What makes NN good learners ?

- 2L NN

RQ: Why are NNs so good at learning from data?

→ NN adapt to STRUCTURE in the DATA

SPOILER Feature learning ↔ Spectral statistics of the weights

1) Motivation and Settings

Supervised Learning

$$D = \{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R} : i \in [n]\}$$
$$= \{1, \dots n\}\}$$

Define some data distribution

$$(x, y_i) \sim P_{x,y}$$

Goal. Learn $F : \mathbb{R}^d \to \mathbb{R}$ "generalizes well"

$$x \longmapsto f(x) = \hat{y}$$

$$\min_F \mathbb{E}\left[\ell(y, f(x))\right]$$

2) Problems: I) Don't know the data distribution P_{XY}

II) How to optimise over $\{ f : \mathbb{R}^d \to \mathbb{R} \}$?

ERM Empirical Risk Minimization

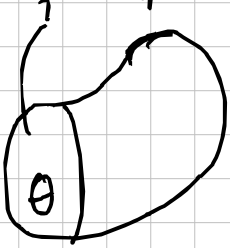
$$\min_{f_\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y, f_\theta(x_i))$$

↓

TREATS Generalization

↓

$\mathcal{H}$ Hypothesis class of Function (OPTIMIZING HERE)



QUESTION: How big n do i need?

Thm (Tsybakov 2008)

Assume $y := f_*(x_i) + \varepsilon_i$ $\quad \mathbb{E}[\varepsilon_i] = 0$

$\mathbb{E}[\varepsilon_i^*] = 0 \ (\alpha$

→ f^* Lipschitziana

→ $x_i \sim \text{UNIF}([0,1]^d)$ Soft assumptions

Then: $\inf_{\hat{f}} \sup_{f^*} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] \gtrsim n^{-\frac{2}{2+d}}$

$R(\hat{f}) < \delta \gtrsim \delta^{\frac{d+2}{2}}$

CURSE OF DIMENSIONALITY

Moral: We need assumptions

→ Data $p_{x,y}$
→ Hypothesis (Architecture)
→ Algorithms

} TYPICAL CASE ANALYSIS

Real question: How much data to learn a NN with SGD/GD

~2018 NTK Theory / Lazy Training (Jacob et al. 2018)

ERM for NNs $\Longleftrightarrow$ Kernel Methods

~2018-2022 Limitation of Kernels in HD

Thm (Mei, Montanari 2022)

Isotropic data $p_x \sim N(0, \mathbb{1}, S^{d-1}, [0,1]^d)$ you have $n = \Theta(d^k)$, $k > 0$. Then a kernel can only learn a polynomial of degree k.

Beyond Kernels — Can we learn better than a kernel?

Focus on 2L NN

$$f_\theta(x) = \sum_{j=1}^{p_1} a_j \sigma\left(\langle w_j, x\rangle\right)$$

$w_j^T x$

Smooth Functio

Data: [smooth index model]

$y_i = f_*(x_i) + \varepsilon$ $\qquad f_*(x) = g(\langle w_1^* x\rangle, \dots \langle w_k^* x\rangle)$

$= g(W^* x)$ (Nonlinear)

where $\langle W_k^* W_k\rangle = \delta_{kk}$ (ORTHOGONAL)

$g : \mathbb{R} \to \mathbb{R}$

MANIFOLD HYPOTHESIS → we hope $r \ll d$

How much data do i need ??

If i only learn $W^* x$, a kernel would be enough

Algorithm: 2 step GD training

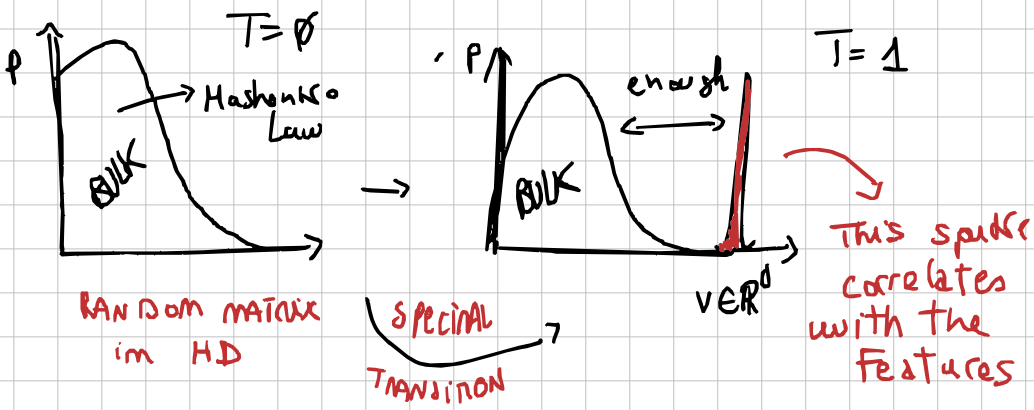
1) Init (a^0, W^0)
2) Train W For T steps with fix a^0.
3) Optimize over a for fixed W^T

~2023 (Ba et al., Bardi et. al.)

$W \in \mathbb{R}^{P \times d}$ Singular values of $\{s_\lambda\}$

$$P(\lambda) = \frac{1}{P} \sum_{i=1}^{P} \delta(\lambda - s_\lambda) \qquad s_\lambda \ll 1$$

=> You get Marchenko Pastur Law (RANDOM MATRIX Follow This)



$$V \approx \sum_{k=1} \mu_k W_k^* \quad \text{Hermite Polynomial}$$

Thm. (Damdi et al. 2023; Cui et al. 2024)

After 1 step $n = O(d)$, f_θ can learn any non-linear component of f^*

$$f^*(x) \approx \varphi(\langle v, x \rangle) + \tilde{\varepsilon}$$

cf Kernel $f^*(x) \approx \langle v, x \rangle + \tilde{\varepsilon}$

[There is also an asymptote in that regime]

3) Quadratic NN

Question: Can we solve ERM problem for a 2L NN? I need to simplify more!!

Assume: $f_\theta(x)$ is quadratic (1L NN)

$$f_\theta(x) = \sum_{j=1}^{p} \left(\langle w_j, x \rangle^2 - \|w_j\|_2^2 \right)$$

Data:

$$f_*(x) = \sum_{k=1}^{r} a_k^* \left(\langle w_k^*, x \rangle - \|w_k^*\|_2^2 \right)$$

$x \sim N(0, \mathbb{1})$

Note: $f_\theta(x) = \sum_{j=1}^{t} w_j^T (xx^T - \mathbb{1}) w_j$ — convex in S

$= T_c(SG)$ where $S = W^T W \in \mathbb{R}^{d \times d}$, $G = xx^T - I$

$= \langle S, G \rangle_F$ → Frobenius Norm

Similarly $f_*(x) = T_c(S_* G)$ where $S_* = W^{*T} W^* \in \mathbb{R}^{d \times d}$

$\therefore$ ERM $\approx \min_{S \succeq 0} \frac{1}{n} \sum \left(\mathrm{Tr}(S - S_*) G_i + \varepsilon \right)^2 + \lambda \mathrm{Tr}(S)$

$\mathrm{Tr}(S) = \sum_j \lambda$

REGULARIZATION [NUCLEAR NORM] = L_1 spectrum

induces SPARSITY differently than from original space

~ Results 2025 (Erba et al. 2025)

Minimizer when $d \to \infty$

$$\hat{S} = \text{soft threshold}\left(S_* + \delta G\right)$$

(δG: Gaussian)