A Survey of Lazy and Feature Learning Regimes

Yilun Kuang | Eugene Choi | Jeff Cui

NYU CSCI-GA 3033 Mathematics of Deep Learning
Spring 2022 Final Project
May 5, 2022

Setup & GFD + FD

Consider a shallow neural network of one hidden layer with width N with $\boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N), \boldsymbol{\theta}_i = (a_i, \boldsymbol{w}_i) \in \mathbb{R}^D, \sigma(\boldsymbol{x}; \boldsymbol{w}_i) = \sigma(\langle \boldsymbol{x}, \boldsymbol{w}_i \rangle)$.

and the population risk defined as [COB19] [MMM19]. Here we are using squared loss.

$$f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\alpha}{N}\sum_{i=1}^{N} a_i \sigma(\boldsymbol{x};\boldsymbol{w}_i) = \frac{\alpha}{N}\sum_{i=1}^{N} \sigma_*(\boldsymbol{x};\boldsymbol{\theta}_i) \quad , \quad R_{\alpha,N}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\left[\ell\Big(f(\boldsymbol{x}), f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta})\Big)\right]$$

Gradient Flow Dynamics (GFD)

$$\frac{d}{dt}\boldsymbol{\theta}_j^t = -\frac{N}{2\alpha^2}\nabla_{\boldsymbol{\theta}_j} R_{\alpha,N}(\boldsymbol{\theta}^t) = \frac{1}{\alpha}\mathbb{E}_{\boldsymbol{x}}\left[\Big(f(\boldsymbol{x}) - f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta})\Big)\nabla_{\boldsymbol{\theta}}\sigma_*(\boldsymbol{x};\boldsymbol{\theta}_j)\right] \text{ (GFD)}$$

Function Dynamics (FD)

$$\partial_t u_t^{\alpha,N}(\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{x}'}\left[\mathcal{K}(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}^t) u_t^{\alpha,N}(\boldsymbol{x}')\right] \text{ (FD).}$$

with the kernel function defined as $\mathcal{K}(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}^t) := \frac{1}{N}\sum_{j=1}^{N}\langle \nabla_{\boldsymbol{\theta}}\sigma_*(\boldsymbol{x};\boldsymbol{\theta}_j^t), \nabla_{\boldsymbol{\theta}}\sigma_*(\boldsymbol{x}';\boldsymbol{\theta}_j^t)\rangle$. Here $u_t^{\alpha,N}(\boldsymbol{x}) := f(\boldsymbol{x}) - f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta}^t)$ is the residual.

Consider **the empirical Dirac reparametrization [RV18b]:**

$\mu_t^{\alpha,N}(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{j=1}^{N}\delta_{\boldsymbol{\theta}_j^t}$. We can then rewrite our function as $f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta}) = f_\alpha(\boldsymbol{x};\mu) = \alpha\int \sigma_*(\boldsymbol{x};\boldsymbol{\theta})\mu(d\boldsymbol{\theta})$

Distributional Dynamics & Residual Dynamics

Consider again GFD. This time we can replace $\boldsymbol{\theta}$ with μ. Since the empirical Dirac measure is related to nonlinear Liouville equations [RV18b], we have the **Distributional Dynamics (DD):**

$$\partial_t \mu_t^{\alpha,N} = \frac{1}{\alpha}\nabla_{\boldsymbol{\theta}} \cdot \left(\mu_t^{\alpha,N}\left[\nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \mu_t^{\alpha,N})\right]\right) \quad \text{(DD)}$$

$$\Psi_\alpha(\boldsymbol{\theta}; \mu) := -\mathbb{E}_{\boldsymbol{x}}\left[(f(\boldsymbol{x}) - f_\alpha(\boldsymbol{x}; \mu))\sigma_*(\boldsymbol{x}; \boldsymbol{\theta})\right]$$

$$\mu_0^{\alpha,N} = \frac{1}{N}\sum_{j=1}^{N} \delta_{\boldsymbol{\theta}_j^0}$$

Residual Dynamics (RD):

Given the empirical Dirac formulation, we can reparametrize the FD as:

$$\partial_t u_t^{\alpha,N}(\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{x}'}\left[\mathcal{K}_{\mu_t^{\alpha,N}}(\boldsymbol{x}, \boldsymbol{x}') u_t^{\alpha,N}(\boldsymbol{x}')\right] \quad \text{(RD)}.$$

$$\mathcal{K}_{\mu_t^{\alpha,N}}(\boldsymbol{x}, \boldsymbol{x}') := \int \langle \nabla_{\boldsymbol{\theta}} \sigma_*(\boldsymbol{x}; \boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} \sigma_*(\boldsymbol{x}'; \boldsymbol{\theta})\rangle \mu_t^{\alpha,N}(d\boldsymbol{\theta})$$

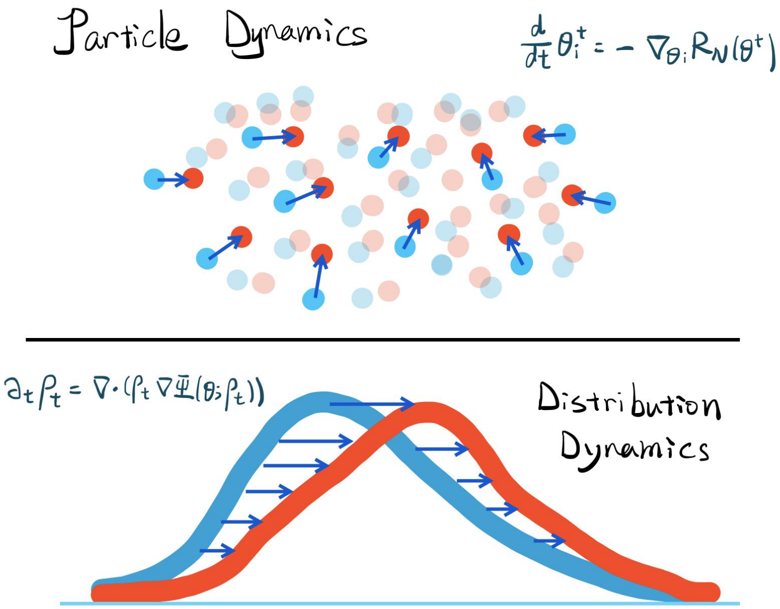

Particle Dynamics
$\frac{d}{dt}\theta_i^t = -\nabla_{\theta_i} R_N(\theta^t)$
$\partial_t \rho_t = \nabla\cdot(\rho_t \nabla \Psi(\theta;\rho_t))$
Distribution Dynamics

Figure: [S19]

Mean Field Limit and NTK Limit

Bring the DD and RD together, we have the following **coupled dynamics**:

$$\partial_t \mu_t^{\alpha,N} = \frac{1}{\alpha}\nabla_{\boldsymbol{\theta}} \cdot \left(\mu_t^{\alpha,N}\left[\nabla_{\boldsymbol{\theta}} \Psi_\alpha(\boldsymbol{\theta}; \mu_t^{\alpha,N})\right]\right) \ \text{(DD)} \quad \partial_t u_t^{\alpha,N}(\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{x}'}\left[\mathcal{K}_{\mu_t^{\alpha,N}}(\boldsymbol{x}, \boldsymbol{x}') u_t^{\alpha,N}(\boldsymbol{x}')\right] \ \text{(RD)}$$

Mean Field Limit

Let $\alpha = \mathcal{O}(1)$, as $N \to \infty$, we have the mean field limit of the residual dynamics [MMM19]

$$\partial_t u_t^{\alpha}(\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{x}'}\left[\mathcal{K}_{\mu_t^{\alpha}}(\boldsymbol{x}, \boldsymbol{x}') u_t^{\alpha}(\boldsymbol{x}')\right] \ \text{(RD-MF Limit)}$$

Notice that the RD-MF Limit is dependent on the kernel $\mathcal{K}_{\mu_t^{\alpha}}(\boldsymbol{x}, \boldsymbol{x}')$ which varies along time. So we are in the feature learning regime under the mean field limit, in comparison to the lazy regime in the neural tangent kernel limit.

Neural Tangent Kernel Limit

Let $\alpha = \mathcal{O}(N^{1/2})$. As $N \to \infty$, we have the neural tangent kernel limit of the residual dynamics $\partial_t u_t^{*}(\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{x}'}\left[\mathcal{K}_{\mu_0}(\boldsymbol{x}, \boldsymbol{x}') u_t^{*}(\boldsymbol{x}')\right]$ (RD-NTK Limit) where $\mathcal{K}_{\mu_0}(\boldsymbol{x}, \boldsymbol{x}')$ is a fixed kernel at initialization.

For $\alpha = \mathcal{O}(1), N \to \infty$, this is mean field limit **[S19]** (p. 82) **[GSJW20]**

$$\lim_{N\to\infty} f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta}) \approx \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \sigma_*(\boldsymbol{x};\boldsymbol{\theta}_i)$$

For $\alpha = N^{1/2}, N \to \infty$, this is neural tangent kernel limit **[S19]** (p. 82) **[GSJW20]**

$$\lim_{N\to\infty} f_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta}) \approx \lim_{N\to\infty} \frac{1}{\sqrt{N}}\sum_{i=1}^{N} \sigma_*(\boldsymbol{x};\boldsymbol{\theta}_i)$$

Separation between neural nets and their linearization

[GMMM19] [GMMM20] Ghorbani, Mei, Misiakiewicz, Montanari

$$f(x) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle)$$

Initial weights iid random: $w \underset{iid.}{\sim} \nu$

$$\xrightarrow{\text{Linearize}}$$

$$f(x) - f_0(x)$$

$$\approx \sum_{i=1}^{N} (a_i - a_{0,i})\sigma(\langle w_{0,i}, x\rangle) + \sum_{i=1}^{N} a_{0,i} \langle w_i - w_{0,i}, x\rangle \sigma'(\langle w_{0,i}, x\rangle)$$

$$x_i \sim \mathtt{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \|x_i\|_2^2 = d$$

$$y_i = f_{*,l}(x_i)$$

Setup: data drawn from uniform ball, transformed by target function f*

$$\mathcal{F}_{RF} := \left\{ f(x) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle) : a_i \in \mathbb{R} \right\}$$

Random features

$$\mathcal{F}_{NT} := \left\{ f(x) = \sum_{i=1}^{N} \langle a_i, x\rangle \sigma'(\langle w_i, x\rangle) : a_i \in \mathbb{R}^d \right\}$$

Neural tangent

Infinite-width limit is kernel regression

$N \to \infty$

$$k(x, x') = \int \sigma(\langle w, x\rangle)\sigma(\langle w, x'\rangle)\nu(dw)$$

$$k(x, x') = \langle x, x'\rangle \int \sigma'(w^T x)\sigma'(w^T x')\nu(dw)$$

Finite-width can be viewed as random approximation of kernel regression

$N \sim d^l$

Approximation error ~ l-th degree polynomial regression

$$\left| R_{\mathsf{RF}}(f_d, \boldsymbol{W}) - R_{\mathsf{RF}}(\mathsf{P}_{\le \ell} f_d, \boldsymbol{W}) - \|\mathsf{P}_{>\ell} f_d\|_{L^2}^2 \right| \le \varepsilon \|f_d\|_{L^2} \|\mathsf{P}_{>\ell} f_d\|_{L^2}$$

$$0 \le R_{\mathsf{RF}}(\mathsf{P}_{\le \ell} f_d, \boldsymbol{W}) \le \varepsilon \|\mathsf{P}_{\le \ell} f_d\|_{L^2}^2$$

Approximation error ~$(l+1)$-th degree polynomial regression

$$\left| R_{\mathsf{NT}}(f_d, \boldsymbol{W}) - R_{\mathsf{NT}}(\mathsf{P}_{\le \ell+1} f_d, \boldsymbol{W}) - \|\mathsf{P}_{>\ell+1} f_d\|_{L^2}^2 \right| \le \varepsilon \|f_d\|_{L^2} \|\mathsf{P}_{>\ell+1} f_d\|_{L^2}$$

$$0 \le R_{\mathsf{NT}}(\mathsf{P}_{\le \ell+1} f_d, \boldsymbol{W}) \le \varepsilon \|\mathsf{P}_{\le \ell+1} f_d\|_{L^2}^2$$

Separation between neural nets and their linearization

[GMMM19] [GMMM20] Ghorbani, Mei, Misiakiewicz, Montanari

In this setup, finite-width linearized neural net cannot fit a single neuron. Assuming activation is not polynomial, width $N \sim d^l$, infinite sample size, approximation error is bounded away from 0.

$$f_*(x) = \sigma(\langle w_*, x\rangle)$$

Neural nets seem better at fitting low effective dimension targets: $d_0 \ll d$

They consider a more general scenario where $U \in \mathbb{R}^{d\times d_0}$ is a low-dim projection:

$$f_*(x) = \varphi(U^T x)$$

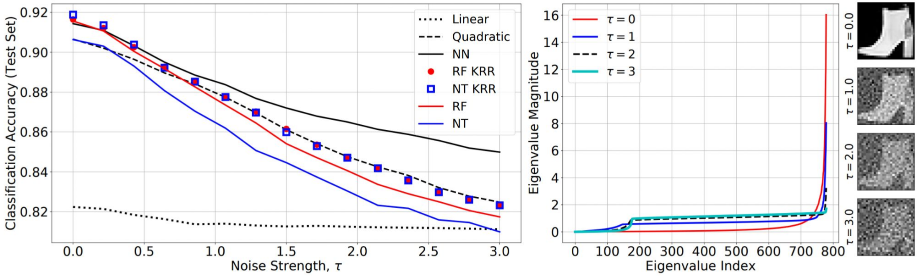
and consider the projection subspace to be the signal; the rest is considered noise. When r = 1, d₀ = 1, we recover the single neuron target setup in [GMMM19].

$$x = U z_0 + U^{\perp} z_1$$

$$z_0 \sim \mathtt{Unif}(\mathbb{S}^{d_0-1}(r\sqrt{d_0}))$$

$$z_1 \sim \mathtt{Unif}(\mathbb{S}^{d-d_0-1}(\sqrt{d_0}))$$

They show that when $d_0 \ll d$, kernel regression performance degrades much faster than neural nets when the input is perturbed by noise, and verify experimentally with Fashion MNIST:



[GMMM19] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint* arXiv:1904.12191, 2019.

[GMMM20] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? *Advances in Neural Information Processing Systems* 33 (2020): 14820-14830.

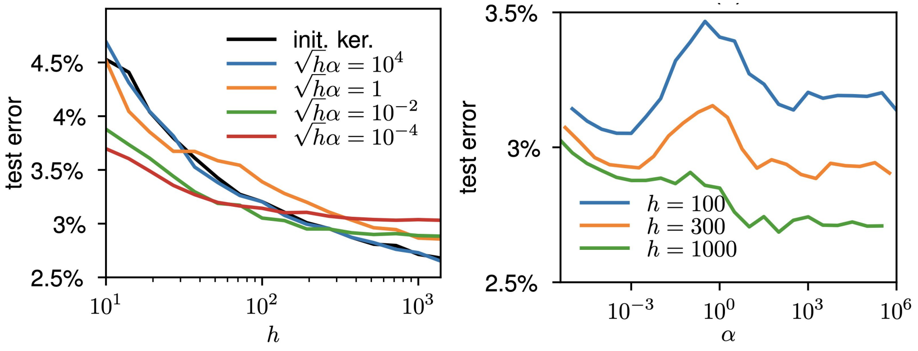
Empirical Findings on the Existence and the Characteristics of the lazy and feature-training regimes **[GSJW20]**

- $F(w,x) = \alpha\big(f(w,x) - f(w_0,x)\big), \quad \text{where } f(w,x) = \frac{1}{\sqrt{h}} W^L z^L$

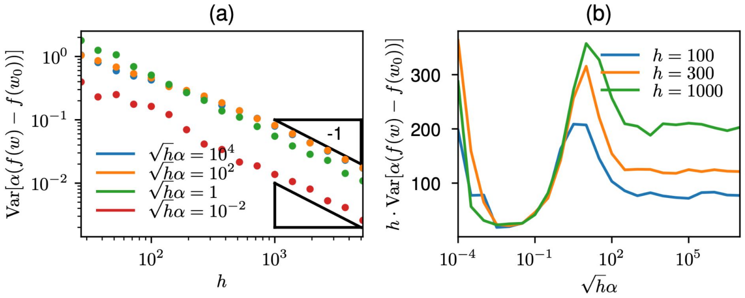
(i) illustrates the existence of two regimes in the overparameterized setting.

(ii) the fluctuations of the output function induced by the initial conditions decreases with h

(i)



(ii)



Setup - NTK Experiments:

- **MNIST** dataset.
- **MLP** and **CNN** architectures with 3-hidden layers + **ReLU** activation function.
- A varying number of network width:
    - **MLP (h)**: 1024, 2048, 4096
    - **CNN (ch)**: 32, 64, 128, 256
    - NTK with w_0 :
        - a larger width for a better approximation of the NTK limit of the lazy regime.
    - NTK with w :
        - a larger width for a better approximation of the mean-field limit of the feature-training regime.
        - weights were trained using **Adam**.
- Each kernel used 30 training samples per digit (300 training samples in total).
- NTK was implemented using **pytorch** and **functorch** libraries.
- Followed the classification approach in **[ADH+19]**, which uses the **argmax** as prediction:

$$f^*(\boldsymbol{x}) = \big(\ker(\boldsymbol{x}, \boldsymbol{x}_1), \ker(\boldsymbol{x}, \boldsymbol{x}_2), \ldots, \ker(\boldsymbol{x}, \boldsymbol{x}_n)\big) \cdot \big(\boldsymbol{H}^*\big)^{-1} \boldsymbol{Y}$$

Results - NTK Experiments:

	MLP (S)	**MLP (M)**	**MLP (L)**	**CNN (S)**	**CNN (M)**	**CNN (L)**	**CNN (XL)**
h/ch	1,024	2,048	4,096	32	64	128	256
N	2,913,290	10,020,874	36,818,954	173,706	384,266	915,978	2,421,770
Lazy	0.7705	0.7851	0.7885	0.5983	0.6299	0.6433	0.6522
Feature	0.9609	0.9618	0.9509	0.7774	0.8187	0.7911	0.7335
Regular Inf.	0.9859	0.9863	0.9861	0.9903	0.9905	0.9885	0.9855

Link to the code.

Conclusion:

- Coupled dynamics in the prelimit and the limit settings
- Separation between neural nets and their linearization
- Empirical findings on the existence and the characteristics of lazy and feature-training regimes

fin. Thank you for your attention!

References:

- **[ADH+19]** Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., & Wang, R. (2019). On exact computation with an infinitely wide neural net. Advances in Neural Information Processing Systems, 32.
- **[AGS08]** Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. Gradient flows: in metric spaces and in the space of probability measures. Springer Science & Business Media, 2008.
- **[CMV+03]** José A Carrillo, Robert J McCann, Cédric Villani, et al. Kinetic equilibration rates for granular media and related equations: entropy dissipation and mass transportation estimates. *Revista Matematica Iberoamericana*, 19(3):971–1018, 2003.
- **[COB19]** Chizat, Lenaic, Edouard Oyallon, and Francis Bach. "On lazy training in differentiable programming." Advances in Neural Information Processing Systems 32 (2019).
- **[GSJW20]** Geiger, M., Spigler, S., Jacot, A., & Wyart, M. (2020). Disentangling feature and lazy training in deep neural networks. Journal of Statistical Mechanics: Theory and Experiment, 2020(11), 113301.
- **[JGH18]** Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31.
- **[JKO98]** Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. SIAM journal on mathematical analysis, 29(1):1–17, 1998.
- **[MMM19]** Mei, S., Misiakiewicz, T., & Montanari, A. (2019, June). Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory* (pp. 2388-2464). PMLR.
- **[MMN18]** Mei, S., Montanari, A., & Nguyen, P. M. (2018). A mean field view of the landscape of two-layer neural networks. Proceedings of the National Academy of Sciences, 115(33), E7665-E7671.
- **[RV18b]** Rotskoff, G. M., & Vanden-Eijnden, E. (2018). Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv preprint arXiv:1805.00915*.
- **[S19]** Mei, S. Mean field theory and tangent kernel theory of neural networks. https://stats385.github.io/assets/lectures/MF_dynamics_Stanford.pdf