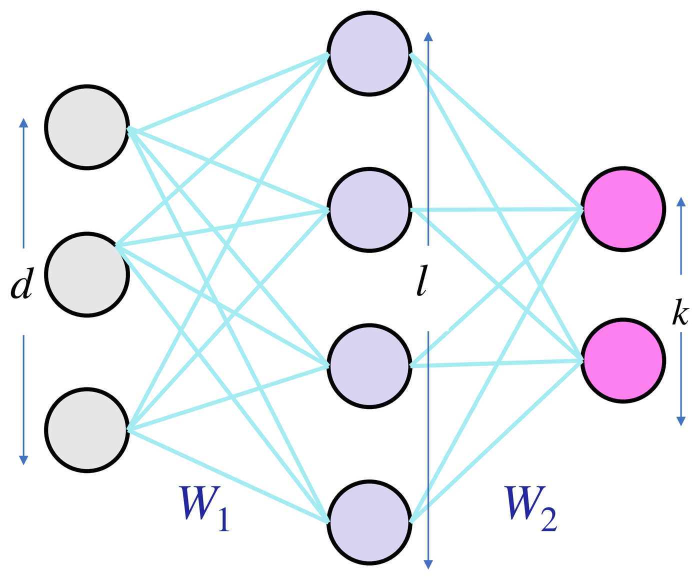
On the spectral bias of two-layer linear networks

Aditya Varre, Maria Vladarean, Loucas Pillaud-Vivien, Nicolas Flammarion @ Theory of Machine Learning Lab (TML), Lausanne

Regression with Linear Network

Setup: The $(x_i)_{i=1}^n$ from $\mathbb{R}^d$ and w.l.o.g assume the labels are generated by $y_i = \mathbf{U}_*^\top x_i \in \mathbb{R}^k$ and $\mathbf{U}_* \in \mathbb{R}^{d\times k}$.



A linear network representing function f

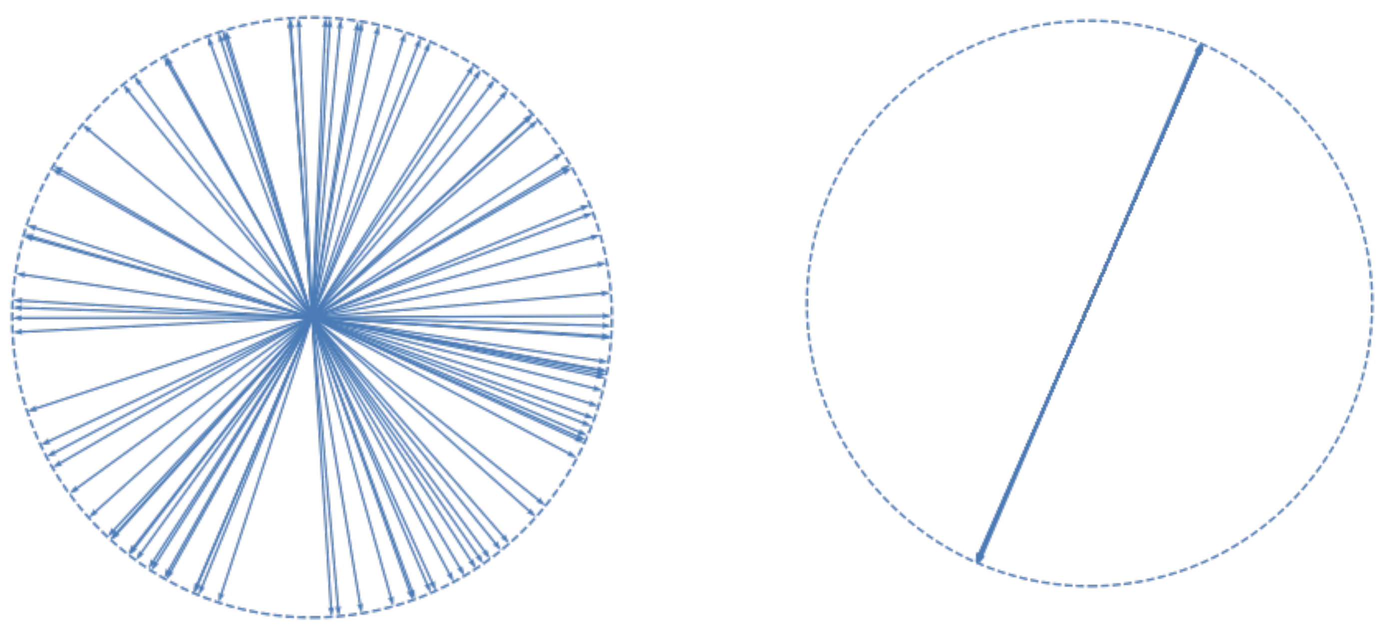
Linear Network: Let hidden layer be $W_1 \in \mathbb{R}^{d\times l}$ and weight layer $W_2 \in \mathbb{R}^{l\times k}$. The network represents the function

$$f(x) = W_2^\top W_1^\top x.$$

Square Loss: $L(W_1, W_2) = \frac{1}{2}\|Y - XW_1W_2\|^2$ which is **non-convex** in W_1, W_2 and X, Y denotes the data in a matrix form.

Dependence on initialization : Consider the problem of scalar regression, i.e., k = 1.

The direction of neurons of W_1 at the end of training.



W_1, W_2 initialized a **large** scale — W_1, W_2 initialized at a **small** scale

Different regimes: For higher scale of initialization, the neurons at convergence point in many directions (**lazy training**). For smaller scale of initialization, the neurons point towards a few feature directions (**feature learning**).

The aim of the paper: To rigorously study the impact of the initialization on the learning dynamics while training linear networks with gradient methods.

Implicit bias of the Parameters

Initialization (I) : $W = \sqrt{2\gamma}P$ where $P \in \mathbb{R}^{d\times l}$ and $PP^\top = I_d$ and $W_2 = 0$. Here γ refers to the **scale** of initialization.

Gradient Flow: We train with gradient flow (GF) on the loss L,

$$\dot{W}_1 = -\nabla_{W_1} L(W_1, W_2) = (X^\top(Y - XW_1W_2))W_2^\top,$$
$$\dot{W}_2 = -\nabla_{W_2} L(W_1, W_2) = W_1^\top(X^\top(Y - XW_1W_2)) .$$

Theorem : When trained with GF and initialized as (I), let $\beta = W_1W_2$,

a) Convergence to the solution: $\lim_{t\to\infty} L\left(W_1(t), W_2(t)\right) \to 0$

b) Implicit bias:

- ⋆ β : $\beta_\infty = \lim_{t\to\infty} \beta(t) = \operatorname{argmin}_{X\beta \in Y} \|\beta\|_2 = \beta_*$
- ⋆ W_1, W_2: $W_1^\infty, W_2^\infty = \operatorname{argmin}_{XW_1W_2 = Y} \|W_1\|_F^2 + \|W_2\|_F^2 - 2\gamma \log\det W_1^\top W_1$

Comments:

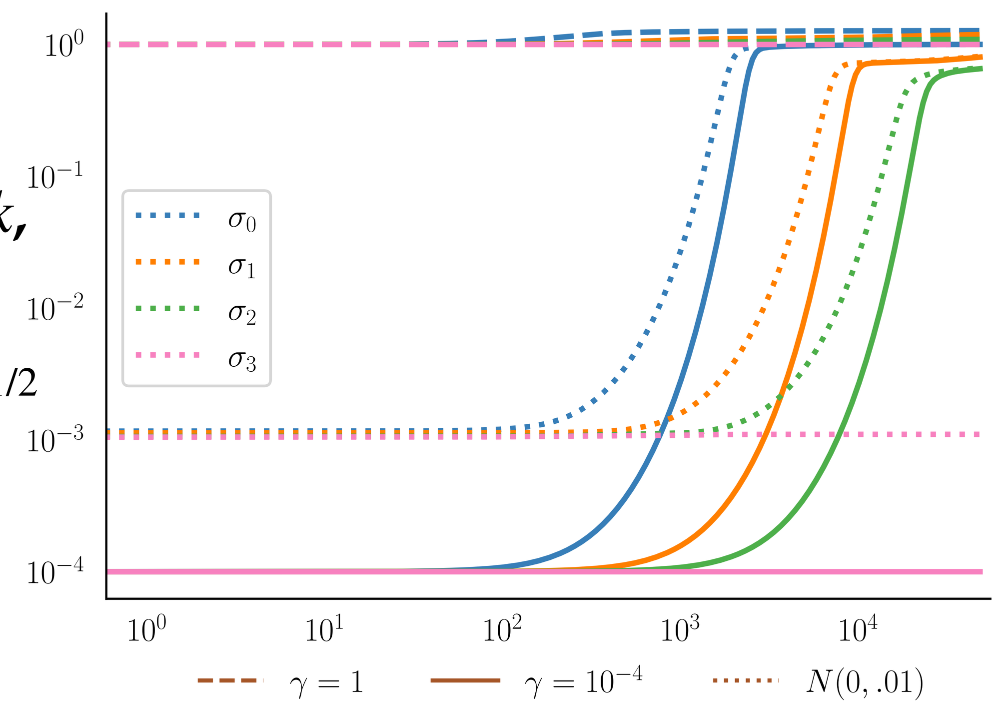
- when $\gamma \to 0$, GF converges to **min-norm interpolator** ensuring that $rank(W_1^\infty) = k$.
- $\log\det(\,.\,)$ is a smooth approximation of rank and intuitively larger γ pushes to large rank.

Corollary: We have the following expressions of final **singular values.** For $1 \le i \le k$,

$$\sigma_i\left(W_1^\infty\right) = \left(\sqrt{\sigma_i(\beta_*) + \gamma^2} + \gamma\right)^{1/2},$$

and for $k < i \le d$,

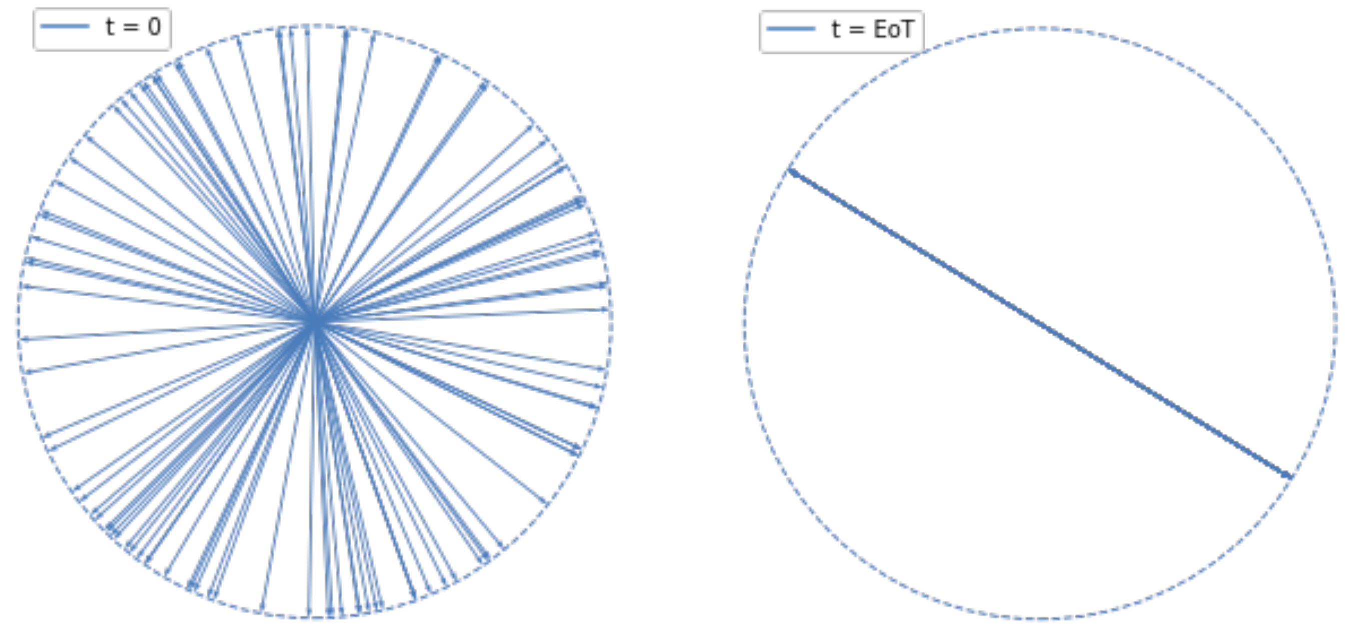
$$\sigma_i\left(W_1^\infty\right) = (2\gamma)^{1/2}$$



Evolution of singular values along the train when trained with different scales and init. shape.

GF converges to a **low-rank** W_1 at small γ (**rich** regime), whereas converges to a **high-rank** W_1 at large γ (**lazy** regime).

Going beyond small γ- noisy dynamics



Evolution of direction of neurons when trained with LNGD, shows **neuron alignment** at even **large** init. scale

LNGD: Let $\mathbf{B}_t$ denote a n-dimensional brownian motion,

$$dW = a(X^\top(Y - XW^\top a))^\top dt + a(X^\top(d\mathbf{B}_t))^\top,$$
$$da = W(X^\top(Y - XW^\top a))dt + W(X^\top d\mathbf{B}_t) .$$

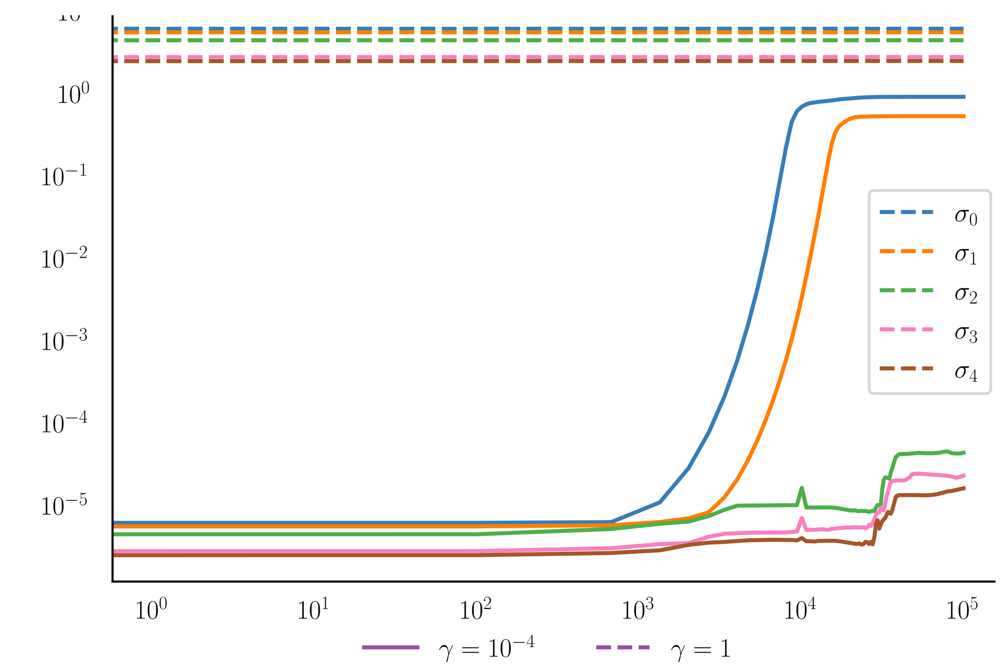
Non-linear ReLU activation

A simplified setup: Let u, v be any two orthogonal directions. We sample inputs $(x_i)_{i=1}^n$ from standard normal distribution, i.e., $x_i \sim \mathcal{N}(0, I)$ and we generate labels by a teacher network,

$$y_i = \sigma(u^\top x_i) + \sigma(v^\top x_i) .$$

We train a student network with 20 neurons on this data.

Small initialization is effective in recovering **features** even with ReLU activation.



Evolution of singular values along the train when trained with different scales for the ReLU network

References:

Chizat et.al. On Lazy Training in Differentiable Programming, NeurIPS 2019

Woodworth et. al. Kernel and rich regimes in overparametrized models, COLT 2020.

Blanc et. al. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process, COLT 2020.