Learning in high dimension: inter-disciplinary insights

Aurélien Garivier

March 1st, 2021

Table of contents

1. Approximation: a Separation Result

2. Convergence of Wide Depth-2 Networks: Mean-Field Insight

3. Statistical Physics View on Generalization

What we want to do: Predictions

Phenomenon: observations $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in a product of measurables spaces $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}^q$.

Goal: predict y from x. Prediction error measure by *loss* $\ell(\hat{y}, y) = \|\hat{y} - y\|^2/2$ typically.

Statistical hypothesis: there exists $F : \mathcal{X} \times \Omega \to \mathcal{Y}$ such that the observations are distributed as (X, Y) where X has distribution $\mathbb{P}_X$ and $Y = F(X, \omega)$. Typically, $Y = f(X) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Examples:

- classification (OCR, image recognition, text classification, etc.)
- regression (response to a drug, weather or stock price forecast, etc.)

Target: best possible guess of Y given X: $f(X) = \mathbb{E}[Y|X]$.

Machine Learning

Mechanism of f is complex or hidden. Access to f only thru **examples** i.e. a sample $S_n = \big((X_1, Y_1), \ldots, (X_n, Y_n)\big)$ of random pairs

Learning algorithm $\mathcal{A}_n : S_n \mapsto \hat{f}_n \quad$ where $\hat{f}_n \in \mathcal{F} \subset \mathcal{Y}^{\mathcal{X}} \subset (\mathbb{R}^q)^{\mathbb{R}^p}$

$\mathcal{F}$ = **hypothesis class** = model. Example: linear regression

$$\mathcal{F} = \left\{ f_\theta : x \mapsto \left(\theta_{i,0} + \sum_{j=1}^{p} \theta_{i,j} x_j\right)_{1 \le i \le q} : \theta \in \mathcal{M}_{q,1+p}(\mathbb{R}) \right\}$$

Quality of prediction $\hat{y}$: **loss function** $\ell : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}_+$ e.g. $\ell(\hat{y}, y) = \frac{(\hat{y}-y)^2}{2}$

Quality of hypothesis $f \in \mathcal{F}$: **generalization error** = average loss

$$L(f) = \mathbb{E}\big[\ell(f(X), Y)\big]$$

expectation is on new observation (X,Y)

Quality of the learning algorithm $\mathcal{A}$: **risk** = average average loss

$$R_n(\mathcal{A}_n) = \mathbb{E}\left[L(\hat{f}_n)\right]$$

expectation is on sample S_n

Empirical Risk Minimization

Learning = how to find the best possible $f \in \mathcal{F}$?

→ Minimize the **empirical loss = training error**

$$L_n(f) = \frac{1}{n}\sum_{k=1}^{n} \ell\big(f(X_k), Y_k\big) \qquad \text{average loss on the sample}$$

= unbiased estimator of the generalization error $L(f)$

Empirical Risk Minimizer: $\hat{f}_n \in \arg\min_{f \in \mathcal{F}} L_n(f)$

Example: linear regression with quadratic loss (dates back at least to Gauss) $\hat{f}_n = f_{\hat{\theta}_n}$ where $\hat{\theta}_n^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, with

$$\mathbf{X} = \begin{pmatrix} 1 & X_1^1 & \dots & X_1^p \\ & \dots & & \\ 1 & X_n^1 & \dots & X_n^p \end{pmatrix} \text{ and } \mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

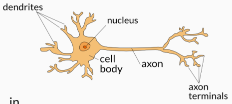
Regression by polynomials of degrees $1, 2, \dots, n-1$ → more parameters is not necessarily better, bias / variance tradeoff, Structural Risk Minimization (penalize empirical risk by model complexity)

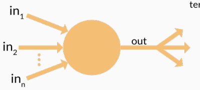
Feedforward Neural Networks: Mimicking Brains?

Neuron: $x \mapsto \sigma\big(\langle w, x\rangle + b\big)$ with

- parameter $w \in \mathbb{R}^p, b \in R$
- (non-linear) activation function $\sigma : \mathbb{R} \to \mathbb{R}$

  typically $\sigma(x) = \frac{1}{1+\exp(-x)}$ or $\sigma(x) = \max(x, 0)$ called ReLU





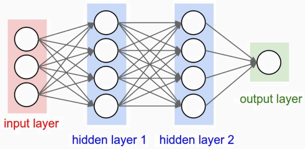
Layer: $x \mapsto \boldsymbol{\sigma}\big(Mx + \mathbf{b}\big)$ with

- parameter $M \in M_{q,p}(\mathbb{R}), \mathbf{b} \in \mathbb{R}^q$
- component-wise activation function $\boldsymbol{\sigma} = \sigma^{\otimes q}$

Network: composition of layers $f_\theta = \boldsymbol{\sigma}_D \circ T_D \circ \cdots \circ \boldsymbol{\sigma}_1 \circ T_1$ with

- architecture $A = \big(D, (p_1, \ldots, p_{D-1})\big)$
- $x_0 = x, \quad x_d = \boldsymbol{\sigma}_d\big(T_d x_{d-1}\big) \in \mathbb{R}^{p_d}$
- $T_d x = M_d x + \mathbf{b}_d$
- parameter $\theta = (M_1, \mathbf{b}_1, \ldots, \ldots, M_D, \mathbf{b}_D)$

  $\theta \in \Theta_A = \prod_{d=1}^{D} \mathcal{M}_{p_{d-1}, p_d}(\mathbb{R}) \times \mathbb{R}^{p_d}$
- <u>*depth*</u> D (⚠st. nb layers), <u>*width*</u> $\max_{1 \le d \le D} p_d$



How to learn with feedforward neural networks?

Choose architecture $A = \big[D, (p_1, \dots, p_{D-1})\big]$

- depth D?
- what architectures are good if f has some with given properties?
- activation function? sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$ or ReLU $\sigma(x) = \max(x, 0)$

→ approximation theory

Learn = find the good coefficients using S_n

- Empirical Risk Minimization: $\hat{f}_n$ solution of

$$\min_{\substack{T_k \in \mathcal{M}_{p_d, 1+p_{d-1}}(\mathbb{R}) \\ 1 \leq d \leq D}} \quad \frac{1}{n} \sum_{k=1}^{n} \ell\big(\boldsymbol{\sigma}_D \circ T_D \circ \cdots \circ \boldsymbol{\sigma}_1 \circ T_1(X_k), Y_k\big)$$

- non convex, high-dimensional optimization problem
- but gradient can be computed by **back-propagation**

→ does gradient descent work?

Apply $\hat{f}_n$ to new data (X, Y)

- how to bound the generalization error $L(\hat{f}_n)$?
- should we regularize = penalize large coefficients?

→ no overfitting?

→ How to explain the huge empirical success of deep learning?

Approximation: a Separation Result

Lipschitz function approximation

Every Lipschitz function can be ε-approximated by a poly-sized depth-2 NN:

- $\sigma(x) = \max\{0, x\}$ is the ReLU activation function
- $f : [-R, R] \to \mathbb{R}$ is an L-Lipschitz function
- There is a function (implemented by a depth-2 neural network)

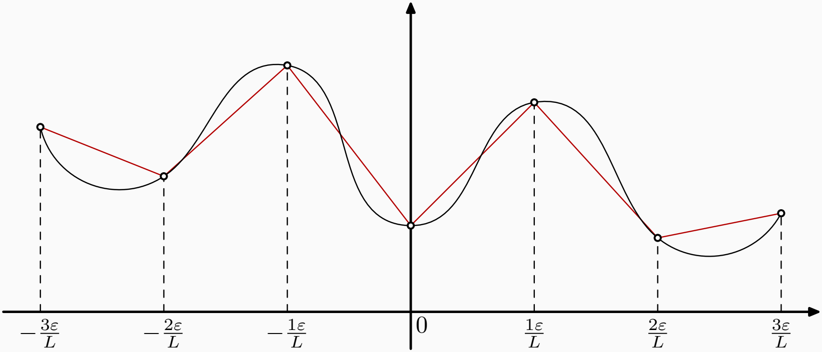
$$N_2(x) = f(0) + \sum_{i=1}^{m} \gamma_i \sigma(\alpha_i x + \beta_i)$$

- $\|f - N_2\|_\infty \leq \varepsilon$
- N_2 is L-Lipschitz on all $\mathbb{R}$
- Width bounded as $m \leq \frac{2RL}{\varepsilon}$
- $\alpha_i \in \{-1, 1\}$
- $|\beta_i| \leq R$
- $|\gamma_i| \leq 2L$

[Cybenko 1989] [Hornik et al. 1989] [Funahashi 1989] - N_2 is a universal approximator

Lipschitz function approximation - proof

$$N_2(x) = f(0) + \sum_{i=1}^{m} \gamma_i \sigma(\alpha_i x + \beta_i)$$



Lipschitz function approximation - proof

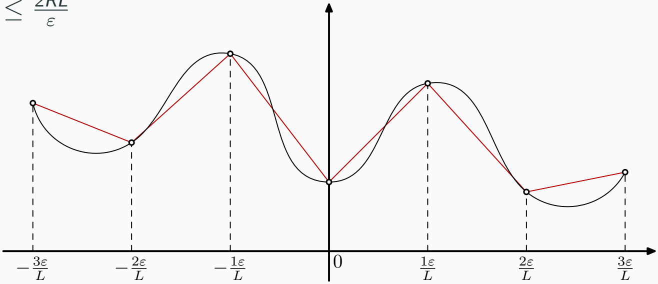
For every x, x_1, $x_2 \in \left\langle \frac{i\varepsilon}{L}, \frac{(i+1)\varepsilon}{L} \right\rangle$

$$|f(x_1) - f(x_2)| \leq L|x_1 - x_2| \leq L\frac{\varepsilon}{L} = \varepsilon$$

Therefore we have

$$\left.\begin{aligned} |N_2(\tfrac{i\varepsilon}{L}) - f(x)| \leq \varepsilon \\ |N_2(\tfrac{(i+1)\varepsilon}{L}) - f(x)| \leq \varepsilon \end{aligned}\right\} |N_2(x) - f(x)| \leq \varepsilon$$

And $m \leq \frac{2RL}{\varepsilon}$

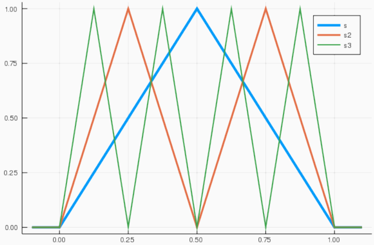


Why is Depth important? Sawteeth Function

$$\text{Let } s(x) = \begin{cases} 2x & \text{if } 0 \leq x \leq \frac{1}{2} \\ 2 - 2x & \text{if } \frac{1}{2} \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$= 2r(x) - 4r\left(x - \frac{1}{2}\right) + 2r(x-1)$$

and for all $m \geq 1$ let $s_m = \underbrace{s \circ \cdots \circ s}_{m \text{ times}}$



Lemma

For all $m \geq 1$, all $k \in \{0, \ldots, 2^{m-1} - 1\}$ and all $t \in [0, 1]$,

$$s_m\left(\frac{k+t}{2^{m-1}}\right) = \begin{cases} 2t & \text{if } t \leq \frac{1}{2} \\ 2 - 2t & \text{if } t \geq \frac{1}{2} \end{cases}$$

Why is Depth important? Square Function

Let $g(x) = x^2$, and for $m \geq 0$ let $g_m(x)$ be such that $\forall k \in \{0, \ldots, 2^m\}$:

- $g_m\left(\frac{k}{2^m}\right) = g\left(\frac{k}{2^m}\right)$
- g_m is linear on $\left[\frac{k}{2^m}, \frac{k+1}{2^m}\right]$

Lemma

For all $k \in \{0, \ldots, 2^m - 1\}$ and all $t \in [0, 1]$,

$$g_m\left(\frac{k+t}{2^m}\right) - g\left(\frac{k+t}{2^m}\right) = \frac{t(1-t)}{4^m}$$

In particular, $\|g - g_m\|_\infty = \frac{1}{4^{m+1}}$ and for all $m \geq 2$,

$$g_m = g_{m-1} - \frac{1}{4^m}\, s_m = id - \sum_{j=1}^{m} \frac{1}{4^j}\, s_j$$

Corollary

For every $\epsilon > 0$, there exists a neural network f of depth $\lceil \log_4(1/\epsilon) \rceil$, width 3 and coefficients in $[-4, 2]$ such that $\|f - g\|_\infty \leq \epsilon$ on $[0, 1]$

Why is Depth important? Square Function

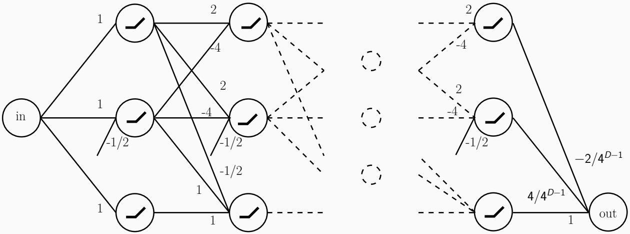
Lemma

$\|g - g_m\|_\infty = \frac{1}{4^{m+1}}$ and for all $m \geq 2$,

$$g_m = g_{m-1} - \frac{1}{4^m}\, s_m = id - \sum_{j=1}^{m} \frac{1}{4^j}\, s_j$$

Corollary

For every $\epsilon > 0$, there exists a neural network f of depth $\lceil \log_4(1/\epsilon) \rceil$, width 3 and coefficients in $[-4, 2]$ such that $\|f - g\|_\infty \leq \epsilon$ on $[0, 1]$



$x_0 = x$ $x_1 = x$ $x_2 = x - \frac{s(x)}{4}$ $x_D = x - \frac{s(x)}{4} - \cdots - \frac{s_{D-1}(x)}{4^{D-1}}$

Outline

Approximation: a Separation Result

A separation result

Depth-2 neural network

Convergence of Wide Depth-2 Networks: Mean-Field Insight

Statistical Physics View on Generalization

Statistical Learning: Recap in a Nutshell

Linear Models

Beyond Linear Models

Exponential Depth Separation

A. Daniely proved that there is a function $F : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to \mathbb{R}$ such that

- there exist a poly(d)-sized depth-3 network N_3 s.t.

$$\|N_3 - F\|_{L^2(\mathbb{S}^{d-1} \times \mathbb{S}^{d-1})} \leq \varepsilon$$

- for every poly(d)-sized depth-2 neural network N_2

$$\|N_2 - F\|_{L^2(\mathbb{S}^{d-1} \times \mathbb{S}^{d-1})} > \varepsilon$$

He showed this for an **inner product functions** i.e.

$$F(\mathbf{x}, \mathbf{x}') = f(\langle \mathbf{x}, \mathbf{x}' \rangle)$$

where $f : [-1, 1] \to \mathbb{R}$.

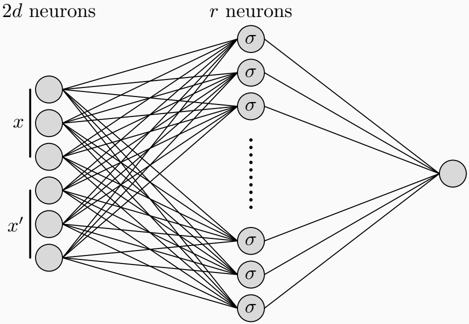
[Martens et al. 2013] [Eldan and Shamir 2016] - similar results

Depth-2 σ-network

Function $N : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to \mathbb{R}$ is implementing a depth-2 σ-network of width r and weights bounded by B if

$$N(\mathbf{x}, \mathbf{x}') = w_2^\top \sigma(W_1 \mathbf{x} + W_1' \mathbf{x}' + b1) + b_2$$

$W_1,\ W_1' \in [-B,\ B]^{r \times d},\ w_2,\ b_1 \in [-B,\ B]^r,\ b_2 \in [-B,\ B]$.

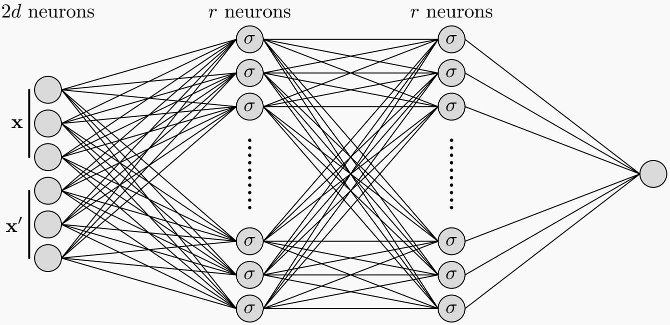


Depth-3 σ-network

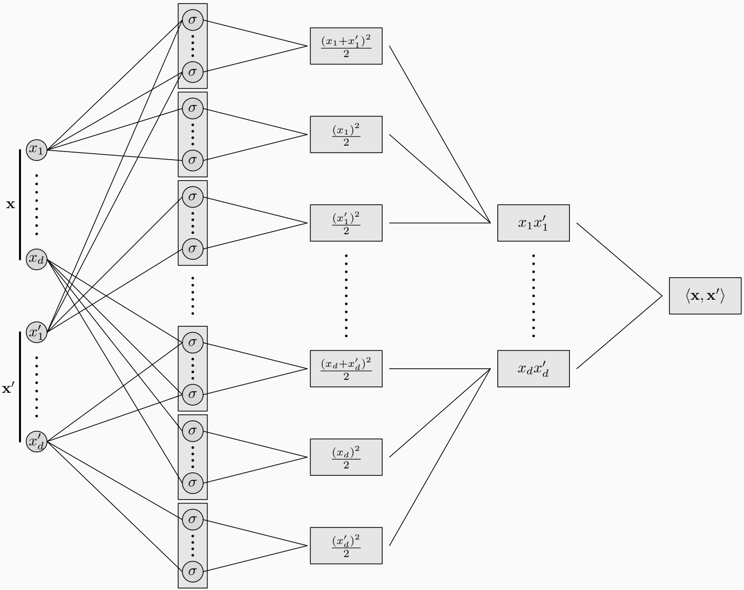
Function $N : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to \mathbb{R}$ is implementing a depth-3 σ-network of width r and weights bounded by B if

$$N(\mathbf{x}, \mathbf{x}') = w_3^\top \sigma(W_2 \sigma(W_1 \mathbf{x} + W_1' \mathbf{x}' + b1) + b_2) + b_3$$

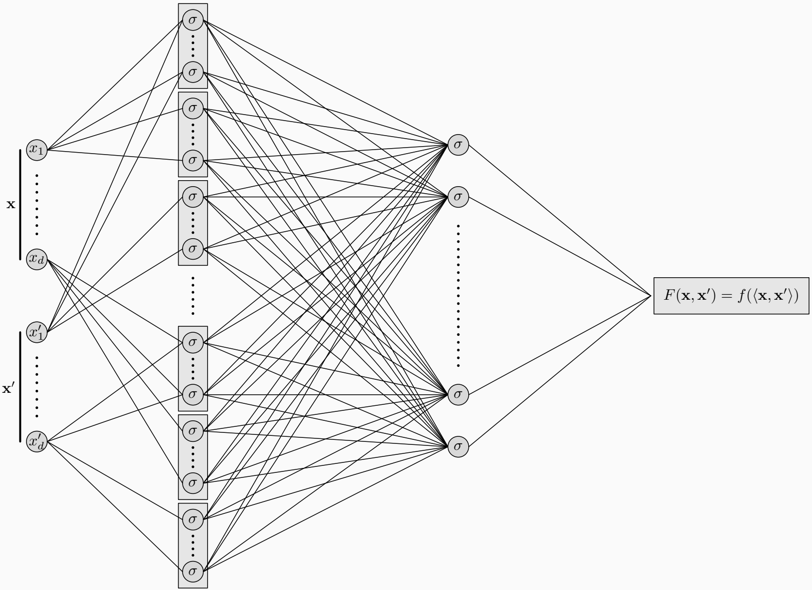
$W_1,\ W_1' \in [-B,\ B]^{r \times d},\ W_2 \in [-B,\ B]^{r \times r},\ b_1,\ b_2 \in [-B,\ B]^r,\ b_3 \in [-B,\ B].$



Inner product approximation



Inner product function approximation



Inner product function approximation

Inner product approximated by N_i

- Approximation precision: $\frac{\varepsilon}{2L}$
- Width of approximation N_i: $\frac{16d^2L}{\varepsilon}$

L-Lipschitz function f approximated by N_f

- Approximation precision: $\frac{\varepsilon}{2}$
- Width of approximation N_f: $\frac{4L}{\varepsilon}$

Inner product function approximated by $N_F = N_f \circ N_i$.

- Width of approximation N_F: $\frac{16d^2L}{\varepsilon}$
- Approximation precision:

$$
\begin{aligned}
|N_F(\mathbf{x}, \mathbf{x}') - F(\mathbf{x}, \mathbf{x}')| &= |N_f(N_i(\mathbf{x}, \mathbf{x}')) - f(\langle \mathbf{x}, \mathbf{x}' \rangle)| \\
&\leq |N_f(N_i(\mathbf{x}, \mathbf{x}')) - N_f(\langle \mathbf{x}, \mathbf{x}' \rangle)| + |N_f(\langle \mathbf{x}, \mathbf{x}' \rangle) - f(\langle \mathbf{x}, \mathbf{x}' \rangle)| \\
&\leq L|N_i(\mathbf{x}, \mathbf{x}') - \langle \mathbf{x}, \mathbf{x}' \rangle| + \frac{\varepsilon}{2} \leq L\frac{\varepsilon}{2L} + \frac{\varepsilon}{2} = \varepsilon
\end{aligned}
$$

Example

Highly oscillating inner product function:

$$F(\mathbf{x}, \mathbf{x}') = f(\langle \mathbf{x}, \mathbf{x}' \rangle) = \sin(\pi d^3 \langle \mathbf{x}, \mathbf{x}' \rangle)$$

$$\sin(x) \text{ is 1-Lipschitz} \quad \Longrightarrow \quad \sin(\pi d^3 x) \text{ is } (\pi d^3)\text{-Lipschitz}$$

We can ε-approximate F by a depth-3 neural network of width at most

$$\frac{16 d^2 L}{\varepsilon} = \frac{16 \pi d^5}{\varepsilon}$$

Outline

Approximation: a Separation Result

A separation result

Depth-2 neural network

Convergence of Wide Depth-2 Networks: Mean-Field Insight

Statistical Physics View on Generalization

Statistical Learning: Recap in a Nutshell

Linear Models

Beyond Linear Models

Legendre polynomials

$P_0(x) = 1,\ P_1(x) = x$

$$P_n(x) = \frac{2n+d-4}{n+d-3} x P_{n-1}(x) - \frac{n-1}{n+d-3} P_{n-2}(x)$$

Sequence $\{\sqrt{N_{d,n}} P_n\}_{n \geq 0}$ is **orthonormal basis** of $L^2(\mu_d)$ where

$$N_{n,d} = \binom{d+n-1}{d-1} - \binom{d+n-3}{d-1}$$

and μ_d is defined by pushing forward the uniform measure on $\mathbb{S}^{d-1}$ using function $\mathbf{x} \to x_1$

$$d\mu_d(x) = \frac{\Gamma(\frac{d}{2})}{\sqrt{\pi}\Gamma(\frac{d-1}{2})}(1-x^2)^{\frac{d-3}{2}} dx$$

Legendre polynomials

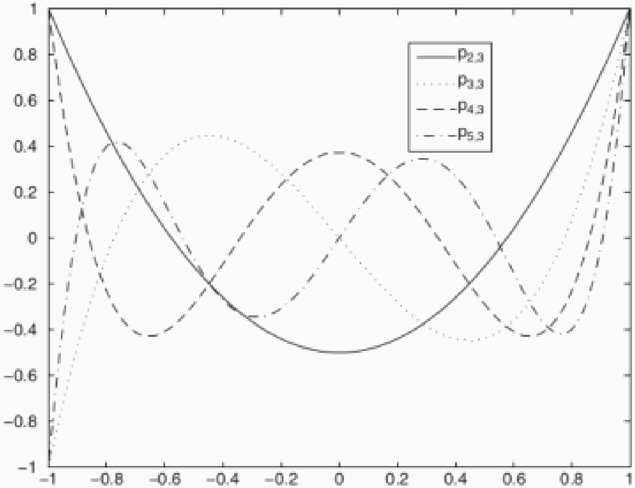


Fig. 2.2 Legendre polynomials for dimension 3

Inner product functions

Denote

$$h_n(\mathbf{x}, \mathbf{x}') = \sqrt{N_{d,n}} P_n(\langle \mathbf{x}, \mathbf{x}' \rangle)$$

Then

$\{h_n\}_{n \geq 0}$ is a **basis** of the space of inner product functions

Let $F(\mathbf{x}, \mathbf{x}') = f(\langle \mathbf{x}, \mathbf{x}' \rangle)$ be any inner product function. Then

$$F(\mathbf{x}, \mathbf{x}') = \sum_{i=0}^{\infty} \alpha_i h_i(\mathbf{x}, \mathbf{x}')$$

Separable functions

Function $g(\mathbf{x}, \mathbf{x}')$ is **(v, v')-separable function** if

$$g(\mathbf{x}, \mathbf{x}') = \psi(\langle \mathbf{v}, \mathbf{x}\rangle, \langle \mathbf{v}', \mathbf{x}'\rangle)$$

Denote

$$L_n^{\mathbf{x}}(\mathbf{x}') = h_n(\mathbf{x}, \mathbf{x}') = \sqrt{N_{d,n}} P_n(\langle \mathbf{x}, \mathbf{x}'\rangle)$$

$\left\{ L_i^{\mathbf{v}}(\mathbf{x}) L_j^{\mathbf{v}'}(\mathbf{x}') \right\}_{i,j\geq 0}$ – **basis** of $(\mathbf{v}, \mathbf{v}')$–separable functions

Any $(\mathbf{v}, \mathbf{v}')$–separable function $g(\mathbf{x}, \mathbf{x}')$ can be written as

$$g(\mathbf{x}, \mathbf{x}') = \sum_{i,j\geq 0} \beta_{i,j} L_i^{\mathbf{v}}(\mathbf{x}) L_j^{\mathbf{v}'}(\mathbf{x}')$$

Note: neuron $\sigma(\langle \mathbf{v}, \mathbf{x}\rangle + \langle \mathbf{v}', \mathbf{x}'\rangle + \mathbf{b})$ is a separable function

Main result

Theorem

Let $F\colon \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to \mathbb{R}$ be an inner product function and let $g_1, g_2, ..., g_r$ be separable functions. Then we have

$$\left\| F - \sum_{j=1}^{r} g_j \right\|^2 = \|\mathcal{P}_n F\| \left(\|\mathcal{P}_n F\| - \frac{2\sum_{j=1}^{r} \|g_j\|}{\sqrt{N_{d,\,n}}} \right).$$

where $\mathcal{P}_n$ is a projection operator such that

$$\mathcal{P}_n \left(\sum_{i=0}^{\infty} \alpha_i h_i \right) = \sum_{i=n}^{\infty} \alpha_i h_i$$

Note: whenever F has heavy Legendre tail, N_2 needs to be wide

Main result - proof

$$\|F - N_2\|^2 = \sum_{i=0}^{\infty}\left\|\alpha_i h_i - \sum_{j=1}^{r}\beta_i^j L_i^{\mathbf{v}_j}\otimes L_i^{\mathbf{v}'_j}\right\|^2 \geq \sum_{i=n}^{\infty}\left\|\alpha_i h_i - \sum_{j=1}^{r}\beta_i^j L_i^{\mathbf{v}_j}\otimes L_i^{\mathbf{v}'_j}\right\|^2$$

$$\geq \sum_{i=n}^{\infty}\alpha_i^2 - 2\sum_{i=n}^{\infty}\sum_{j=1}^{r}\langle \alpha_i h_i,\, \beta_i^j L_i^{\mathbf{v}_j}\otimes L_i^{\mathbf{v}'_j}\rangle$$

$$= \|\mathcal{P}_n F\|^2 - 2\sum_{i=n}^{\infty}\sum_{j=1}^{r}\frac{\beta_i^j \alpha_i P_i(\langle \mathbf{v}_j, \mathbf{v}'_j\rangle)}{\sqrt{N_{d,i}}}$$

$$\geq \|\mathcal{P}_n F\|^2 - 2\sum_{j=1}^{r}\sum_{i=n}^{\infty}\frac{|\beta_i^j||\alpha_i|}{\sqrt{N_{d,n}}}$$

$$\geq \|\mathcal{P}_n F\|^2 - 2\sum_{j=1}^{r}\frac{1}{\sqrt{N_{d,n}}}\sqrt{\sum_{i=n}^{\infty}|\alpha_i|^2}\sqrt{\sum_{i=n}^{\infty}|\beta_i^j|^2}$$

$$\geq \|\mathcal{P}_n F\|^2 - \frac{2\,\|\mathcal{P}_n F\|\sum_{j=1}^{r}\|g_j\|}{\sqrt{N_{d,n}}}$$

Example

We are looking for a function that can not be well approximated by a low degree polynomial. For example:

$$\sin(\pi\sqrt{d}mx)$$

Lemma

Let $s_{d,m}(x) = \sin\left(\pi\sqrt{d}mx\right)$. Then for any $d > d_0$ and for any degree k polynomial p we have

$$\int_{-1}^{1}(s_{d,m}(x) - p(x))^2 d\mu(x) \geq \frac{m-k}{4e\pi m}$$

Example

Proof of the Lemma

For large enough d and $|x| \leq \frac{1}{\sqrt{d}}$ we have

$$1 - x^2 \geq e^{-2x^2} \geq e^{-\frac{2}{d}} \quad \Longrightarrow \quad (1-x^2)^{\frac{d-3}{2}} \geq e^{-\frac{d-3}{d}} \geq e^{-1}$$

This, together with the fact that $\Gamma(\frac{d}{2})/\Gamma(\frac{d-1}{2}) \approx \sqrt{\frac{d}{2}}$, gives us

$$d\mu(x) = \frac{\Gamma(\frac{d}{2})}{\sqrt{\pi}\Gamma(\frac{d-1}{2})}(1-x^2)^{\frac{d-3}{2}} dx \geq \frac{\sqrt{d}}{2e\pi} \mathbb{1}_{\left[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\right]}(x)\, dx$$

and therefore, for every $f \geq 0$:

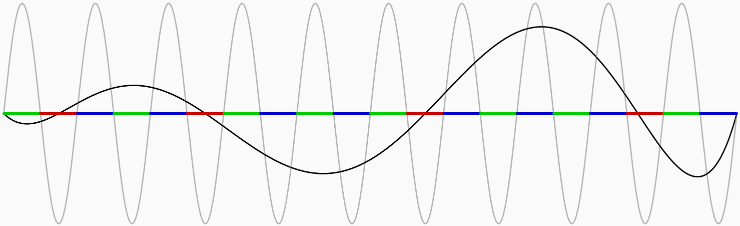
$$\int_{-1}^{1} f(x) d\mu_d(x) \geq \frac{\sqrt{d}}{2e\pi} \int_{-\frac{1}{\sqrt{d}}}^{\frac{1}{\sqrt{d}}} f(x) dx = \frac{1}{2e\pi} \int_{-1}^{1} f\left(\frac{t}{\sqrt{d}}\right) dt$$

Example

$$\int_{-1}^{1} f(x)d\mu_d(x) \geq \frac{1}{2e\pi}\int_{-1}^{1} f\left(\frac{t}{\sqrt{d}}\right) dt$$

Setting $f(x) = (\sin(\pi\sqrt{d}mx) - p(x))^2$ we obtain

$$\int_{-1}^{1} \left(\sin(\pi\sqrt{d}mx) - p\,(x)\right)^2 d\mu(x) \geq \frac{1}{2e\pi}\int_{-1}^{1} \left(\sin(\pi mx) - p\left(\frac{x}{\sqrt{d}}\right)\right)^2 dx$$



$$\frac{1}{2e\pi}\int_{-1}^{1} \left(\sin(\pi mx) - p\left(\frac{x}{\sqrt{d}}\right)\right)^2 dx \geq \frac{1}{2e\pi}\frac{m-k}{2m}$$

Example - conclusion

Setting

- $f(x) = \sin(\pi d^3 x)$
- $n = d^2$
- $B = 2^d$

and using our main theorem, we get

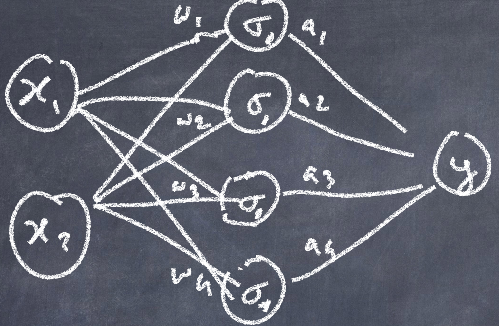
- $\mathcal{P}_n F \geq \frac{1}{5e\pi}$
- To get $\frac{1}{50e^2\pi^2}$–approximatopn of F, the width of NN should be

$$\frac{\sqrt{N_{d,d^2}}}{20e\pi 2^{2d}(1+\sqrt{4d}) + 2^{d+1}} = 2^{\Omega(d\log(d))}$$

Convergence of Wide Depth-2 Networks: Mean-Field Insight

Neural Nets

1 hidden layer



$$y = f(x;\theta) = \frac{1}{N}\sum_{i=1}^{N} \sigma(x;\theta_i)$$

$$\theta_i = (a_i, b_i, w_i)$$

$$\sigma(x,\theta) = a\,\sigma_0(\langle w, x\rangle + b)$$

$\sigma_0(x) = x_+$ or $\sigma_0(x) = \frac{1}{1+e^{-2x}}$ ...

Th [Cybenko '89]

if $\mathbb{E}[f(x)^2] < \infty$

if $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous, $\begin{cases} \sigma(x) \xrightarrow[x \to +\infty]{} 1 \\ \sigma(x) \xrightarrow[x \to -\infty]{} 0 \end{cases}$

then $\forall \varepsilon > 0$, $\exists\, N(\varepsilon)$ s.t.

$$\inf_{(a_i, b_i, w_i)_{1 \le i \le N}} \mathbb{E}\left[f(x) - \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle + b_i) \right] \le \varepsilon$$

PB: how to find the good approximator?

→ Gradient Descent:

$$\theta^{k+1} = \theta^{k} + s_k v_k$$

(stepsize → s_k)

$$v_k = -\nabla R(\theta^k)$$

SGD:

$$v_k = -\nabla R(\theta^k) + \varepsilon^k$$

(ε^k: noise)

Pb: $\underset{\sim}{R}(\theta)$ is not convex in θ
it has local minima, etc...

BUT

still, (S)GD works
specially when the network is
over-parameterized → WHY?

Here: $R_N(\theta) = \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\sigma(x;\theta) - y\right)^2\right]$

$$\rightarrow \theta_i^{k+1} = \theta_i^k + 2\eta_k \mathbb{E}\left[\sigma(x,\theta_i)\left(y - \frac{1}{N}\sum_{j=1}^{N}\sigma(x;\theta_j)\right)\right]$$

$$R_N(\theta) = \mathbb{E}[y^2] + \frac{2}{N}\sum_{i=1}^{N} -\mathbb{E}[y\sigma(x,\theta_i)] + \frac{1}{N^2}\sum_{i,j=1}^{N}\mathbb{E}[\sigma(x,\theta_i)\sigma(x,\theta_j)]$$

$$R_N(\theta) = R_\# + \frac{2}{N}\sum_{i=1}^{N} V(\theta_i) + \frac{1}{N^2}\sum_{i,j=1}^{N} U(\theta_i,\theta_j)$$

energy

external potential

pairwise potential

Important: the kernel U is semi-definite:

$\forall h$ bounded, compactly supported

$$\iint U(\theta_1, \theta_2)\, h(\theta_1)\, h(\theta_2)\, d\theta_1\, d\theta_2 \geq 0$$

repulsive interaction (in average sense)

Convexification

$$f(x,\theta) = \frac{1}{N}\sum_{i=1}^{N} \sigma(x,\theta_i)$$

$$= \int \sigma(x,\theta)\, \rho(d\theta)$$

when N is large, can be modelled by a <u>density ρ</u>

$\to R_N(\theta)$

$$\simeq \; R(\rho) := R_{\#} + 2\int v(\theta)\,\rho(d\theta) + \int u(\theta_1,\theta_2)\,\rho(d\theta_1)\,\rho(d\theta_2)$$

<u>Prop</u>: if $\exists \varepsilon_0 : \forall \rho$; $R(\rho) \leq \inf_{\rho} R(\rho) + \varepsilon_0$

$$\Rightarrow \int u(\theta,\theta)\,\rho(d\theta) \leq K$$

then $\left|\inf_{\theta} R_N(\theta) - \inf_{\rho} R(\rho)\right| \leq K/N$

Remark:

$$R(\rho) = R_{\#} + 2\int V(\theta)\,\rho(d\theta) + \int U(\theta_1,\theta_2)\,\rho(d\theta_1)\,\rho(d\theta_2)$$

is <u>convex in ρ</u> !

cvx optimization in ∞-dim $\mathcal{M}_1(\Theta)$

∞- dim approach

functional derivative

$$\psi(\theta, \rho) = \frac{1}{2} \frac{\delta R(\rho)}{\delta \rho(\theta)} = V(\theta) + \int U(\theta, \theta') \rho(d\theta')$$

= variation of energy when adding 1 particule at θ

ρ_* is a minimum if

$$\mathrm{supp}(\rho_*) \subset \underset{\theta \in \mathbb{R}^D}{\mathrm{argmin}} \; \psi(\theta, \rho^*)$$

→ Idea 1:

- Discretize $\Theta \to \Theta_N$
- Minimize $R(\rho_N)$ on $\mathcal{M}_1(\Theta_N) \simeq (\mathbb{R}^N)'$

$$R(\rho_N) = (R_\# +) + \rho_N V_N + \rho_N U_N \rho_N'$$

where $V_N(i) = V(\theta_i)$ and $U_N(i,j) = U(\theta_i, \theta_j)$

Pb: curse of dimensionality: requires gigantic N.

→ What follows proves that a good N does not need to be exponential in D.

Idea 2: particular approach

$$\rho_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{\theta_i}$$

each particle θ_i moves according to the forces of the system: at time $t = k\varepsilon$

speed $\mathbb{E}(v_i^k | \mathcal{F}_k) = -\nabla_{\theta_i} V(\theta_i^k) - \frac{1}{N} \sum_{j=1}^{N} \nabla_{\theta_i} U(\theta_i^k, \theta_j^k)$

$$= \mathbb{E}[y\sigma(x, \theta_i)] - \frac{1}{N} \sum_{j=1}^{N} \nabla_{\theta_i} \mathbb{E}[\sigma(x, \theta_i^k)\sigma(x, \theta_j^k)]$$

→ this is exactly (S)GD !

→ prove that the particle system behaves like its continuous equivalent (statistical mechanics)

$$\mathbb{E}\left[v_i^k / \mathcal{F}_k^\varepsilon \right] = -\nabla \psi(\theta_i^k, \rho_{k\varepsilon})$$

$(\rho_t)_{t \geq 0}$ $t = k\varepsilon$ continuous time limit.

Continuity equation

div

$$\partial_t \rho_t = - \nabla \cdot \left(\rho_t(\theta) \, v(\theta, \rho_t) \right)$$

enters: $\rho_t(\theta) \, v(\theta, \rho_t)$

leaves: $\rho_t(\theta + d\theta) \, v(\theta + d\theta, \rho_t)$

→ variation of ρ_t: $\frac{\partial}{\partial \theta} \left(\rho_t(\theta) \, v(\theta, \rho_t) \right)$

$$\rightarrow \partial_t \rho_t(\theta) = \nabla_\theta \cdot \left(\rho_t(\theta) \nabla_\theta \Psi(\theta, \rho_t)\right)$$

Fixed points = densities ρ_* s.t. all mass sits on zero velocity positions:

$$\text{supp}(\rho_*) \subset \{\theta \in \mathbb{R}^D : \nabla \Psi(\theta, \rho_*) = 0\}$$

Thm: if $\|\sigma_*\|_\infty \leq K_2$, $\|\nabla_\theta \sigma_*(x;\theta)\|_2 \leq K_2$

$|y_k| \leq K_2$

if $\|\nabla_\theta V(\theta)\|_2 \leq K_3$, $\|\nabla_{\theta_1} U(\theta_1,\theta_2)\|_2 \leq K_3$

$\|\nabla_\theta V(\theta) - \nabla_\theta V(\theta')\|_2 \leq K_3 \|\theta - \theta'\|_2$

$\|\nabla_{\theta_1} U(\theta_1,\theta_2) - \nabla_{\theta_1} U(\theta_1',\theta_2')\|_2 \leq K_3 \left\| \begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} - \begin{pmatrix}\theta_1'\\ \theta_2'\end{pmatrix}\right\|_2$

$\rho_0 \in \mathcal{P}(\mathbb{R}^D)$. SGD with initialization $(\theta_i^0)_{1 \leq i \leq N} \sim \rho_0$

step size $s_k = \varepsilon/2$

ρ_t : solution of the PDE

Then $\exists C = C(K_i)$, $\forall f : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}$ st $\|f\|_\infty \leq 1$, $\|f\|_{Lip} \leq 1$, $\varepsilon \leq 1$

$$\sup_{k=0 \ldots T/\varepsilon} \left| R_N(\theta^k) - R(\rho_{k\varepsilon}) \right| \leq C e^{CT} \sqrt{\frac{1}{N} \vee \varepsilon} \left[\sqrt{D + \log \frac{N}{\varepsilon}} + 3 \right]$$

with proba $\geq 1 - e^{-3^2}$

→ PDE accurate as soon as $N \gg D$
$\varepsilon \ll 1/D$

→ no curse of dimensionality.

numerical experiments → PDE approx very accurate in practice.

→ global convergence can be proved in some cases.

Gradient Flows

$$\dot{x}(t) = -\nabla F(x(t))$$

"continuous time gradient descent"

$$x(t+\varepsilon) = \arg\min_{z \in \mathbb{R}^d} \left\{ F(z) + \frac{1}{2\varepsilon} \|z - x(t)\|_2^2 \right\}$$

cf GD: $x_{k+1} = \arg\min_{z \in \mathbb{R}^d} \left\{ F(z) + \frac{1}{2s_k} \|z - x_k\|^2 \right\}$

→ general definition: for a distance $d(\cdot,\cdot)$

$$x_\varepsilon((k+1)\varepsilon) = \underset{z \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ F(z) + \frac{1}{2\varepsilon} d(z, x_\varepsilon(t)) \right\}$$

$$x(t) = \lim_{\varepsilon \to 0} x_\varepsilon(t)$$

is the gradient flow of the cost function F on $\mathcal{X}$ for the metric d

Prop:

$$\partial_t \rho(t) = \nabla_\theta \cdot \left(\rho_t(\theta) \nabla_\theta \Psi(\theta, \rho_t) \right)$$

is the gradient flow for the cost $R(\rho)$ in Wasserstein metric

$$W_2(\mu, \nu) = \left(\inf_{\gamma \in \mathcal{C}(\mu, \nu)} \int \|x - y\|_2^2 \, \gamma(dx, dy) \right)^{1/2}$$

couplings of μ and ν = prob. γ on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and ν

Noisy GD

$$\theta_i^{k+1} = \theta_i^k + 2s_k \nabla_{\theta_i} \sigma(x_k, \theta_i^k)\left(y_k - \frac{1}{N}\sum_{j=1}^{N} \sigma(x_k, \theta_j^k)\right)$$

$$+ \sqrt{2\tau s_k}\; g_i^k \leftarrow \sim \mathcal{N}(0, I_D)$$

$$\rightarrow \partial_t \rho_t(\theta) = \nabla_\theta \cdot \left(\rho_t(\theta) \nabla_\theta \Psi(\theta, \rho_t)\right) + \tau \Delta \rho_t(\theta)$$

$$F(\rho) = \frac{1}{2} R(\rho) - \tau S(\rho)$$ free energy

where $$S(\rho) = -\int \rho(\theta) \log \rho(\theta)\, d\theta$$ entropy

$$\rho_*(\theta) = \frac{1}{Z(\beta)} \exp\left(-\beta \Psi(\theta, \rho_*)\right)$$ Boltzmann equation

One can prove convergence in a time that depends on D but not on N.

→ SGD reaches a near-optimum in time independent of the number of neurons

Statistical Physics View on Generalization

Outline

Approximation: a Separation Result

A separation result

Depth-2 neural network

Convergence of Wide Depth-2 Networks: Mean-Field Insight

Statistical Physics View on Generalization

Statistical Learning: Recap in a Nutshell

Linear Models

Beyond Linear Models

Empirical Risk Minimization

Goal: find θ minimizing $L(\theta) = \mathbb{E}_P\Big[\ell\big(f_\theta(X), Y\big)\Big]$

But the learnt rule $\hat{f}_n = f_{\hat{\theta}_n}$ depends only on the sample S_n

PAC learning: for every $\epsilon, \delta > 0$, find the *sample size* $n(\epsilon, \delta)$ such that *whatever the law P*, if $n \geq n(\delta, \epsilon)$ then with probability at least $1 - \delta$ one has $L(\hat{\theta}_n) < \min_{\theta \in \Theta} L(\theta) + \epsilon$

Idea: Empirical Risk Minimization (ERM):

$$\hat{\theta}_n = \arg\min_{\theta \in \Theta} L_n(\theta) := \frac{1}{n} \sum_{i=1}^{n} \ell\big(f_\theta(X_i), Y_i\big)$$

PAC learning theory (pessimistic): uniform law of large numbers

$$\mathbb{P}\left(\forall \theta \in \Theta, \ \ \big|L_n(\theta) - L(\theta)\big| \leq c\sqrt{\frac{\dim \Theta + \log \frac{1}{\delta}}{n}}\right) \geq 1 - \delta$$

Bias-variance tradeoff

The classical PAC theory does not work unless $n \gg \dim \Theta$

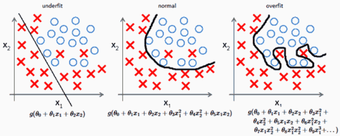
Consider different models $(\Theta_d)_{d \geq 1}$ Example: images at different resolutions

Decomposition of the (quadratic) risk

$$\mathbb{E}\left[L(\hat{\theta}_n)\right] = b_d^2 + v_d$$

- Bias: $b_d = \min_{\theta \in \Theta_d} L(\theta)$ decreases with d
- Variance term: $v_d = \frac{\dim \Theta_d}{n}$ increases with d

$\Longrightarrow$ best choice = bias-variance balance think: polynomial regression



src: `https://miro.medium.com`

Bias-variance tradeoff

The classical PAC theory does not work unless $n \gg \dim \Theta$

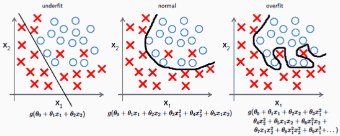
Consider different models $(\Theta_d)_{d \geq 1}$ Example: images at different resolutions

Decomposition of the (quadratic) risk

$$\mathbb{E}\left[L(\hat{\theta}_n)\right] = b_d^2 + v_d$$

– Bias: $b_d = \min_{\theta \in \Theta_d} L(\theta)$ decreases with d

– Variance term: $v_d = \frac{\dim \Theta_d}{n}$ increases with d



$\Longrightarrow$ best choice = bias-variance balance think: polynomial regression

This statement is challenged by numerical experiments on neural networks

src: `https://miro.medium.com`

Outline

Approximation: a Separation Result

- A separation result
- Depth-2 neural network

Convergence of Wide Depth-2 Networks: Mean-Field Insight

Statistical Physics View on Generalization

- Statistical Learning: Recap in a Nutshell
- Linear Models
- Beyond Linear Models

Linear models

$\hat{Y} = X \cdot \theta,\ \theta \in \mathbb{R}^d$

Matrix notation: $\mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} \in M_{n,d}(\mathbb{R}),\ Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \in \mathbb{R}^n$

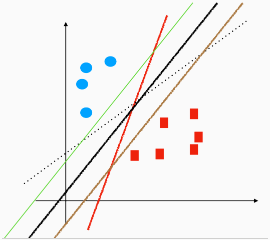
L^2 loss $\ell\big(\hat{Y}_i, Y_i\big) = \big(\hat{Y}_i - Y_i\big)^2$ solution: Ordinary Least Square (OLS)

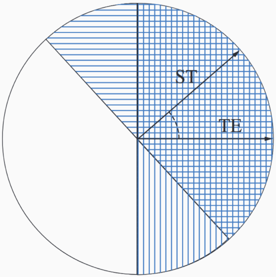
$\hat{\theta}_n \in \underset{\theta \in \mathbb{R}^d}{\arg\min} \left\| Y - X\theta \right\|_2^2$ satisfies the normal equations $XX^T\theta = X^TY$

If $rank(X^TX) = d$ **(requires** $n \geq d$**)**	**Otherwise, many solutions**
there is a unique solution	Popular: *least-norm* solution
$$\hat{\theta}_n = \big(X^TX\big)^{-1}X^TY$$	$$\hat{\theta}_n = X^T\big(XX^T\big)^{-1}Y$$
Classical statistics theory	Statistically spurious

Physics model: the teacher-student framework

- The model is true: $Y_i = \text{sign}(X_i \cdot \theta^*)$
- $X_i \sim \mathcal{N}(0, I_d)$ cf images?
- $\theta^* \sim \mathcal{N}(0, I_d)$ cf Bayesian approach?
- High-dimensional limit as $n, d \to \infty$ with $\alpha = n/d$ fixed





src: *Learning to generalize*, Opper'01

• volume of students with generalization error ϵ: $v(\epsilon) \propto \epsilon^{d \times \text{entropy}(\epsilon)}$

• probability that a student with generalization error ϵ makes no mistake: $p(\epsilon) \propto \epsilon^{n \times \text{energy}(\epsilon)}$

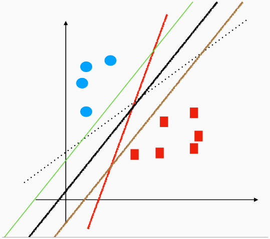
$\Longrightarrow$ average generalization error of ERM:

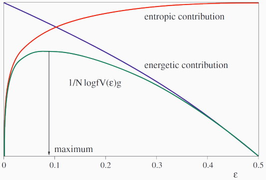
$$\int v(\epsilon) p(\epsilon) d\epsilon \approx \arg\max_{\epsilon} \text{entropy}(\epsilon) + \alpha\, \text{energy}(\epsilon)$$

in the limit when $\alpha = n/d$ fixed

Physics model: the teacher-student framework

- The model is true: $Y_i = \text{sign}(X_i \cdot \theta^*)$
- $X_i \sim \mathcal{N}(0, I_d)$ cf images?
- $\theta^* \sim \mathcal{N}(0, I_d)$ cf Bayesian approach?
- High-dimensional limit as $n, d \to \infty$ with $\alpha = n/d$ fixed





src: *Learning to generalize*, Opper'01

• volume of students with generalization error ϵ: $v(\epsilon) \propto \epsilon^{d \times \text{entropy}(\epsilon)}$

• probability that a student with generalization error ϵ makes no mistake: $p(\epsilon) \propto \epsilon^{n \times \text{energy}(\epsilon)}$

$\Longrightarrow$ average generalization error of ERM:

$$\int v(\epsilon)p(\epsilon)d\epsilon \approx \arg\max_{\epsilon} \text{entropy}(\epsilon) + \alpha\, \text{energy}(\epsilon)$$

in the limit when $\alpha = n/d$ fixed

Statistical physics analysis

$\Longrightarrow$ "spin glasses", physics analysis in the 1990's (Opper & Kinzel, etc.) rigorous proofs recently (Florent Krzakala, Lenka Zdeborova, etc.): can compute

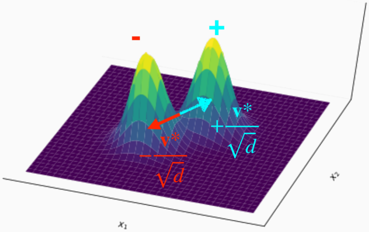
- the "Bayes risk" $\mathbb{E}\left[L(\theta)|S_n\right] =$ mean risk of ERM
- the risk of the minimal-norm ERM

in the limit, as a function of α.

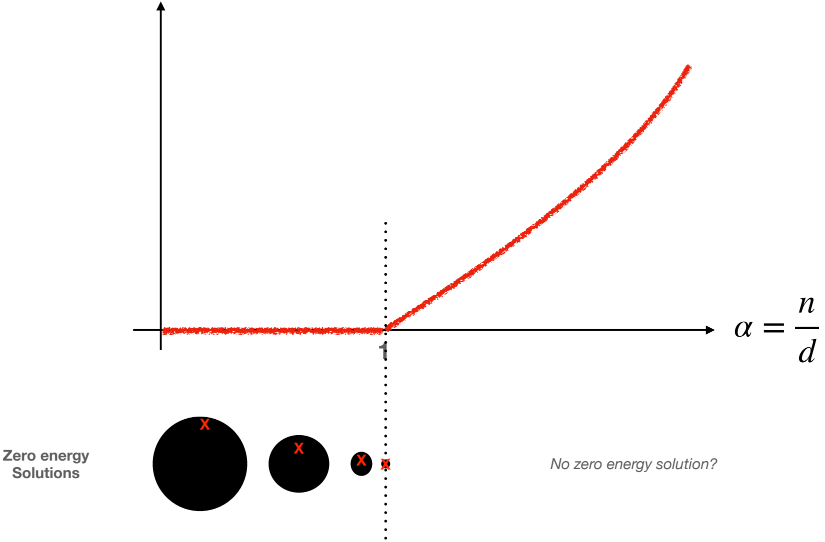
Other model discussed: Gaussian Mixture

$$X_i \sim \mathcal{N}\left(\frac{Y_i \mathbf{v}^*}{\sqrt{d}}, \Delta\right)$$

Similar results

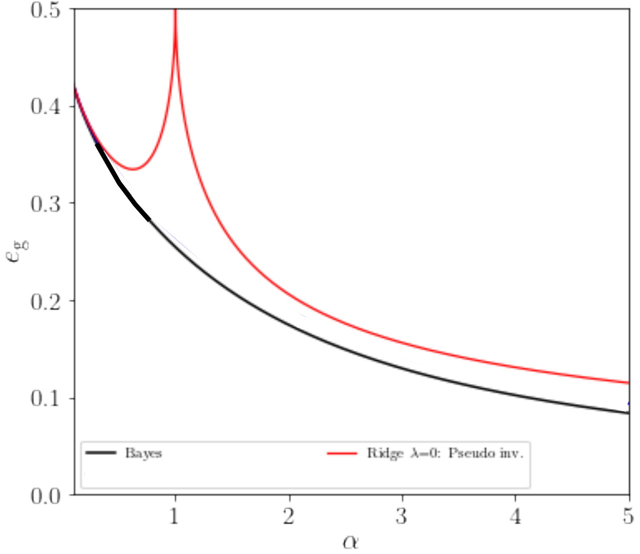


Least norm solution



slide by Florent Krzakala

Least norm solution: “double descent”



slide by Florent Krzakala

Gradient Descent

Iterative optimization of θ:

$$\theta^t = \theta^{t-1} - \eta_t \nabla_\theta L_n(\theta^{t-1})$$

Here $f_\theta(x) = x \cdot \theta \implies$

$$\nabla_\theta L_n(\theta^{t-1}) = \frac{1}{n} \sum_{i=1}^{n} \partial_1 \ell\big(f_\theta(X_i), Y_i\big) X_i \in \mathrm{span}(X_1, \ldots, X_n)$$

Representer theorem

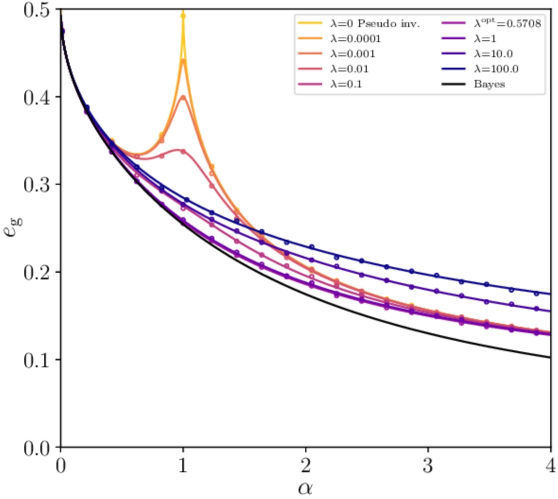
$$\min_{\theta \in \mathbb{R}^d} L_n(\theta) = \min_{\theta \in \mathrm{span}(X_1, \ldots, X_n)} L_n(\theta)$$

In fact, if $\theta = \theta_X + \theta_{X^\perp}$, then $L_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(X_i \cdot \theta, Y_i\big) = L_n(\theta_X)$

$\implies$ if $\theta^0_{X^\perp} = 0$ gradient descent finds the solution with minimal norm

$\implies$ other approach: minimize *ridge loss* $L_n^\lambda(\theta) = L_n(\theta) + \lambda \|\theta\|^2$

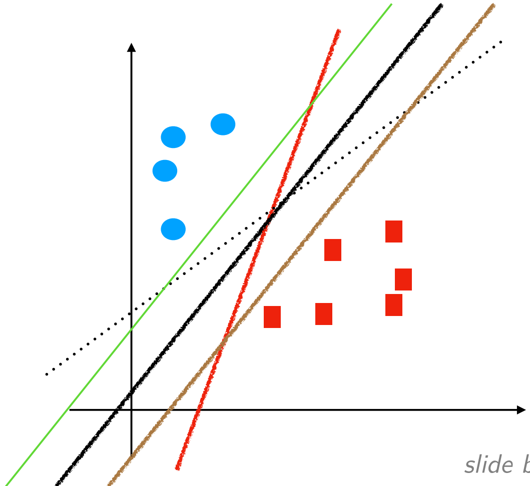
Ridge loss



[Aubin, Krzakala, Lu, Zdeborova'20]

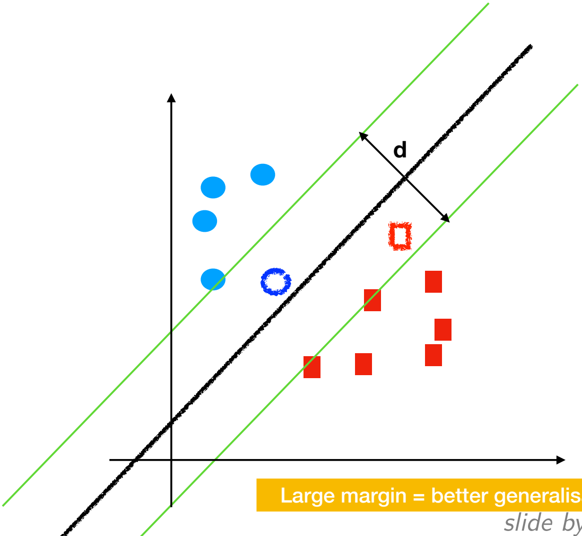
Pushing the boundaries

Which frontier should we choose?



slide by Florent Krzakala

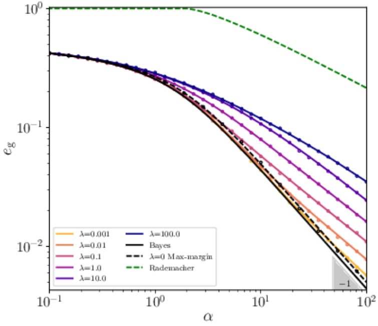
Pushing the boundaries

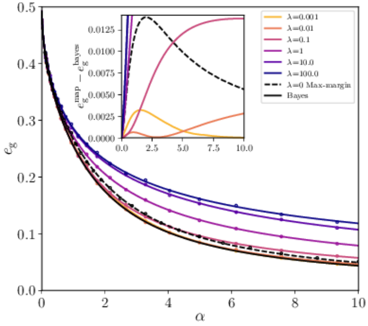


Large margin = better generalisation properties!

slide by Florent Krzakala

Chasing the Bayes optimal result





Regularized logistic losses (almost) achieve Bayes optimal results!

[Aubin, Krzakala, Lu, Zdeborova'20]

Outline

Approximation: a Separation Result

A separation result

Depth-2 neural network

Convergence of Wide Depth-2 Networks:
Mean-Field Insight

Statistical Physics View on Generalization

Statistical Learning: Recap in a Nutshell

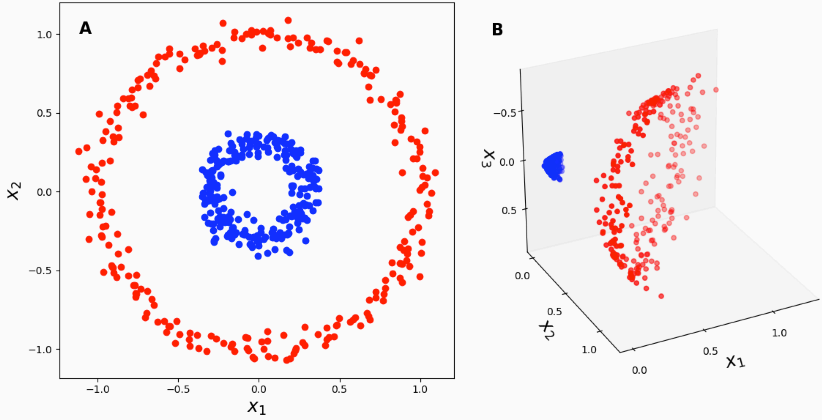
Linear Models

Beyond Linear Models

Lifting the data: feature map

Find a better representation of the data that makes it linearly separable

$$X_i \mapsto \Phi(X_i)$$



$$\Phi(x_1, x_2) = \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$$

src: http://gregorygundersen.com/

Template matching

Representer theorem: can search for

$$\hat{\theta}_n = \sum_{i=1}^{n} \beta_i X_i$$

The resulting prediction is hardly more than comparison with data:

$$f_{\hat{\theta}_n}(x) = \sum_{i=1}^{n} \beta_i X_i \cdot x$$

(cf nearest neighbor method)

$\implies$ can consider more general similarity functions than scalar product:

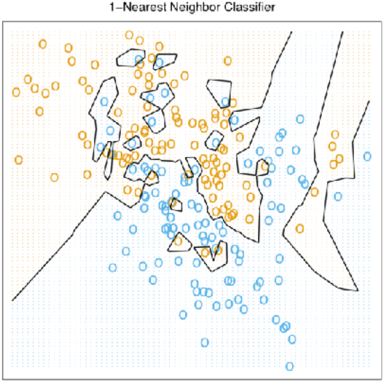
$$f_\theta(x) = \sum_{i=1}^{n} \beta_i \, K(X_i, x)$$

where K is an carefully chosen *kernel*

Ex: Gaussian Kernel

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\beta \|\mathbf{X}_i - \mathbf{X}_j\|_2^2}$$

As $\beta \to \infty$ converges to 1NN methods



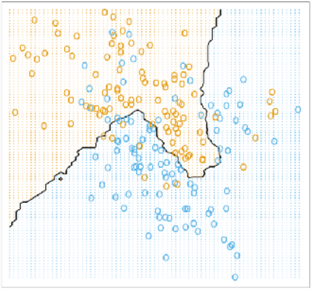
Warning: $\beta =$ inverse temperature $\neq$ the one of previous slide

slide by Florent Krzakala

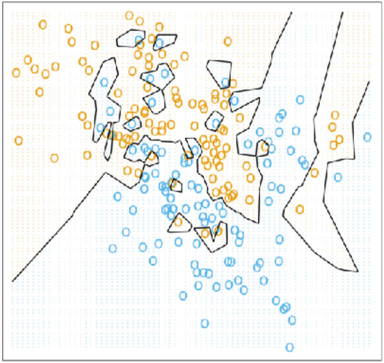
Ex: Gaussian Kernel

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\beta \|\mathbf{X}_i - \mathbf{X}_j\|_2^2}$$

For lower values, interpolate between neighbours



1–Nearest Neighbor Classifier



Warning: β = inverse temperature $\neq$ the one of previous slide

slide by Florent Krzakala

Mercer's Theorem & the feature map

If K(s,t) is symmetric and positive-definite, then there is an orthonormal basis $\{e_i\}_i$ of $L^2[a, b]$ consisting of « **eigenfunctions** » such that the corresponding sequence of eigenvalues $\{\lambda_i\}_i$ is nonnegative.

$$K(s,t) = \sum_{j=1}^{\infty} \lambda_j \, e_j(s) \, e_j(t)$$

All symmetric positive-definite Kernels can be seen as a projection in an infinite dimensional space

Orignal space (data space) dimension d

Feature map

$$\Phi = g(X)$$

Features space (After projection) dimension D (*possibly infinite*)

$$X_i$$

$$K(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{\Phi}_i \cdot \mathbf{\Phi}_j$$

$$\Phi_i = \begin{pmatrix} \sqrt{\lambda_1} e_1(X_i) \\ \sqrt{\lambda_2} e_2(X_i) \\ \sqrt{\lambda_3} e_4(X_i) \\ \dots \\ \sqrt{\lambda_D} e_D(X_i) \end{pmatrix}$$

slide by Florent Krzakala

Example: Gaussian Kernel, 1D

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\frac{1}{2\sigma^2}\|\mathbf{X}_i - \mathbf{X}_j\|_2^2}$$

$$e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2} = e^{\frac{-x_i^2 - x_j^2}{2\sigma^2}} \left(1 + \frac{2x_i x_j}{1!} + \frac{(2x_i x_j)^2}{2!} + \ldots\right) \tag{1.25}$$

$$= e^{\frac{-x_i^2 - x_j^2}{2\sigma^2}} \left(1 \cdot 1 + \sqrt{\frac{2}{1!}}x_i \cdot \sqrt{\frac{2}{1!}}x_j + \sqrt{\frac{(2)^2}{2!}}(x_i)^2 \cdot \sqrt{\frac{(2)^2}{2!}}(x_j)^2 + \ldots\right)$$

$$= \phi(x_i)^T \phi(x_j)$$

$$\text{where, } \phi(x) = e^{\frac{-x^2}{2\sigma^2}} \left(1, \sqrt{\frac{2}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \ldots\right)$$

Infinite dimensional feature (polynomial) map!

slide by Florent Krzakala

Kernel methods

$$\mathcal{R} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, f_\beta(\mathbf{X}_i))$$

$$f_\beta(\mathbf{X}) = \sum_{j=1}^{n} \beta_j K(\mathbf{X}_j, \mathbf{X}) \quad \beta \in \mathbb{R}^n$$

Gradient descent

$$\beta^t = \beta^{t-1} - \eta \nabla_\beta \mathcal{R}$$

Gradient flow

$$\dot{\beta}^t = -\nabla_\beta \mathcal{R}$$

$$\mathcal{R} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(\mathbf{\Phi}_i))$$

Feature map $\Phi = g(X)$

$$f_\theta(\mathbf{\Phi}) = \theta \cdot \mathbf{\Phi} \qquad \theta \in \mathbb{R}^{D=\infty}$$

Gradient descent

$$\theta^t = \theta^{t-1} - \eta \nabla_\theta \mathcal{R}$$

Gradient flow

$$\dot{\theta}^t = -\nabla_\theta \mathcal{R}$$

slide by Florent Krzakala

$$K(X_i, X_j) = \Phi_i \cdot \Phi_j$$

$$\mathbf{K} = \begin{pmatrix} K(X^1, X^1) & K(X^1, X^2) & \dots & K(X^1, X^N) \\ K(X^2, X^1) & K(X^2, X^2) & \dots & K(X^2, X^N) \\ \dots & \dots & \dots & \dots \\ K(X^N, X^1) & K(X^N, X^2) & \dots & K(X^N, X^N) \end{pmatrix}$$

Say you have one million examples....

slide by Florent Krzakala

Kernel methods

$$\mathcal{R} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, f_\beta(\mathbf{X}_i))$$

$$f_\beta(\mathbf{X}) = \sum_{j=1}^{n} \beta_j K(\mathbf{X}_j, \mathbf{X}) \quad \beta \in \mathbb{R}^n$$

Gradient descent

$$\beta^t = \beta^{t-1} - \eta \nabla_\beta \mathcal{R}$$

Gradient flow

$$\dot{\beta}^t = - \nabla_\beta \mathcal{R}$$

$$\mathcal{R} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(\mathbf{\Phi}_i))$$

Feature map $\Phi = g(X)$

$$f_\theta(\mathbf{\Phi}) = \theta \cdot \mathbf{\Phi} \qquad \theta \in \mathbb{R}^{D=\infty}$$

Gradient descent

$$\theta^t = \theta^{t-1} - \eta \nabla_\theta \mathcal{R}$$

Gradient flow

$$\dot{\theta}^t = - \nabla_\theta \mathcal{R}$$

slide by Florent Krzakala

Kernel methods

$$\mathscr{R} = \frac{1}{n}\sum_{i=1}^{n} \mathscr{L}(y_i, f_\theta(\mathbf{\Phi}_i))$$

$$f_\theta(\mathbf{\Phi}) = \theta \cdot \mathbf{\Phi}$$

Feature map $\Phi = g(X)$

$$\theta \in \mathbb{R}^{D=\infty}$$

Gradient descent

$$\theta^t = \theta^{t-1} - \eta \nabla_\theta \mathscr{R}$$

Gradient flow

$$\dot{\theta}^t = -\nabla_\theta \mathscr{R}$$

Idea 1: truncate the expansion of the feature map (e.g. polynomial features)

Idea 2: approximate the feature map by sampling

slide by Florent Krzakala

Random Fourier Features [Recht-Rahimi '07]

Take $F_1, \ldots, F_D \overset{iid}{\sim} Q$ Fourier coefficients in $\mathbb{R}^d$ and choose feature map

$$\Phi(x) = \frac{1}{\sqrt{D}} \begin{pmatrix} e^{i F_1 \cdot x} \\ \vdots \\ e^{i F_D \cdot x} \end{pmatrix}$$

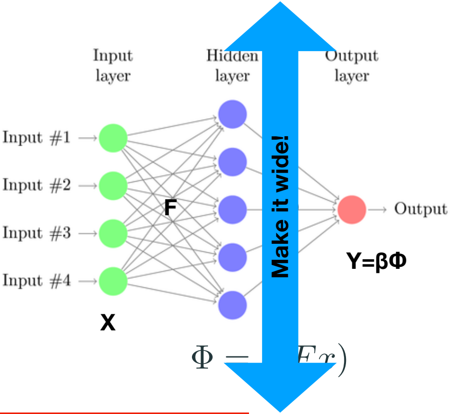
Then

$$K(X_j, X_k) = \Phi(X_j) \cdot \Phi(X_k) = \frac{1}{D} \sum_{\ell=1}^{D} e^{i F_\ell \cdot (X_j - X_k)} \xrightarrow{D \to \infty} \int e^{i f \cdot (X_j - X_k)} dQ(f)$$

$\to$ if $dQ/df =$ Fourier transform of κ, then $K(X_j, X_k) \approx \kappa(X_j - X_k)$

Kernel	$\kappa(\Delta)$	$dQ(f)$
Gaussian	$e^{-\|\Delta\|^2/2}$	$(2\pi)^{-d/2} e^{-\|f\|^2/2}$
Laplacian	$e^{-\|\Delta\|_1}$	$\prod_k \frac{1}{\pi(1+f_k^2)}$
Cauchy	$\prod_k \frac{2}{1+\Delta_k^2}$	$e^{-\|f\|_1}$

Equivalent representation:
A WIIIIIIIDE random 2-layer neural network



Fix the « weights » in the first layer randomly…
… and to learn only the weights in the second layer

Infinitly wide neural net with random weights converges to kernel methods
(Neal '96, Williams 98, Recht-Rahimi '07)

Deep connection with genuine neural networks in the "Lazy regime"
[Jacot, Gabriel, Hongler '18; Chizat, Bach '19; Geiger et al. '19]

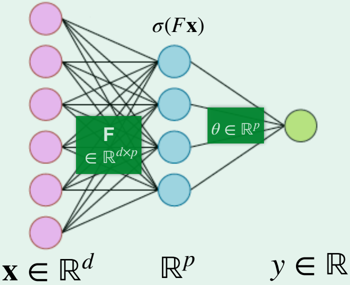
slide by Florent Krzakala

Random feature model...

Dataset:

- n vector $\mathbf{x}_i \in \mathbb{R}^d$, drawn randomly from $\mathcal{N}(0,\mathbf{1}_d)$
- n labels y_i given by a function $y_i^0 = f^0(\mathbf{x} \cdot \theta^*)$

Architecture: Two-layers neural network with fixed first layer **F**



Cost function:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, y_i^0) + \lambda \|\theta\|_2^2 \qquad \ell(\,.\,) = \begin{array}{l}\textit{Logistic loss}\\ \textit{Hinge loss}\\ \textit{Square loss}\\ \dots\end{array}$$

What is the training error & the generalisation error in the high dimensional limit $(d,p,n) \to \infty$?

slide by Florent Krzakala

… and its solution

Mention COUILLET HERE

[Loureiro, Gerace, **FK**, Mézard, Zdeborova, '20]

Definitions:

Consider the unique fixed point of the following system of equations

$$\begin{cases} \hat{V}_s = \frac{\alpha}{\gamma}\kappa_1^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}(y,\omega_0)\frac{\partial_\omega \eta(y,\omega_1)}{V}\right], \\ \hat{q}_s = \frac{\alpha}{\gamma}\kappa_1^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}(y,\omega_0)\frac{\left(\eta(y,\omega_1)-\omega_1\right)^2}{V^2}\right], \\ \hat{m}_s = \frac{\alpha}{\gamma}\kappa_1 \mathbb{E}_{\xi,y}\left[\partial_\omega \mathcal{Z}(y,\omega_0)\frac{\left(\eta(y,\omega_1)-\omega_1\right)}{V}\right], \\ \hat{V}_w = \alpha\kappa_\star^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}(y,\omega_0)\frac{\partial_\omega \eta(y,\omega_1)}{V}\right], \\ \hat{q}_w = \alpha\kappa_\star^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}(y,\omega_0)\frac{\left(\eta(y,\omega_1)-\omega_1\right)^2}{V^2}\right], \end{cases} \qquad \begin{cases} V_s = \frac{1}{\hat{V}_s}\left(1 - z\, g_\mu(-z)\right), \\ q_s = \frac{\hat{m}_s^2+\hat{q}_s}{\hat{V}_s}\left[1 - 2z g_\mu(-z) + z^2 g_\mu'(-z)\right] \\ \qquad - \frac{\hat{q}_w}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g_\mu'(-z)\right], \\ m_s = \frac{\hat{m}_s}{\hat{V}_s}\left(1 - z\, g_\mu(-z)\right), \\ V_w = \frac{\gamma}{\lambda+\hat{V}_w}\left[\frac{1}{\gamma} - 1 + z g_\mu(-z)\right], \\ q_w = \gamma\frac{\hat{q}_w}{(\lambda+\hat{V}_w)^2}\left[\frac{1}{\gamma} - 1 + z^2 g_\mu'(-z)\right], \\ \qquad + \frac{\hat{m}_s^2+\hat{q}_s}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g_\mu'(-z)\right], \end{cases}$$

$$\begin{cases} \eta(y,\omega) = \underset{x\in\mathbb{R}}{\operatorname{argmin}}\left[\frac{(x-\omega)^2}{2V} + \ell(y,x)\right] \\ \mathcal{Z}(y,\omega) = \int \frac{dx}{\sqrt{2\pi V^0}} e^{-\frac{1}{2V^0}(x-\omega)^2}\delta\left(y - f^0(x)\right) \end{cases}$$

where $V = \kappa_1^2 V_s + \kappa_\star^2 V_w$, $V^0 = \rho - \frac{M^2}{Q}$, $Q = \kappa_1^2 q_s + \kappa_\star^2 q_w$, $M = \kappa_1 m_s$, $\omega_0 = M/\sqrt{Q}\xi$, $\omega_1 = \sqrt{Q}\xi$ and g_μ is the Stieltjes transform of FF^T

$\kappa_0 = \mathbb{E}\left[\sigma(z)\right], \kappa_1 \equiv \mathbb{E}\left[z\sigma(z)\right], \kappa_\star \equiv \mathbb{E}\left[\sigma(z)^2\right] - \kappa_0^2 - \kappa_1^2$ and $\vec{z}^\mu \sim \mathcal{N}(\vec{0}, \mathbf{I_p})$

Then in the high-dimensional limit:

$$\epsilon_{gen} = \mathbb{E}_{\lambda,\nu}\left[(f^0(\nu) - \hat{f}(\lambda))^2\right]$$

with $(\nu,\lambda) \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\rho & M^\star \\ M^\star & Q^\star\end{pmatrix}\right)$

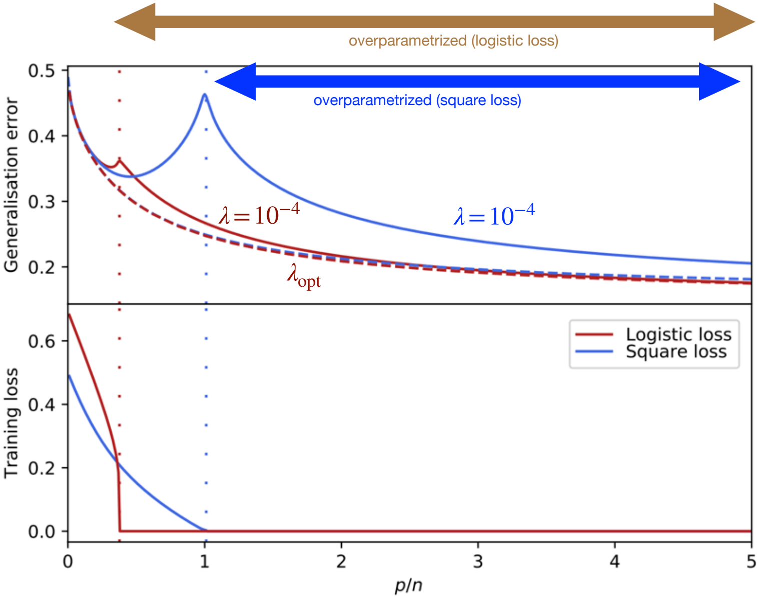
$$\mathcal{L}_{\text{training}} = \frac{\lambda}{2\alpha}q_w^\star + \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0^\star\right)\ell\left(y,\eta(y,\omega_1^\star)\right)\right]$$

with $\omega_0^\star = M^\star/\sqrt{Q^\star}\xi, \omega_1^\star = \sqrt{Q^\star}\xi$

Agrees with **[Louart , Liao , Couillet'18 & Mei-Montanari '19]** who solved a particular case using random matrix theory: linear function f^0, $\ell(x,y) = \|x-y\|_2^2$ & Gaussian random weights **F**.

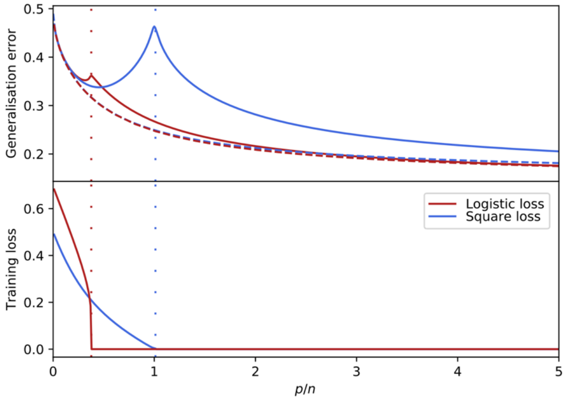
slide by Florent Krzakala

A classification task



slide by Florent Krzakala

A classification task



Implicit regularisation of gradient descent [Rosset, Zhy, Hastie, '04] [Neyshabur, Tomyoka, Srebro, '15]

As $\lambda \to 0$, in the overparametrized regime,
Logistic converges to max-margin, ℓ_2 converges to least norm

slide by Florent Krzakala

References

Separation Result

Amit Daniely, Depth Separation for Neural Networks. Proceedings of the 2017 Conference on Learning Theory.

Presentation by Tomáš Kocák

Mean-Field proof of convergence

B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, Linearized two-layers neural networks in high dimension, 2019.

L. Chizat, F. Bach. On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport. Advances in Neural Information Processing Systems (NeurIPS), 2018

Statistical Physics view on Generalization

Florent Krzakala's website and work: `https://florentkrzakala.com/files/leshouches2020/courses/florent`

Manfred Opper, Learning to generalize, in Model Neural Networks for Computation and Learning, 2001.