

- Risk of *initial* CK (random features): $\mathcal{R}_0(\lambda) = \mathbb{E}_{\boldsymbol{x}}\left(\langle \sigma(\boldsymbol{W}_0^\top \boldsymbol{x}), \hat{\boldsymbol{a}}_0\rangle - f^*(\boldsymbol{x})\right)^2$.
- Risk of *trained* CK (after one step): $\mathcal{R}_{\mathrm{GD}}(\lambda) = \mathbb{E}_{\boldsymbol{x}}\left(\langle \sigma(\boldsymbol{W}_1^\top \boldsymbol{x}), \hat{\boldsymbol{a}}_1\rangle - f^*(\boldsymbol{x})\right)^2$.

Theorem (BES+22)

For $\eta = \Theta(1)$ and $\lambda > 0$, as $n/d \to \psi_1$, $N/d \to \psi_2$, we have

$$\mathcal{R}_0(\lambda) - \mathcal{R}_{\mathrm{GD}}(\lambda) \xrightarrow{\mathbb{P}} \delta(\eta, \lambda, \psi_1, \psi_2).$$

- *$\delta(\eta, \lambda, \psi_1, \psi_2)$ is a* ***non-negative*** *function of $\eta, \lambda, \psi_1, \psi_2 \in (0, +\infty)$;*
- *δ vanishes if and only if (at least) one of μ_1^*, μ_1 and η is zero.*

Provable improvement over the initial CK model!

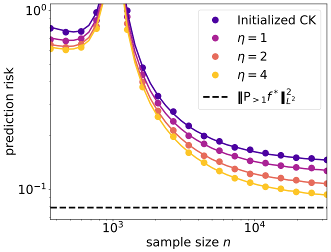
Note: this does not require the student and teacher to have the same nonlinearity

Small learning Rate (continued)

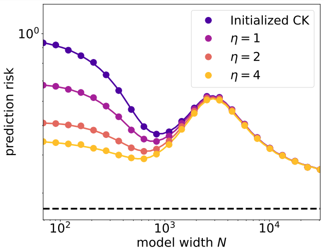
In some special cases, the expression of δ can be further simplified.

Proposition (BES+22)

- **[Large sample limit]** *As* $\psi_1 \to \infty$, δ *is* ***increasing*** *with respect to* η.
- **[Large width limit]** *As* $\psi_2 \to \infty$, $\delta(\eta, \lambda, \psi_1, \psi_2) \to 0$.



Risk vs. sample size.



Risk vs. model width.

Note: In all cases, $\mathcal{R}_0(\lambda) \geq \mathcal{R}_{\mathrm{GD}}(\lambda) \geq \|P_{>1} f^*\|_{L^2}^2$ due to the GET under $\eta = \Theta(1)$.

Analysis of Large Learning Rate ($\eta = \Theta(\sqrt{d})$)

Finally, we consider the large learning rate regime with $\eta = \Theta(\sqrt{d})$.

- $\boldsymbol{W}_1$ travels far away from initialization $\Rightarrow$ CK can be "nonlinear" ☺
- In the absence of GET, precise analysis of prediction risk is difficult ☹

Alternative: *upper-bound* $\mathcal{R}_{\mathrm{GD}}(\lambda)$ and compare against *kernel lower bound*.

We define: $\tau^* := \inf_\eta \mathbb{E}_{\xi_1}\left(\sigma^*(\xi_1) - \mathbb{E}_{\xi_2}(\sigma(\eta\xi_1 + \xi_2))\right)^2$

Lemma (BES+22)

[Informal] *Given* ***bounded*** *activation σ, after one GD step on $\boldsymbol{W}$ with $\eta = \Theta(\sqrt{N})$, there exists some $\tilde{f}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\tilde{\boldsymbol{a}}^\top \sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$ that achieves prediction risk "close" to τ^*.*

- τ^* can be interpreted as some measure of "model misspecification".
- **Note:** the definition of τ^* *does not involve the specific value of step size η.*

Large Learning Rate (continued)

Theorem (BES+22)

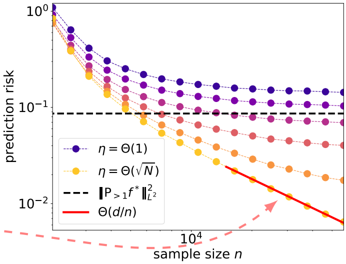
After one GD step on $\boldsymbol{W}$ with $\eta = \Theta(\sqrt{N})$, there exist constants $C, \psi_1^* > 0$ such that for any $\psi_1 > \psi_1^*$, and $n^{\epsilon-1} < N^{-1}\lambda < n^{-\epsilon}$ for some small $\epsilon > 0$, we have

$$\mathcal{R}_{\mathrm{GD}}(\lambda) \leq 16\tau^* + C\left(\sqrt{\tau^*} \cdot \psi_1^{-1/2} + \psi_1^{-1}\right),$$

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

If $\tau^* \ll \|\mathrm{P}_{>1} f^*\|_{L^2}^2$, CK ridge regression after *one feature learning step* outperforms the kernel ridge lower bound:

- $\sigma = \sigma^* = \tanh$: $\mathcal{R}_{\mathrm{GD}}(\lambda) < \|\mathrm{P}_{>1} f^*\|_{L^2}^2$
- $\sigma = \sigma^* = \mathrm{erf}$: there exists constant $C > 0$ s.t. $\mathcal{R}_{\mathrm{GD}}(\lambda) \leq C \cdot \psi_1^{-1} = \Theta(d/n)$



$\sigma = \sigma^* = \mathrm{erf}$, $\eta = N^{\alpha}$, $\alpha \in [0, 1/2]$.

Caution: separation only present in specific (σ, σ^*)

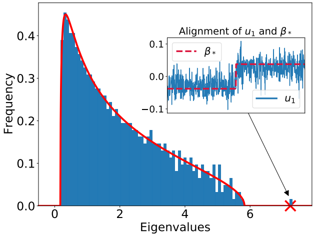
Summary of Results

How Does One Gradient Step Change the Weights?

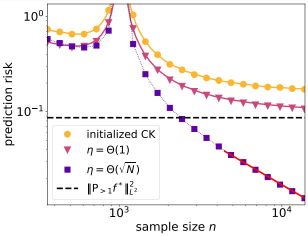
- The isolated singular vector of $\boldsymbol{W}_1$ aligns with *linear component of* f^*.
- The top eigenvector of CK matrix aligns with *training labels* $\boldsymbol{y}$ (conjecture).

How Do the Learned Features Improve Generalization?

- $\eta = \Theta(1)$ – **Linear Regime**. Precise analysis via GET; $\mathcal{R}_0 \geq \mathcal{R}_{\mathrm{GD}} \geq \|\mathrm{P}_{>1} f^*\|_{L^2}^2$.
- $\eta = \Theta(\sqrt{d})$ – **Nonlinear Regime**. For certain f^*, $\mathcal{R}_0 \geq \|\mathrm{P}_{>1} f^*\|_{L^2}^2 \geq \mathcal{R}_{\mathrm{GD}}$.



Signal+noise structure of $\boldsymbol{W}_1$.

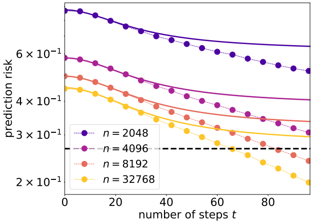


Improvement of prediction risk.

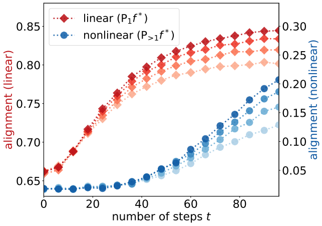
Future Directions

Some questions to consider:

1. A *spiked model* for the kernel (CK) matrix after one gradient step?
2. "Phase transition" in the Gaussian equivalence property?
3. *Precise asymptotics* beyond Gaussian equivalence?



Prediction risk vs. time step t.



Alignment with teacher f^*.

Thank you!

References

- El Karoui, 2010. *The spectrum of kernel random matrices*.
- Louart, Liao, and Couillet, 2018. *A random matrix approach to neural networks*.
- Mei and Montanari, 2019. *The generalization error of random features regression: Precise asymptotics and double descent curve*.
- Ghorbani et al., 2019. *Linearized two-layer neural networks in high dimensions*.
- Hu and Lu, 2020. *Universality laws for high-dimensional learning with random features*.
- Fan and Wang, 2020. *Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks*.
- Gerace et al., 2020. *Generalisation error in learning with random features and the hidden manifold model*.
- Yang and Hu, 2021. *Feature learning in infinite-width neural networks*.
- Loureiro et al., 2021. *Learning curves of generic features maps for realistic datasets with a teacher-student model*.
- Bartlett et al., 2021. *Deep learning: a statistical viewpoint*.