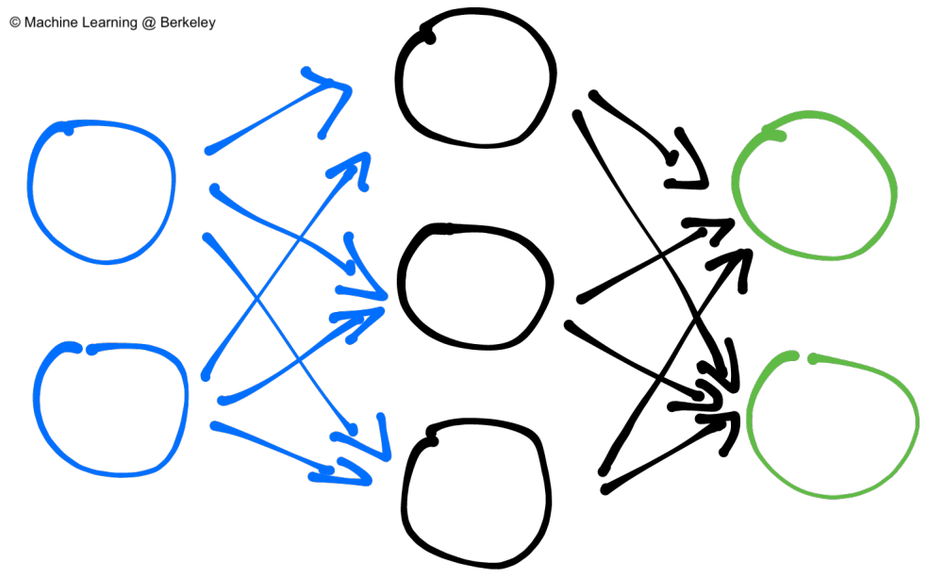
Trainability and Accuracy of Artificial Neural Networks



Eric Vanden-Eijnden
Courant Institute

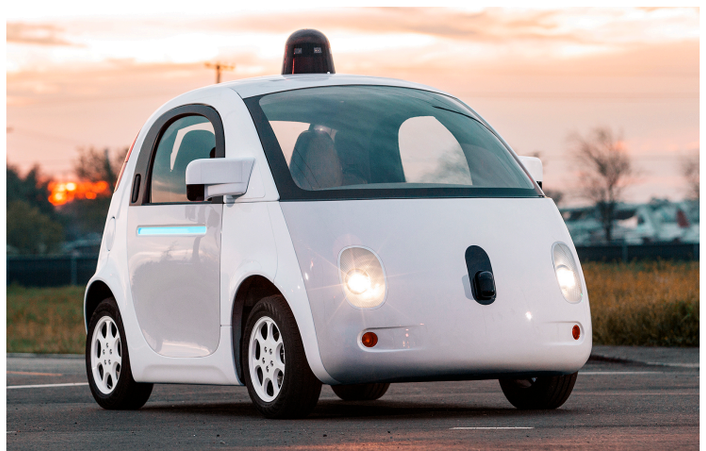
Using Physical Insights for Machine Learning
IPAM, UCLA, Nov 2019

With Grant Rotskoff, Samy Jelassi, Zhengdao Chen, and Joan Bruna.

Breaking the curse of dimensionality with ML?

- Machine learning has led to extraordinary progress in speech and image recognition, language processing and translation, object detection, data processing, etc.
- Problems assumed to be intractable a decade ago are now routine. *Why?*

- Are image / speech recognition, cognitive tasks etc. special (e.g. inherently low dimensional)?
- Can (deep) neural networks accurately represent other high-dimensional data / functions?
- *Can we beat / alleviate the curse of dimensionality in scientific computing?*



ABC of neural network representation and training

- Approximate a *target function* $f : \Omega \to \mathbb{R}$ defined on $\Omega \subseteq \mathbb{R}^d$ by a *neural network (NN) representation*,

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \widehat{\varphi}(\boldsymbol{x}, \boldsymbol{z}_i) \equiv \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

where $\{\boldsymbol{\theta}_i = (c_i, \boldsymbol{z}_i)\}_{i=1}^{n}$ are fitting parameters and $\widehat{\varphi} : \Omega \times \widehat{D} \to \mathbb{R}$ is an *activation function/unit*, e.g.

$$\text{ReLU} : \qquad \varphi(\boldsymbol{x}, \boldsymbol{z}) = \max(\boldsymbol{a} \cdot \boldsymbol{x} + b, 0), \qquad \boldsymbol{z} = (\boldsymbol{a}, b) \in \Omega \times \mathbb{R}.$$

ABC of neural network representation and training

- Approximate a *target function* $f : \Omega \to \mathbb{R}$ defined on $\Omega \subseteq \mathbb{R}^d$ by a *neural network (NN) representation*,

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \widehat{\varphi}(\boldsymbol{x}, \boldsymbol{z}_i) \equiv \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

where $\{\boldsymbol{\theta}_i = (c_i, \boldsymbol{z}_i)\}_{i=1}^n$ are fitting parameters and $\widehat{\varphi} : \Omega \times \widehat{D} \to \mathbb{R}$ is an *activation function/unit*, e.g.

$$\text{ReLU} : \qquad \varphi(\boldsymbol{x}, \boldsymbol{z}) = \max(\boldsymbol{a} \cdot \boldsymbol{x} + b, 0), \qquad \boldsymbol{z} = (\boldsymbol{a}, b) \in \Omega \times \mathbb{R}.$$

- Measure the approximation error via the *loss function / risk*

$$\ell(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \tfrac{1}{2} \int_\Omega |f(\boldsymbol{x}) - f_n(\boldsymbol{x})|^2 \, d\nu(\boldsymbol{x}) = \tfrac{1}{2}\mathbb{E}_{\text{data}} \, |f - f_n|^2 \, .$$

ABC of neural network representation and training

- Approximate a *target function* $f : \Omega \to \mathbb{R}$ defined on $\Omega \subseteq \mathbb{R}^d$ by a *neural network (NN) representation*,

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \widehat{\varphi}(\boldsymbol{x}, \boldsymbol{z}_i) \equiv \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

where $\{\boldsymbol{\theta}_i = (c_i, \boldsymbol{z}_i)\}_{i=1}^n$ are fitting parameters and $\widehat{\varphi} : \Omega \times \hat{D} \to \mathbb{R}$ is an *activation function/unit*, e.g.

$$\text{ReLU} : \qquad \varphi(\boldsymbol{x}, \boldsymbol{z}) = \max(\boldsymbol{a} \cdot \boldsymbol{x} + b, 0), \qquad \boldsymbol{z} = (\boldsymbol{a}, b) \in \Omega \times \mathbb{R}.$$

- Measure the approximation error via the *loss function / risk*

$$\ell(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \tfrac{1}{2} \int_\Omega |f(\boldsymbol{x}) - f_n(\boldsymbol{x})|^2 \, d\nu(\boldsymbol{x}) = \tfrac{1}{2}\mathbb{E}_{\text{data}} \, |f - f_n|^2 \, .$$

- In practice, estimate $\ell(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ via the *empirical loss function / risk*

$$\ell_P(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \frac{1}{P}\sum_{p=1}^{P} | \underbrace{f(\boldsymbol{x}_p)}_{=y_P} - f_n(\boldsymbol{x}_p)|^2, \qquad \{\boldsymbol{x}_p\}_{p=1}^P = \textit{iid drawn from } \nu \; = \textit{batch.}$$

ABC of neural network representation and training

- Approximate a *target function* $f : \Omega \to \mathbb{R}$ defined on $\Omega \subseteq \mathbb{R}^d$ by a *neural network (NN) representation*,

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \widehat{\varphi}(\boldsymbol{x}, \boldsymbol{z}_i) \equiv \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

where $\{\boldsymbol{\theta}_i = (c_i, \boldsymbol{z}_i)\}_{i=1}^n$ are fitting parameters and $\widehat{\varphi} : \Omega \times \hat{D} \to \mathbb{R}$ is an *activation function/unit*, e.g.

$$\text{ReLU} : \qquad \varphi(\boldsymbol{x}, \boldsymbol{z}) = \max(\boldsymbol{a} \cdot \boldsymbol{x} + b, 0), \qquad \boldsymbol{z} = (\boldsymbol{a}, b) \in \Omega \times \mathbb{R}.$$

- Measure the approximation error via the *loss function / risk*

$$\ell(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \tfrac{1}{2}\int_\Omega |f(\boldsymbol{x}) - f_n(\boldsymbol{x})|^2 \, d\nu(\boldsymbol{x}) = \tfrac{1}{2}\mathbb{E}_{\text{data}} |f - f_n|^2 .$$

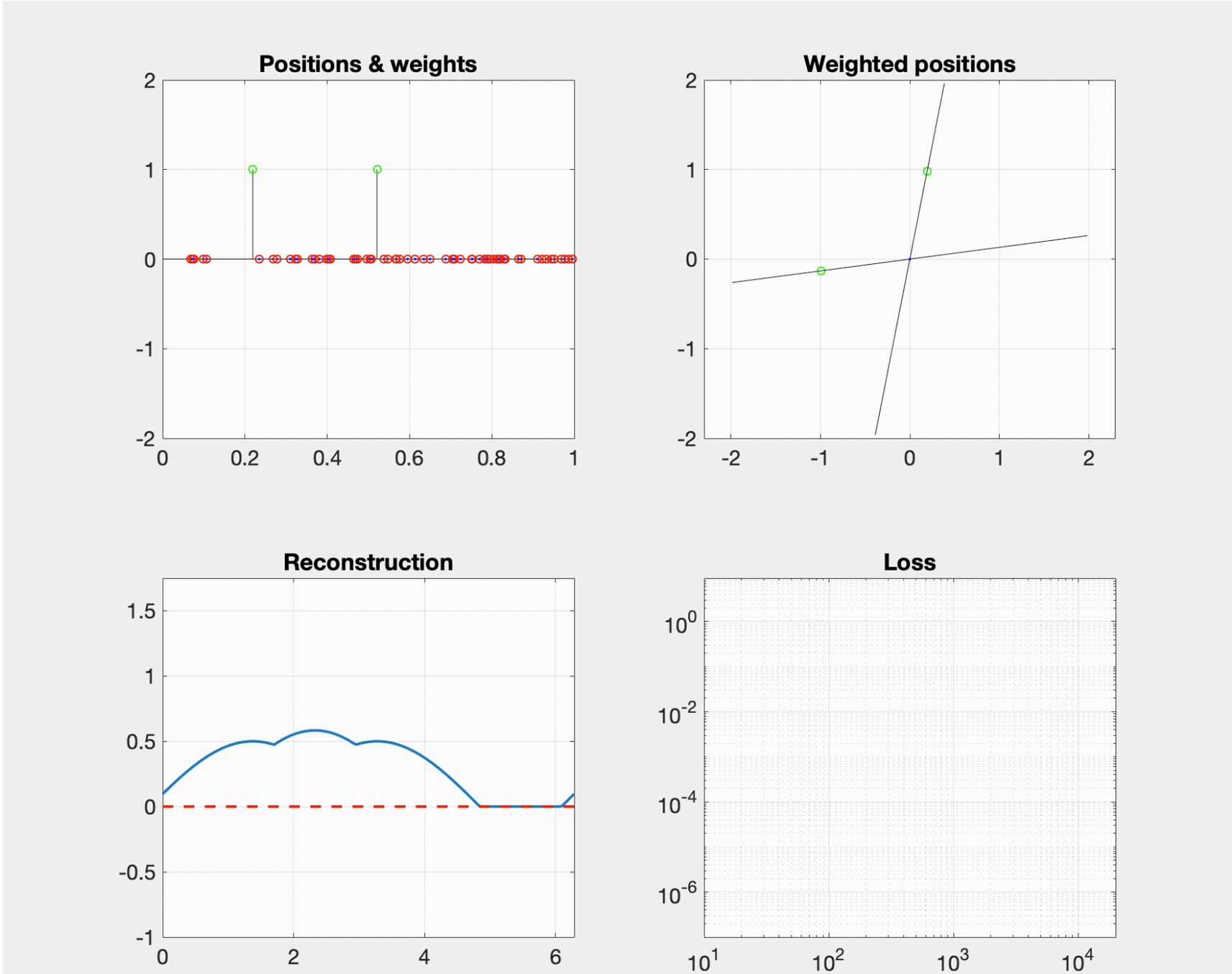
- In practice, estimate $\ell(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ via the *empirical loss function / risk*

$$\ell_P(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \frac{1}{P}\sum_{p=1}^{P} | \underbrace{f(\boldsymbol{x}_p)}_{=y_P} - f_n(\boldsymbol{x}_p)|^2, \qquad \{\boldsymbol{x}_p\}_{p=1}^P = \textit{iid drawn from } \nu \; = \textit{batch.}$$

- Train the network via *stochastic gradient descent (SGD)* to minimize the loss over the parameters

$$\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \Delta t_n \, \nabla_{\boldsymbol{\theta}_i} \ell_P(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n), \qquad i = 1, \ldots, n$$

ABC of neural network training



Planted neuron example

Several puzzles

- Main building blocks:

  1. *NN representation*: $f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \widehat{\varphi}(\boldsymbol{x}, \boldsymbol{z}_i) \equiv \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{x}, \boldsymbol{\theta}_i);$

  2. *Empirical loss:* $\ell_P(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n) = \frac{1}{P}\sum_{p=1}^{P} |f(\boldsymbol{x}_p) - f_n(\boldsymbol{x}_p)|^2,$ $\{\boldsymbol{x}_p\}_{p=1}^{P}$ = *batch.*

  3. *SGD:* $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \Delta t_n \, \nabla_{\boldsymbol{\theta}_i} \ell_P(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n), \qquad i = 1, \dots, n$

n = # of units;
P = # of data points

- Several puzzles:
  - ▷ *Optimization:* Does SGD converge, i.e. how well can the network be trained in practice?
  - ▷ *Approximation:* How does the error of the trained network scale with its size n / architecture?
  - ▷ *Generalization:* How does it scale with the data size P?

- *NN representation = non-linear approximation* $\Rightarrow$ *NN training = non-convex optimization problem*
  (as opposed to linear approximations, $f_n(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \phi_i(\boldsymbol{x})$, used in kernel / Galerkin methods)

Nonlinearity improves approximation power (?),
but renders analysis more complicated !

Theoretical approximation power of NN

Bach, JMLR (2017) x2.

Universal Approximation Theorem (UAT) (Barron, Cybenko, Park,...) *If the unit $\widehat{\varphi}$ is discriminatory, then given any $f \in L^2(\Omega, \nu)$ and $\epsilon > 0$:*

$$\exists \gamma^* = \textit{Radon measure} \quad \textit{such that} \quad \|f - f^*\|_{L^2(\Omega,\nu)} \leq \epsilon \quad \textit{with} \quad f^* = \int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma^*(\boldsymbol{z})$$

In addition, the function f^* can be realized as $f^* = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} c_j \widehat{\varphi}(\cdot, \boldsymbol{z}_j)$ by drawing every pair in $\{c_i, \boldsymbol{z}_i\}_{i\in\mathbb{N}}$ independently from a probability measure μ^* on $\mathbb{R} \times \hat{D}$ such that $\int_{\mathbb{R}} c\mu^*(dc, \cdot) = \gamma^*$.

Theoretical approximation power of NN

Bach, JMLR (2017) x2.

Universal Approximation Theorem (UAT) (Barron, Cybenko, Park,...) *If the unit $\widehat{\varphi}$ is discriminatory, then given any $f \in L^2(\Omega, \nu)$ and $\epsilon > 0$:*

$$\exists \gamma^* = \textit{Radon measure} \quad \textit{such that} \quad \|f - f^*\|_{L^2(\Omega,\nu)} \leq \epsilon \quad \textit{with} \quad f^* = \int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma^*(\boldsymbol{z})$$

In addition, the function f^* can be realized as $f^* = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} c_j \widehat{\varphi}(\cdot, \boldsymbol{z}_j)$ by drawing every pair in $\{c_i, \boldsymbol{z}_i\}_{i\in\mathbb{N}}$ independently from a probability measure μ^* on $\mathbb{R} \times \hat{D}$ such that $\int_{\mathbb{R}} c\mu^*(dc, \cdot) = \gamma^*$.

- Universal Approximation Theorem (UAT) quantifies the *theoretical* approximation power of a NN via its capacity to represent functions in the normed space

$$\mathcal{F}_1^{\varphi} = \{f : \Omega \to \mathbb{R} \mid f = \textstyle\int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma(\boldsymbol{z})\},$$

- UAT gives Monte-Carlo error bounds on the loss, scaling as C/n for the approximation error and $C'/P^{1/2}$ for the generalization error with data set of batch size P, with constant C and C' related to the norm of $f^* \in \mathcal{F}_1^{\varphi}$:

$$\|f^*\|_{\mathsf{TV}} = \inf\{\|\gamma^*\|_{\mathsf{TV}} \mid \textstyle\int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma^*(\boldsymbol{z}) = f^*\}$$

Colloquially:

- Consider

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\varphi(\boldsymbol{x}, \boldsymbol{\theta}_i)$$

  and assume that the paramters $\boldsymbol{\theta}_i$ are drawn independently from some probability distribution μ.

- Then as $n \to \infty$:
  - ▷ The *Law of Large Numbers (LLN)* tells us that

$$f_n(\boldsymbol{x}) \to f(\boldsymbol{x}) = \int_D \varphi(\boldsymbol{x}, \boldsymbol{\theta}) d\mu(\boldsymbol{\theta})$$

  - ▷ The *Central Limit Theorem (CLT)* tells us that

$$\sqrt{n}\,(f_n(\boldsymbol{x}) - f(\boldsymbol{x})) \to \text{some Gaussian field}$$

- That is,

$$f_n(\boldsymbol{x}) \sim f(\boldsymbol{x}) + n^{-1/2}\eta(\boldsymbol{x}) \qquad \text{for } n \gg 1$$

- Standard Monte-Carlo error scaling $O(n^{-1/2})$ rather than $O(n^{-1/d})$ (if d is the input dimension, $\boldsymbol{x} \in \mathbb{R}^d$)

Theoretical approximation power of NN

Bach, JMLR (2017) x2.

Universal Approximation Theorem (UAT) (Barron, Cybenko, Park,...) *If the unit $\widehat{\varphi}$ is discriminatory, then given any $f \in L^2(\Omega, \nu)$ and $\epsilon > 0$:*

$$\exists \gamma^* = \textit{Radon measure} \quad \textit{such that} \quad \|f - f^*\|_{L^2(\Omega,\nu)} \leq \epsilon \quad \textit{with} \quad f^* = \int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma^*(\boldsymbol{z})$$

In addition, the function f^* can be realized as $f^* = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} c_j \widehat{\varphi}(\cdot, \boldsymbol{z}_j)$ by drawing every pair in $\{c_i, \boldsymbol{z}_i\}_{i\in\mathbb{N}}$ independently from a probability measure μ^* on $\mathbb{R} \times \hat{D}$ such that $\int_{\mathbb{R}} c\mu^*(dc, \cdot) = \gamma^*$.

- Universal Approximation Theorem (UAT) quantifies the *theoretical* approximation power of a NN via its capacity to represent functions in the normed space

$$\mathcal{F}_1^{\varphi} = \{f : \Omega \to \mathbb{R} \mid f = \textstyle\int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma(\boldsymbol{z})\},$$

- UAT gives Monte-Carlo error bounds on the loss, scaling as C/n for the approximation error and $C'/P^{1/2}$ for the generalization error with data set of batch size P, with constant C and C' related to the norm of $f^* \in \mathcal{F}_1^{\varphi}$:

$$\|f^*\|_{\mathsf{TV}} = \inf\{\|\gamma^*\|_{\mathsf{TV}} \mid \textstyle\int_{\hat{D}} \widehat{\varphi}(\cdot, \boldsymbol{z}) d\gamma^*(\boldsymbol{z}) = f^*\}$$

- UAT gives, in principle, a way to quantify the quality/adequacy of the unit $\widehat{\varphi}$ (i.e. NN architecture).

- But: UAT does not indicate how to obtain γ^* and construct the NN representation in practice, i.e. it does not give approximation and generalization errors after training.

Set-up: training via double-lifting

- Hard to learn γ^* directly (see however recent work by Chizat), simpler to learn in the lifted space of probability measures μ^* on D such that $\gamma^* = \int_{\mathbb{R}} c\mu^*(dc, \cdot)$.

- To guarantee convergence of SGD we lift one more time and introduce

$$\rho^*(dw, d\boldsymbol{\theta}) \text{ = prob. meas. on } (\mathbb{R}_+, D) \quad \text{such that} \quad \mu^* = \int_0^\infty w\rho^*(dw, \cdot) \text{ = prob. meas. on } D$$

Set-up: training via double-lifting

- Hard to learn γ^* directly (see however recent work by Chizat), simpler to learn in the lifted space of probability measures μ^* on D such that $\gamma^* = \int_{\mathbb{R}} c\mu^*(dc, \cdot)$.

- To guarantee convergence of SGD we lift one more time and introduce

$$\rho^*(dw, d\boldsymbol{\theta}) \text{ = prob. meas. on } (\mathbb{R}_+, D) \quad \text{such that} \quad \mu^* = \int_0^\infty w\rho^*(dw, \cdot) \text{ = prob. meas. on } D$$

- In practice we use the function representation

$$f_n = \frac{1}{n}\sum_{i=1}^n w_i\varphi(\cdot, \boldsymbol{\theta}_i) = \frac{1}{n}\sum_{i=1}^n w_i c_i\widehat{\varphi}(\cdot, \boldsymbol{z}_i)$$

and the *regularized empirical loss*:

$$\ell_P^\lambda(f_n) \equiv \ell_P^\lambda(w_i, \boldsymbol{\theta}_1, \dots, w_n, \boldsymbol{\theta}_n) = \tfrac{1}{2}\mathbb{E}_{\nu_P}|f - f_n|^2 + \frac{\lambda}{n}\sum_{i=1}^n w_i|c_i|^q \qquad (q > 2)$$

- Regularizing term added to control the TV of $\gamma = \int_{\mathbb{R}} c\mu(dc, \cdot)$. Explicitly:

$$\ell_P^\lambda(f_n) = C_f + \frac{1}{n}\sum_{i=1}^n w_i F(\boldsymbol{\theta}_i) + \frac{1}{2n^2}\sum_{i,j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$$

where $C_f = \frac{1}{2}\mathbb{E}_{\nu_P}|f|^2$, $F(\boldsymbol{\theta}) = -\mathbb{E}_{\nu_P}[f\varphi(\cdot, \boldsymbol{\theta})] + \lambda n^{-1}\sum_{i=1}^n w_i|c_i|^q$, $K(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbb{E}_{\nu_P}[\varphi(\cdot, \boldsymbol{\theta})\varphi(\cdot, \boldsymbol{\theta}')]$.

Training by SGD— parameters = particles

- NN training by SGD over loss function under the constraint $n^{-1}\sum_{i=1}^n w_i = 1$:

$$\dot{\boldsymbol{\theta}}_i = -nw_i^{-1}\partial_{\boldsymbol{\theta}_i}\ell_P^\lambda = -\nabla F(\boldsymbol{\theta}_i) - \frac{1}{n}\sum_{j=1}^n w_j \nabla K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j),$$

$$\dot{w}_i = -n\alpha w_i\left(\partial_{w_i}\ell_P^\lambda - g\right) = -\alpha w_i F(\boldsymbol{\theta}_i) - \frac{\alpha}{n}\sum_{j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) - \alpha g w_i,$$

with $g = -n^{-1}\sum_{i=1}^n w_i F(\boldsymbol{\theta}_i) - n^{-2}\sum_{i,j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$.

Different metrics used for $\boldsymbol{\theta}$ and w!

Very easy to implement in practice — simple patch on SGD.

Training by SGD— parameters = particles

- NN training by SGD over loss function under the constraint $n^{-1}\sum_{i=1}^n w_i = 1$:

$$\dot{\boldsymbol{\theta}}_i = -nw_i^{-1}\partial_{\boldsymbol{\theta}_i}\ell_P^\lambda = -\nabla F(\boldsymbol{\theta}_i) - \frac{1}{n}\sum_{j=1}^n w_j \nabla K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j),$$

$$\dot{w}_i = -n\alpha w_i\left(\partial_{w_i}\ell_P^\lambda - g\right) = -\alpha w_i F(\boldsymbol{\theta}_i) - \frac{\alpha}{n}\sum_{j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) - \alpha g w_i,$$

with $g = -n^{-1}\sum_{i=1}^n w_i F(\boldsymbol{\theta}_i) - n^{-2}\sum_{i,j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$.

- Using interchangeability of parameters / particles, represent f_n by their empirical distribution

$$f_n(t) = \int_{\mathbb{R}_+\times D} w\varphi(\cdot, \boldsymbol{\theta}) d\rho_t^{(n)}(w, \boldsymbol{\theta}) \quad \text{with} \quad \rho_t^{(n)} = \frac{1}{n}\sum_{i=1}^n \delta_{w_i(t)}\delta_{\boldsymbol{\theta}_i(t)} = \text{probability measure}$$

Training by SGD— parameters = particles

- NN training by SGD over loss function under the constraint $n^{-1}\sum_{i=1}^n w_i = 1$:

$$\dot{\boldsymbol{\theta}}_i = -nw_i^{-1}\partial_{\boldsymbol{\theta}_i}\ell_P^\lambda = -\nabla F(\boldsymbol{\theta}_i) - \frac{1}{n}\sum_{j=1}^n w_j \nabla K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j),$$

$$\dot{w}_i = -n\alpha w_i\left(\partial_{w_i}\ell_P^\lambda - g\right) = -\alpha w_i F(\boldsymbol{\theta}_i) - \frac{\alpha}{n}\sum_{j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) - \alpha g w_i,$$

with $g = -n^{-1}\sum_{i=1}^n w_i F(\boldsymbol{\theta}_i) - n^{-2}\sum_{i,j=1}^n w_i w_j K(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$.

- Using interchangeability of parameters / particles, represent f_n by their empirical distribution

$$f_n(t) = \int_{\mathbb{R}_+\times D} w\varphi(\cdot, \boldsymbol{\theta}) d\rho_t^{(n)}(w, \boldsymbol{\theta}) \quad \text{with} \quad \rho_t^{(n)} = \frac{1}{n}\sum_{i=1}^n \delta_{w_i(t)}\delta_{\boldsymbol{\theta}_i(t)} = \text{probability measure}$$

- GD dynamics of $\rho_t^{(n)}$ = nonlinear Liouville equation

$$\partial_t \rho_t^{(n)} = \nabla\cdot\left(\rho_t^{(n)}\nabla V\right) + \alpha\partial_w\left(w(V-\bar{V})\rho_t^{(n)}\right),$$

where

$$V(\boldsymbol{\theta}, [\mu_t^{(n)}]) = F(\boldsymbol{\theta}) + \int_D K(\boldsymbol{\theta}, \boldsymbol{\theta}')\mu_t^{(n)}(d\boldsymbol{\theta}'), \quad \bar{V}[\mu_t^{(n)}] = \int_D V(\boldsymbol{\theta}, [\mu_t^{(n)}])\mu_t^{(n)}(d\boldsymbol{\theta}),$$

with $\mu_t^{(n)} = \int_0^\infty w\rho_t^{(n)}(dw, \cdot)$.

Hydrodynamic (mean field) limit

- The measure $\mu_t^{(n)} = \int_0^\infty w \rho_t^{(n)}(dw, \cdot)$ satisfies the unbalanced transport equation (Chizat, Peyré, ...)

$$\partial_t \mu_t^{(n)} = \nabla \cdot \left(\mu_t^{(n)} \nabla V\right) - \alpha(V - \bar{V})\mu_t^{(n)} \longrightarrow \text{Nonlocal transport}$$

  where V is the functional derivative of a quadratic energy $\mathcal{E}[\mu]$ = loss viewed by μ:

$$V(\boldsymbol{\theta}, [\mu]) = \frac{\delta \mathcal{E}}{\delta \mu} \quad \text{where} \quad \mathcal{E}[\mu] = \tfrac{1}{2}\mathbb{E}_{\nu_P}\left| f - \int_D \varphi(\cdot, \boldsymbol{\theta})\mu(d\boldsymbol{\theta})\right|^2 + \frac{\lambda}{q}\int_D |c|^q \mu(d\boldsymbol{\theta})$$

Quadratic objective function $\Rightarrow$ unique minimum.

Hydrodynamic (mean field) limit

- The measure $\mu_t^{(n)} = \int_0^\infty w \rho_t^{(n)}(dw, \cdot)$ satisfies the unbalanced transport equation (Chizat, Peyré, ...)

$$\partial_t \mu_t^{(n)} = \nabla \cdot \left(\mu_t^{(n)} \nabla V\right) - \alpha(V - \bar{V})\mu_t^{(n)}$$

  where V is the functional derivative of a quadratic energy $\mathcal{E}[\mu]$ = loss viewed by μ:

$$V(\boldsymbol{\theta}, [\mu]) = \frac{\delta \mathcal{E}}{\delta \mu} \quad \text{where} \quad \mathcal{E}[\mu] = \tfrac{1}{2}\mathbb{E}_{\nu_P}\left| f - \int_D \varphi(\cdot, \boldsymbol{\theta})\mu(d\boldsymbol{\theta})\right|^2 + \frac{\lambda}{q}\int_D |c|^q \mu(d\boldsymbol{\theta})$$

- Mean-field limit: if $\theta_i(0) \sim \mu_0$, then $\mu_t^{(n)} \rightharpoonup \mu_t$ as $n \to \infty$, where μ_t satisfies (McKean, Varadhan, Serfaty, ...):

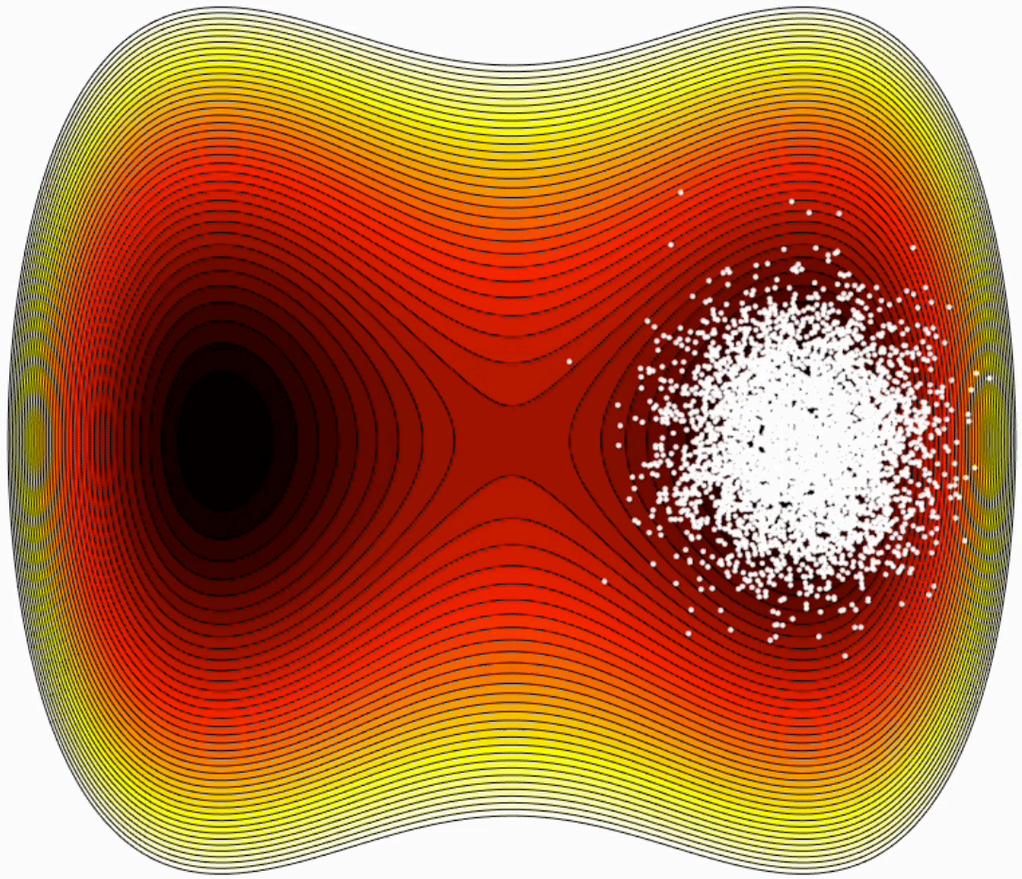
$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V) - \alpha(V - \bar{V})\mu_t$$

Propagation of chaos

Rotskoff & V.-E. arXiv:1805.00915 & NIPS18.

Similar results in:
Mei, Montanari & Nguyen arXiv:1804.06561;
Sirigano & Spiliopoulos arXiv:1805.01053.

Hydrodynamic limit in picture



1e4 particles moving in a double well-potential and repelling each other with a long range interaction potential

Hydrodynamic (mean field) limit

- The measure $\mu_t^{(n)} = \int_0^\infty w \rho_t^{(n)}(dw, \cdot)$ satisfies the unbalanced transport equation (Chizat, Peyré, ...)

$$\partial_t \mu_t^{(n)} = \nabla \cdot \left(\mu_t^{(n)} \nabla V\right) - \alpha(V - \bar{V})\mu_t^{(n)}$$

  where V is the functional derivative of a quadratic energy $\mathcal{E}[\mu]$ = loss viewed by μ:

$$V(\boldsymbol{\theta}, [\mu]) = \frac{\delta \mathcal{E}}{\delta \mu} \quad \text{where} \quad \mathcal{E}[\mu] = \tfrac{1}{2}\mathbb{E}_{\nu_P}\left| f - \int_D \varphi(\cdot, \boldsymbol{\theta})\mu(d\boldsymbol{\theta})\right|^2 + \frac{\lambda}{q}\int_D |c|^q \mu(d\boldsymbol{\theta})$$

- Mean-field limit: if $\theta_i(0) \sim \mu_0$, then $\mu_t^{(n)} \rightharpoonup \mu_t$ as $n \to \infty$, where μ_t satisfies (McKean, Varadhan, Serfaty, ...):

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V) - \alpha(V - \bar{V})\mu_t$$

Propagation of chaos

- Ruggedness of the loss landscape viewed by particles / parameters disappears at the level of their empirical distribution

- Unbalanced transport equation = GD flow in Wasserstein-Fisher-Rao metric (Otto, Villani, Ambrosio, ...)

$$\text{JKO scheme} \quad : \quad \begin{aligned} &\mu_{t+\tau/2} \in \operatorname{argmin}\left(\mathcal{E}[\mu] + \tfrac{1}{2}\tau^{-1} W_2^2(\mu, \mu_t)\right), \\ &\mu_{t+\tau} \in \operatorname{argmin}\left(\mathcal{E}[\mu] + (\alpha\tau)^{-1} D_{\mathrm{KL}}(\mu || \mu_{t+\tau/2})\right), \end{aligned} \qquad \tau \to 0.$$

Implementable at particle level

- Nonlocal transport and regularizing terms guarantee global convergence towards unique minimum, with a minimizer whose TV norm is controlled by that of the target function.

Global convergence and dynamical UAT

Prop 1 (LLN) *Let* $f_n(t) = n^{-1}\sum_{i=1}^n w_i(t)\varphi(\cdot,\boldsymbol{\theta}_i(t))$ *with* $\{w_i(t),\boldsymbol{\theta}_i(t)\}_{i=1}^n$ *evolving by SGD from well-prepared initial conditions on the empirical loss involving the target function* f^**. Then*

$$\lim_{n\to\infty} f_n(t) = f(t) = \int_D \varphi(\cdot,\boldsymbol{\theta})d\mu_t(\boldsymbol{\theta}) \qquad \textit{almost surely}$$

and $f(t)$ *satisfies*

$$\lim_{t\to\infty} f(t) = f_\lambda$$

Limits commute

where f_λ *is unique and such that*

$$\|f_\lambda\|_{TV} < \|f^*\|_{TV}, \qquad \tfrac{1}{2}\mathbb{E}_\nu|f^* - f_\lambda|^2 < \lambda\|f^*\|_{TV}^q$$

- Note that the parameters $\{\boldsymbol{\theta}_i(t)\}_{i=1}^n$ may converge to many different values (i.e. the model is not identifiable).

Very different way to approximate functions than in standard numerical analysis.

Rotskoff & V.-E. arXiv:1805.00915 & NIPS18;
Rotskoff, Jelassi, Bruna & V.-E. arXiv:1902.01843 & ICML 19.

See also: Chizat & Bach arXiv:1805.09545 & NIPS18;
Chizat arXiv:1907.10300

Approximation error via CLT

Bruna, Chen, Rotskoff & V.-E.

- For well-prepared initial conditions (i.e. iid parameters initially), CLT with usal scaling $O(n^{-1/2})$ holds at finite time

$$\sqrt{n}\big(\mu_t^{(n)} - \mu_t\big) \rightharpoonup \omega_t = \text{Gaussian measure} \quad \text{as } n \to \infty, \text{ in law}$$

- Accordingly:

$$\sqrt{n}\,(f_n(t) - f(t)) \to g(t) = \text{Gaussian field} \quad \text{as } n \to \infty, \text{ in law}$$

- We can write down an evolution equation for ω_t (Braun-Hepp, Sznitman, ...) and use it to show that the covariance of $g(t)$ is controlled as $t \to \infty$.

Prop 2 (Approximation error) *Let $f_n(t) = n^{-1}\sum_{i=1}^n w_i(t)\varphi(\cdot, \boldsymbol{\theta}_i(t))$ with $\{w_i(t), \boldsymbol{\theta}_i(t)\}_{i=1}^n$ evolving by SGD from well-prepared initial conditions on the regularized empirical loss. Then*

$$\lim_{t\to\infty}\lim_{n\to\infty} n\mathbb{E}_\nu\,|f_n(t) - f(t)|^2 = C \leq \infty$$

with C related to the TV norm of the target f^*.

- Validate Monte-Carlo error bound (thanks to propagation of chaos), with constant *a priori* known.

Rotskoff & V.-E. arXiv:1805.00915 & NIPS18.

See also: Sirignano & Spiliopoulos arXiv:1808.09372

Generalization error

Bruna, Chen, Rotskoff & V.-E.

- How good is the NN trained by SGD on the empirical loss outside of the training set?
- Standard results (see e.g. Bach JMLR 17) assert that

Prop 3 (Generalization error) *Let $f_\lambda^P = \int_D \varphi(\cdot,\boldsymbol{\theta})\mu_\lambda^P(d\theta)$ be the unique minimizer of the regularized empirical loss with a data set (batch) of size P and target function f^*. Then*

$$\mathbb{E}_{batch}\,\mathbb{E}_\nu \left|f_\lambda^P - f^*\right|^2 \le C' P^{-1/2} + \lambda |f^*|_{TV}^q$$

no overfitting at $n = \infty$!

with explicit constant C' depending on the TV norm of the target function f^*.

- Our results suggest that *this error is achievable by training.* That is, if $f_n(t) = n^{-1}\sum_{i=1}^n w_i(t)\varphi(\cdot,\boldsymbol{\theta}_i(t))$ with $\{w_i(t),\boldsymbol{\theta}_i(t)\}_{i=1}^n$ evolving by SGD from well-prepared initial conditions on the regularized empirical loss, and $f_n = \lim_{t\to\infty} f_n(t)$, we expect

$$\mathbb{E}_{\text{batch}}\,\mathbb{E}_\nu \,|f_n - f^*|^2 \le Cn^{-1} + C' P^{-1/2} + \lambda |f^*|_{\text{TV}}^q$$

with explicit constant C and C' depending on the TV norm of the target function f^*.

- Main difficulty here is to show that the CLT provides an estimate of the approximation error at finite n.

Generalization to deep neural networks

- Mean field approach generalizable e.g. to CNN of the type

$$f_n : \mathbb{R}^d \to \mathbb{R}; \quad \boldsymbol{x} \mapsto f_n(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{X}_K^{(n)}(\boldsymbol{x})$$

where $\boldsymbol{w} \in \mathbb{R}^{d_K}$ is a fixed vector $\boldsymbol{X}_K^{(n)}(\boldsymbol{x})$ is the final output of an iteration through K layers

$$\boldsymbol{X}_0^{(n)} = \boldsymbol{x} \in \mathbb{R}^d, \qquad \boldsymbol{X}_{k+1}^{(n)} = \frac{1}{n}\sum_{i=1}^{n} \varphi_k(\boldsymbol{X}_k^{(n)}, \boldsymbol{\theta}_{k,i}) \in \mathbb{R}^{d_{k+1}}, \quad k = 0, \dots, K-1.$$

- Requires one measure μ_k per layer.

- However, loss function no longer convex in μ_k.
  - ▷ Can we prove convergence of unbalanced SGD?
  - ▷ Can we determine approximation and generalization errors?

Comparison with NTK

Jacot, Gabriel, Hongler, arXiv:1806.07572.

- NTK uses the scaling $$f_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \varphi(\cdot, \boldsymbol{\theta}_i)$$

- Existence of a (deterministic) limit for the function requires that $\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^0 + n^{-1/2}\tilde{\boldsymbol{\theta}}_i$, where $\boldsymbol{\theta}_i^0$ satisfies

  centering condition: $$\sum_{i=1}^{n} \varphi(\cdot, \boldsymbol{\theta}_i^0) = 0$$

- Implies that the function is $$f_n = \frac{1}{n} \sum_{i=1}^{n} \tilde{\boldsymbol{\theta}}_i \cdot \nabla \varphi(\cdot, \boldsymbol{\theta}_i^0) + O(n^{-1/2}).$$

- Brings us back in the mean-field scaling, but with the important difference that only the $\tilde{\boldsymbol{\theta}}_i$'s are being trained. Representation is effectively linear in the training parameters $\Rightarrow$ *random feature kernel*.

$$\partial_t f_t = \int_\Omega M(\boldsymbol{x}, \boldsymbol{x}', [\mu_t])(f(\boldsymbol{x}) - f_t(\boldsymbol{x}'))d\nu(\boldsymbol{x}') \qquad \text{mean-field scaling}$$

$$\partial_t f_t = \int_\Omega M(\boldsymbol{x}, \boldsymbol{x}', [\mu_0])(f(\boldsymbol{x}) - f_t(\boldsymbol{x}'))d\nu(\boldsymbol{x}') \qquad \text{NTK scaling}$$

where

$$M(\boldsymbol{x}, \boldsymbol{x}', [\mu]) = \int_D \nabla \varphi(\boldsymbol{x}, \boldsymbol{\theta}) \cdot \nabla \varphi(\boldsymbol{x}', \boldsymbol{\theta}) d\mu(\boldsymbol{\theta})$$

Comparison with NTK

Jacot, Gabriel, Hongler, arXiv:1806.07572.

- Training in the NTK regime is easier. However:

- Lead to different spaces for learning
  - ▷ Mean-field learns in the space $\mathcal{F}_1^\varphi \subset L^2(\Omega, \nu)$ representable as

$$f = \int_D \varphi(\cdot, \boldsymbol{z}) d\gamma(\boldsymbol{z}) \quad \text{with } \gamma = \text{Radon measure such that} \quad \|\gamma\|_{\mathsf{TV}} = \int_D |d\gamma(\boldsymbol{z})| < \infty$$

  - ▷ NTK learns in the space of functions in $L^2(\Omega, \nu)$ representable as

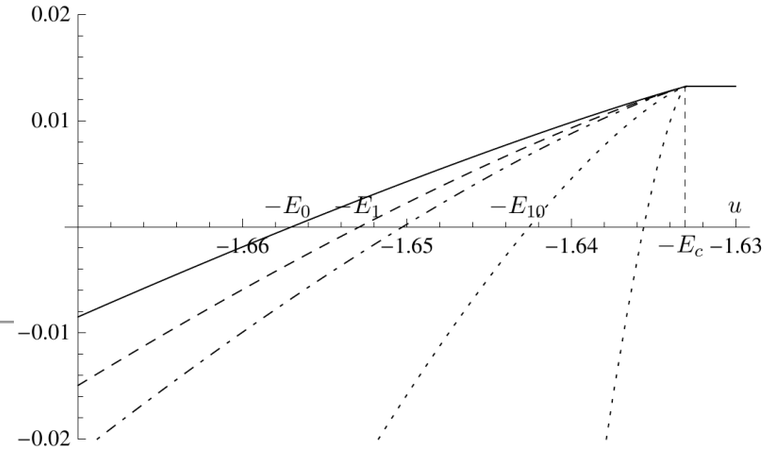
$$f = \int_D g(\boldsymbol{z}) \varphi(\cdot, \boldsymbol{z}) d\widehat{\mu}_0(\boldsymbol{z}) \quad \text{with} \quad \int_D |g(\boldsymbol{z})|^2 d\widehat{\mu}_0(\boldsymbol{z}) < \infty$$

- NTK space is smaller, which leads to larger (possibly infinite) approximation and generalization errors.

No free lunch!

Similar conclusion in:
Chizat, Oyallon & Bach arXiv:1812.07956;
Ghorbani, Mei, Misiakiewicz, & Montanari arXiv:1906.08899.

A test case: Spherical 3-spin model



e.g. Auffinger, Ben Arous, and Černý

- Spherical 3-spin model: $f : S^{d-1}(\sqrt{d}) \to \mathbb{R}$ given by

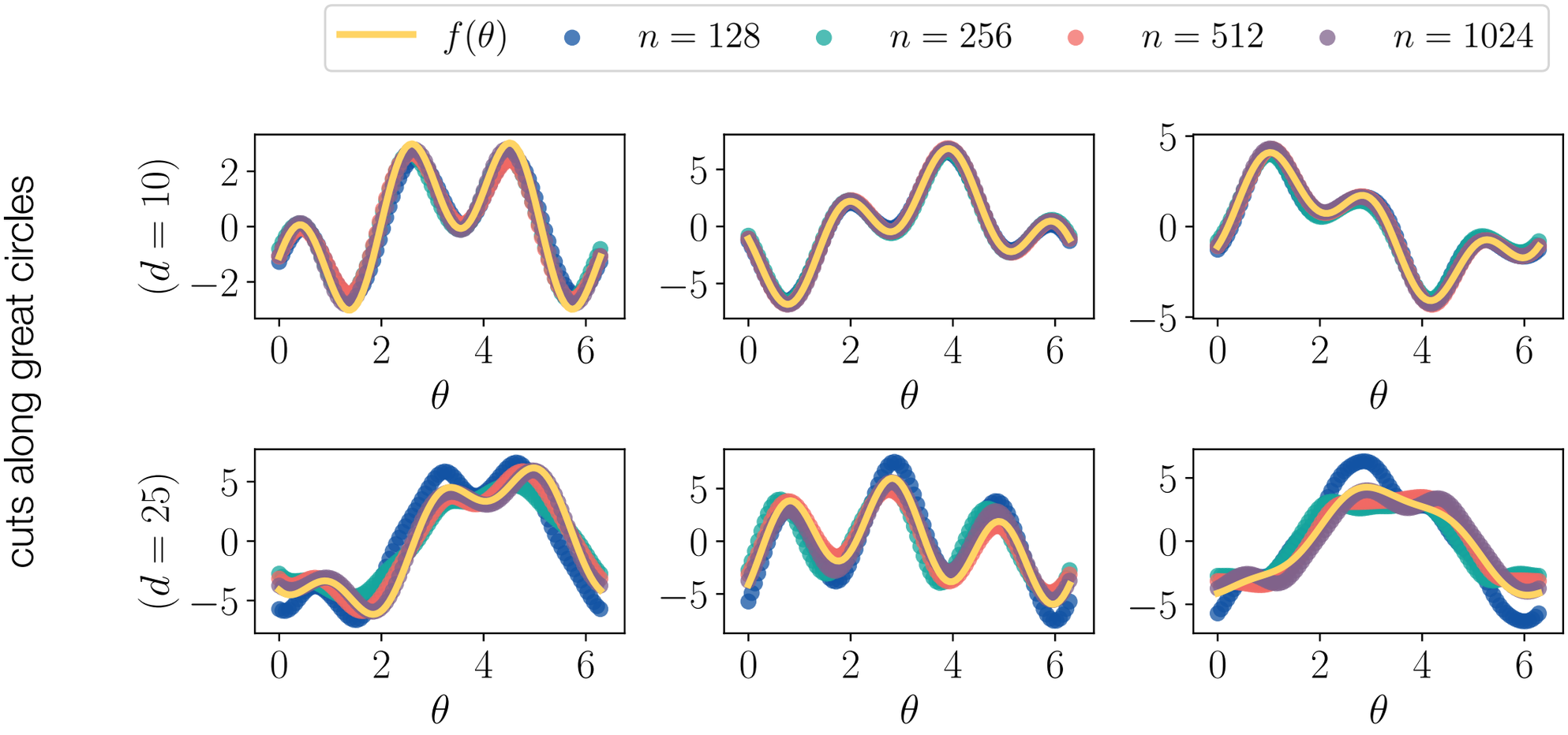
$$f(\boldsymbol{x}) = \frac{1}{d} \sum_{p,q,r=1}^{d} a_{p,q,r} x_p x_q x_r, \qquad \boldsymbol{x} \in S^{d-1}\left(\sqrt{d}\right) \subset \mathbb{R}^d$$

  where the coefficients $\{a_{p,q,r}\}_{p,q,r=1}^{d}$ are independent Gaussian random variables with mean zero and variance one.

- Complex function with a number of critical points that grows exponentially with the dimensionality d.
- Previous works drew a parallel between the glassy 3-spin function and generic loss functions.
- In contrast, we use the 3-spin function as a difficult target for approximation by neural networks, that is:
  - ▷ we train networks to learn f with a particular realization of $a_{p,q,r}$, and;
  - ▷ we study the accuracy of that representation as a function of the number n of units.

Numerical results on 3-spin model

$$f(\boldsymbol{x}) = \frac{1}{d} \sum_{p,q,r=1}^{d} a_{p,q,r} x_p x_q x_r, \qquad \boldsymbol{x} \in S^{d-1}\left(\sqrt{d}\right) \subset \mathbb{R}^d \qquad a_{p,q,r} \sim N(0,1)$$

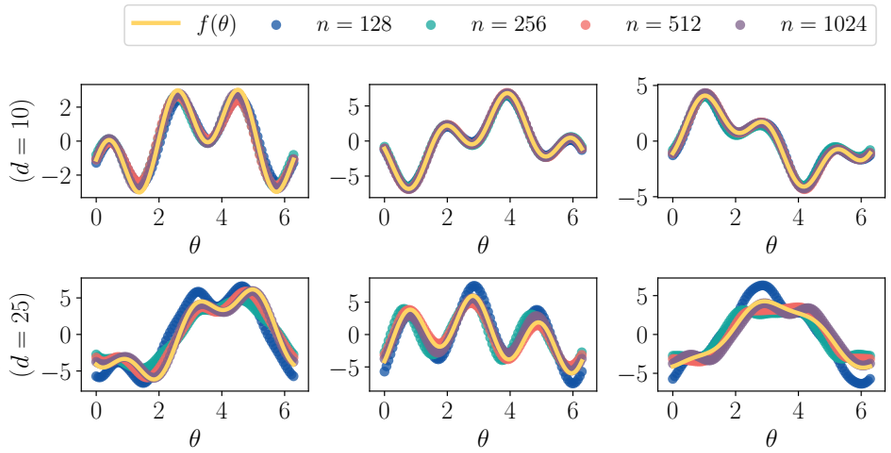


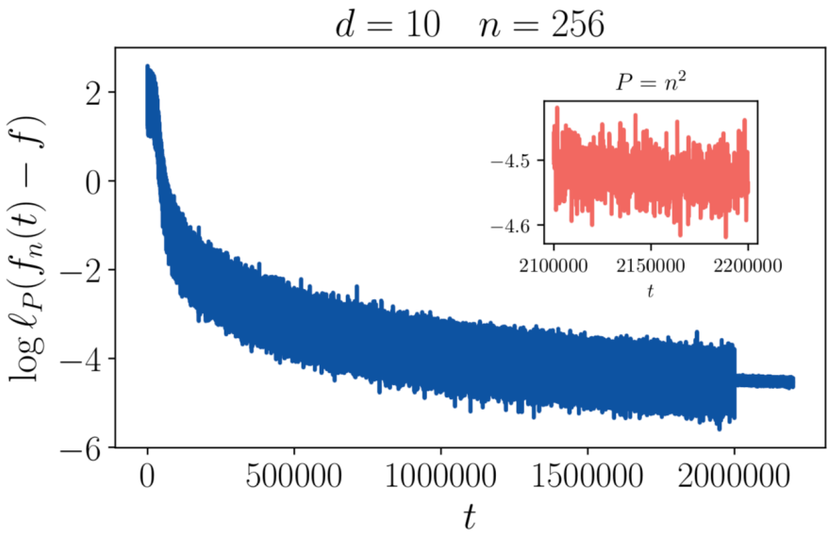
$$x_i(\theta) = \sqrt{d}\cos(\theta), \qquad x_j(\theta) = \sqrt{d}\sin(\theta), \qquad x_k(\theta) = 0 \quad \forall k \neq i, j.$$

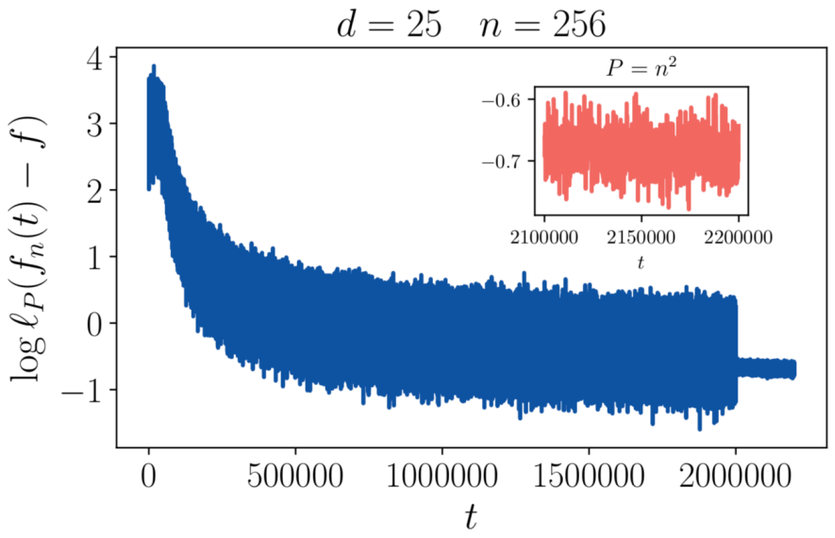
NB: Two grid points per dimension requires $2^{25} = 33,554,432$ grid points

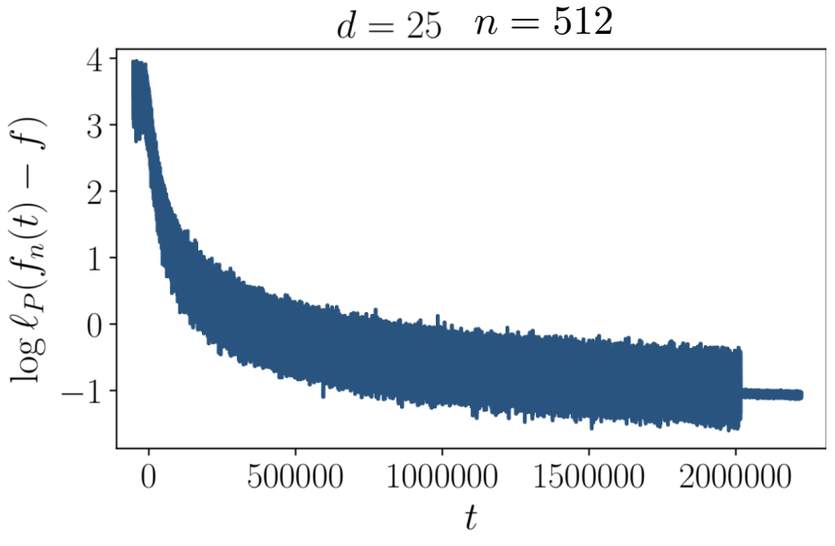
Numerical results on 3-spin model

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i h(\boldsymbol{a}_i \cdot \boldsymbol{x} + b_i), \qquad h(z) = 1/(1+e^{-z}).$$

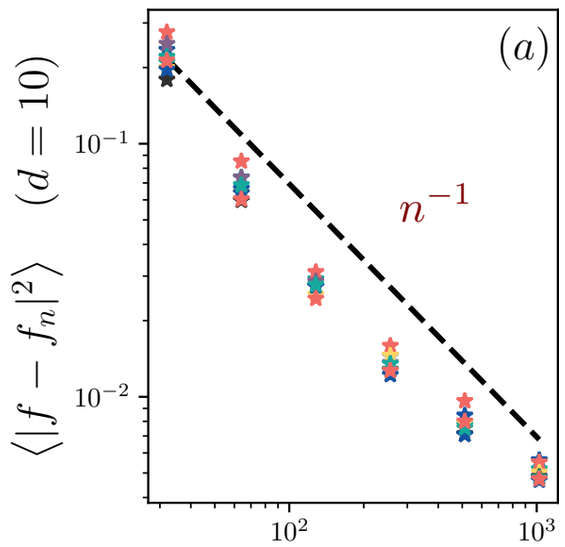


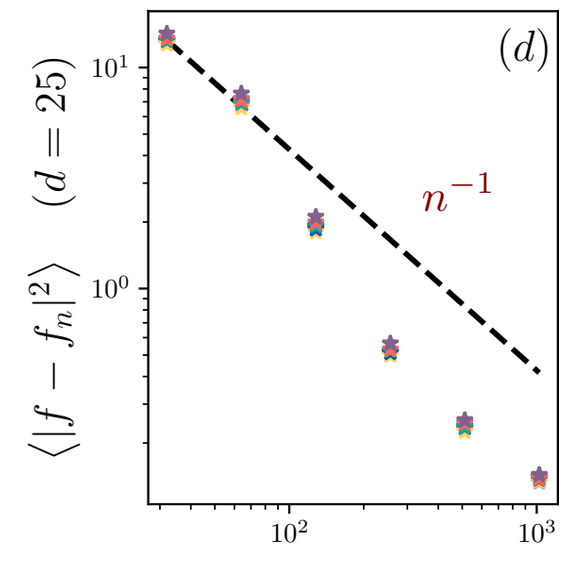






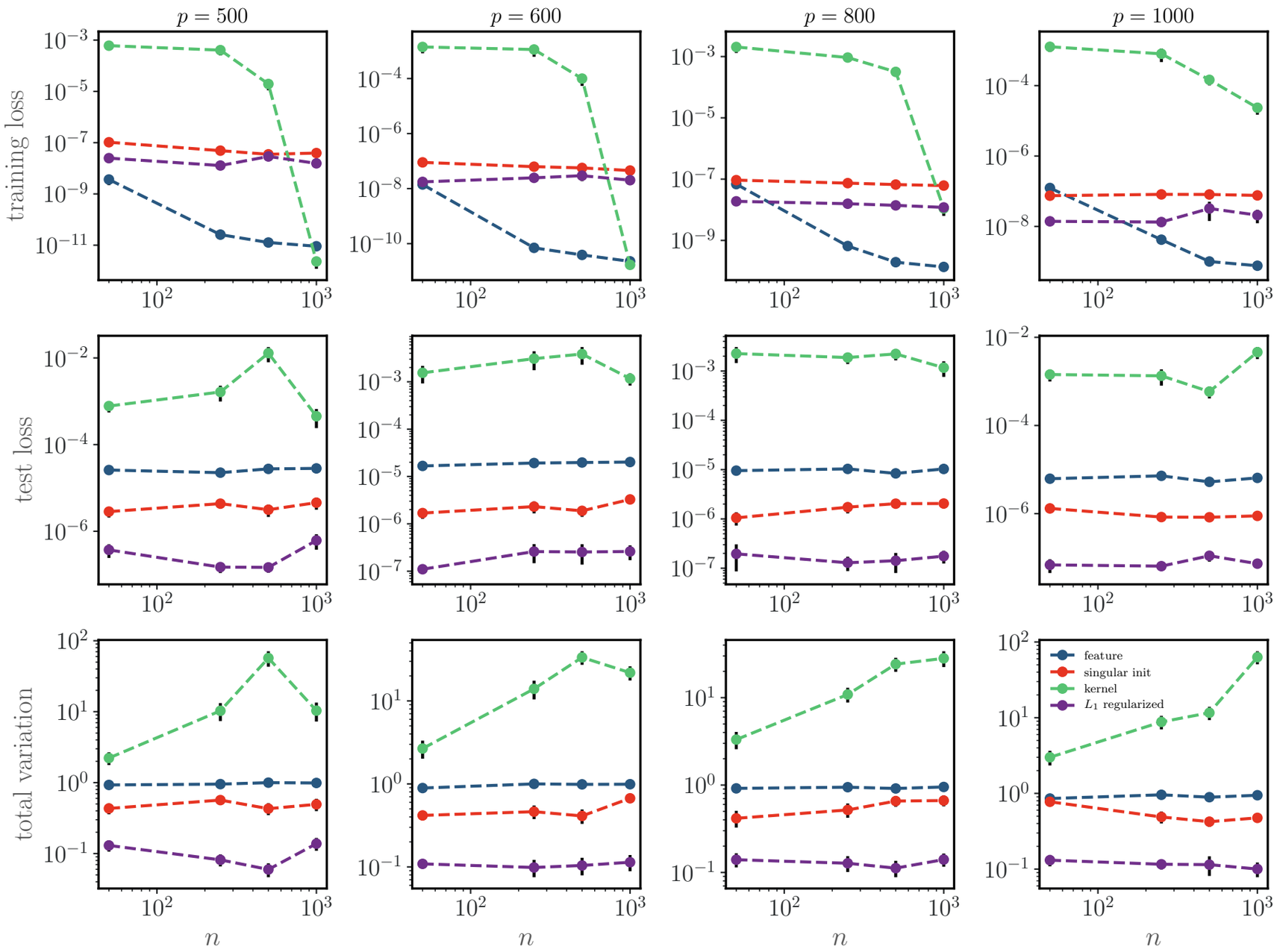
Empirical loss vs SGD training time.
At time 2E6, the batch size is increased to initiate an additional quench.





Error scaling

Testing generalization error on student-teacher model



ReLU teacher on the sphere

Concluding remarks

- Theoretical results support the empirical evidence that NN can outperform standard interpolation methods and massively reduce the cost of representing functions in high dimensional spaces.

- Gives a framework to understand how to carefully design of the NN used in the representation: how to use the data, in which loss, with which unit, etc.

  — NN must be tailored to the problem at hand to be accurate and efficient!

- *In particular, it is key to design NN such that the constant C and C' in*

$$\mathbb{E}_{\text{batch}}\, \mathbb{E}_{\nu}\, |f_n - f^*|^2 \leq C n^{-1} + C' P^{-1/2} + \lambda |f^*|^q_{\text{TV}}$$

  depend "gently" on input dimension d.