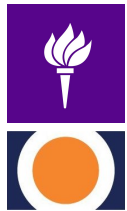
Understanding modern machine learning models through the lens of high-dimensional statistics

Denny Wu

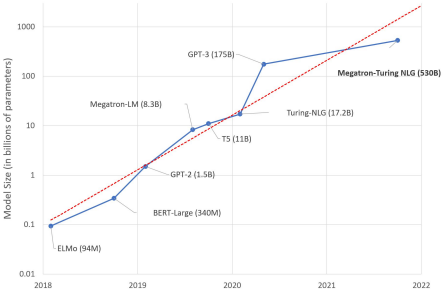
Center for Data Science, New York University
Center for Computational Mathematics, Flatiron Institute



High-dimensionality of Modern ML Systems

Modern ML tasks require searching over a *high-dimensional* parameter space.

Curse of dimensionality? Larger neural networks often achieve better performance.



LLM parameter count (Hugging Face blogpost)

Overparameterization: # parameters > #training data.

High-dimensionality of Modern ML Systems

Modern ML tasks require searching over a *high-dimensional* parameter space.

Curse of dimensionality? Larger neural networks often achieve better performance.

Understanding the success of deep learning

(i) **Optimization:** standard gradient-based methods work, despite the non-convexity.
- benefit of overparameterization (NTK, mean-field, etc.)

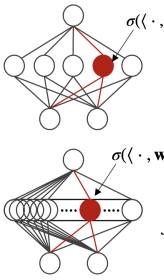
(ii) **Generalization:** model generalizes well, despite the overparameterization.
- implicit regularization, benign overfitting.

(iii) **Why neural networks?** NN often outperforms classical methods (e.g., kernels).
- adaptivity, representation (feature) learning.

My research: quantitative understanding of (*i*)-(*iii*) via *high-dimensional statistics*.

Intuition: theoretical analysis may simplify if we take the *dimensionality to infinity*.

Scaling ① – Large Width Limit



$$f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \sigma(\langle \mathbf{x}, \mathbf{w}_i \rangle)$$

$$N \to \infty$$

$$f(\mathbf{x}) = \mathbb{E}_p\big[\sigma(\langle \mathbf{x}, \mathbf{w} \rangle)\big] = \int \sigma(\langle \mathbf{x}, \mathbf{w} \rangle) p(\mathbf{w}) \mathrm{d}\mathbf{w}$$

For convex loss L, learning is

☹ non-convex w.r.t. $\boldsymbol{w}_i$

☺ convex w.r.t. distribution p

Perspective: study optimization in the space of measures (Wasserstein gradient flow, functional inequalities (LSI), etc.)

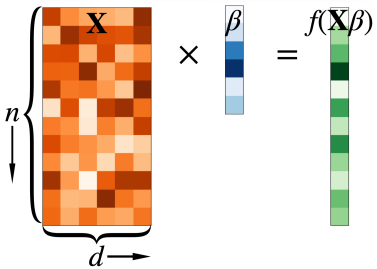
- Convergence rate of mean-field Langevin dynamics and propagation of chaos [NWS22][SWN23]
- Learnability guarantees for low-dimensional target functions [SWO+23][NOS+23]
- New algorithms for optimization in the space of measures [NWS21] [OSN+22] [NOW+23]
- .....

Mathematical Models for High-dimensional Problems

Intuition: theoretical analysis may simplify if we take the *dimensionality to infinity*.

Scaling ② – Proportional Asymptotic Limit

$$n, d \to \infty, d/n = \gamma \in (0, \infty)$$



Diverging dimensionality & *fixed aspect ratio*.

- Captures the *overparameterized* regime (by setting $\gamma > 1$)

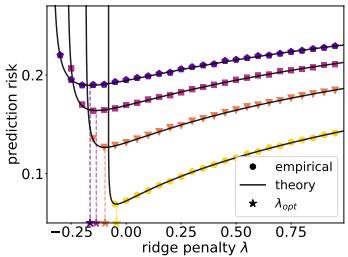
Performance of simple ML models can be **precisely analyzed** via *random matrix theory* (the study of large-dimensional matrices with certain *random* structures)

This talk: two examples of precise analysis using random matrix theory (RMT).
(*i*) optimal regularization in linear regression. (*ii*) feature learning in neural network.

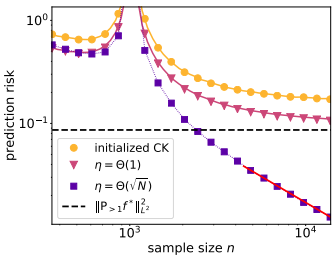
Precise Analysis of Learning in High Dimensions

What are the advantages of a **precise analysis**?

- Enables *accurate comparison* between estimators/algorithms.
  - ❒ positive vs. negative ridge penalty, gradient descent vs. natural gradient, etc.
- Captures refined properties of the learning curve.
  - ❒ phase transitions, (non-)monotonicity, etc.



"Negative ridge" phenomenon.



Benefit of representation learning.

Ridge Regression in High Dimensions

Problem Setting & Assumptions

- **Data Generation:** $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}_* + \varepsilon_i$, $1 \le i \le n$. $\boldsymbol{x}_i \in \mathbb{R}^d$.
  i.i.d. label noise satisfies $\mathbb{E}[\varepsilon] = 0$, $\mathrm{Var}(\varepsilon) = \sigma_\varepsilon^2$.
- **Random Design:** $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \boldsymbol{\Sigma})$. Also holds for $\boldsymbol{x}_i$ with bounded $(4+\epsilon)$ moment
- **Signal (Ground Truth):** $\boldsymbol{\beta}_*$ can be both *fixed* or *random* (i.e., $\mathbb{E}[\boldsymbol{\beta}_* \boldsymbol{\beta}_*^\top] = \boldsymbol{\Sigma}_\beta$)
- **Proportional Asymptotics:** $n, d \to \infty$, $d/n \to \gamma \in (0, \infty)$.

Ridge regression estimator: $\hat{\boldsymbol{\beta}}_\lambda = \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}\right)^\dagger \boldsymbol{X}^\top \boldsymbol{y}$.

- **Goal:** compute the *prediction risk* (test error) $\mathcal{R}(\lambda) = \mathbb{E}(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\lambda)^2$.

Remark: When $\lambda \ge 0$, $\hat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^n (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|^2$ (Gaussian prior)

Asymptotic Risk Formulae

Theorem ([WX20] Precise generalization error of ridge regression)

The test error admits a bias-variance decomposition $\mathcal{R}(\lambda) = \mathcal{B}(\lambda) + \mathcal{V}(\lambda)$, *where*

$$\mathcal{B}(\lambda) \xrightarrow{p} \frac{\partial \kappa_\lambda}{\partial \lambda} \cdot \kappa_\lambda^2 \big\langle \boldsymbol{\beta}_*, \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\lambda \boldsymbol{I})^{-2} \boldsymbol{\beta}_* \big\rangle, \quad \mathcal{V}(\lambda) \xrightarrow{p} \sigma_\varepsilon^2 \frac{\partial \kappa_\lambda}{\partial \lambda},$$

and $\kappa_\lambda \geq \lambda$ *is the* ***effective regularization*** *given by the non-negative solution of*

$$\frac{1}{n} \mathrm{Tr}\big(\boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\lambda \boldsymbol{I})^{-1}\big) = 1 - \frac{\lambda}{\kappa_\lambda}.$$

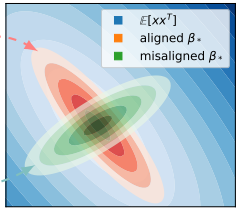
- **Bias** $\mathcal{B}(\lambda)$: learning of *signal* $\boldsymbol{\beta}_*$.
- **Variance** $\mathcal{V}(\lambda)$: "overfitting" to *label noise*.

Given *eigendecomposition* $\boldsymbol{\Sigma} = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$, prediction risk $\mathcal{R}(\lambda)$ depends on:

- **Capacity condition:** eigenvalues of the population covariance $\{\lambda_i\}_{i=1}^d$.
- **Source condition:** projection of signal (teacher) $\boldsymbol{\beta}_*$ onto the feature eigenbasis $\{\rho_i\}_{i=1}^d$, where $\rho_i = \langle \boldsymbol{\beta}_*, \boldsymbol{u}_i \rangle$.

Alignment between Features and Signal

- *Aligned* feature & signal: large $\lambda_i \Leftrightarrow$ large $\langle \boldsymbol{\beta}_*, \boldsymbol{u}_i \rangle$
  - ☺ Features are well-engineered $\Rightarrow$ easy problem
- *Misaligned* feature & signal: large $\lambda_i \Leftrightarrow$ small $\langle \boldsymbol{\beta}_*, \boldsymbol{u}_i \rangle$
  - ☹ Features are uninformative $\Rightarrow$ hard problem



Theorem ([WX20] Sign of Optimal Ridge Penalty λ_{opt})

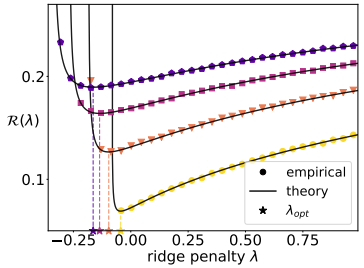
Recall that $\gamma = d/n$.

- ❒ $\gamma < 1$ (***underparameterized***): $\lambda_{\text{opt}} \geq 0$ *in all cases.*
- ❒ $\gamma > 1$ (***overparameterized***): *the sign of λ_{opt} depends on the alignment between the features and the signal.*

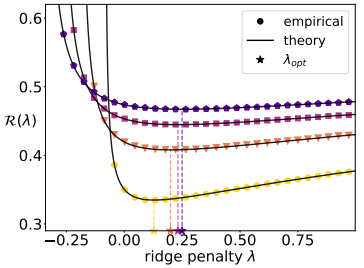
The "Negative Ridge" Phenomenon

Corollary ([WX20] Sign of λ_{opt} in the Overparameterized Regime)

- ***Negative*** λ *is beneficial under alignment (informative features); hence interpolation* ($\lambda = 0$) *can be optimal even if* $\sigma_\varepsilon > 0$.
- ***Positive*** λ *is beneficial under misalignment (hard problem), even in the absence of label noise* ($\sigma_\varepsilon = 0$).



Aligned: $\lambda_{\text{opt}} < 0$.



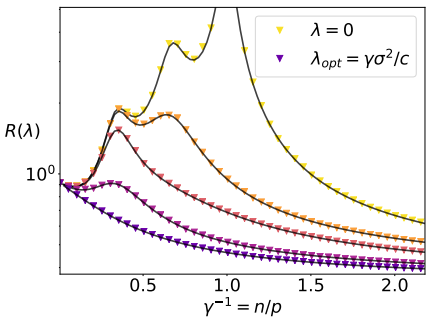
Misaligned: $\lambda_{\text{opt}} > 0$.

Regularization Suppresses "Multiple Descent"

Without appropriate regularization, $\mathcal{R}(\lambda)$ may exhibit *multiple peaks*...

Theorem ([WX20] Monotonicity of $\mathcal{R}(\lambda_{\text{opt}})$)

*Given $\mathbb{E}[\boldsymbol{\beta}_*\boldsymbol{\beta}_*^\top] \propto \boldsymbol{I}$ (isotropic prior), the optimally regularized prediction risk $\mathcal{R}(\lambda_{opt})$ is a **decreasing** function of $\gamma^{-1} = n/d \in (0, \infty)$.*



Message: if we tune λ, more training data *always helps* the test performance.

Implication I: Implicit Bias of Optimizers

Update rule: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{P}(\boldsymbol{\theta}_t) \nabla_{\boldsymbol{\theta}_t} L(\boldsymbol{\theta}_t), \quad t = 0, 1, \ldots.$

Geometric Intuition: $\boldsymbol{P}$ alleviates pathological curvature and speed up **optimization**.

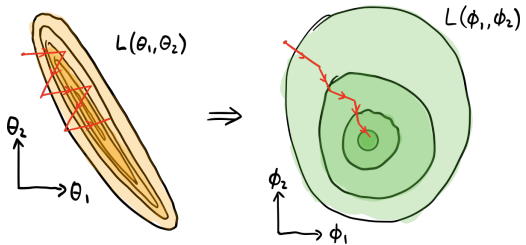


Figure from Xanadu blog post.

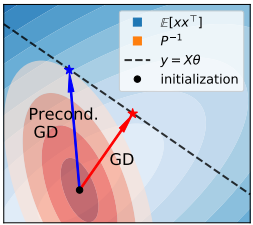
Question: in the *interpolation setting* (i.e. absence of explicit regularization), how does preconditioning influence the **generalization** performance?

Implicit Bias in Overparameterized Linear Regression

Theoretical Setting: preconditioned gradient descent (flow) on the *overparameterized* least squares objective: $L(\beta) = \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\beta\|_2^2$.

Implicit Bias ($t \to \infty$):

- **Gradient descent:** min ℓ_2 norm solution.
- **Preconditioned GD:** for *time-independent* and full-rank $\boldsymbol{P}$, min $\|\beta\|_{\boldsymbol{P}^{-1}}$ norm solution.



Example.

Natural gradient descent with *population Fisher*: $\boldsymbol{P} = \boldsymbol{\Sigma}^{-1}$

- **Goal I:** use the asymptotic risk formulae (taking $\lambda \to 0$) to *precisely* compare the generalization of GD vs. NGD.
- **Goal II:** validate our predictions in neural network experiments.

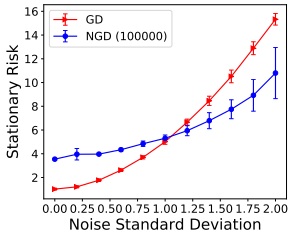
Comparison of Generalization Performance

Theorem ([ABG+21] Prediction Risk of GD vs. NGD)

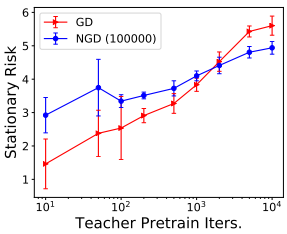
- ***Variance*** $\lim_{\lambda \to 0} \mathcal{V}(\lambda)$*: NGD (population,* $\boldsymbol{P} = \boldsymbol{\Sigma}^{-1}$*) is optimal.*
- ***Bias*** $\lim_{\lambda \to 0} \mathcal{B}(\lambda)$*: GD generalizes better when signal is isotropic (*$\boldsymbol{\Sigma}_\beta = \boldsymbol{I}$*); NGD generalizes better under misalignment ("difficult problem").*

Remark: bias-variance tradeoff achieved by "interpolating" between optimizers.

Two-layer MLP: student-teacher setup (CIFAR-10)



Label noise (variance).



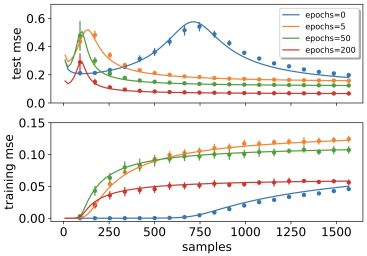
Misspecification (bias).

Implication II: Beyond Gaussian Features

Question: does our risk formula have predictive power in *practical settings*, e.g., neural network representations?

- **Decomposition of kernel:** $k(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \lambda_i \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{y})$.
- **Decomposition of target function:** $f^*(\boldsymbol{x}) = \sum_i \rho_i \phi_i(\boldsymbol{x})$, $\rho_i = \langle \phi_i, f^* \rangle_{L^2}$.
- **Leap of faith:** estimate $\{\lambda_i, \rho_i\}_{i=1}^{\infty}$ from data, and plug in the risk formulae.

Universality: RMT prediction empirically accurate for many feature maps, including **trained** *neural network features*.



[Loureiro et al. 2021]

Observation: trained NN achieves lower risk

$\Rightarrow$ advantage of *representation learning*.

❒ **Spoiler:** this benefit will be precisely analyzed!

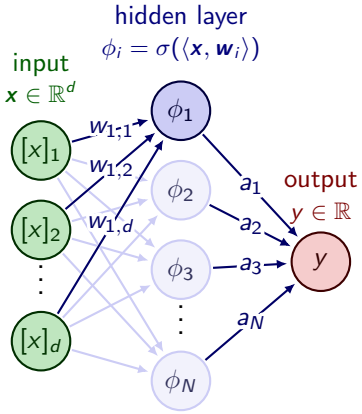
Two-layer Neural Network

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

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

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

- Increase $\psi_1 \Rightarrow$ larger **sample size**.
- Increase $\psi_2 \Rightarrow$ **overparameterization**.



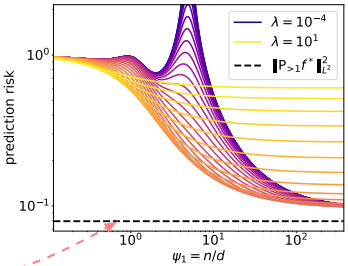
Motivation: rigorously show that the *learned representation* (via gradient descent) achieves better performance in the proportional limit.

Prior Works: Asymptotics of Random Features Model

Fix 1st layer $\boldsymbol{W}$ at initialization, learn 2nd layer $\boldsymbol{a}$
$\Rightarrow$ **random features (RF)** model.

☺ Prediction risk precisely characterized in the *proportional regime* via **random matrix theory**.

☹ The (*nonlinear*) RF estimator cannot even outperform linear functions on the input...



$\mathsf{P}_{>1}$ is the projector orthogonal to constant & linear functions in $L_2(P_x)$

Where does this gap come from?

Feature (representation) learning!

❒ When $\boldsymbol{W}$ is optimized, NN can "adapt" to data and learn useful features.

- Mei and Montanari, 2019. *The generalization error of random features regression: Precise asymptotics and double descent curve.*
- Gerace et al., 2020. *Generalisation error in learning with random features and the hidden manifold model.*

Feature Learning via One Gradient Descent Step

"Early Phase" Feature Learning: Does the *first* gradient descent step on the first-layer $\boldsymbol{W}$ already learn useful representations?

- **One-step GD on 1st Layer**. Gradient update $\boldsymbol{W}_1 = \boldsymbol{W}_0 + \eta\sqrt{N}\cdot\boldsymbol{G}$, where

$$\boldsymbol{G} = -\nabla_{\boldsymbol{W}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - f_{\mathsf{NN}}(\boldsymbol{x}_i)\right)^2\right] = \frac{1}{n}\boldsymbol{X}^\top\left[\left(\frac{1}{\sqrt{N}}(\boldsymbol{y} - f_{\mathsf{NN}}(\boldsymbol{X}))\boldsymbol{a}^\top\right)\odot\sigma'(\boldsymbol{X}\boldsymbol{W}_0)\right].$$

- **Ridge Regression for 2nd Layer**. Regression using *trained* kernel features:

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

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

Goal: Precise analysis of $\mathcal{R}_{\mathrm{GD}}(\lambda)$ to show its *improvement* over the initialized RF $\mathcal{R}_{\mathsf{RF}}(\lambda)$, and potentially over the *kernel lower bound* $\|\mathsf{P}_{>1}f^*\|_{L^2}^2$.

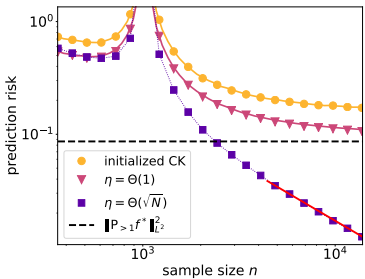
Our Results: Precise Asymptotics of Feature Learning

- **Student-teacher Setup.** $y_i = f^*(\langle \boldsymbol{x}_i, \boldsymbol{\beta}_* \rangle) + \varepsilon_i$, where $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \boldsymbol{I})$.
- **Gaussian Initialization.** $[\boldsymbol{W}_0]_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/d)$, $[\boldsymbol{a}]_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/N)$.

Small lr ("lazy" regime): $\eta = \Theta(1) \;\Rightarrow\; \left|[\boldsymbol{W}_1 - \boldsymbol{W}_0]_{ij}\right| \ll \left|[\boldsymbol{W}_0]_{ij}\right|$

Large lr (μP scaling): $\eta = \Theta(\sqrt{N}) \;\Rightarrow\; \left|[\boldsymbol{W}_1 - \boldsymbol{W}_0]_{ij}\right| \asymp \left|[\boldsymbol{W}_0]_{ij}\right|$

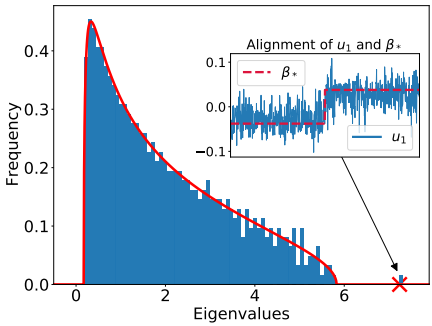
- Small lr $\eta = \Theta(1)$: trained kernel **always improve** upon the initial RF estimator, but the model remains "*linear*".
- Large lr $\eta = \Theta(\sqrt{N})$: regression on trained features can learn certain **nonlinear** f^*.



- Jacot et al, 2018. *Neural tangent kernel: convergence and generalization in neural networks.*
- Yang and Hu, 2021. *Feature learning in infinite-width neural networks.*

A Spiked Model for the Trained Weight Matrix

Challenge: learned $\boldsymbol{W}_1$ *no longer i.i.d.*; can we still apply *RMT* tools?



Blue: empirical simulation

Red: analytic prediction (*BBP Phase Transition*)

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

Observation: after *one feature learning step* on the first-layer $\boldsymbol{W}$:

- The **bulk** of the spectrum of $\boldsymbol{W}_1$ remains unchanged
- A **spike** ($\times$) appears in $\boldsymbol{W}_1$, which aligns with signal $\boldsymbol{\beta}^*$

$\eta = \Theta(1)$ – Precise Analysis via Gaussian Equivalence

Intuition – Universality: replace *nonlinear* NN features with *linear* Gaussian features with *matching first two moments* does not change the risk.

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

where $\mu_1 = \mathbb{E}[z\sigma(z)]$, $\mu_2 = \sqrt{\mathbb{E}[\sigma(z)^2] - \mu_1^2}$ $\Rightarrow$ $\mathbb{E}[\phi_{\rm NN}\phi_{\rm NN}^\top] = \mathbb{E}[\phi_{\rm GE}\phi_{\rm GE}^\top]$

Theorem ([BES+22] Gaussian Equivalence for Trained Features)

After one feature learning step on $\boldsymbol{W}$ with small learning rate $\eta = \Theta(1)$,

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

Implications of Gaussian Equivalence (GET):

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

Precise Characterization of Feature Learning

Theorem ([BES+22] Benefit of Feature Learning)

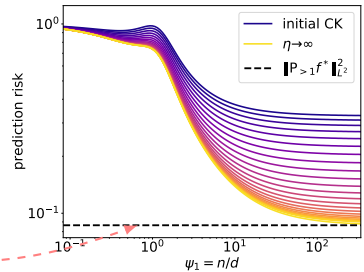
The prediction risk difference $\delta := \lim_{n,d,N\to\infty} \mathcal{R}_{\mathrm{RF}}(\lambda) - \mathcal{R}_{\mathrm{GD}}(\lambda)$ *is*

- *a* ***non-negative*** *function of* $\eta, \lambda, \psi_1, \psi_2 \in (0, +\infty)$*;*
- *an* ***increasing*** *function with respect to learning rate* η*.*

Provable improvement over the initial RF model!

Observations:

- For $\eta = \Theta(1)$, feature learning always helps.
- Larger step size $\Rightarrow$ *greater improvement*.
- Improvement also limited by the GET, i.e., the learned kernel is still "linear".



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

$\eta = \Theta(\sqrt{N})$ – Upper Bound via Nonparametric Analysis

Sufficiently large $\eta \Rightarrow \boldsymbol{W}_1$ travels far away from initialization.

☺ learned kernel can be "nonlinear". ☹ Gaussian equivalence no longer holds.

Theorem ([BES+22] Upper Bound on Prediction Risk)

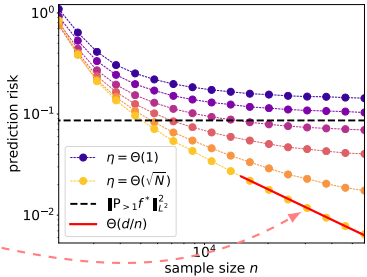
After one GD step on $\boldsymbol{W}$ *with* $\eta = \Theta(\sqrt{N})$*, for appropriate* λ *and* $\psi_1 > \psi^*$*,*

$$\mathcal{R}_{\mathrm{GD}}(\lambda) \le 10\tau^* + \Theta(\psi_1^{-1}), \quad w.h.p.,$$

where constant τ^* *depends on* σ, f^**, but not the specific value of step size* η*.*

If $\tau^* \ll \|\mathrm{P}_{>1} f^*\|_{L^2}^2$, *one feature learning step* can outperform kernel lower bound:

- $\sigma = f^* = \tanh$: $\mathcal{R}_{\mathrm{GD}}(\lambda) < \|\mathrm{P}_{>1} f^*\|_{L^2}^2$
- $\sigma = f^* = \mathrm{erf}$: $\mathcal{R}_{\mathrm{GD}}(\lambda) = O(d/n)$



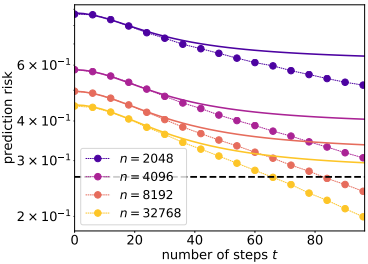
Conclusion and Future Directions

Random matrix theory allows us to characterize

- ❒ Precise conditions that determine the *sign of optimal ridge penalty*.
- ❒ Benefit of *representation learning* in the "early phase" of gradient descent.

Open Questions

- **Universality:** Under what conditions on the representation do we expect the RMT predictions to hold?
- **Beyond Universality:** What theoretical tools can we employ when the RMT predictions fail?



Failure case of RMT prediction.

Conclusion and Future Directions

Deep learning phenomena $\rightarrow$ interesting mathematical problems

- New models of (nonlinear) random matrix theory.
  - properties of neural net representation, beyond the proportional regime, ...
- What functions can be efficiently learned by neural network + gradient descent?
  - sparsity & low-dimensional structure, information exponent, ...
  - the role of architecture (depth, normalization, etc.) and optimization method (stochastic gradient, preconditioning, etc.)

Theoretical advances $\rightarrow$ principled guidance in practical settings

- How do we scale hyperparameters in the overparameterized setting?
  - selection of learning rate, regularization parameters, etc.
- "neural scaling laws" beyond the kernel regime.
  - How many samples, parameters, and optimization steps is required to achieve a desired test performance?

Conclusion

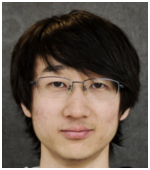
Thank you! Happy to take questions:)


Shun-ichi Amari


Jimmy Ba


Murat A. Erdogdu

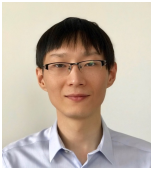

Roger Grosse


Xuechen Li


Atsushi Nitanda


Taiji Suzuki


Zhichao Wang


Ji Xu


Greg Yang

References

This Talk:

- [WX20], *Optimal weighted ℓ_2 regularization in overparameterized linear regression.*
- [ABG+21], *When does preconditioning help or hurt generalization?*
- [BES+22], *High-dimensional asymptotics of feature learning: how one gradient step improves the representation.*

Additional References:

- [NWS21], *Particle dual averaging: optimization of mean-field neural networks with global convergence rate analysis.*
- [NWS22], *Convex analysis of the mean-field Langevin dynamics.*
- [BES+23], *Learning in the presence of low-dimensional structure: a spiked random matrix perspective.*
- [MWS+23], *Gradient-based feature learning under structured data.*
- [SWN23], *Mean-field Langevin dynamics: time and space discretization, stochastic gradient, and variance reduction.*
- [SWO+23], *Feature learning via mean-field Langevin dynamics: classifying sparse parities and beyond.*