Empirical risk minimization over deep neural networks overcomes the curse of dimensionality in the numerical approximation of Kolmogorov equations

Julius Berner[1], Philipp Grohs[1,2], Arnulf Jentzen[3]

[1]Faculty of Mathematics, University of Vienna
[2]Research Platform DataScience@UniVienna, University of Vienna
[3]Institute for Analysis and Numerics, University of Münster

The Power of Deep Learning [10]

- automatic generation of photo-realistic images (deep generative adversarial networks)

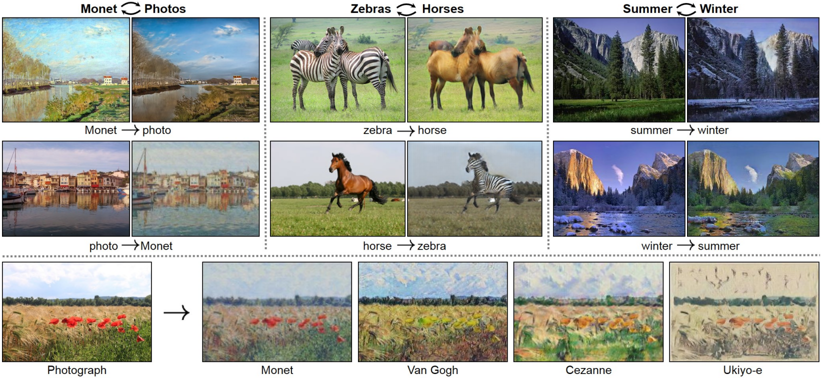


Figure: render natural photographs into different styles - Zhu et al. '17

The Power of Deep Learning [8]

- automatic game playing with super-human performance (deep Q-learning)

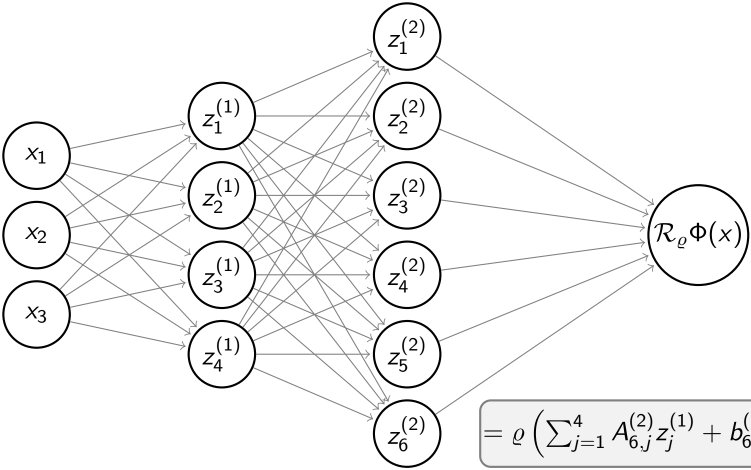
Video: Learning to play 'ATARI outbreak' - Mnih et al. '15 (https://youtu.be/V1eYniJ0Rnk)

The Power of Deep Learning

'Machine learning works spectacularly well, but mathematicians aren't quite sure why.' - Daubechies '15

Artificial Feed-Forward Neural Network

- stacking together artificial neurons



$$= \varrho\left(\sum_{j=1}^{4} A_{6,j}^{(2)} z_j^{(1)} + b_6^{(2)}\right)$$

Artificial Feed-Forward Neural Network

- stacking together artificial neurons
- network architecture $N = (N_0, N_1, \ldots, N_L)$ specifying the number of artifical neurons N_l in each of the L layers

♣ setting: input dimension $N_0 = d$, output dimension $N_L = n$

Artificial Feed-Forward Neural Network

- stacking together artificial neurons
- network architecture $N = (N_0, N_1, \dots, N_L)$ specifying the number of artifical neurons N_l in each of the L layers

♣ setting: input dimension $N_0 = d$, output dimension $N_L = n$

♣ ϱ is Lipschitz continuous, e.g.

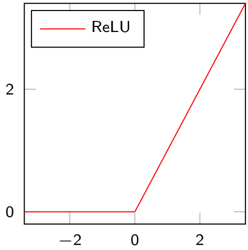
Artificial Feed-Forward Neural Network

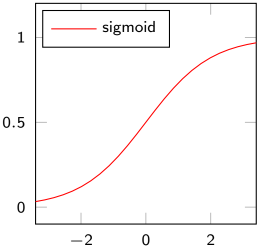
- stacking together artificial neurons
- network architecture $N = (N_0, N_1, \ldots, N_L)$ specifying the number of artifical neurons N_l in each of the L layers

♣ setting: input dimension $N_0 = d$, output dimension $N_L = n$

♣ ϱ is Lipschitz continuous, e.g.

  - rectified linear unit $\varrho(x) = \mathrm{ReLU}(x) = max\{x, 0\}$
  - sigmoid (logistic) $\varrho(x) = \frac{1}{1+e^{-x}}$





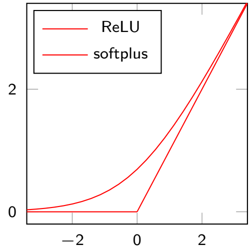
Artificial Feed-Forward Neural Network

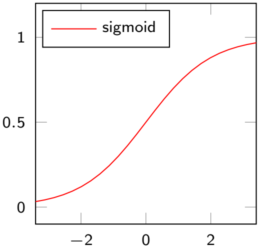
- stacking together artificial neurons
- network architecture $N = (N_0, N_1, \ldots, N_L)$ specifying the number of artifical neurons N_l in each of the L layers

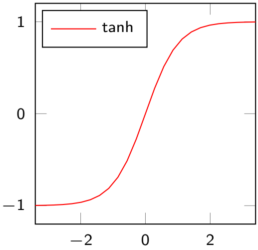
♣ setting: input dimension $N_0 = d$, output dimension $N_L = n$

♣ ϱ is Lipschitz continuous, e.g.

- rectified linear unit $\varrho(x) = \mathrm{ReLU}(x) = max\{x, 0\}$
- sigmoid (logistic) $\varrho(x) = \frac{1}{1+e^{-x}}$

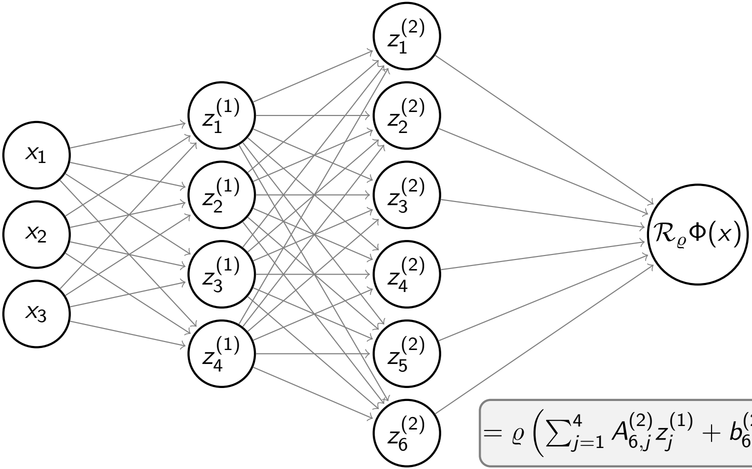






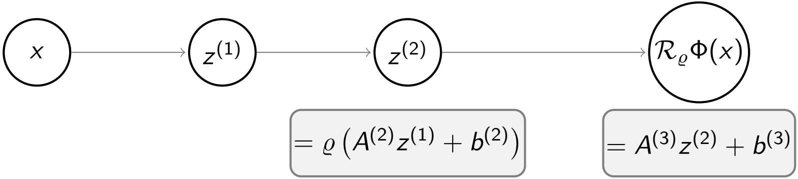
Artificial Feed-Forward Neural Network

- example: $N = (3, 4, 6, 1)$, $d = 3$, $n = 1$, $L = 3$ ('deep')



$$= \varrho\left(\sum_{j=1}^{4} A_{6,j}^{(2)} z_j^{(1)} + b_6^{(2)}\right)$$

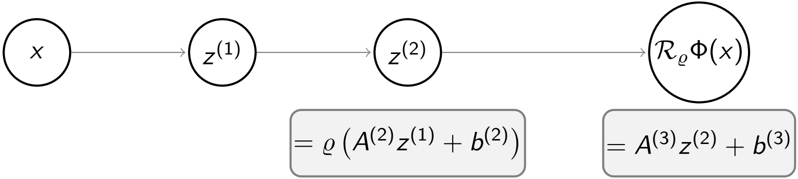
Artificial Feed-Forward Neural Network



Artificial Feed-Forward Neural Network

- set of parametrizations with architecture N and parameter bound R

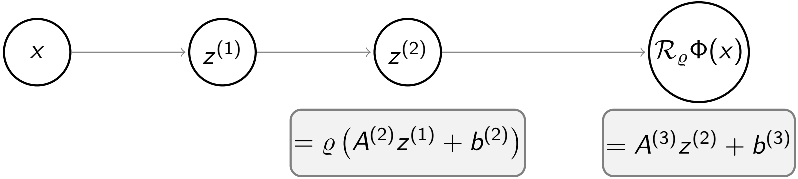
$$\mathcal{P}_N^R := \left\{ \Phi = ((A^{(\ell)}, b^{(\ell)}))_{\ell=1}^L \;\middle|\; \begin{array}{l} A^{(\ell)} \in [-R, R]^{N_\ell \times N_{\ell-1}}, \\ b^{(\ell)} \in [-R, R]^{N_\ell} \end{array} \right\}$$



Artificial Feed-Forward Neural Network

- set of parametrizations with architecture N and parameter bound R

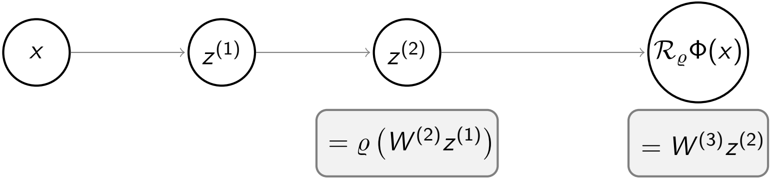
$$\mathcal{P} = \mathcal{P}_N^R := \left\{ \Phi = ((A^{(\ell)}, b^{(\ell)}))_{\ell=1}^L \;\middle|\; \begin{array}{l} A^{(\ell)} \in [-R, R]^{N_\ell \times N_{\ell-1}}, \\ b^{(\ell)} \in [-R, R]^{N_\ell} \end{array} \right\}$$



Artificial Feed-Forward Neural Network

- set of parametrizations with architecture N and parameter bound R

$$\mathcal{P} = \mathcal{P}_N^R := \left\{ \Phi = ((A^{(\ell)}, b^{(\ell)}))_{\ell=1}^L \;\middle|\; \begin{array}{l} A^{(\ell)} \in [-R, R]^{N_\ell \times N_{\ell-1}}, \\ b^{(\ell)} \in [-R, R]^{N_\ell} \end{array} \right\}$$



- realization map with activation function ϱ on compact space $K \subseteq \mathbb{R}^d$

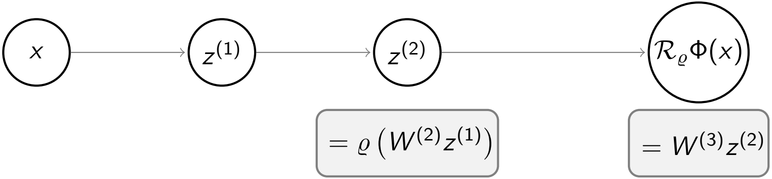
$$\begin{aligned} \mathcal{R}_\varrho^K \colon \mathcal{P} &\to \mathcal{W}^{1,\infty}(K) \subseteq \mathcal{C}(K) \\ \Phi &\mapsto W^{(L)} \circ \varrho \circ W^{(L-1)} \circ \cdots \circ \varrho \circ W^{(1)}, \end{aligned}$$

where $W^{(\ell)}(z) := A^{(\ell)} z + b^{(\ell)}$ and ϱ is applied component-wise

Artificial Feed-Forward Neural Network

- set of parametrizations with architecture N and parameter bound R

$$\mathcal{P} = \mathcal{P}_N^R := \left\{ \Phi = \left((A^{(\ell)}, b^{(\ell)}) \right)_{\ell=1}^L \;\middle|\; \begin{array}{l} A^{(\ell)} \in [-R, R]^{N_\ell \times N_{\ell-1}}, \\ b^{(\ell)} \in [-R, R]^{N_\ell} \end{array} \right\}$$



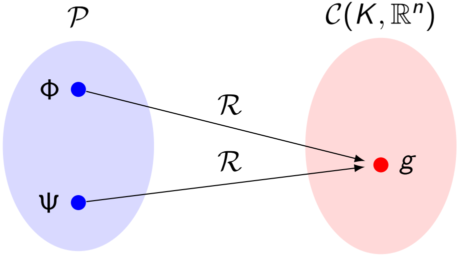
- realization map with activation function ϱ on compact space $K \subseteq \mathbb{R}^d$

$$\mathcal{R} = \mathcal{R}_\varrho^K : \mathcal{P} \to \mathcal{W}^{1,\infty}(K) \subseteq \mathcal{C}(K)$$
$$\Phi \mapsto W^{(L)} \circ \varrho \circ W^{(L-1)} \circ \dots \circ \varrho \circ W^{(1)},$$

where $W^{(\ell)}(z) := A^{(\ell)} z + b^{(\ell)}$ and ϱ is applied component-wise

(Undesirable) Properties of the Realization Map [3, 4, 9]

- not injective



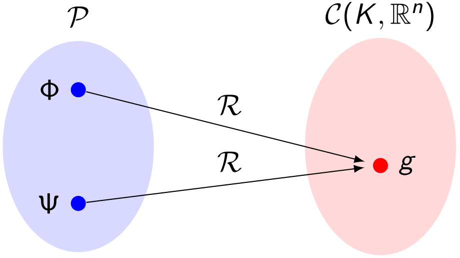
(Undesirable) Properties of the Realization Map [3, 4, 9]

- not injective

Example

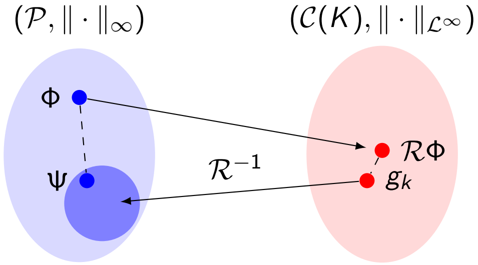
$\mathcal{R}(\Phi) = \mathcal{R}(\Psi) \equiv 0$ with

$$\Phi = \big((A_1, b_1), \dots, (A_{L-1}, b_{L-1}), (0,0)\big)$$
$$\Psi = \big((B_1, c_1), \dots, (B_{L-1}, c_{L-1}), (0,0)\big)$$



(Undesirable) Properties of the Realization Map [3, 4, 9]

- not inverse stable w.r.t. $\|\cdot\|_{\mathcal{L}^\infty}$ norm



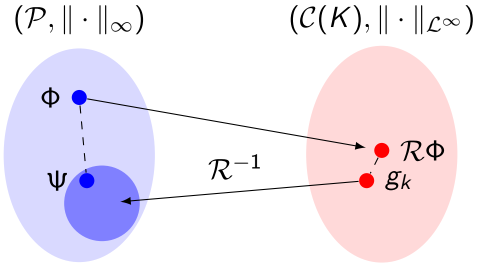
(Undesirable) Properties of the Realization Map [3, 4, 9]

- not inverse stable w.r.t. $\|\cdot\|_{\mathcal{L}^\infty}$ norm

Theorem (failure of inverse stability - Petersen et al. '18)

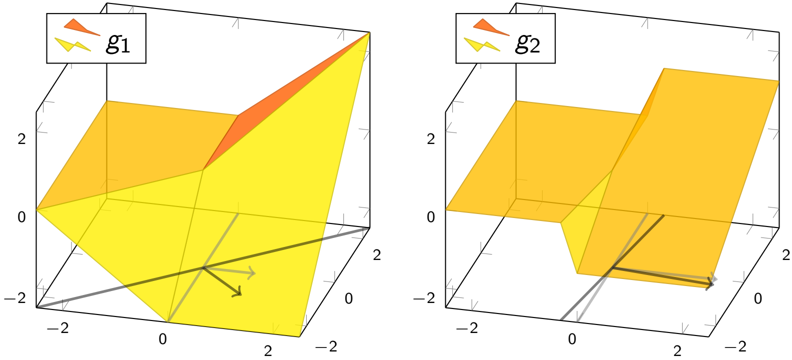
There exist $\Phi \in \mathcal{P}$ and $(g_k) \subseteq \mathcal{R}(\mathcal{P})$ with

$$\|\mathcal{R}\Phi - g_k\|_{\mathcal{L}^\infty} \to 0 \quad \text{and} \quad \inf_{k\in\mathbb{N},\ \Psi\in\mathcal{R}^{-1}(g_k)} \|\Phi - \Psi\|_\infty \geq c.$$



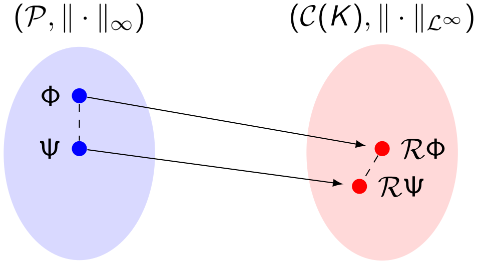
(Undesirable) Properties of the Realization Map [3, 4, 9]

- not inverse stable w.r.t. $\|\cdot\|_{\mathcal{L}^\infty}$ norm



Properties of the Realization Map [1, 4, 9]

- Lipschitz continuous w.r.t. $\|\cdot\|_{\mathcal{L}^\infty}$ norm



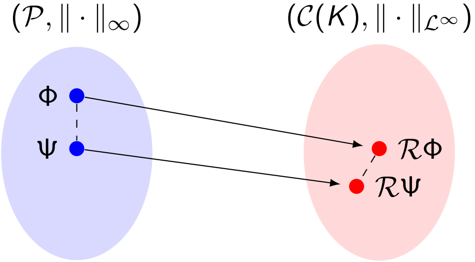
Properties of the Realization Map [1, 4, 9]

- Lipschitz continuous w.r.t. $\|\cdot\|_{\mathcal{L}^\infty}$ norm

Lemma (quantitative version for ReLU activation)

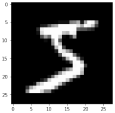
For every $\Phi, \Psi \in \mathcal{P}$ it holds that

$$\left\|\mathcal{R}_{\mathrm{ReLU}}\Phi - \mathcal{R}_{\mathrm{ReLU}}\Psi\right\|_{\mathcal{L}^\infty} \leq c(K)\left(6R\|N\|_\infty\right)^L \|\Phi - \Psi\|_\infty.$$



(Deep) Learning

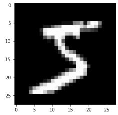
- training data $z^i = (x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^n$, $i = 1, \dots, m$



$= x^i \quad \longmapsto \quad y^i = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$

(Deep) Learning

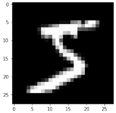
- training data $z^i = (x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^n$, $i = 1, \dots, m$
- loss function $\mathcal{E}_z \colon \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n) \to \mathbb{R}_+$



$$= x^i \quad \longmapsto \quad y^i = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$$

(Deep) Learning

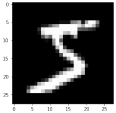
- training data $z^i = (x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^n$, $i = 1, \ldots, m$
- loss function $\mathcal{E}_z \colon \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n) \to \mathbb{R}_+$, e.g.
  - quadratic loss $\mathcal{E}_z(g) = \left\| g(x) - y \right\|_2^2$



$= x^i \quad \longmapsto \quad y^i = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$

(Deep) Learning

- training data $z^i = (x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^n$, $i = 1, \dots, m$
- loss function $\mathcal{E}_z \colon \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n) \to \mathbb{R}_+$, e.g.
  - quadratic loss $\mathcal{E}_z(g) = \|g(x) - y\|_2^2$
  - softmax + cross-entropy $\mathcal{E}_z(g) = \sum_{j=1}^{n} -y_j \log\left(\frac{\exp g_j(x)}{\sum_{k=1}^{n} \exp g_k(x)}\right)$



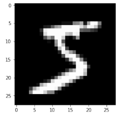
$= x^i \quad \longmapsto \quad y^i = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$

(Deep) Learning

- training data $z^i = (x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^n$, $i = 1, \dots, m$
- loss function $\mathcal{E}_z \colon \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n) \to \mathbb{R}_+$, e.g.
  - quadratic loss $\mathcal{E}_z(g) = \left\| g(x) - y \right\|_2^2$
  - softmax + cross-entropy $\mathcal{E}_z(g) = \sum_{j=1}^{n} -y_j \log \left(\frac{\exp g_j(x)}{\sum_{k=1}^{n} \exp g_k(x)} \right)$

Definition (empirical risk minimization (ERM) ⇒ empirical target network)

$$\Phi^{\text{emp}} \in \underset{\Phi \in \mathcal{P}}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \mathcal{E}_{z^i}(\mathcal{R}\Phi)$$



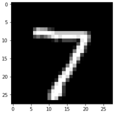
$= x^i \quad \xmapsto{\mathcal{R}\Phi^{\text{emp}}} \quad y^i = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$

(Deep) Learning

- training data $z^i = (x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^n$, $i = 1, \dots, m$
- loss function $\mathcal{E}_z \colon \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n) \to \mathbb{R}_+$, e.g.
  - quadratic loss $\mathcal{E}_z(g) = \left\| g(x) - y \right\|_2^2$
  - softmax + cross-entropy $\mathcal{E}_z(g) = \sum_{j=1}^{n} -y_j \log \left(\frac{\exp g_j(x)}{\sum_{k=1}^n \exp g_k(x)} \right)$

Definition (empirical risk minimization (ERM) $\Rightarrow$ empirical target network)

$$\Phi^{\text{emp}} \in \operatorname*{argmin}_{\Phi \in \mathcal{P}} \frac{1}{m} \sum_{i=1}^{m} \mathcal{E}_{z^i}(\mathcal{R}\Phi)$$



$$= x \quad \xrightarrow{\mathcal{R}\Phi^{\text{emp}}} \quad y = (0, 0.3, 0.1, 0, 0, 0, 0, 0.6, 0, 0)$$

Statistical Learning Theory

- $\big((z^i)\big)_{i=1}^m$ are realizations of i.i.d. samples drawn from the distribution of underlying (unknown) data

$$Z = (X, Y)\colon \Omega \to K \times [-D, D]^n \subseteq \mathbb{R}^d \times \mathbb{R}^n$$

on a suitable probability space $(\Omega, \mathcal{F}, \mathbb{P})$

Statistical Learning Theory

♣ $\big((z^i)\big)_{i=1}^m$ are realizations of i.i.d. samples drawn from the distribution of underlying (unknown) data

$$Z = (X, Y)\colon \Omega \to K \times [-D, D]^n \subseteq \mathbb{R}^d \times \mathbb{R}^n$$

on a suitable probability space $(\Omega, \mathcal{F}, \mathbb{P})$

Definition (learning problem ⇒ regression function)

$$\widehat{g} \in \underset{g \in \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n)}{\operatorname{argmin}} \mathbb{E}\left[\mathcal{E}_Z(g)\right]$$

Statistical Learning Theory

- $\big((z^i)\big)_{i=1}^m$ are realizations of i.i.d. samples drawn from the distribution of underlying (unknown) data

$$Z = (X, Y) \colon \Omega \to K \times [-D, D]^n \subseteq \mathbb{R}^d \times \mathbb{R}^n$$

on a suitable probability space $(\Omega, \mathcal{F}, \mathbb{P})$

Definition (learning problem ⇒ regression function)

$$\widehat{g} \in \operatorname*{argmin}_{g \in \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n)} \mathbb{E}\left[\mathcal{E}_Z(g)\right]$$

Definition (deep learning ⇒ best approximation)

$$\Phi^{\text{best}} \in \operatorname*{argmin}_{\Phi \in \mathcal{P}} \mathbb{E}\left[\mathcal{E}_Z(\mathcal{R}\Phi)\right]$$

(Colloquial) Error Analysis

underlying data $Z = (X, Y)$

$\widehat{g} \in \operatorname{argmin}_{g \in \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n)} \mathbb{E}\left[\mathcal{E}_Z(g)\right]$

(Colloquial) Error Analysis

underlying data $Z = (X, Y)$

$\widehat{g} \in \operatorname{argmin}_{g \in \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n)} \mathbb{E}\left[\mathcal{E}_Z(g)\right]$

hypothesis class

approximation error

neural networks given by $\mathcal{P} = \mathcal{P}_N^R$, activation ϱ

$\Phi^{\text{best}} \in \operatorname{argmin}_{\Phi \in \mathcal{P}} \mathbb{E}\left[\mathcal{E}_Z(\mathcal{R}\Phi)\right]$

(Colloquial) Error Analysis

underlying data $Z = (X, Y)$

hypothesis class

neural networks given by $\mathcal{P} = \mathcal{P}_N^R$, activation ϱ

sampling

$Z^i \sim Z$ i.i.d. $(i = 1, \dots, m)$

$\widehat{g} \in \operatorname{argmin}_{g \in \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n)} \mathbb{E}\left[\mathcal{E}_Z(g)\right]$

approximation error

$\Phi^{\text{best}} \in \operatorname{argmin}_{\Phi \in \mathcal{P}} \mathbb{E}\left[\mathcal{E}_Z(\mathcal{R}\Phi)\right]$

estimation error

$\Phi^{\text{emp}} \in \operatorname{argmin}_{\Phi \in \mathcal{P}} \frac{1}{m} \sum_{i=1}^m \mathcal{E}_{Z^i}(\mathcal{R}\Phi)$

(Colloquial) Error Analysis

underlying data $Z = (X, Y)$	$\widehat{g} \in \operatorname{argmin}_{g \in \mathcal{L}^0(\mathbb{R}^d, \mathbb{R}^n)} \mathbb{E}\left[\mathcal{E}_Z(g)\right]$
hypothesis class	approximation error
neural networks given by $\mathcal{P} = \mathcal{P}_N^R$, activation ϱ	$\Phi^{\text{best}} \in \operatorname{argmin}_{\Phi \in \mathcal{P}} \mathbb{E}\left[\mathcal{E}_Z(\mathcal{R}\Phi)\right]$
sampling	estimation error
$Z^i \sim Z$ i.i.d. $(i = 1, \dots, m)$	$\Phi^{\text{emp}} \in \operatorname{argmin}_{\Phi \in \mathcal{P}} \frac{1}{m} \sum_{i=1}^m \mathcal{E}_{Z^i}(\mathcal{R}\Phi)$
stoch. gradient descent	optimization error
n iterations, batches (I_n), learning rate λ	$\Phi_{n+1} = \Phi_n - \frac{\lambda}{\lvert I_n \rvert} \sum_{i \in I_n} \nabla_\Phi \left[\mathcal{E}_{Z^i}(\mathcal{R}\Phi)\right]$

Towards a Mathematical Error Analysis [4, 6]

♣ quadratic loss, $n = 1$

- $\mathcal{L}^2 := \mathcal{L}^2(K; \mathbb{P}_X)$ where $\mathbb{P}_X$ denotes image measure of X

Theorem (Bias-Variance-Decomposition)

$$\left\|\mathcal{R}\Phi^{\mathrm{emp}} - \widehat{g}\right\|_{\mathcal{L}^2}^2 = E_{m,\mathcal{P}} + A_{\mathcal{P}}$$

Towards a Mathematical Error Analysis [4, 6]

- quadratic loss, $n = 1$
- $\mathcal{L}^2 := \mathcal{L}^2(K; \mathbb{P}_X)$ where $\mathbb{P}_X$ denotes image measure of X

Theorem (Bias-Variance-Decomposition)

$$\left\|\mathcal{R}\Phi^{\text{emp}} - \widehat{g}\right\|_{\mathcal{L}^2}^2 = E_{m,\mathcal{P}} + A_{\mathcal{P}}$$

with

- approximation error (bias)

$$A_{\mathcal{P}} = \left\|\mathcal{R}\Phi^{\text{best}} - \widehat{g}\right\|_{\mathcal{L}^2}^2 = \min_{\Phi \in \mathcal{P}} \left\|\mathcal{R}\Phi - \widehat{g}\right\|_{\mathcal{L}^2}^2$$

Towards a Mathematical Error Analysis [4, 6]

♣ quadratic loss, $n = 1$

- $\mathcal{L}^2 := \mathcal{L}^2(K; \mathbb{P}_X)$ where $\mathbb{P}_X$ denotes image measure of X

Theorem (Bias-Variance-Decomposition)

$$\left\| \mathcal{R}\Phi^{\text{emp}} - \widehat{g} \right\|_{\mathcal{L}^2}^2 = E_{m,\mathcal{P}} + A_{\mathcal{P}}$$

with

- approximation error (bias)

$$A_{\mathcal{P}} = \left\| \mathcal{R}\Phi^{\text{best}} - \widehat{g} \right\|_{\mathcal{L}^2}^2 = \min_{\Phi \in \mathcal{P}} \left\| \mathcal{R}\Phi - \widehat{g} \right\|_{\mathcal{L}^2}^2$$

- estimation error (variance)

$$E_{m,\mathcal{P}} = \mathbb{E}\left[\mathcal{E}_Z\left(\mathcal{R}\Phi^{\text{emp}} \right) \right] - \mathbb{E}\left[\mathcal{E}_Z\left(\mathcal{R}\Phi^{\text{best}} \right) \right]$$

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} = \mathbb{E}\left[\mathcal{E}_Z\left(\mathcal{R}\Phi^{\mathsf{emp}}\right)\right] - \mathbb{E}\left[\mathcal{E}_Z\left(\mathcal{R}\Phi^{\mathsf{best}}\right)\right]$$

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \leq \mathbb{E}\left[\mathcal{E}_Z\left(\mathcal{R}\Phi^{\mathrm{emp}}\right)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{emp}})$$

$$+\frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{best}}) - \mathbb{E}\left[\mathcal{E}_Z\left(\mathcal{R}\Phi^{\mathrm{best}}\right)\right]$$

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \le \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\mathrm{emp}}\big)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{emp}}) + \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{best}}) - \mathbb{E}\big[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\mathrm{best}}\big)\big]$$

- goal: bound $\sup_{g\in\mathcal{R}(\mathcal{P})} \mathbb{E}\left[\mathcal{E}_Z(g)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(g)$

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \le \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\mathrm{emp}}\big)\right] - \frac{1}{m}\sum_{i=1}^{m} \mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{emp}})$$
$$+ \frac{1}{m}\sum_{i=1}^{m} \mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{best}}) - \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\mathrm{best}}\big)\right]$$

- goal: bound $\sup_{g\in\mathcal{R}(\mathcal{P})} \mathbb{E}\left[\mathcal{E}_Z(g)\right] - \frac{1}{m}\sum_{i=1}^{m} \mathcal{E}_{Z^i}(g)$
- $\mathbb{E}\left[\mathcal{E}_{Z^i}(g)\right] = \mathbb{E}\left[\mathcal{E}_Z(g)\right]$

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \leq \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\text{emp}}\big)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\text{emp}})$$
$$+ \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\text{best}}) - \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\text{best}}\big)\right]$$

- goal: bound $\sup_{g\in\mathcal{R}(\mathcal{P})}\mathbb{E}\left[\mathcal{E}_Z(g)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(g)$
- $\mathbb{E}\left[\mathcal{E}_{Z^i}(g)\right] = \mathbb{E}\left[\mathcal{E}_Z(g)\right]$
- reduction to finite case $\Rightarrow$ complexity measure of $\mathcal{R}(\mathcal{P})$

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \leq \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\text{emp}}\big)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\text{emp}})$$
$$+ \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\text{best}}) - \mathbb{E}\big[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\text{best}}\big)\big]$$

- goal: bound $\sup_{g\in\mathcal{R}(\mathcal{P})}\mathbb{E}\left[\mathcal{E}_Z(g)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(g)$
- $\mathbb{E}\left[\mathcal{E}_{Z^i}(g)\right] = \mathbb{E}\left[\mathcal{E}_Z(g)\right]$
- reduction to finite case $\Rightarrow$ complexity measure of $\mathcal{R}(\mathcal{P})$
- regularity of $\mathcal{E}_{Z^i}(g)$ $\Rightarrow$ Concentration inequality

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \le \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\mathrm{emp}}\big)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{emp}}) + \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\mathrm{best}}) - \mathbb{E}\big[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\mathrm{best}}\big)\big]$$

- goal: bound $\sup_{g\in\mathcal{R}(\mathcal{P})} \mathbb{E}\left[\mathcal{E}_Z(g)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(g)$
- $\mathbb{E}\left[\mathcal{E}_{Z^i}(g)\right] = \mathbb{E}\left[\mathcal{E}_Z(g)\right]$
- reduction to finite case $\Rightarrow$ covering number of $\mathcal{R}(\mathcal{P})$
- regularity of $\mathcal{E}_{Z^i}(g) \Rightarrow$ Concentration inequality

Generalization Result [1, 4, 6]

$$E_{m,\mathcal{P}} \leq \mathbb{E}\left[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\text{emp}}\big)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\text{emp}})$$
$$+\frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(\mathcal{R}\Phi^{\text{best}}) - \mathbb{E}\big[\mathcal{E}_Z\big(\mathcal{R}\Phi^{\text{best}}\big)\big]$$

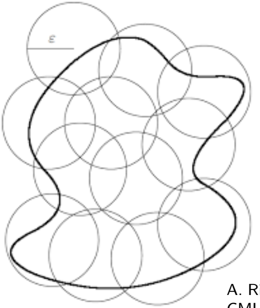
- goal: bound $\sup_{g\in\mathcal{R}(\mathcal{P})}\mathbb{E}\left[\mathcal{E}_Z(g)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathcal{E}_{Z^i}(g)$
- $\mathbb{E}\left[\mathcal{E}_{Z^i}(g)\right] = \mathbb{E}\left[\mathcal{E}_Z(g)\right]$
- reduction to finite case $\Rightarrow$ covering number of $\mathcal{R}(\mathcal{P})$
- boundedness of $\mathcal{E}_{Z^i}(g)$ $\Rightarrow$ Hoeffdings inequality

Assumption (uniformly bounded realization functions)

Replace $\mathcal{R}$ by clipped realization map $\bar{\mathcal{R}}$ given by

$$\bar{\mathcal{R}}\Phi := \big(\min\{|\cdot|, D\}\,\text{sgn}(\cdot)\big)\circ\mathcal{R}\Phi$$

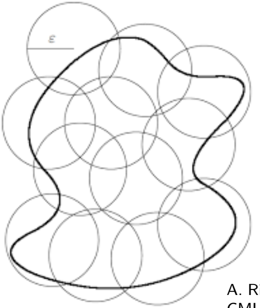
Generalization Result [1, 4, 6]



A. Rinaldo. Lecture Notes. CMU, 2016.

Figure: $\operatorname{cov}\left(\bar{\mathcal{R}}(\mathcal{P}), \varepsilon\right)$ denotes the minimal number of balls of radius ε covering $\bar{\mathcal{R}}(\mathcal{P})$.

Generalization Result [1, 4, 6]



A. Rinaldo. Lecture Notes. CMU, 2016.

Figure: $\mathrm{cov}\left(\bar{\mathcal{R}}(\mathcal{P}), \varepsilon\right)$ denotes the minimal number of balls of radius ε covering $\bar{\mathcal{R}}(\mathcal{P})$.

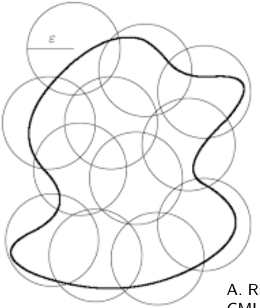
Theorem (Haussler '92, Vapnik '98, Cucker and Smale '02)

With

$$m \lesssim D^4 \varepsilon^{-2} \ln \left[\delta^{-1} \underbrace{\mathrm{cov}\left(\bar{\mathcal{R}}(\mathcal{P}), \tfrac{\varepsilon}{32D}\right)}_{\text{covering number}}\right]$$

samples it holds that $\mathbb{P}\left[E_{m,\mathcal{P}} \le \varepsilon\right] \ge 1 - \delta$.

Generalization Result [1, 4, 6]



A. Rinaldo. Lecture Notes. CMU, 2016.

Figure: $\operatorname{cov}\left(\bar{\mathcal{R}}(\mathcal{P}), \varepsilon\right)$ denotes the minimal number of balls of radius ε covering $\bar{\mathcal{R}}(\mathcal{P})$.

Lemma

$$\operatorname{cov}\left(\bar{\mathcal{R}}(\mathcal{P}), \varepsilon\right) \leq \operatorname{cov}\left(\mathcal{P}, \tfrac{\varepsilon}{\operatorname{Lip}(\mathcal{R})}\right) \leq \left(\tfrac{4R\operatorname{Lip}(\mathcal{R})}{\varepsilon}\right)^{\dim(\mathcal{P})}$$

Theorem (Haussler '92, Vapnik '98, Cucker and Smale '02)

With

$$m \lesssim D^4 \varepsilon^{-2} \ln \left[\delta^{-1} \underbrace{\operatorname{cov}\left(\bar{\mathcal{R}}(\mathcal{P}), \tfrac{\varepsilon}{32D}\right)}_{\text{covering number}}\right]$$

samples it holds that $\mathbb{P}\left[E_{m,\mathcal{P}} \leq \varepsilon\right] \geq 1 - \delta$.

Deep Learning without Curse [4]

- learning problems for every dimension $d \in \mathbb{N}$

Deep Learning without Curse [4]

- learning problems for every dimension $d \in \mathbb{N}$

♣ $\operatorname{size}(\mathcal{P}) := \max\Big\{ \ln(R), \underbrace{\sum_{\ell=1}^{L} N_\ell N_{\ell-1} + N_\ell}_{\dim(\mathcal{P})} \Big\}$, $\varrho = \mathrm{ReLU}$

Deep Learning without Curse [4]

- learning problems for every dimension $d \in \mathbb{N}$

♣ $\mathrm{size}(\mathcal{P}) := \max\Big\{ \ln(R), \underbrace{\sum_{\ell=1}^{L} N_\ell N_{\ell-1} + N_\ell}_{\mathrm{dim}(\mathcal{P})} \Big\}$, $\varrho = \mathrm{ReLU}$

Assumption (Approximation without curse)

Assume there are $\mathcal{P}$ with $\mathrm{size}(\mathcal{P}) \lesssim poly(d, \varepsilon^{-1})$ and $A_{\mathcal{P}} \leq \varepsilon$.

Theorem (Deep Learning without curse - B., Grohs, Jentzen '18)

Then with $m \lesssim poly(d, \varepsilon^{-1}) \ln(\delta^{-1})$ samples it holds that

$$\mathbb{P}\left[\left\| \bar{\mathcal{R}} \Phi^{\mathrm{emp}} - \widehat{g} \right\|_{\mathcal{L}^2}^2 \leq \varepsilon \right] \geq 1 - \delta.$$

Partial Summary

Assume

- underlying data $(X, Y)\colon \Omega \to K \times [-D, D]$
- i.i.d. training data $(X^i, Y^i) \sim (X, Y)$, $i = 1, \dots, m$
- $\widehat{g}$ can be approximated by ReLU networks in $\mathcal{L}^2$ without curse

Partial Summary

Assume

- underlying data $(X, Y)\colon \Omega \to K \times [-D, D]$
- i.i.d. training data $(X^i, Y^i) \sim (X, Y)$, $i = 1, \ldots, m$
- $\widehat{g}$ can be approximated by ReLU networks in $\mathcal{L}^2$ without curse

Then $\bar{\mathcal{R}}(\Phi^{\mathrm{emp}})$

- approximates $\widehat{g}$ in $\mathcal{L}^2$ within accuracy ε with high probability
- with size$(\mathcal{P})$ and m scaling polynomially in d and ε^{-1}

Partial Summary

Assume

- underlying data $(X, Y)\colon \Omega \to K \times [-D, D]$
- i.i.d. training data $(X^i, Y^i) \sim (X, Y)$, $i = 1, \dots, m$
- $\widehat{g}$ can be approximated by ReLU networks in $\mathcal{L}^2$ without curse

Then $\bar{\mathcal{R}}(\Phi^{\mathrm{emp}})$

- approximates $\widehat{g}$ in $\mathcal{L}^2$ within accuracy ε with high probability
- with $\mathrm{size}(\mathcal{P})$ and m scaling polynomially in d and ε^{-1}

Can the assumptions be satisfied?

Application to Kolmogorov PDEs [2, 4]

- initial condition: $\varphi \in \mathcal{C}(\mathbb{R}^d, [-D, D])$
- coefficient functions: $\sigma \colon \mathbb{R}^d \to R^{d \times d}$, $\mu \colon \mathbb{R}^d \to \mathbb{R}^d$ affine linear

Definition (Kolmogorov equation)

$$\begin{cases} \partial_t u(t,x) = \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\big) + \mu(x) \cdot \nabla_x u(t,x) \\ u(0,x) = \varphi(x) \end{cases}$$

for $t \in [0, T]$, $x \in \mathbb{R}^d$

Application to Kolmogorov PDEs [2, 4]

- initial condition: $\varphi \in \mathcal{C}(\mathbb{R}^d, [-D, D])$
- coefficient functions: $\sigma\colon \mathbb{R}^d \to R^{d\times d}$, $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$ affine linear

Definition (Kolmogorov equation)

$$\begin{cases} \partial_t u(t,x) = \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\big) + \mu(x)\cdot\nabla_x u(t,x) \\ u(0,x) = \varphi(x) \end{cases}$$

for $t \in [0, T]$, $x \in \mathbb{R}^d$

$\Rightarrow$ goal: approximately compute the function (end value)

$$K \ni x \mapsto u(T,x)$$

Learning Problem [2]

- $X \sim \mathcal{U}(K) \Rightarrow \mathbb{P}_X = \frac{1}{|K|}\lambda_K$

Learning Problem [2]

- $X \sim \mathcal{U}(K) \Rightarrow \mathbb{P}_X = \frac{1}{|K|}\lambda_K$
- $Y := \varphi(S_T^X)$ where S^X is the solution processes to the stochastic differential equation (SDE)

$$\begin{cases} dS_t^X = \sigma(S_t^X)dB_t + \mu(S_t^X)dt \\ S_0^X = X \end{cases}$$

$\Rightarrow \; \|Y\|_{\mathcal{L}^\infty} \leq D$

Learning Problem [2]

- $X \sim \mathcal{U}(K) \Rightarrow \mathbb{P}_X = \frac{1}{|K|}\lambda_K$
- $Y := \varphi(S_T^X)$ where S^X is the solution processes to the stochastic differential equation (SDE)

$$\begin{cases} dS_t^X = \sigma(S_t^X)dB_t + \mu(S_t^X)dt \\ S_0^X = X \end{cases}$$

$\Rightarrow \ \|Y\|_{\mathcal{L}^\infty} \leq D$

Theorem (learning problem - Beck, Becker, Grohs, Jaafari, Jentzen '18)

For a.e. $x \in K$ it holds that

$$u(T, x) = \widehat{g}(x).$$

Learning Problem [2]

- $X \sim \mathcal{U}(K) \Rightarrow \mathbb{P}_X = \frac{1}{|K|}\lambda_K$
- $Y := \varphi(S_T^X)$ where S^X is the solution processes to the stochastic differential equation (SDE)

$$\begin{cases} dS_t^X = \sigma(S_t^X)dB_t + \mu(S_t^X)dt \\ S_0^X = X \end{cases}$$

$\Rightarrow$ $\|Y\|_{\mathcal{L}^\infty} \leq D$

Theorem (learning problem - Beck, Becker, Grohs, Jaafari, Jentzen '18)

For a.e. $x \in K$ it holds that

$$u(T,x) = \widehat{g}(x).$$

Proof: Feynman-Kac formula $u(T,x) = \mathbb{E}[\varphi(S_T^x)]$ and representation of regression function $\widehat{g}(x) = \mathbb{E}[Y|X = x]$

Approximation without Curse [7]

- ♣ assume φ can be approximated by ReLU networks without curse of dimensionality

$\Rightarrow$ satisfied for applications in financial engineering

Approximation without Curse [7]

- ♣ assume φ can be approximated by ReLU networks without curse of dimensionality
- $\Rightarrow$ satisfied for applications in financial engineering

Theorem (approximation without curse - Grohs et al. '18)

Then there are $\mathcal{P}$ with size$(\mathcal{P}) \lesssim poly(d, \varepsilon^{-1})$ and

$$A_{\mathcal{P}} = \min_{\Phi \in \mathcal{P}} \frac{1}{|K|} \left\| \bar{\mathcal{R}}\Phi - u(T, \cdot) \right\|^2_{\mathcal{L}^2(K)} \leq \varepsilon.$$

Approximation without Curse [7]

- ♣ assume φ can be approximated by ReLU networks without curse of dimensionality
- $\Rightarrow$ satisfied for applications in financial engineering

Theorem (approximation without curse - Grohs et al. '18)

Then there are $\mathcal{P}$ with $\text{size}(\mathcal{P}) \lesssim poly(d, \varepsilon^{-1})$ and

$$A_{\mathcal{P}} = \min_{\Phi \in \mathcal{P}} \frac{1}{|K|} \left\| \bar{\mathcal{R}}\Phi - u(T, \cdot) \right\|^2_{\mathcal{L}^2(K)} \leq \varepsilon.$$

Proof: representation of SDE solution and simulation of Monte-Carlo sampling by neural networks

Solving the Kolmogorov PDE without Curse [4]

Our assumptions are satisfied!

Solving the Kolmogorov PDE without Curse [4]

Our assumptions are satisfied!

Corollary (ERM solves the Kolmogorov PDE without curse)

There exists $\mathcal{P}$ and m with

- $\text{size}(\mathcal{P}) \lesssim poly(d, \varepsilon^{-1})$
- $m \lesssim poly(d, \varepsilon^{-1}) \ln(\delta^{-1})$
- $\mathbb{P}\left[\frac{1}{|K|} \left\|\bar{\mathcal{R}}\Phi^{\text{emp}} - u(T, \cdot)\right\|^2_{\mathcal{L}^2(K)} \leq \varepsilon\right] \geq 1 - \delta.$

Pricing of European Options without Curse [4]

- capped European put option:
  $\varphi(x) = \min\left\{\max\left\{D - \sum_{i=1}^{d} c_i x_i, 0\right\}, D\right\}$

$\Rightarrow$ exactly representable by a ReLU network with size scaling linearly in d

Pricing of European Options without Curse [4]

- capped European put option:
  $\varphi(x) = \min\left\{\max\left\{D - \sum_{i=1}^{d} c_i x_i, 0\right\}, D\right\}$

$\Rightarrow$ exactly representable by a ReLU network with size scaling linearly in d

$\Rightarrow$ quantitative version: there exist $\mathcal{P}$ and m with

- $\text{size}(\mathcal{P}) \lesssim d^2\varepsilon^{-2}$
- $m \lesssim d^2\varepsilon^{-4}\ln(d\varepsilon^{-1}\varrho^{-1})$
- $\mathbb{P}\left[\frac{1}{|K|}\left\|\bar{\mathcal{R}}\Phi^{\text{emp}} - u(T,\cdot)\right\|^2_{\mathcal{L}^2(K)} \leq \varepsilon\right] \geq 1-\varrho.$

Numerical Experiments (Beck et al. '18) [2]

- Black-Scholes equation from financial engineering (option pricing)
- $N = (100, 200, 200, 1)$

Number of descent steps n	Relative $\mathcal{L}^1$ error	Relative $\mathcal{L}^\infty$ error	Runtime in seconds
0	1.004285	1.009524	1
100000	0.371515	0.387978	437.9
250000	0.001220	0.010039	1092.6
500000	0.000949	0.005105	2183.8

Table: Error between $\mathcal{R}_{\mathrm{ReLU}}\Phi_n$ and $u(T,\cdot)$ on $[90,110]^{100}$

Possible Extensions

- learn solution map $(\varphi, \sigma, \mu, t, x) \mapsto u(t, x)$
- combined Dirichlet-Poisson problem

$$\begin{cases} \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(x)\big) + \nabla_x u(x) \cdot \mu(x) = \vartheta(x), & x \in D \\ u(x) = \varphi(x), & x \in \partial D \end{cases}$$

- high dimensional functions that admit a probabilistic representation and that can be approximated by an iterative scheme

Towards an Analysis of the Optimization Error [3]

Theorem (inverse stability on a subset - B., Elbrächter, Grohs)

There exists $\Omega \subseteq \mathcal{P}_{(d,N_1,1)}$ such that for every $\Phi \in \Omega$ and $g \in \mathcal{R}(\Omega)$ there exists a parametrization $\Psi \in \Omega$ with

$$\mathcal{R}\Psi = g \quad \text{and} \quad \|\Psi - \Phi\|_\infty \leq 4|g - \mathcal{R}\Phi|_{\mathcal{W}^{1,\infty}}^{\frac{1}{2}}.$$

Corollary (parameter minimum ⇒ realization minimum)

Let $\Phi_* \in \Omega$ be a local minimum of

$$\min_{\Phi \in \Omega} \frac{1}{m} \sum_{i=1}^{m} \mathcal{E}_{z^i}(\mathcal{R}\Phi).$$

Then $\mathcal{R}\Phi_*$ is a local minimum (w.r.t. $|\cdot|_{W^{1,\infty}}$) of

$$\min_{g \in \mathcal{R}(\Omega)} \frac{1}{m} \sum_{i=1}^{m} \mathcal{E}_{z^i}(g)$$

Thank you for your Attention!

[1] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

[2] C. Beck, S. Becker, P. Grohs, N. Jaafari, and A. Jentzen. "Solving stochastic differential equations and Kolmogorov equations by means of deep learning". In: *arXiv:1806.00421* (2018).

[3] J. Berner, D. Elbrächter, and P. Grohs. "How degenerate is the parametrization of neural networks with the ReLU activation function?". In: *arXiv:1905.09803* (2019). Accepted at NeurIPS 2019.

[4] J. Berner, P. Grohs, and A. Jentzen. "Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations". In: *arXiv:1809.03062* (2018).

[5] J. Berner, D. Elbrächter, P. Grohs, and A. Jentzen. "Towards a regularity theory for ReLU networks–chain rule and global error estimates". In: *arXiv:1905.04992* (2019). Accepted for presentation at SampTA 2019.

[6] F. Cucker and S. Smale. "On the mathematical foundations of learning". In: *Bulletin of the American mathematical society* 39.1 (2002), pp. 1–49.

[7] P. Grohs, F. Hornung, A. Jentzen, and P. von Wurstemberger. "A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations". In: *arXiv:1809.02362* (2018).

[8] V. Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), p. 529.

[9] P. Petersen, M. Raslan, and F. Voigtlaender. "Topological properties of the set of functions generated by neural networks of fixed size". In: *arXiv:1806.08459* (2018).

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.