Deep Neural Networks for PDEs

Philipp Grohs

universität wien
Faculty of Mathematics

Bedlewo, Nov 2019

Short Reading List

1. Weinan E, Bing Yu: The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems; Communications in Mathematics and Statistics, 2018
2. Christian Beck, Sebastian Becker, Philipp Grohs, Nor Jaafari, Arnulf Jentzen: Solving stochastic differential equations and Kolmogorov equations by means of deep learning; arXiv:1806.00421
3. Philipp Grohs, Fabian Hornung, Arnulf Jentzen and Philippe Von Wurstemberger: A proof that ANNs overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations; Memoirs of the AMS, 2020
4. Julius Berner, Philipp Grohs, Arnulf Jentzen: Analysis of the generalization error: ERM over deep ANNs overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations; SIAM Journal on Mathematics of Data Science; 2020
5. Dennis Elbrächter, Philipp Grohs, Arnulf Jentzen, Christoph Schwab: DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing; arXiv:1809.07669

Syllabus

PDEs and the Curse of Dimensionality

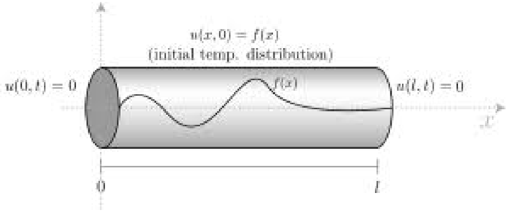
PDEs

A PDE for the function $u(x_1, \ldots, x_d)$ is an equation of the form

$$\mathcal{F}\left(x_1, \ldots, x_d, u, \frac{\partial u}{\partial x_1}, \ldots \frac{\partial u}{\partial x_d}, \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_d}, \ldots\right) = 0.$$

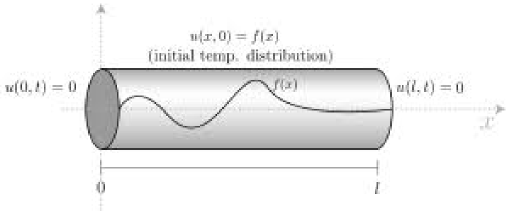
together with suitable boundary conditions.

Heat Equation in 1D



$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \quad u(x,0) = f(x)$$

Heat Equation in 1D

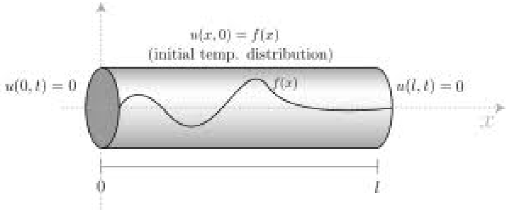


$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \quad u(x,0) = f(x)$$

or informally:

$$u(x, t+\delta t) \approx u(x,t) + \frac{\delta t}{2(\delta x)^2}\left(\frac{u(x+\delta x, t) + u(x-\delta x, t)}{2} - u(x,t)\right)$$

Heat Equation in 1D

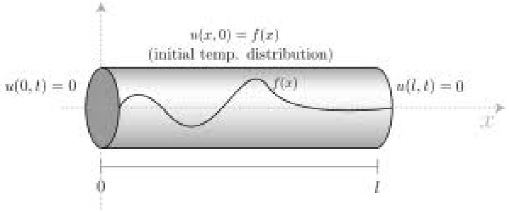


$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \quad u(x,0) = f(x)$$

or informally:

$$\begin{aligned} u(x, t+\delta t) &\approx u(x,t) + \frac{\delta t}{2(\delta x)^2}\left(\frac{u(x+\delta x,t)+u(x-\delta x,t)}{2} - u(x,t)\right) \\ &= \left(1 - \frac{\delta t}{2(\delta x)^2}\right)u(x,t) + \frac{\delta t}{2(\delta x)^2}\,\frac{u(x+\delta x,t)+u(x-\delta x,t)}{2} \end{aligned}$$

Heat Equation in 1D



$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \quad u(x,0) = f(x)$$

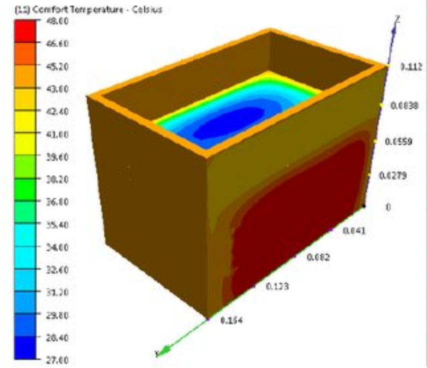
or informally:

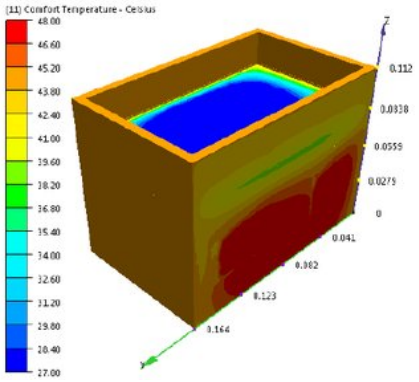
$u(x, t+\delta t)$

The temperature at x is pushed to the average temperature around x $\rightsquigarrow$ diffusion

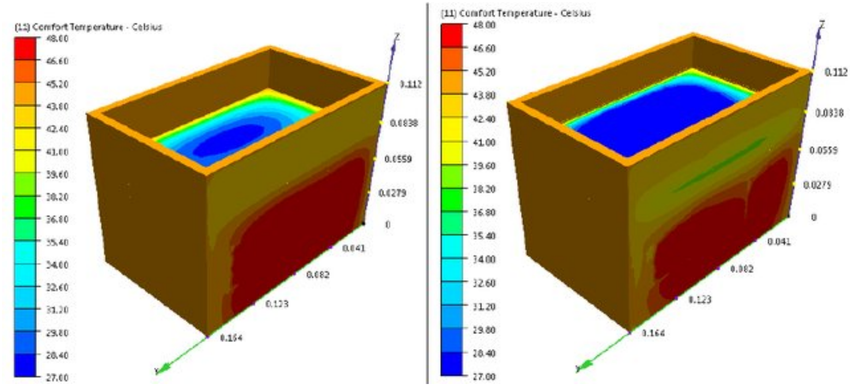
$$= \left(1 - \frac{\delta t}{2(\delta x)^2}\right) u(x,t) + \frac{\delta t}{2(\delta x)^2} \frac{u(x+\delta x, t) + u(x-\delta x, t)}{2}$$

Heat Equation





Heat Equation



$$\frac{\partial u}{\partial t}(t,x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3} + g(t,x), \quad u(0,x) = \varphi(x)$$

$t \in (0,\infty), x \in \mathbb{R}^3$; $d = 4$.

Explicit Solution of Heat Equation if $g = 0$

Let $u(t, x)$ satisfy

$$\frac{\partial u}{\partial t}(t, x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3}, \quad u(0, x) = \varphi(x)$$

$t \in (0, \infty), x \in \mathbb{R}^3$; $d = 4$.

Explicit Solution of Heat Equation if $g = 0$

Let $u(t, x)$ satisfy

$$\frac{\partial u}{\partial t}(t,x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3}, \quad u(0,x) = \varphi(x)$$

$t \in (0, \infty), x \in \mathbb{R}^3$; $d = 4$.
Then

$$u(t,x) = \frac{1}{(4\pi t)^{3/2}} \int_{\mathbb{R}^3} \varphi(y) \exp(-|x-y|^2/4t) dy.$$

Fluid Dynamics

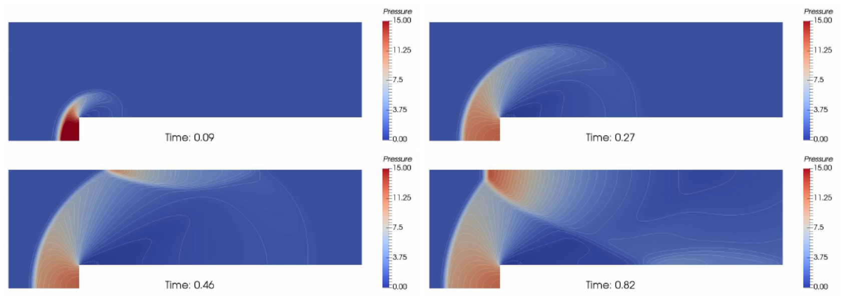


Figure 8: Mach 3 wind tunnel: Polynomial degree $K = 40$, $35k$ Vertices, Maxwellian molecules, $28.9M$ total DoFs. Coloring: pressure, contour lines: density. Computations were carried out on the Euler cluster of ETH Zurich (Xeon E5-2697 v2) with 360 cores.

$$\frac{\partial u}{\partial t}(t,x,v) + v \cdot \nabla u(t,x,v) = Qu(t,x,v)$$

$t \in (0,\infty), x, v \in \mathbb{R}^3$; $d = 7$.

(Multi Electron) Schrödinger Equation

Wave function of non-relativistic quantum mechanical system of N electons in a field of K nuclei of charge Z_ν and fixed position $R_\mu \in \mathbb{R}^3$

$$
\begin{aligned}
i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N;t) = -\frac{1}{2}\sum_{\xi=1}^{N}\Delta_i\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N;t) - \\
\sum_{\xi=1}^{N}\sum_{\nu=1}^{K}\frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N;t) + \\
\frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N}\frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N;t),
\end{aligned}
$$

$t \in (0,\infty)$, $\mathbf{r}_1,\ldots,\mathbf{r}_N \in \mathbb{R}^3$; $d = 3N+1$.

Black-Scholes Equation

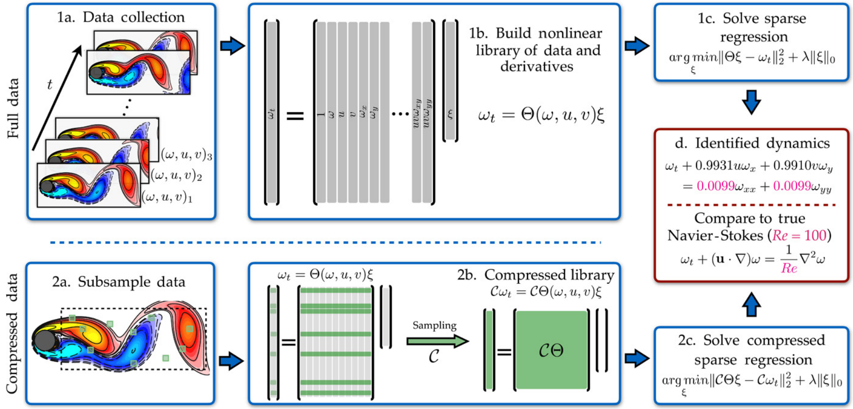
Pricing a portfolio of N financial derivatives

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^N} \left(\frac{\partial^2 u}{\partial x_i \partial x_j}\right)(t,x) + \sum_{i=1}^{N} \mu_i x_i \left(\frac{\partial u}{\partial x_i}\right)(t,x)$$

$$u(0,x) = \max\{K - \sum_{i=1}^{N} c_i x_i, 0\}$$

$t \in (0,\infty)$, $x \in \mathbb{R}^N$; $d = N + 1$.

Learning the PDE [Rudy et.al. (2017)]


Full data
1a. Data collection
t
$(\omega, u, v)_3$
$(\omega, u, v)_2$
$(\omega, u, v)_1$
ω_t
$=$
1 ω u v ω_x ω_y $\cdots$ $uv\omega_{xy}$ $uv\omega_{yy}$
ξ
1b. Build nonlinear library of data and derivatives
$\omega_t = \Theta(\omega, u, v)\xi$
1c. Solve sparse regression
$\arg\min_{\xi} \|\Theta\xi - \omega_t\|_2^2 + \lambda\|\xi\|_0$
d. Identified dynamics
$\omega_t + 0.9931u\omega_x + 0.9910v\omega_y = 0.0099\omega_{xx} + 0.0099\omega_{yy}$
Compare to true Navier-Stokes ($Re = 100$)
$\omega_t + (\mathbf{u}\cdot\nabla)\omega = \frac{1}{Re}\nabla^2\omega$
Compressed data
2a. Subsample data
$\omega_t = \Theta(\omega, u, v)\xi$
2b. Compressed library
$\mathcal{C}\omega_t = \mathcal{C}\Theta(\omega, u, v)\xi$
Sampling
$\mathcal{C}$
$\mathcal{C}\Theta$
2c. Solve compressed sparse regression
$\arg\min_{\xi} \|\mathcal{C}\Theta\xi - \mathcal{C}\omega_t\|_2^2 + \lambda\|\xi\|_0$

Finite Difference Approach

Want to approximate $u(x)$ for $x \in [0,1]^d$.

Finite Difference Approach

Want to approximate $u(x)$ for $x \in [0,1]^d$.

- Let

$$u_{i_1,\ldots,i_d} \sim u(i_1\epsilon, \ldots, i_d\epsilon), \quad (i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d,$$

Finite Difference Approach

Want to approximate $u(x)$ for $x \in [0,1]^d$.

- Let

$$u_{i_1,\dots,i_d} \sim u(i_1\epsilon,\dots,i_d\epsilon), \quad (i_1,\dots,i_d) \in \{0,\dots,\lfloor \epsilon^{-1} \rfloor\}^d,$$

-

$$\frac{u_{i_1,\dots,i_l+1,\dots,i_d} - u_{i_1,\dots,i_l,\dots,i_d}}{\epsilon} \sim \frac{\partial}{\partial x_l} u(i_1\epsilon,\dots,i_d\epsilon),$$
$$(i_1,\dots,i_d) \in \{0,\dots,\lfloor \epsilon^{-1} \rfloor\}^d,$$

  and so on,

Finite Difference Approach

Want to approximate $u(x)$ for $x \in [0,1]^d$.

- Let
$$u_{i_1,\ldots,i_d} \sim u(i_1\epsilon,\ldots,i_d\epsilon), \quad (i_1,\ldots,i_d) \in \{0,\ldots,\lfloor \epsilon^{-1} \rfloor\}^d,$$

- $$\frac{u_{i_1,\ldots,i_l+1,\ldots,i_d} - u_{i_1,\ldots,i_l,\ldots,i_d}}{\epsilon} \sim \frac{\partial}{\partial x_l} u(i_1\epsilon,\ldots,i_d\epsilon), \qquad (i_1,\ldots,i_d) \in \{0,\ldots,\lfloor \epsilon^{-1} \rfloor\}^d,$$

  and so on,
- and solve the discrete system
$$\mathcal{F}\left(i_1\epsilon,\ldots,i_d\epsilon, u_{i_1,\ldots,i_d}, \frac{u_{i_1+1,\ldots,i_d} - u_{i_1,\ldots,i_d}}{\epsilon},\ldots\right) = 0 \qquad (i_1,\ldots,i_d) \in \{0,\ldots,\lfloor \epsilon^{-1} \rfloor\}^d.$$

Curse of Dimensionality

The system

$$\mathcal{F}\left(i_1\epsilon, \ldots, i_d\epsilon, u_{i_1,\ldots,i_d}, \frac{u_{i_1+1,\ldots,i_d} - u_{i_1,\ldots,i_d}}{\epsilon}, \ldots\right) = 0$$
$$(i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d.$$

requires us to solve an equation in $u_{i_1,\ldots,i_d}$ for $(i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d$.

Curse of Dimensionality

The system

$$\mathcal{F}\left(i_1\epsilon, \ldots, i_d\epsilon, u_{i_1,\ldots,i_d}, \frac{u_{i_1+1,\ldots,i_d} - u_{i_1,\ldots,i_d}}{\epsilon}, \ldots\right) = 0$$

$$(i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d.$$

requires us to solve an equation in $u_{i_1,\ldots,i_d}$ for $(i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d$.

Exponential Dependence on the Dimension

Let $\epsilon = \frac{1}{2}$ (take two samples in each coordinate). Then these are 2^d unknowns.

Curse of Dimensionality

The system

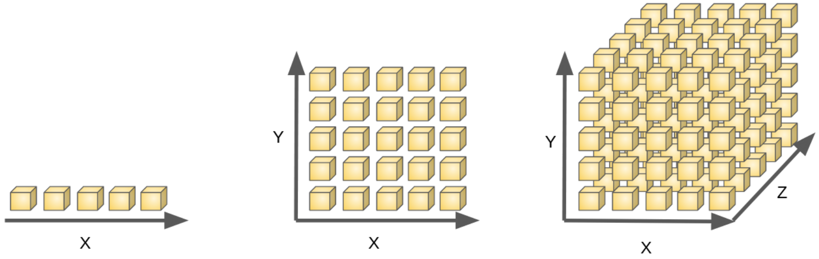
$$\mathcal{F}\left(i_1\epsilon, \ldots, i_d\epsilon, u_{i_1,\ldots,i_d}, \frac{u_{i_1+1,\ldots,i_d} - u_{i_1,\ldots,i_d}}{\epsilon}, \ldots\right) = 0$$
$$(i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d.$$

requires us to solve an equation in $u_{i_1,\ldots,i_d}$ for $(i_1, \ldots, i_d) \in \{0, \ldots, \lfloor \epsilon^{-1} \rfloor\}^d$.

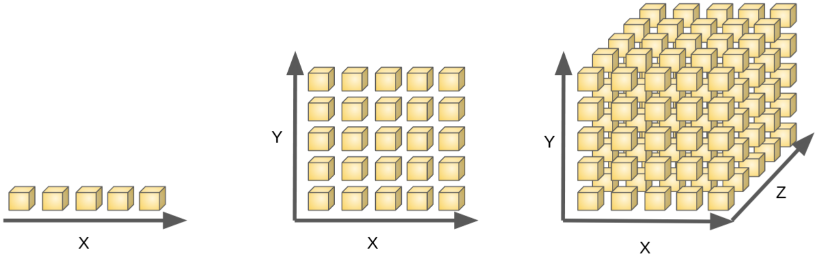
Exponential Dependence on the Dimension

Let $\epsilon = \frac{1}{2}$ (take two samples in each coordinate). Then these are 2^d unknowns. $\rightsquigarrow$ intractable for high-dimensional problems!

Curse of Dimensionality

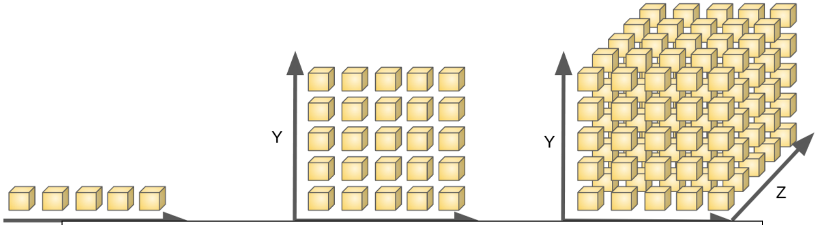


Curse of Dimensionality



The complexity of approximating a general d-dimensional function scales exponentially in d.

Curse of Dimensionality



The ction

Suppose we have a problem where we aim to approximate a d-dimensional function. An algorithm to solve the problem suffers from the curse of dimensionality if its computational complexity depends exponentially on the dimension d.

Black-Scholes Equation

- Pricing a portfolio of N financial derivatives

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^N} \left(\frac{\partial^2 u}{\partial x_i \partial x_j}\right)(t,x) + \sum_{i=1}^{N} \mu_i x_i \left(\frac{\partial u}{\partial x_i}\right)(t,x)$$

$$u(0,x) = \max\{K - \sum_{i=1}^{N} c_i x_i, 0\}$$

$t \in (0,\infty)$, $x \in \mathbb{R}^N$; $d = N+1$.

Black-Scholes Equation

- Pricing a portfolio of N financial derivatives

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^N} \left(\frac{\partial^2 u}{\partial x_i \partial x_j}\right)(t,x) + \sum_{i=1}^{N} \mu_i x_i \left(\frac{\partial u}{\partial x_i}\right)(t,x)$$

$$u(0,x) = \max\{K - \sum_{i=1}^{N} c_i x_i, 0\}$$

  $t \in (0,\infty)$, $x \in \mathbb{R}^N$; $d = N+1$.

- Realistic values: $d = 100 - 1000$.

Black-Scholes Equation

- Pricing a portfolio of N financial derivatives

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^N} (\frac{\partial^2 u}{\partial x_i \partial x_j})(t,x) + \sum_{i=1}^{N} \mu_i x_i (\frac{\partial u}{\partial x_i})(t,x)$$

$$u(0,x) = \max\{K - \sum_{i=1}^{N} c_i x_i, 0\}$$

$t \in (0,\infty)$, $x \in \mathbb{R}^N$; $d = N+1$.

- Realistic values: $d = 100 - 1000$.
- Complexity of finite difference method: $2^{100} - 2^{1000}$.

Black-Scholes Equation

- Pricing a portfolio of N financial derivatives

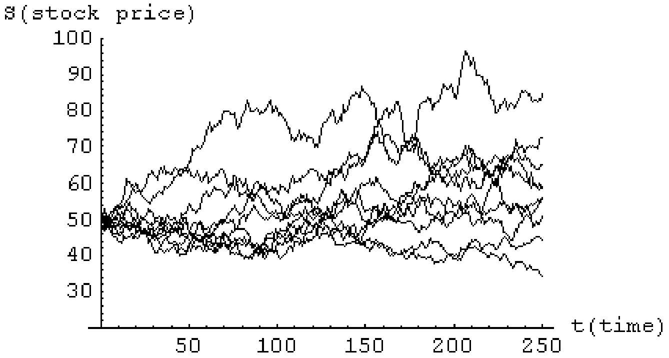
$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^N} \Big(\frac{\partial^2 u}{\partial x_i \partial x_j}\Big)(t,x) + \sum_{i=1}^{N} \mu_i x_i \Big(\frac{\partial u}{\partial x_i}\Big)(t,x)$$

$$u(0,x) = \max\{K - \sum_{i=1}^{N} c_i x_i, 0\}$$

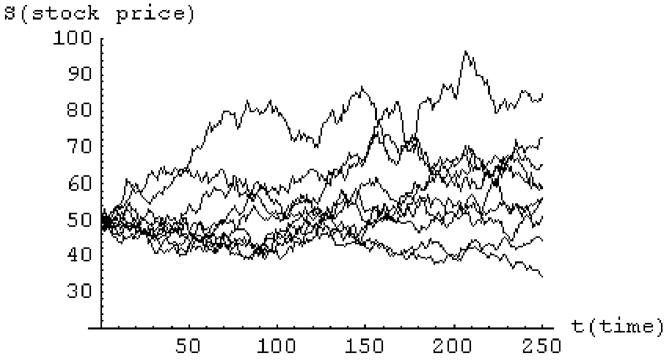
$t \in (0,\infty)$, $x \in \mathbb{R}^N$; $d = N+1$.

- Realistic values: $d = 100 - 1000$.
- Complexity of finite difference method: $2^{100} - 2^{1000}$.
- Number of atoms in the universe: 2^{250}.

Black-Scholes Equation

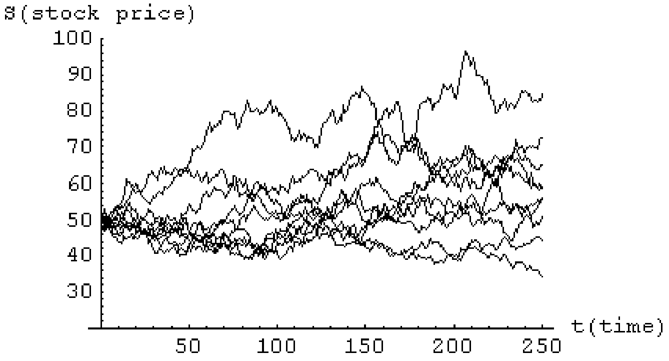


Black-Scholes Equation



- Option pricing is extremely relevant and has to be done every day in the financial industry

Black-Scholes Equation



- Option pricing is extremely relevant and has to be done every day in the financial industry
- All algorithms for the solution of the Black-Scholes equation suffer from the curse of dimensionality!

MNIST

MNIST Database for hand-written digit recognition http://yann.lecun.com/exdb/mnist/

MNIST

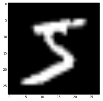
MNIST Database for hand-written digit recognition http://yann.lecun.com/exdb/mnist/

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28 \times 28} \sim \mathbb{R}^{784}$:

MNIST

MNIST Database for handwritten digit recognition http://yann.lecun.com/exdb/mnist/

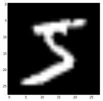
- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:



MNIST

MNIST Database for handwritten digit recognition `http://yann.lecun.com/exdb/mnist/`

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:

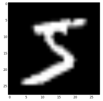


- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit

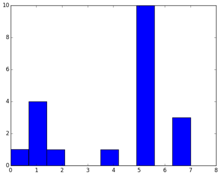
MNIST

MNIST Database for hand-written digit recognition `http://yann.lecun.com/exdb/mnist/`

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:

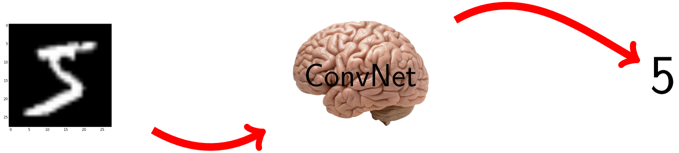


- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit

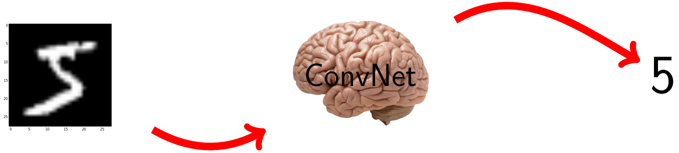


MNIST

MNIST

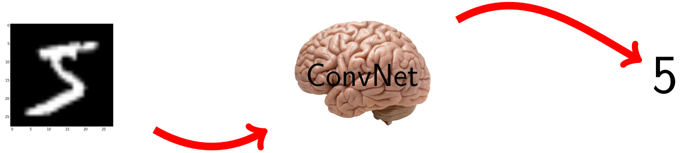


MNIST



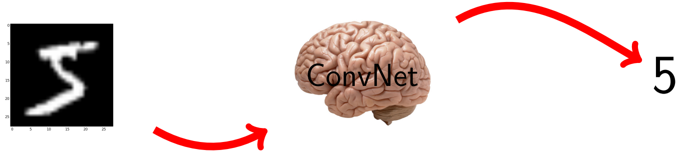
- This is a 784-dimensional function

MNIST



- This is a 784-dimensional function
- Apparently, deep learning does not suffer from the curse of dimensionality for certain classification problems!

MNIST



- This is a 784-dimensional function
- Apparently, deep learning does not suffer from the curse of dimensionality for certain classification problems!

Can this also be used for the solution of PDEs?

The Deep Ritz Method [E - Yu (2018)]

Given an elliptic ($a > 0$) PDE

$$\begin{aligned} \operatorname{div}(a(x)\nabla u(x)) = f(x) \quad & x \in \Omega \\ u(x) = 0 \quad & x \in \partial\Omega. \end{aligned}$$

The Deep Ritz Method [E - Yu (2018)]

Given an elliptic ($a > 0$) PDE

$$\begin{aligned} \operatorname{div}(a(x)\nabla u(x)) = f(x) \quad & x \in \Omega \\ u(x) = 0 \quad & x \in \partial\Omega. \end{aligned}$$

in variational form

$$u = \operatorname*{argmin}_{v \in H_0^1(\Omega)} J(v) \quad \text{where}$$

$$J(v) := \frac{1}{2}\int_\Omega a(x)|\nabla v(x)|^2 dx - \int_\Omega f(x)v(x)dx.$$

The Deep Ritz Method [E - Yu (2018)]

Given an elliptic ($a > 0$) PDE

$$\begin{aligned} \operatorname{div}(a(x)\nabla u(x)) &= f(x) \quad x \in \Omega \\ u(x) &= 0 \quad x \in \partial\Omega. \end{aligned}$$

in variational form

$$u = \operatorname*{argmin}_{v \in H_0^1(\Omega)} J(v) \quad \text{where}$$

$$J(v) := \frac{1}{2}\int_\Omega a(x)|\nabla v(x)|^2 dx - \int_\Omega f(x)v(x)dx.$$

Solve this minimization problem using deep learning!

The Deep Ritz Method [E - Yu (2018)]

To solve

$$u = \operatorname*{argmin}_{v \in H_0^1(\Omega)} J(v)$$

The Deep Ritz Method [E - Yu (2018)]

To solve

$$u = \operatorname*{argmin}_{v \in H_0^1(\Omega)} J(v)$$

we make a NN ansatz $u = R_{ReLU}(\Phi)$ and minimize

$$\hat{\Phi} = \operatorname*{argmin}_{\Phi \in \times_{i=1}^{L} \left(\mathbb{R}^{N_i \times N_{i-1}} \times \mathbb{R}^{N_i}\right)} J(R_{ReLU}(\Phi))$$

The Deep Ritz Method [E - Yu (2018)]

To solve

$$u = \operatorname*{argmin}_{v \in H_0^1(\Omega)} J(v)$$

we make a NN ansatz $u = R_{ReLU}(\Phi)$ and minimize

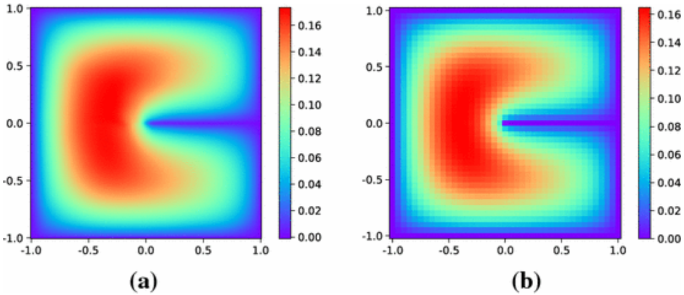
$$\hat{\Phi} = \operatorname*{argmin}_{\Phi \in \times_{i=1}^{L} \left(\mathbb{R}^{N_i \times N_{i-1}} \times \mathbb{R}^{N_i}\right)} J(R_{ReLU}(\Phi))$$

using an (SGD) update

$$\Phi^{k+1} = \Phi^k - \eta \nabla_\Phi a(x^k) |\nabla R_{ReLU}(\Phi^k)(x^k)|^2,$$

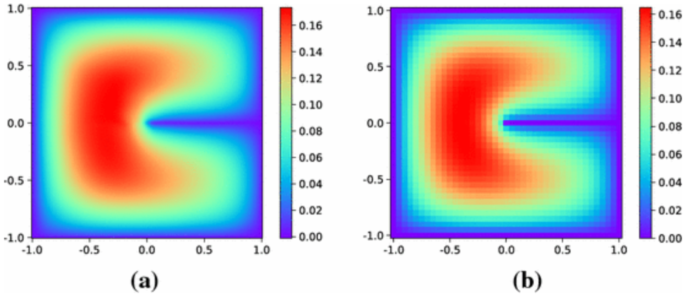
where x^k is chosen uniformly at random in Ω.

The Deep Ritz Method [E - Yu (2018)]



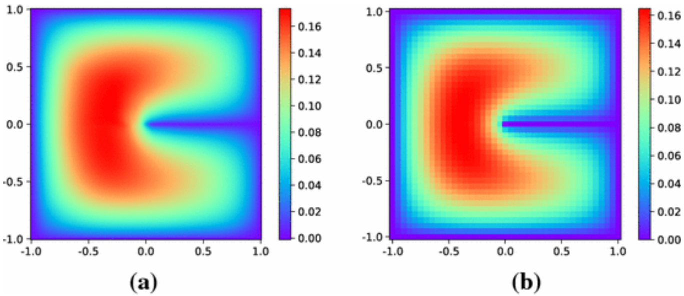
(a) Solution of Deep Ritz Method with 811 parameters, (b) solution of finite difference method with 1681 parameters.

The Deep Ritz Method [E - Yu (2018)]



(a) Solution of Deep Ritz Method with 811 parameters, (b) solution of finite difference method with 1681 parameters. ...but...

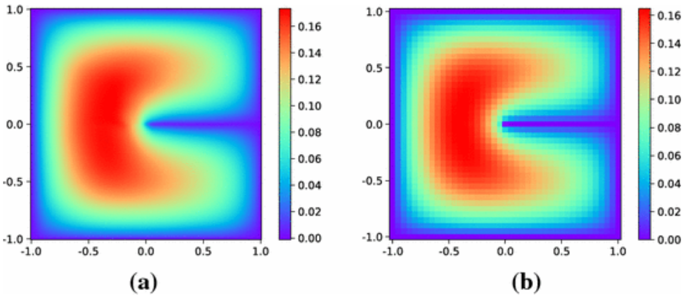
The Deep Ritz Method [E - Yu (2018)]



(a) Solution of Deep Ritz Method with 811 parameters, (b) solution of finite difference method with 1681 parameters. ...but...

- only qualitatively correct solution; no convergence!!
- no good way to enforce boundary conditions

The Deep Ritz Method [E - Yu (2018)]



(a) Solution of Deep Ritz Method with 811 parameters, (b) solution of finite difference method with 1681 parameters. ...but...

- only qualitatively correct solution; no convergence!!
- no good way to enforce boundary conditions
- still a long way to go...

Computing Ground State Energies

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

Computing Ground State Energies

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

where

$$H\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = -\frac{1}{2}\sum_{\xi=1}^{N} \Delta_i \Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) -$$
$$\sum_{\xi=1}^{N}\sum_{\nu=1}^{K} \frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) + \frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N} \frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N).$$

Computing Ground State Energies

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

where

$$H\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) = -\frac{1}{2}\sum_{\xi=1}^{N}\Delta_i\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) - \sum_{\xi=1}^{N}\sum_{\nu=1}^{K}\frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) + \frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N}\frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N).$$

Challenges:

Computing Ground State Energies

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

where

$$H\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) = -\frac{1}{2}\sum_{\xi=1}^{N}\Delta_i\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) - \sum_{\xi=1}^{N}\sum_{\nu=1}^{K}\frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) + \frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N}\frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N).$$

Challenges:

- The wave function Φ needs to satisfy the Pauli exclustion principle (if two electrons have the same spin, the wave function has to be antisymmetric w.r.t. the respective coordinates)

Computing Ground State Energies

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

where

$$H\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) = -\frac{1}{2}\sum_{\xi=1}^{N}\Delta_i\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) -$$

$$\sum_{\xi=1}^{N}\sum_{\nu=1}^{K}\frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) + \frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N}\frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N).$$

Challenges:

- The wave function Φ needs to satisfy the Pauli exclustion principle (if two electrons have the same spin, the wave function has to be antisymmetric w.r.t. the respective coordinates)
- The high dimensionality

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi\rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi\rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Variational MC

At the current iterate Ψ^k (from a usually very simple ansatz) pick (MCMC) sampling points to discretize the inner products and minimize over this Raleigh Ritz quotient.

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Variational MC

At the current iterate Ψ^k (from a usually very simple ansatz) pick (MCMC) sampling points to discretize the inner products and minimize over this Raleigh Ritz quotient. This can of course also be done by using NNs as ansatz! $\rightsquigarrow$ [Han-Zhang-E (2018)]

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Variational MC

At the current iterate Ψ^k (from a usually very simple ansatz) pick (MCMC) sampling points to discretize the inner products and minimize over this Raleigh Ritz quotient. This can of course also be done by using NNs as ansatz! $\rightsquigarrow$ [Han-Zhang-E (2018)]

Challenges:

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Variational MC

At the current iterate Ψ^k (from a usually very simple ansatz) pick (MCMC) sampling points to discretize the inner products and minimize over this Raleigh Ritz quotient. This can of course also be done by using NNs as ansatz! $\rightsquigarrow$ [Han-Zhang-E (2018)]

Challenges:

- Very slow (no) convergence

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Variational MC

At the current iterate Ψ^k (from a usually very simple ansatz) pick (MCMC) sampling points to discretize the inner products and minimize over this Raleigh Ritz quotient. This can of course also be done by using NNs as ansatz! $\rightsquigarrow$ [Han-Zhang-E (2018)]

Challenges:

- Very slow (no) convergence
- Enforcing the Pauli exclusion principle on NNs

Computing Ground State Energies

Variational Formulation

$$\lambda = \min_{\Psi \in H^1} \frac{\langle H\Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}{\langle \Psi, \Psi \rangle_{L^2(\mathbb{R}^{3N})}}.$$

but these quantities cannot even be computed!

Variational MC

At the current iterate Ψ^k (from a usually very simple ansatz) pick (MCMC) sampling points to discretize the inner products and minimize over this Raleigh Ritz quotient. This can of course also be done by using NNs as ansatz! $\rightsquigarrow$ [Han-Zhang-E (2018)]

Challenges:

- Very slow (no) convergence
- Enforcing the Pauli exclusion principle on NNs
- still a long way to go...

Computing Energy Surfaces

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

Computing Energy Surfaces

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

where

$$H\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N) = -\frac{1}{2}\sum_{\xi=1}^{N}\Delta_i\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N) - \sum_{\xi=1}^{N}\sum_{\nu=1}^{K}\frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N) + \frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N}\frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N).$$

Energy surface

The ground state Energy λ as a function of the locations R_ν, $\nu = 1,\dots,K$ is called the *energy surface*.

Computing Energy Surfaces

The ground state energy of a molecule is the minimal Eigenvalue of the Eigenvalue problem

$$H\Psi = \lambda\Psi,$$

where

$$H\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N) = -\frac{1}{2}\sum_{\xi=1}^{N}\Delta_i\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N) - \sum_{\xi=1}^{N}\sum_{\nu=1}^{K}\frac{Z_\nu}{|\mathbf{r}_\xi - R_\nu|}\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N) + \frac{1}{2}\sum_{\xi=1}^{N}\sum_{\eta=1}^{N}\frac{1-\delta_{\xi,\eta}}{|\mathbf{r}_\xi - \mathbf{r}_\eta|}\Psi(\mathbf{r}_1,\dots,\mathbf{r}_N).$$

Energy surface

The ground state Energy λ as a function of the locations R_ν, $\nu = 1,\dots,K$ is called the *energy surface*.

There are very successful applications which compute the energy surface using deep learning regression (and training samples generated by off the shelf numerical solvers) $\rightsquigarrow$ [Schmitz-Godtliebsen-Christiansen (2019)].

Recap of Statistical Learning Theory

Data Generating Distribution

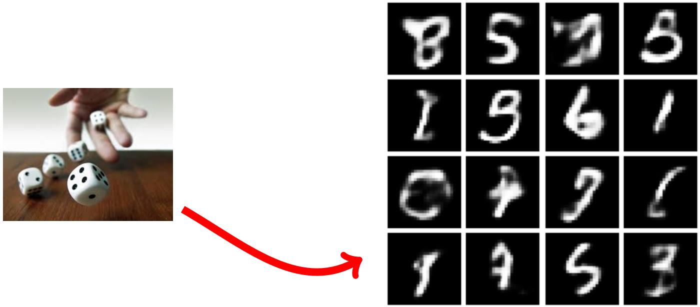
Suppose that there exists a probability distribution on $\mathbb{R}^{784}$ that randomly generates handwritten digits.

Data Generating Distribution

Suppose that there exists a probability distribution on $\mathbb{R}^{784}$ that randomly generates handwritten digits.

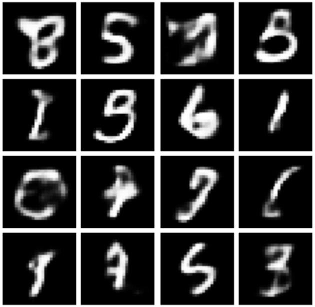
Data Generating Distribution

Suppose that there exists a probability distribution on $\mathbb{R}^{784}$ that randomly generates handwritten digits.



Data Generating Distribution

Suppose that there exists a probability distribution on $\mathbb{R}^{784}$ that randomly generates handwritten digits.



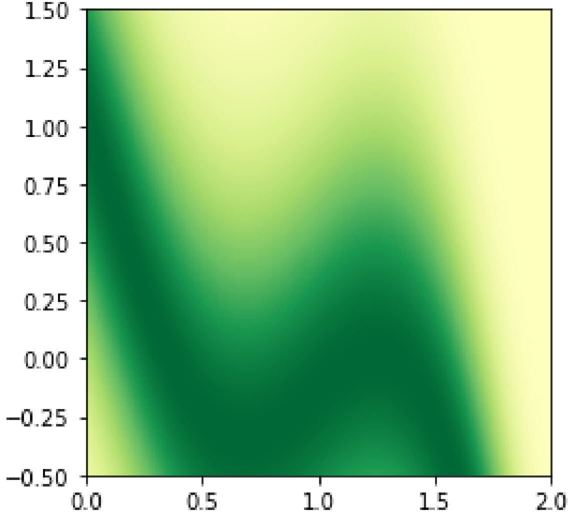
$\boxed{\rightsquigarrow \textbf{Variational Autoencoder Demo}}$

The Statistical Learning Theory View

Suppose that our training data consists of samples according to a given data distribution (X, Y)

The Statistical Learning Theory View

Suppose that our training data consists of samples according to a given data distribution (X, Y)

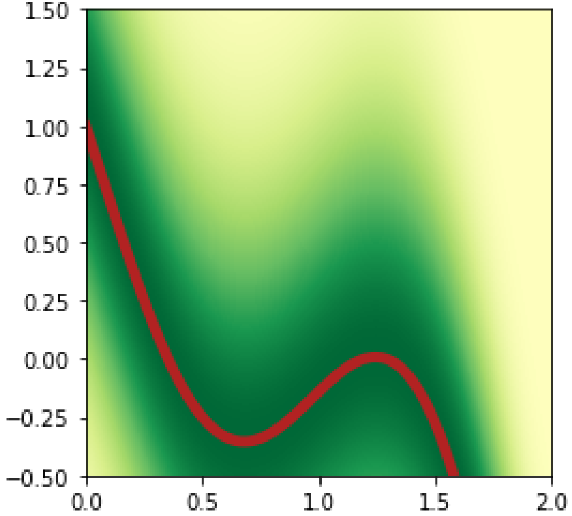


The Statistical Learning Theory View

If we knew the data distribution (X, Y), the best functional relation between X and Y would simply be $\mathbb{E}[Y|X = x]$!

The Statistical Learning Theory View

If we knew the data distribution (X, Y), the best functional relation between X and Y would simply be $\mathbb{E}[Y|X = x]$!

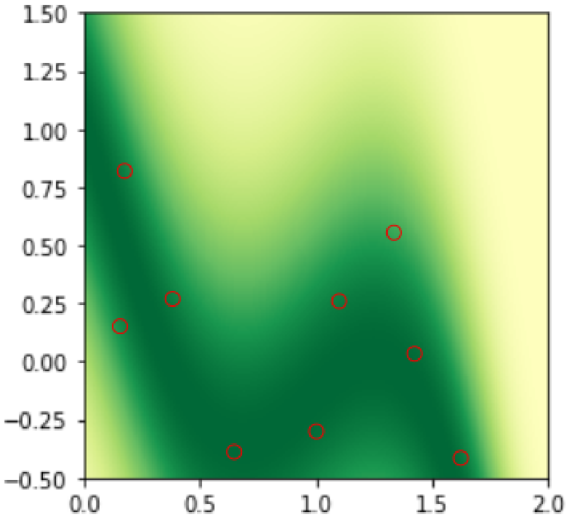


The Statistical Learning Theory View

But we only have samples and do not know the distribution (X, Y)

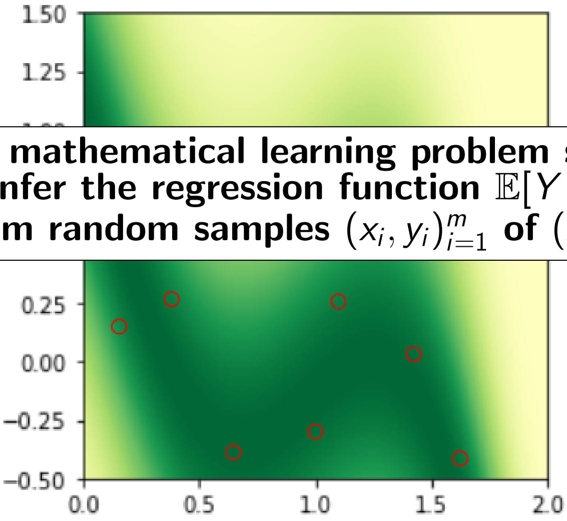
The Statistical Learning Theory View

But we only have samples and do not know the distribution (X, Y)



The Statistical Learning Theory View

But we only have samples and do not know the distribution (X, Y)



A mathematical learning problem seeks to infer the regression function $\mathbb{E}[Y|X = x]$ from random samples $(x_i, y_i)_{i=1}^m$ of (X, Y).

The Mathematical Learning Problem

The Mathematical Learning Problem

Let $(\Sigma, \mathcal{G}, \mathbb{P})$ probability space. Given (Borel measurable) random vectors $X : \Sigma \to \mathbb{R}^d$, $Y : \Sigma \to \mathbb{R}^k$ with $\mathrm{im}(X) \subseteq \Omega$ for $\Omega \subset \mathbb{R}^d$ compact.

The Mathematical Learning Problem

Let $(\Sigma, \mathcal{G}, \mathbb{P})$ probability space. Given (Borel measurable) random vectors $X : \Sigma \to \mathbb{R}^d$, $Y : \Sigma \to \mathbb{R}^k$ with $\mathrm{im}(X) \subseteq \Omega$ for $\Omega \subset \mathbb{R}^d$ compact.
For any (Borel measurable) $f : \Omega \to \mathbb{R}^k$ define the *least squares error*

$$\mathcal{E}(f) := \mathbb{E}[(f(X) - Y)^2].$$

The Mathematical Learning Problem

Let $(\Sigma, \mathcal{G}, \mathbb{P})$ probability space. Given (Borel measurable) random vectors $X : \Sigma \to \mathbb{R}^d$, $Y : \Sigma \to \mathbb{R}^k$ with $\mathrm{im}(X) \subseteq \Omega$ for $\Omega \subset \mathbb{R}^d$ compact.

For any (Borel measurable) $f : \Omega \to \mathbb{R}^k$ define the *least squares error*

$$\mathcal{E}(f) := \mathbb{E}[(f(X) - Y)^2].$$

The learning problem asks for the function $\widehat{f}$ which minimizes $\mathcal{E}$.

The Mathematical Learning Problem

Let $(\Sigma, \mathcal{G}, \mathbb{P})$ probability space. Given (Borel measurable) random vectors $X : \Sigma \to \mathbb{R}^d$, $Y : \Sigma \to \mathbb{R}^k$ with $\mathrm{im}(X) \subseteq \Omega$ for $\Omega \subset \mathbb{R}^d$ compact.
For any (Borel measurable) $f : \Omega \to \mathbb{R}^k$ define the *least squares error*

$$\mathcal{E}(f) := \mathbb{E}[(f(X) - Y)^2].$$

The learning problem asks for the function $\widehat{f}$ which minimizes $\mathcal{E}$.

Theorem (Main Regression Theorem)

Let $\widehat{f} := \mathbb{E}[Y|X]$ be the regression function and $\sigma^2 := \mathcal{E}(\widehat{f})$. It holds that

$$\mathcal{E}(f) = \|f - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} + \sigma^2$$

Empirical Error

Given $\mathbf{z} = ((X^{(1)}, Y^{(1)}), \dots, (X^{(m)}, Y^{(m)}))$ be i.i.d. with $(X^{(1)}, Y^{(1)}) \sim (X, Y)$. Define the *empirical error*

$$\mathcal{E}_{\mathbf{z}}(f) := \frac{1}{m} \sum_{i=1}^{m} (f(X^{(i)}) - Y^{(i)})^2.$$

Empirical Error

Given $\mathbf{z} = ((X^{(1)}, Y^{(1)}), \dots, (X^{(m)}, Y^{(m)}))$ be i.i.d. with $(X^{(1)}, Y^{(1)}) \sim (X, Y)$. Define the *empirical error*

$$\mathcal{E}_{\mathbf{z}}(f) := \frac{1}{m} \sum_{i=1}^{m} (f(X^{(i)}) - Y^{(i)})^2.$$

Hypothesis Space

Let $\mathcal{H}$ be a compact subset of the Banach space $\{f : X \to Y, \text{ continuous}\}$ with norm $\|f\| := \max_{x \in X} |f(x)|$. We call $\mathcal{H}$ *hypothesis space* or *model space*.

Empirical Error

Given $\mathbf{z} = ((X^{(1)}, Y^{(1)}), \ldots, (X^{(m)}, Y^{(m)}))$ be i.i.d. with $(X^{(1)}, Y^{(1)}) \sim (X, Y)$. Define the *empirical error*

$$\mathcal{E}_{\mathbf{z}}(f) := \frac{1}{m} \sum_{i=1}^{m} (f(X^{(i)}) - Y^{(i)})^2.$$

Hypothesis Space

Let $\mathcal{H}$ be a compact subset of the Banach space $\{f : X \to Y, \text{ continuous}\}$ with norm $\|f\| := \max_{x \in X} |f(x)|$. We call $\mathcal{H}$ *hypothesis space* or *model space*.

Best Approximation in $\mathcal{H}$

$$\widehat{f}_{\mathcal{H}} := \operatorname{argmin}_{f \in \mathcal{H}} \mathcal{E}(f) = \operatorname{argmin}_{f \in \mathcal{H}} \|\widehat{f} - f\|_{L^2(\mathbb{R}^d, d\mathbb{P}_X)}.$$

Empirical Error

Given $\mathbf{z} = ((X^{(1)}, Y^{(1)}), \dots, (X^{(m)}, Y^{(m)}))$ be i.i.d. with $(X^{(1)}, Y^{(1)}) \sim (X, Y)$. Define the *empirical error*

$$\mathcal{E}_{\mathbf{z}}(f) := \frac{1}{m} \sum_{i=1}^{m} (f(X^{(i)}) - Y^{(i)})^2.$$

Hypothesis Space

Let $\mathcal{H}$ be a compact subset of the Banach space $\{f : X \to Y,\ \text{continuous}\}$ with norm $\|f\| := \max_{x \in X} |f(x)|$. We call $\mathcal{H}$ *hypothesis space* or *model space*.

Best Approximation in $\mathcal{H}$

$$\widehat{f}_{\mathcal{H}} := \operatorname{argmin}_{f \in \mathcal{H}} \mathcal{E}(f) = \operatorname{argmin}_{f \in \mathcal{H}} \|\widehat{f} - f\|_{L^2(\mathbb{R}^d, d\mathbb{P}_X)}.$$

Empirical Regression Function

$$\widehat{f}_{\mathcal{H},\mathbf{z}} := \operatorname{argmin}_{f \in \mathcal{H}} \mathcal{E}_{\mathbf{z}}(f).$$

Theorem (Bias-Variance Decomposition)

$$\|\widehat{f}_{\mathcal{H},\mathbf{z}} - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} = \Big(\mathcal{E}(\widehat{f}_{\mathcal{H},\mathbf{z}}) - \mathcal{E}(\widehat{f}_{\mathcal{H}})\Big) + \|\widehat{f}_{\mathcal{H}} - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)}.$$

"*error = generalization error + approximation error.*"

Theorem (Bias-Variance Decomposition)

$$\|\widehat{f}_{\mathcal{H},\mathbf{z}} - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} = \left(\mathcal{E}(\widehat{f}_{\mathcal{H},\mathbf{z}}) - \mathcal{E}(\widehat{f}_{\mathcal{H}})\right) + \|\widehat{f}_{\mathcal{H}} - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)}.$$

"*error = generalization error + approximation error.*"

Covering Number

Let S be a metric space and $s > 0$. $\mathcal{N}(S, s)$ is the minimal $l \in \mathcal{N}$ such that there exist l disks in S with radius s covering S.

Theorem (Bias-Variance Decomposition)

$$\|\widehat{f}_{\mathcal{H},\mathbf{z}} - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} = \Big(\mathcal{E}(\widehat{f}_{\mathcal{H},\mathbf{z}}) - \mathcal{E}(\widehat{f}_{\mathcal{H}})\Big) + \|\widehat{f}_{\mathcal{H}} - \widehat{f}\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)}.$$

"*error = generalization error + approximation error.*"

Covering Number

Let S be a metric space and $s > 0$. $\mathcal{N}(S, s)$ is the minimal $l \in \mathcal{N}$ such that there exist l disks in S with radius s covering S.

Theorem C

Let $\mathcal{H}$ be a hypothesis class. Assume that for all $f \in \mathcal{H}$ it holds that $|f(X) - Y| < M$ a.e. Then, for all $\varepsilon > 0$,

$$\mathbb{P}\left(\mathcal{E}(\widehat{f}_{\mathcal{H},\mathbf{z}}) - \mathcal{E}(\widehat{f}_{\mathcal{H}}) \leq \varepsilon\right) \geq 1 - \mathcal{N}(\mathcal{H}, \frac{\varepsilon}{16M})2e^{-\frac{m\varepsilon^2}{8(2\sigma^2 + \frac{1}{3}M^2\varepsilon)}},$$

where $\sigma^2 := \sup_{f \in \mathcal{H}} \sigma_f^2$.

Covering Numbers of ReLU Networks

For $\mathfrak{D} \in (0, \infty)$, $k, n \in \mathbb{N}$, $W \in \mathbb{R}^{k \times n}$, $B \in \mathbb{R}^k$ let

$$\mathcal{A}_{W,B}(x) = Wx + B \quad \text{and} \quad \mathcal{C}_{n,\mathfrak{D}}(x) = \big(\min\{|x_i|, \mathfrak{D}\} \operatorname{sgn}(x_i)\big)_{i=1}^n.$$

For $l \in \mathbb{N}_0$, $\mathfrak{D} \in (0, \infty)$ and a *network architecture*

$$\mathbf{a} = (a_0, a_1, \dots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$$

let the *number of hidden layers* and the *number of parameters* be given by

$$\mathcal{L}(\mathbf{a}) = l \quad \text{and} \quad \mathcal{P}(\mathbf{a}) = \sum_{i=0}^{l} a_{i+1}a_i + a_{i+1}$$

and for $x \in \mathbb{R}^{a_0}$ and *parameters*

$$\theta = ((W_i, B_i))_{i=0}^l \in \bigtimes_{i=0}^{l} \left(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\right) \simeq \mathbb{R}^{\mathcal{P}(\mathbf{a})}$$

let the *neural network* $\mathcal{F}_{\mathbf{a}} : \mathbb{R}^{\mathcal{P}(\mathbf{a})} \times \mathbb{R}^{a_0} \to \mathbb{R}^{a_{l+1}}$ be defined as

$$\mathcal{F}_{\mathbf{a}}(\theta, x) = \mathcal{A}_{W_l,B_l} \circ \mathrm{ReLU}_{a_l} \circ \mathcal{A}_{W_{l-1},B_{l-1}} \circ \mathrm{ReLU}_{a_{l-1}} \circ \dots \circ \mathrm{ReLU}_{a_1} \circ \mathcal{A}_{W_0,B_0}(x),$$

and let the *clipped neural network* $\mathcal{F}_{\mathbf{a},\mathfrak{D}} : \mathbb{R}^{\mathcal{P}(\mathbf{a})} \times \mathbb{R}^{a_0} \to \mathbb{R}^{a_{l+1}}$ be defined as

$$\mathcal{F}_{\mathbf{a},\mathfrak{D}}(\theta, x) = \mathcal{C}_{a_{l+1},\mathfrak{D}} \circ \mathcal{F}_{\mathbf{a}}(\theta, x).$$

For $l \in \mathbb{N}_0$, $R, \mathfrak{D} \in (0, \infty)$, $u \in \mathbb{R}$, $v \in (u, \infty)$, $\mathbf{a} = (a_0, a_1, \dots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$ let

$$\mathcal{NN}_{\mathbf{a},R}^{u,v} = \left\{ \left([u, v]^{a_0} \ni x \mapsto \mathcal{F}_{\mathbf{a}}(\theta, x)\right) : \ \theta \in \bigtimes_{i=0}^{l} \left(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\right), \ \|\theta\|_\infty \le R \right\}$$

$$\mathcal{NN}_{\mathbf{a},R,\mathfrak{D}}^{u,v} = \left\{ \left([u, v]^{a_0} \ni x \mapsto \mathcal{F}_{\mathbf{a},\mathfrak{D}}(\theta, x)\right) : \ \theta \in \bigtimes_{i=0}^{l} \left(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\right), \ \|\theta\|_\infty \le R \right\}$$

and for $d \in \mathbb{N}$ let $\mathbf{A}_d = \bigcup_{l \in \mathbb{N}_0} \{(a_0, a_1, \dots, a_l, a_{l+1}) \in \mathbb{R}^{l+2} : \ a_0 = d, \ a_{l+1} = 1\}$.

For $\mathfrak{D} \in (0, \infty)$, $k, n \in \mathbb{N}$, $W \in \mathbb{R}^{k \times n}$, $B \in \mathbb{R}^{k}$ let

$$\mathcal{A}_{W,B}(x) = Wx + B \quad \text{and} \quad \mathcal{C}_{n,\mathfrak{D}}(x) = \big(\min\{|x_i|, \mathfrak{D}\} \operatorname{sgn}(x_i) \big)_{i=1}^{n}.$$

For $l \in \mathbb{N}_0$, $\mathfrak{D} \in (0, \infty)$ and a *network architecture*

$$\mathbf{a} = (a_0, a_1, \ldots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$$

le

Clipping is needed since Statistical Learning Theory requires Hypothesis class to consist of bounded functions.

and for $x \in \mathbb{R}^{a_0}$ and *parameters*

$$\theta = \big((W_i, B_i)\big)_{i=0}^{l} \in \bigtimes_{i=0}^{l} \left(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\right) \simeq \mathbb{R}^{\mathcal{P}(\mathbf{a})}$$

let the *neural network* $\mathcal{F}_{\mathbf{a}} : \mathbb{R}^{\mathcal{P}(\mathbf{a})} \times \mathbb{R}^{a_0} \to \mathbb{R}^{a_{l+1}}$ be defined as

$$\mathcal{F}_{\mathbf{a}}(\theta, x) = \mathcal{A}_{W_l, B_l} \circ \mathrm{ReLU}_{a_l} \circ \mathcal{A}_{W_{l-1}, B_{l-1}} \circ \mathrm{ReLU}_{a_{l-1}} \circ \cdots \circ \mathrm{ReLU}_{a_1} \circ \mathcal{A}_{W_0, B_0}(x),$$

and let the *clipped neural network* $\mathcal{F}_{\mathbf{a},\mathfrak{D}} : \mathbb{R}^{\mathcal{P}(\mathbf{a})} \times \mathbb{R}^{a_0} \to \mathbb{R}^{a_{l+1}}$ be defined as

$$\mathcal{F}_{\mathbf{a},\mathfrak{D}}(\theta, x) = \mathcal{C}_{a_{l+1},\mathfrak{D}} \circ \mathcal{F}_{\mathbf{a}}(\theta, x).$$

For $l \in \mathbb{N}_0$, $R, \mathfrak{D} \in (0, \infty)$, $u \in \mathbb{R}$, $v \in (u, \infty)$, $\mathbf{a} = (a_0, a_1, \ldots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$ let

$$\mathcal{N}\mathcal{N}_{\mathbf{a},R}^{u,v} = \left\{ \left([u, v]^{a_0} \ni x \mapsto \mathcal{F}_{\mathbf{a}}(\theta, x)\right) : \ \theta \in \bigtimes_{i=0}^{l} \left(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\right), \ \|\theta\|_\infty \le R \right\}$$

$$\mathcal{N}\mathcal{N}_{\mathbf{a},R,\mathfrak{D}}^{u,v} = \left\{ \left([u, v]^{a_0} \ni x \mapsto \mathcal{F}_{\mathbf{a},\mathfrak{D}}(\theta, x)\right) : \ \theta \in \bigtimes_{i=0}^{l} \left(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\right), \ \|\theta\|_\infty \le R \right\}$$

and for $d \in \mathbb{N}$ let $\mathbf{A}_d = \bigcup_{l \in \mathbb{N}_0} \{(a_0, a_1, \ldots, a_l, a_{l+1}) \in \mathbb{R}^{l+2} : \ a_0 = d, \ a_{l+1} = 1\}$.

For $\mathfrak{D} \in (0, \infty)$, $k, n \in \mathbb{N}$, $W \in \mathbb{R}^{k \times n}$, $B \in \mathbb{R}^k$ let

$$\mathcal{A}_{W,B}(x) = Wx + B \quad \text{and} \quad \mathcal{C}_{n,\mathfrak{D}}(x) = \big(\min\{|x_i|, \mathfrak{D}\} \operatorname{sgn}(x_i)\big)_{i=1}^n.$$

For $l \in \mathbb{N}_0$, $\mathfrak{D} \in (0, \infty)$ and a *network architecture*

$$\mathbf{a} = (a_0, a_1, \ldots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$$

le

Clipping is needed since Statistical Learning Theory requires Hypothesis class to consist of bounded functions.

and for $x \in \mathbb{R}^{a_0}$ and *parameters*

Exercise: Clipped networks are standard networks.

$$i=0$$

let the *neural network* $\mathcal{F}_{\mathbf{a}} : \mathbb{R}^{\mathcal{P}(\mathbf{a})} \times \mathbb{R}^{a_0} \to \mathbb{R}^{a_{l+1}}$ be defined as

$$\mathcal{F}_{\mathbf{a}}(\theta, x) = \mathcal{A}_{W_l,B_l} \circ \mathrm{ReLU}_{a_l} \circ \mathcal{A}_{W_{l-1},B_{l-1}} \circ \mathrm{ReLU}_{a_{l-1}} \circ \cdots \circ \mathrm{ReLU}_{a_1} \circ \mathcal{A}_{W_0,B_0}(x),$$

and let the *clipped neural network* $\mathcal{F}_{\mathbf{a},\mathfrak{D}} : \mathbb{R}^{\mathcal{P}(\mathbf{a})} \times \mathbb{R}^{a_0} \to \mathbb{R}^{a_{l+1}}$ be defined as

$$\mathcal{F}_{\mathbf{a},\mathfrak{D}}(\theta, x) = \mathcal{C}_{a_{l+1},\mathfrak{D}} \circ \mathcal{F}_{\mathbf{a}}(\theta, x).$$

For $l \in \mathbb{N}_0$, $R, \mathfrak{D} \in (0, \infty)$, $u \in \mathbb{R}$, $v \in (u, \infty)$, $\mathbf{a} = (a_0, a_1, \ldots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$ let

$$\mathcal{NN}^{u,v}_{\mathbf{a},R} = \left\{ \big([u,v]^{a_0} \ni x \mapsto \mathcal{F}_{\mathbf{a}}(\theta, x)\big) : \ \theta \in \bigtimes_{i=0}^{l} \big(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\big), \ \|\theta\|_\infty \le R \right\}$$

$$\mathcal{NN}^{u,v}_{\mathbf{a},R,\mathfrak{D}} = \left\{ \big([u,v]^{a_0} \ni x \mapsto \mathcal{F}_{\mathbf{a},\mathfrak{D}}(\theta, x)\big) : \ \theta \in \bigtimes_{i=0}^{l} \big(\mathbb{R}^{a_{i+1} \times a_i} \times \mathbb{R}^{a_{i+1}}\big), \ \|\theta\|_\infty \le R \right\}$$

and for $d \in \mathbb{N}$ let $\mathbf{A}_d = \bigcup_{l \in \mathbb{N}_0} \{(a_0, a_1, \ldots, a_l, a_{l+1}) \in \mathbb{R}^{l+2} : \ a_0 = d, \ a_{l+1} = 1\}$.

Lipschitz Property

Theorem

Let $u \in \mathbb{R}$, $v \in (u, \infty)$, $R \in [1, \infty)$, $l \in \mathbb{N}_0$, let

$$\mathfrak{m} = \max\left\{1, |u|, |v|\right\}$$

and let $\mathbf{a} = (a_0, a_1, a_2, \ldots, a_l, a_{l+1}) \in \mathbb{N}^{l+2}$. Then for every $\theta, \eta \in \mathbb{R}^{\mathcal{P}(\mathbf{a})}$ with

$$\max\{\|\theta\|_\infty, \|\eta\|_\infty\} \leq R$$

it holds that

$$\sup_{x \in [u,v]^{a_0}} |\mathcal{F}_{\mathbf{a}}(\theta, x) - \mathcal{F}_{\mathbf{a}}(\eta, x)| \leq \|\theta - \eta\|_\infty \frac{\mathfrak{m} R^l \left(3\|\mathbf{a}\|_\infty + 3\right)^{l+1}}{2}.$$

Lipschitz Implies Covering Estimates

Lemma

Let $n \in \mathbb{N}$, $R \in [1, \infty)$, $r \in (0, 1)$ and define $B_R = \left\{\theta \in \mathbb{R}^n : \|\theta\|_\infty \leq R\right\}$ with its metric induced by the norm $\|\cdot\|_\infty$. Then it holds that

$$\ln \mathcal{N}(B_R, r) \leq n \ln\left(\frac{4R}{r}\right).$$

Lipschitz Implies Covering Estimates

Lemma

Let $n \in \mathbb{N}$, $R \in [1, \infty)$, $r \in (0, 1)$ and define $B_R = \left\{\theta \in \mathbb{R}^n : \|\theta\|_\infty \leq R\right\}$ with its metric induced by the norm $\|\cdot\|_\infty$. Then it holds that

$$\ln \mathcal{N}(B_R, r) \leq n \ln \left(\frac{4R}{r}\right).$$

Theorem

Let $u \in \mathbb{R}$, $v \in (u, \infty)$, $d \in \mathbb{N}$, $R \in [1, \infty)$, $r \in (0, 1)$, let $\mathfrak{m} = \max\left\{1, |u|, |v|\right\}$ and let $\mathbf{a} \in \mathbf{A}_d$. Then it holds that

$$\begin{aligned}\ln \mathcal{N}\left(\mathcal{N}\mathcal{N}^{u,v}_{\mathbf{a},R,\mathfrak{D}}, r\right) &\leq \ln \mathcal{N}\left(\mathcal{N}\mathcal{N}^{u,v}_{\mathbf{a},R}, r\right) \\ &\leq \mathcal{P}(\mathbf{a})\Big[\ln\left(\frac{2\mathfrak{m}}{r}\right) + (\mathcal{L}(\mathbf{a}) + 1)\ln\left(3R\|\mathbf{a}\|_\infty + 3R\right)\Big].\end{aligned}$$

Proof of the Theorem.

Use Lemma and Lipschitz property (see blackboard). $\square$

Implications for Learning Problem

Theorem A

Let $(X, Y) : \Sigma \to (\mathbb{R}^d, \mathbb{R})$ define a learning problem. Assume that for $\varepsilon \in (0, 1)$ there exist $\mathbf{a}_\varepsilon \in \mathbf{A}_d$, $R_\varepsilon \in [1, \infty)$ and a clipped neural network $g_\varepsilon \in \mathcal{H}_\varepsilon := \mathcal{N}^{u,v}_{\mathbf{a}_\varepsilon, R_\varepsilon, \mathfrak{D}}$ such that it holds that $\left\| \widehat{f} - g_\varepsilon \right\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} \le \varepsilon/2$. Let

$$h(x) = 128\mathfrak{D}^4 x_1^2 \Big[1 + x_2 + x_4 \Big(\ln\big(64\mathfrak{D}\max\{1, |u|, |v|\} x_1\big) + \big(x_5 + 1\big)\big(x_3 + 2\big) \Big) \Big],$$

let $\varrho \in (0, 1)$ and $\mathbf{z} = ((X^{(i)}, X^{(i)}))_{i=1}^m$ i.i.d. $\sim (X, Y)$ with

$$m \ge h\left(2\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R_\varepsilon \|\mathbf{a}_\varepsilon\|_\infty), \mathcal{P}(\mathbf{a}_\varepsilon), \mathcal{L}(\mathbf{a}_\varepsilon)\right).$$

Then it holds that

$$\mathbb{P}\left[\left\| \widehat{f}_{\mathbf{z}, \mathcal{H}_\varepsilon} - \widehat{f} \right\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} \le \varepsilon \right] \ge 1 - \varrho.$$

Implications for Learning Problem

Same result as for linear regression!

Theorem A

Let $(X,Y):\Sigma \to (\mathbb{R}^d,\mathbb{R})$ define a learning problem. Assume that for $\varepsilon \in (0,1)$ there exist $\mathbf{a}_\varepsilon \in \mathbf{A}_d$, $R_\varepsilon \in [1,\infty)$ and a clipped neural network $g_\varepsilon \in \mathcal{H}_\varepsilon := \mathcal{N}^{u,v}_{\mathbf{a}_\varepsilon, R_\varepsilon, \mathfrak{D}}$ such that it holds that $\left\| \widehat{f} - g_\varepsilon \right\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} \le \varepsilon/2$. Let

$$h(x) = 128\mathfrak{D}^4 x_1^2 \Big[1+x_2+x_4\Big(\ln\big(64\mathfrak{D}\max\{1,|u|,|v|\}x_1\big)+\big(x_5+1\big)\big(x_3+2\big)\Big)\Big],$$

let $\varrho \in (0,1)$ and $\mathbf{z} = ((X^{(i)}, X^{(i)}))_{i=1}^m$ i.i.d. $\sim (X,Y)$ with

$$m \ge h\left(2\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R_\varepsilon \|\mathbf{a}_\varepsilon\|_\infty), \mathcal{P}(\mathbf{a}_\varepsilon), \mathcal{L}(\mathbf{a}_\varepsilon)\right).$$

Then it holds that

$$\mathbb{P}\left[\left\| \widehat{f}_{\mathbf{z},\mathcal{H}_\varepsilon} - \widehat{f} \right\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} \le \varepsilon\right] \ge 1-\varrho.$$

Implications for Learning Problem

Theorem A

Let $(X, Y) : \Sigma \to (\mathbb{R}^d, \mathbb{R})$ define a learning problem. Assume that for $\varepsilon \in (0,1)$ there exist $\mathbf{a}_\varepsilon \in \mathbf{A}_d$, $R_\varepsilon \in [1,\infty)$ and a clipped neural network $g_\varepsilon \in \mathcal{H}_\varepsilon := \mathcal{N}^{u,v}_{\mathbf{a}_\varepsilon, R_\varepsilon, \mathfrak{D}}$ such that it holds that $\left\|\widehat{f} - g_\varepsilon\right\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} \leq \varepsilon/2$. Let

$$h(x) = 128\mathfrak{D}^4 x_1^2 \Big[1 + x_2 + x_4 \Big(\ln\left(64\mathfrak{D} \max\{1, |u|, |v|\} x_1\right) + \left(x_5 + 1\right)\left(x_3 + 2\right)\Big)\Big],$$

let $\varrho \in (0,1)$ and $\mathbf{z} = ((X^{(i)}, X^{(i)}))_{i=1}^m$ i.i.d. $\sim (X, Y)$ with

$$m \geq h\left(2\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R_\varepsilon \|\mathbf{a}_\varepsilon\|_\infty), \mathcal{P}(\mathbf{a}_\varepsilon), \mathcal{L}(\mathbf{a}_\varepsilon)\right).$$

Then it holds that

$$\mathbb{P}\left[\left\|\widehat{f}_{\mathbf{z},\mathcal{H}_\varepsilon} - \widehat{f}\right\|^2_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} \leq \varepsilon\right] \geq 1 - \varrho.$$

Proof.

Combine Bias-Variance decomposition, Covering number estimate and genearlization result. □

PDEs as Learning Problems

Explicit Solution of Heat Equation if $g = 0$

Let $u(t, x)$ satisfy

$$\frac{\partial u}{\partial t}(t, x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3}, \quad u(0, x) = \varphi(x)$$

$t \in (0, \infty), x \in \mathbb{R}^3$; $d = 4$.

Explicit Solution of Heat Equation if $g = 0$

Let $u(t, x)$ satisfy

$$\frac{\partial u}{\partial t}(t,x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3}, \quad u(0,x) = \varphi(x)$$

$t \in (0, \infty), x \in \mathbb{R}^3$; $d = 4$.
Then

$$u(t,x) = \int_{\mathbb{R}^3} \varphi(y) \frac{1}{(4\pi t)^{3/2}} \exp(-|x-y|^2/4t)dy.$$

Explicit Solution of Heat Equation if $g = 0$

Let $u(t,x)$ satisfy

$$\frac{\partial u}{\partial t}(t,x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3}, \quad u(0,x) = \varphi(x)$$

$t \in (0,\infty), x \in \mathbb{R}^3$; $d = 4$.
Then

$$u(t,x) = \int_{\mathbb{R}^3} \varphi(y) \frac{1}{(4\pi t)^{3/2}} \exp(-|x-y|^2/4t) dy.$$

In other words

$$u(t,x) = \mathbb{E}\left[\varphi(Z_t^x)\right], \quad Z_t^x \sim \mathcal{N}(x, t^{1/2} I).$$

Explicit Solution of Heat Equation if $g = 0$

Let $u(t,x)$ satisfy

$$\frac{\partial u}{\partial t}(t,x) = \frac{\partial^2 u}{\partial x_1 \partial x_1} + \frac{\partial^2 u}{\partial x_2 \partial x_2} + \frac{\partial^2 u}{\partial x_3 \partial x_3}, \quad u(0,x) = \varphi(x)$$

$t \in (0,\infty), x \in \mathbb{R}^3$; $d = 4$.
Then

$$u(t,x) = \int_{\mathbb{R}^3} \varphi(y) \frac{1}{(4\pi t)^{3/2}} \exp(-|x-y|^2/4t) dy.$$

In other words

$$u(t,x) = \mathbb{E}\left[\varphi(Z_t^x)\right], \quad Z_t^x \sim \mathcal{N}(x, t^{1/2} I).$$

In other words, for $x \in [u,v]^3$ and $X \sim \mathcal{U}[u,v]^3$ and $Y = \varphi\left(Z_t^X\right)$ we have

$$u(t,x) = \mathbb{E}\left[Y | X = x\right].$$

Ex

The solution $u(t,x)$ of the PDE can be interpreted as solution to the learning problem with data distribution (X,Y), where $X \sim \mathcal{U}[u,v]^3$ and $Y = \varphi(Z_t^X)$ and $Z_t^X \sim \mathcal{N}(x, t^{1/2} I)$!

$t \in (0,\infty), x \in \mathbb{R}^3$; $d = 4$.

Then

$$u(t,x) = \int_{\mathbb{R}^3} \varphi(y) \frac{1}{(4\pi t)^{3/2}} \exp(-|x-y|^2/4t) dy.$$

In other words

$$u(t,x) = \mathbb{E}\left[\varphi(Z_t^x)\right], \quad Z_t^x \sim \mathcal{N}(x, t^{1/2} I).$$

In other words, for $x \in [u,v]^3$ and $X \sim \mathcal{U}[u,v]^3$ and $Y = \varphi\left(Z_t^X\right)$ we have

$$u(t,x) = \mathbb{E}\left[Y | X = x\right].$$

Ex

The solution $u(t,x)$ of the PDE can be interpreted as solution to the learning problem with data distribution (X,Y), where $X \sim \mathcal{U}[u,v]^3$ and $Y = \varphi(Z_t^X)$ and $Z_t^X \sim \mathcal{N}(x, t^{1/2} I)$!

Contrary to conventional ML problems, the data distribution is now explicitly known – we can simulate as much training data as we want!

In other words, for $x \in [u,v]^3$ and $X \sim \mathcal{U}[u,v]^3$ and $Y = \varphi\left(Z_t^X\right)$ we have

$$u(t,x) = \mathbb{E}\left[Y | X = x\right].$$

Ex

The solution $u(t,x)$ of the PDE can be interpreted as solution to the learning problem with data distribution (X,Y), where $X \sim \mathcal{U}[u,v]^3$ and $Y = \varphi(Z_t^X)$ and $Z_t^X \sim \mathcal{N}(x, t^{1/2}I)$!

Contrary to conventional ML problems, the data distribution is now explicitly known – we can simulate as much training data as we want!

We will see in a minute that similar properties hold for a much more general class of PDEs!

Linear Kolmogorov Equations

Given $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and initial value $\varphi : \mathbb{R}^d \to \mathbb{R}$, find $u : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}$ with

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\mathrm{Trace}\left(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\right) + \mu(x)\cdot\nabla_x u(t,x),$$
$$(t,x) \in [0, T] \times \mathbb{R}^d, \quad u(0,x) = \varphi(x).$$

Linear Kolmogorov Equations

Given $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and initial value $\varphi : \mathbb{R}^d \to \mathbb{R}$, find $u : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}$ with

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\mathrm{Trace}\left(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\right) + \mu(x)\cdot\nabla_x u(t,x),$$
$$(t,x) \in [0,T]\times\mathbb{R}^d, \quad u(0,x) = \varphi(x).$$

- Examples include convection-diffusion equations and Black-Scholes Equation.

Linear Kolmogorov Equations

Given $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and initial value $\varphi : \mathbb{R}^d \to \mathbb{R}$, find $u : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}$ with

$$\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\mathrm{Trace}\left(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\right) + \mu(x)\cdot\nabla_x u(t,x),$$
$$(t,x) \in [0,T]\times\mathbb{R}^d, \quad u(0,x) = \varphi(x).$$

- Examples include convection-diffusion equations and Black-Scholes Equation.
- Standard methods such as sparse grid methods, sparse tensor product methods, spectral methods, finite element methods or finite difference methods are incapable of solving such equations in high dimensions ($d = 100$)!

Special Case: Pricing of Financial Derivatives

Special Case: Pricing of Financial Derivatives

- Given a portfolio consisting of d assets with value $(x_i(t))_{i=1}^d$.

Special Case: Pricing of Financial Derivatives

- Given a portfolio consisting of d assets with value $(x_i(t))_{i=1}^d$.
- *European Max Option:* At time T, exercise option and receive

$$G(x) := \max\left\{\max_{i=1}^{d}(x_i - K_i), 0\right\}$$

Special Case: Pricing of Financial Derivatives

- Given a portfolio consisting of d assets with value $(x_i(t))_{i=1}^d$.
- *European Max Option:* At time T, exercise option and receive

$$G(x) := \max\left\{\max_{i=1}^{d}(x_i - K_i), 0\right\}$$

- (Black-Scholes (1973)): in the absence of correlations the portfolio-value $u(t,x)$ satisfies

$$\left(\frac{\partial}{\partial t}\right) u(t,x) + \frac{\mu}{2}\sum_{i=1}^{d} x_i \left(\frac{\partial}{\partial x_i} u(t,x)\right) + \frac{\sigma^2}{2}\sum_{i=1}^{d} |x_i|^2 \left(\frac{\partial^2}{\partial x_i^2} u(t,x)\right) = 0,$$

$$u(T,x) = G(x).$$

Special Case: Pricing of Financial Derivatives

- Given a portfolio consisting of d assets with value $(x_i(t))_{i=1}^d$.
- *European Max Option:* At time T, exercise option and receive

$$G(x) := \max\left\{\max_{i=1}^{d}(x_i - K_i), 0\right\}$$

- (Black-Scholes (1973)): in the absence of correlations the portfolio-value $u(t,x)$ satisfies

$$\left(\frac{\partial}{\partial t}\right)u(t,x) + \frac{\mu}{2}\sum_{i=1}^{d} x_i\left(\frac{\partial}{\partial x_i}u(t,x)\right) + \frac{\sigma^2}{2}\sum_{i=1}^{d}|x_i|^2\left(\frac{\partial^2}{\partial x_i^2}u(t,x)\right) = 0,$$

$$u(T,x) = G(x).$$

- **Pricing Problem:** $u(0,x) = ??$.

Kolmogorov PDEs as Learning Problems

Kolmogorov PDEs as Learning Problems

For $x \in \mathbb{R}^d$ and $t \in \mathbb{R}_+$ let

$$Z_t^x := x + \int_0^t \mu(Z_s^x)ds + \int_0^t \sigma(Z_s^x)dW_s.$$

Then (Feynman-Kac)

$$u(T, x) = \mathbb{E}(\varphi(Z_T^x)).$$

Kolmogorov PDEs as Learning Problems

For $x \in \mathbb{R}^d$ and $t \in \mathbb{R}_+$ let

$$Z_t^x := x + \int_0^t \mu(Z_s^x)ds + \int_0^t \sigma(Z_s^x)dW_s.$$

Then (Feynman-Kac)

$$u(T, x) = \mathbb{E}(\varphi(Z_T^x)).$$

Lemma (Beck-Becker-G-Jafaari-Jentzen (2018))

Let $X \sim \mathcal{U}_{[a,b]^d}$ and let $Y = \varphi(Z_T^X)$. The solution $\hat{f}$ of the mathematical learning problem with data distribution (X, Y) is given by

$$\hat{f}(x) = u(T, x), \quad x \in [a, b]^d,$$

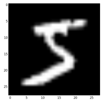
where u solves the corresponding Kolmogorov equation.

Numerical Results

The Vanilla DL Paradigm

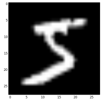
The Vanilla DL Paradigm

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:

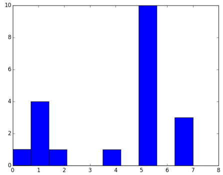


The Vanilla DL Paradigm

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:

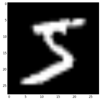


- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit

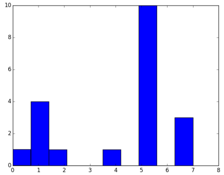


The Vanilla DL Paradigm

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:



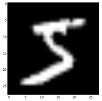
- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit



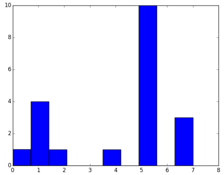
- Given labeled *training data* $(x_i, y_i)_{i=1}^m \subset \mathbb{R}^{784} \times \mathbb{R}^{10}$.

The Vanilla DL Paradigm

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:



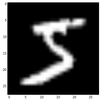
- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit



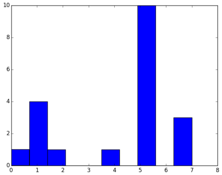
- Given labeled *training data* $(x_i, y_i)_{i=1}^m \subset \mathbb{R}^{784} \times \mathbb{R}^{10}$.
- Fix network architecture, e.g., number of layers (for example $L = 3$) and numbers of neurons ($N_1 = 30,\ N_2 = 30$).

The Vanilla DL Paradigm

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:



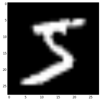
- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit



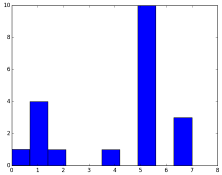
- Given labeled *training data* $(x_i, y_i)_{i=1}^m \subset \mathbb{R}^{784} \times \mathbb{R}^{10}$.
- Fix network architecture, e.g., number of layers (for example $L = 3$) and numbers of neurons ($N_1 = 30,\ N_2 = 30$).
- The learning goal is to find the empirical regression function $\widehat{f}_{\mathbf{z}} \in \mathcal{NN}_{(784,30,30,10)}$.

The Vanilla DL Paradigm

- Every image is given as a 28×28 matrix $x \in \mathbb{R}^{28\times 28} \sim \mathbb{R}^{784}$:



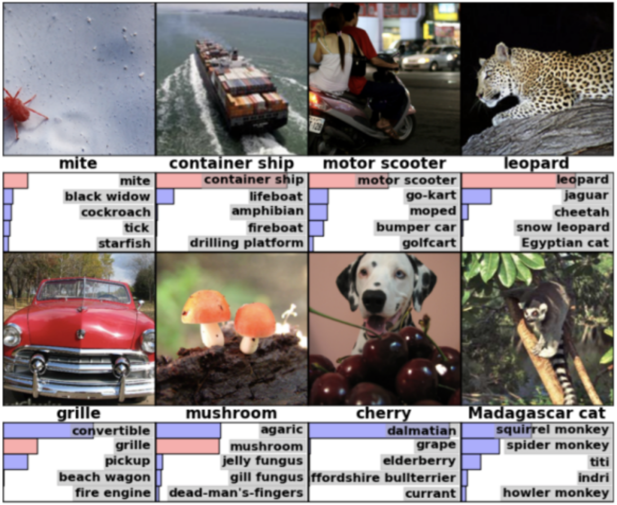
- Every label is given as a 10-dim vector $y \in \mathbb{R}^{10}$ describing the 'probability' of each digit



- Given labeled *training data* $(x_i, y_i)_{i=1}^m \subset \mathbb{R}^{784} \times \mathbb{R}^{10}$.
- Fix network architecture, e.g., number of layers (for example $L = 3$) and numbers of neurons ($N_1 = 30,\ N_2 = 30$).
- The learning goal is to find the empirical regression function $\widehat{f}_{\mathbf{z}} \in \mathcal{NN}_{(784,30,30,10)}$.
- Typically solved by stochastic first order optimization methods.

Description of Image Content

ImageNet Challenge



Deep Learning Algorithm

Deep Learning Algorithm

1. Generate training data $\mathbf{z} = (x_i, y_i)_{i=1}^m \overset{iid}{\sim} (X, \varphi(Z_X^T))$ by simulating Z_X^T with the Euler-Maruyama scheme.

Deep Learning Algorithm

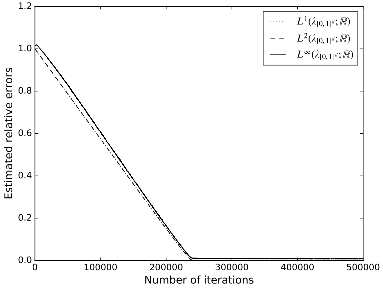
1. Generate training data $\mathbf{z} = (x_i, y_i)_{i=1}^m \overset{iid}{\sim} (X, \varphi(Z_X^T))$ by simulating Z_X^T with the Euler-Maruyama scheme.
2. Apply the Deep Learning Paradigm to this training data

Deep Learning Algorithm

1. Generate training data $\mathbf{z} = (x_i, y_i)_{i=1}^m \overset{iid}{\sim} (X, \varphi(Z_X^T))$ by simulating Z_X^T with the Euler-Maruyama scheme.
2. Apply the Deep Learning Paradigm to this training data ...meaning that
   (i) we pick a network architecture $(N_0 = d, N_1, \dots, N_L = 1)$, and let $\mathcal{H} = \mathcal{NN}_{(N_0,\dots,N_L)}$ and
   (ii) attempt to approximately compute

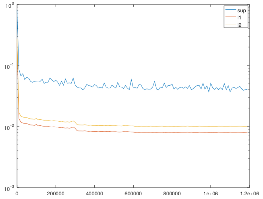
$$\widehat{f}_{\mathcal{H},\mathbf{z}} = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

in Tensorflow.



Number of steps	Relative $L^1(\lambda_{[0,1]^d};\mathbb{R})$-error	Relative $L^2(\lambda_{[0,1]^d};\mathbb{R})$-error	Relative $L^\infty(\lambda_{[0,1]^d};\mathbb{R})$-error	Runtime in seconds
0	0.998253	0.998254	1.003524	0.5
10000	0.957464	0.957536	0.993083	44.6
50000	0.786743	0.786806	0.828184	220.8
100000	0.574013	0.574060	0.605283	440.8
150000	0.361564	0.361594	0.384105	661.0
200000	0.001419	0.001784	0.010423	880.8
500000	0.001419	0.001784	0.010423	2200.7
750000	0.001419	0.001784	0.010423	3300.6

Figure: Estimated errors associated to the solution $u(1,\cdot)$ of the 100-dimensional parabolic PDE $\frac{\partial u}{\partial t}(t,x) = \Delta_x u(t,x)$, $u(0,x) = |x|^2$, $x \in [0,1]^{100}$.



Number of steps	Relative $L^1(\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^2(\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-error	Runtime in seconds
0	1.004285	1.004286	1.009524	1
25000	0.842938	0.843021	0.87884	110.2
50000	0.684955	0.685021	0.719826	219.5
100000	0.371515	0.371551	0.387978	437.9
150000	0.064605	0.064628	0.072259	656.2
250000	0.001220	0.001538	0.010039	1092.6
500000	0.000949	0.001187	0.005105	2183.8
750000	0.000902	0.001129	0.006028	3275.1

Figure: Estimated errors associated to the solution $u(T,\cdot)$ of the 100-dimensional uncorrelated Black Scholes PDE $\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i=1}^{d} |\sigma_i x_i|^2 (\frac{\partial^2 u}{\partial x_i^2})(t,x) + \sum_{i=1}^{d} \mu_i x_i (\frac{\partial u}{\partial x_i})(t,x)$, $u(0,x) = \exp(-rT)\max\big\{\max_{i\in\{1,2,\dots,d\}} x_i - 100, 0\big\}$, $x \in [90,110]^{100}$.

Number of steps	Relative $L^1(\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^2(\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-error	Runtime in seconds
0	1.003383	1.003385	1.011662	0.8
25000	0.631420	0.631429	0.640633	112.1
50000	0.269053	0.269058	0.275114	223.3
100000	0.000752	0.000948	0.00553	445.8
150000	0.000694	0.00087	0.004662	668.2
250000	0.000604	0.000758	0.006483	1119.3
500000	0.000493	0.000615	0.002774	2292.8
750000	0.000471	0.00059	0.002862	3466.8

Figure: Estimated errors associated to the solution $u(T,\cdot)$ of the 100-dimensional correlated Black Scholes PDE $\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{d} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^d} \big(\frac{\partial^2 u}{\partial x_i \partial x_j}\big)(t,x) + \sum_{i=1}^{d} \mu_i x_i \big(\frac{\partial u}{\partial x_i}\big)(t,x)$, $u(0,x) = \exp(-\mu T)\max\big\{110 - \min_{i\in\{1,2,\dots,d\}}\{x_i\}, 0\big\}$, $x \in [90,110]^{100}$.

Number of steps	Relative $L^1(\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^2(\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-error	Runtime in seconds
0	1.003383	1.003385	1.011662	0.8
25000	0.631420	0.631429	0.640633	112.1
50000	0.269053	0.269058	0.275114	223.3
100000	0.000752	0.000948	0.00553	445.8
150000	0.000694	0.00087	0.004662	668.2
250000	0.000604	0.000758	0.006483	1119.3
500000	0.000493	0.000615	0.002774	2292.8
750000	0.000471	0.00059	0.002862	3466.8

Figure: Estimated errors associated to the solution $u(T,\cdot)$ of the 100-dimensional correlated Black Scholes PDE $\frac{\partial u}{\partial t}(t,x) = \frac{1}{2}\sum_{i,j=1}^{d} x_i x_j \beta_i \beta_j \langle \varsigma_i, \varsigma_j \rangle_{\mathbb{R}^d} (\frac{\partial^2 u}{\partial x_i \partial x_j})(t,x) + \sum_{i=1}^{d} \mu_i x_i (\frac{\partial u}{\partial x_i})(t,x)$, $u(0,x) = \exp(-\mu T) \max\{110 - \min_{i\in\{1,2,\dots,d\}}\{x_i\}, 0\}$, $x \in [90,110]^{100}$.

All computations were performed in single precision (float32) on a NVIDIA GeForce GTX 1080 GPU with 1974 MHz core clock and 8 GB GDDR5X memory with 1809.5 MHz clock rate. The underlying system consisted of an Intel Core i7-6800K CPU with 64 GB DDR4-2133 memory running Tensorflow 1.5 on Ubuntu 16.04.

Approximation Results (special cases)

Basic MC Estimate

Lemma

Let $n \in N$ and let $(H_i)_{i=1}^n$ be i.i.d. random variables with $\mathbb{E}[|H_1|] < \infty$. Then

$$\mathbb{E}\left[\left(\mathbb{E}[H_1] - \frac{1}{n}\sum_{i=1}^{n} H_i\right)^2\right] = \frac{\mathbb{V}[H_1]}{n}.$$

Basic MC Estimate

Lemma

Let $n \in N$ and let $(H_i)_{i=1}^n$ be i.i.d. random variables with $\mathbb{E}[|H_1|] < \infty$. Then

$$\mathbb{E}\left[\left(\mathbb{E}[H_1] - \frac{1}{n}\sum_{i=1}^n H_i\right)^2\right] = \frac{\mathbb{V}[H_1]}{n}.$$

Proof.

$$\mathbb{E}\left[\left(\mathbb{E}[H_1] - \frac{1}{n}\sum_{i=1}^n H_i\right)^2\right] = \frac{1}{n^2}\mathbb{E}\left[\left(\sum_{i=1}^n \mathbb{E}[H_i] - H_i\right)^2\right] = \frac{1}{n^2}\sum_{i,j}^n \mathbb{E}\left[\left(\mathbb{E}[H_i] - H_i\right)\left(\mathbb{E}[H_j] - H_j\right)\right]$$
$$= \frac{1}{n}\mathbb{E}\left[\left(\mathbb{E}[H_1] - H_1\right)^2\right],$$

since by independence it holds that for $i \neq j$

$$\mathbb{E}\left[\left(\mathbb{E}[H_i] - H_i\right)\left(\mathbb{E}[H_j] - H_j\right)\right] = 0.$$

$\square$

...applied to Kolmogorv PDEs (special case)

Theorem

Let $n \in \mathbb{N}$ and $\left(Z_x^{T,(i)}\right)_{i=1}^{n}$ i.i.d. $\sim Z_x^T$. Then it holds that

$$\mathbb{E}\left[\int_{[0,1]^d}\left|u(T,x)-\frac{1}{n}\sum_{i=1}^{n}\varphi\left(Z_x^{T,(i)}\right)\right|^2 dx\right]=\frac{\int_{[0,1]^d}\mathbb{V}[\varphi(Z_x^T)]dx}{n}.$$

...applied to Kolmogorv PDEs (special case)

Theorem

Let $n \in \mathbb{N}$ and $\left(Z_x^{T,(i)}\right)_{i=1}^n$ i.i.d. $\sim Z_x^T$. Then it holds that

$$\mathbb{E}\left[\int_{[0,1]^d}\left|u(T,x)-\frac{1}{n}\sum_{i=1}^n \varphi\left(Z_x^{T,(i)}\right)\right|^2 dx\right] = \frac{\int_{[0,1]^d}\mathbb{V}[\varphi(Z_x^T)]dx}{n}.$$

Proof.

$$\mathbb{E}\left[\int_{[0,1]^d}\left|u(T,x)-\frac{1}{n}\sum_{i=1}^n \varphi\left(Z_x^{T,(i)}\right)\right|^2 dx\right] =$$

$$\int_{[0,1]^d}\mathbb{E}\left[\left|u(T,x)-\frac{1}{n}\sum_{i=1}^n \varphi\left(Z_x^{T,(i)}\right)\right|^2\right] dx.$$

Now use basic MC estimate together with $u(T,x)=\mathbb{E}[\varphi(Z_x^T)]$. □

A Concrete Approximation Result

Corollary

There exists an event $\omega \in \Sigma$ such that

$$\int_{[0,1]^d} \left| u(T,x) - \frac{1}{n} \sum_{i=1}^{n} \varphi\left(Z_x^{T,(i)}(\omega) \right) \right|^2 dx \leq \frac{\int_{[0,1]^d} \mathbb{V}[\varphi(Z_x^T)] dx}{n}$$

A Concrete Approximation Result

Corollary

There exists an event $\omega \in \Sigma$ such that

$$\int_{[0,1]^d} \left| u(T,x) - \frac{1}{n}\sum_{i=1}^{n} \varphi\left(Z_x^{T,(i)}(\omega)\right) \right|^2 dx \leq \frac{\int_{[0,1]^d} \mathbb{V}[\varphi(Z_x^T)]dx}{n}$$

Proof.

By the previous theorem we have a bound on the expectation of a nonnegative random variable. Therefore there must exist an event that attains this bound. $\square$

Further specialization: Affine Kolmogorov Equations

Theorem

Suppose that μ and σ are affine functions (this includes heat equation and Black-Scholes!). Then there exist random functions $\mathcal{A}_T : \Sigma \to \mathbb{R}^{d\times d}$ and $\mathcal{B}_T : \Sigma \to \mathbb{R}^d$ with

$$Z_x^T = \mathcal{A}_T x + \mathcal{B}_T$$

Further specialization: Affine Kolmogorov Equations

Theorem

Suppose that μ and σ are affine functions (this includes heat equation and Black-Scholes!). Then there exist random functions $\mathcal{A}_T : \Sigma \to \mathbb{R}^{d\times d}$ and $\mathcal{B}_T : \Sigma \to \mathbb{R}^d$ with

$$Z_x^T = \mathcal{A}_T x + \mathcal{B}_T$$

Corollary

Suppose that μ and σ are affine functions and let $n \in \mathbb{N}$ and $T \in (0,\infty)$. Then there exist $((A_i, b_i))_{i=1}^n \subset \mathbb{R}^{d\times d} \times \mathbb{R}^d$ with

$$\int_{[0,1]^d} \left| u(T,x) - \frac{1}{n}\sum_{i=1}^n \varphi\left(A_i x + b_i\right)\right|^2 dx \le \frac{\int_{[0,1]^d} \mathbb{V}[\varphi(Z_x^T)]dx}{n}.$$

A Priori Estimate

Theorem

Suppose that μ and σ are affine functions wwith $\|\mu(x)\| \leq L(1+\|x\|)$ and $\|\sigma(x)\| \leq L(1+\|x\|)$. Then

$$\mathbb{V}[\varphi(Z_x^T)] \leq \sqrt{2}(\|x\| + L(T+2\sqrt{T}))\exp(TL^2(2+\sqrt{T}).$$

A Priori Estimate

Theorem

Suppose that μ and σ are affine functions wwith $\|\mu(x)\| \leq L(1 + \|x\|)$ and $\|\sigma(x)\| \leq L(1 + \|x\|)$. Then

$$\mathbb{V}[\varphi(Z_x^T)] \leq \sqrt{2}(\|x\| + L(T + 2\sqrt{T}))\exp(TL^2(2 + \sqrt{T}).$$

Corollary

Suppose that μ and σ are affine functions and let $n \in \mathbb{N}$ and $T \in (0, \infty)$. Then there exist $((A_i, b_i))_{i=1}^n \subset \mathbb{R}^{d\times d} \times \mathbb{R}^d$ and C independent of d with

$$\int_{[0,1]^d} \left| u(T,x) - \frac{1}{n}\sum_{i=1}^n \varphi\left(A_i x + b_i\right)\right|^2 dx \leq \frac{Cd^{1/2}}{n}.$$

A Priori Estimate

Theorem

Suppose that μ and σ are affine functions wwith $\|\mu(x)\| \leq L(1+\|x\|)$ and $\|\sigma(x)\| \leq L(1+\|x\|)$. Then

$$\mathbb{V}[\varphi(Z_x^T)] \leq \sqrt{2}(\|x\| + L(T+2\sqrt{T}))\exp(TL^2(2+\sqrt{T}).$$

Corollary

Suppose that μ and σ are affine functions and let $n \in \mathbb{N}$ and $T \in (0,\infty)$. Then there exist $((A_i,b_i))_{i=1}^n \subset \mathbb{R}^{d\times d} \times \mathbb{R}^d$ and C independent of d with

$$\int_{[0,1]^d} \left| u(T,x) - \frac{1}{n}\sum_{i=1}^n \varphi\left(A_i x + b_i\right)\right|^2 dx \leq \frac{Cd^{1/2}}{n}.$$

Proof.

Use that $\int_{[0,1]^d} \|x\| dx \lesssim d^{1/2}$. □

Approximation Estimate (very special case)

Theorem B [Grohs, Hornung, Jentzen (2018)]

Suppose that $\varphi(x) = \max\{\mathfrak{D} - \sum_{i=1}^{d} c_i x_i, 0\}$ (European Put Option). Then for all $\epsilon \in (0,1)$ there exists $\mathbf{a}_\epsilon \in \{d\} \times \mathbb{N}^2 \times \{1\}$ with $\mathcal{P}(\mathbf{a}) \lesssim \frac{d^2}{\epsilon^2}$ and $R_\epsilon \lesssim \frac{d^2}{\epsilon}$ such that

$$\inf_{g \in \mathcal{N}\mathcal{N}^{0,1}_{\mathbf{a}_\epsilon, R_\epsilon, \mathfrak{D}}} \|u(T,\cdot) - g(\cdot)\|^2_{L^2[0,1]^d} \leq \epsilon.$$

Approximation Estimate (very special case)

Theorem B [Grohs, Hornung, Jentzen (2018)]

Suppose that $\varphi(x) = \max\{\mathfrak{D} - \sum_{i=1}^d c_i x_i, 0\}$ (European Put Option). Then for all $\epsilon \in (0,1)$ there exists $\mathbf{a}_\epsilon \in \{d\} \times \mathbb{N}^2 \times \{1\}$ with $\mathcal{P}(\mathbf{a}) \lesssim \frac{d^2}{\epsilon^2}$ and $R_\epsilon \lesssim \frac{d^2}{\epsilon}$ such that

$$\inf_{g \in \mathcal{NN}^{0,1}_{\mathbf{a}_\epsilon, R_\epsilon, \mathfrak{D}}} \|u(T,\cdot) - g(\cdot)\|^2_{L^2[0,1]^d} \leq \epsilon.$$

Proof.

Observe that φ is exactly a small ReLU network and observe that $\frac{1}{n}\sum_{i=1}^n \varphi(A_i \cdot + b_i)$ is also a ReLU network. Now apply the previous corollary. The part about R_ϵ requires an additional argument. □

Approximation Estimate (very special case)

Theorem B [Grohs, Hornung, Jentzen (2018)]

Suppose that $\varphi(x) = \max\{\mathfrak{D} - \sum_{i=1}^d c_i x_i, 0\}$ (European Put Option). Then for all $\epsilon \in (0,1)$ there exists $\mathbf{a}_\epsilon \in \{d\} \times \mathbb{N}^2 \times \{1\}$ with $\mathcal{P}(\mathbf{a}) \lesssim \frac{d^2}{\epsilon^2}$ and $R_\epsilon \lesssim \frac{d^2}{\epsilon}$ such that

$$\inf_{g \in \mathcal{NN}^{0,1}_{\mathbf{a}_\epsilon, R_\epsilon, \mathfrak{D}}} \|u(T,\cdot) - g(\cdot)\|^2_{L^2[0,1]^d} \leq \epsilon.$$

Proof.

We have approximation estimate without curse!

corollary. The part about R_ϵ requires an additional argument.

What we can really show

What we can really show

- μ and σ need not be affine (only well approximable by NNs). Proof becomes much more delicate...

What we can really show

- μ and σ need not be affine (only well approximable by NNs). Proof becomes much more delicate...
- Space-time approximation error estimates.

What we can really show

- μ and σ need not be affine (only well approximable by NNs). Proof becomes much more delicate...
- Space-time approximation error estimates.
- φ only needs to be well approximable by NNs.

Generalization Results (special cases)

A special case

Theorem C [Berner, Grohs, Jentzen (2018)]

Suppose that $\varphi(x) = \max\{\mathfrak{D} - \sum_{i=1}^d c_i x_i, 0\}$ (European Put Option). Then for all $\epsilon \in (0,1)$ there exists $\mathbf{a}_\epsilon \in \{d\} \times \mathbb{N}^2 \times \{1\}$ with $\mathcal{P}(\mathbf{a}) \lesssim \frac{d^2}{\epsilon^2}$ and $R_\epsilon \lesssim \frac{d^2}{\epsilon}$ such that for all $\varrho \in (0,1)$ and $\mathbf{z} = ((X^{(i)}, X^{(i)}))_{i=1}^m$ i.i.d. $\sim (X, Y)$ with

$$m \gtrsim d^2 \epsilon^{-4}(1 + \log(d\epsilon^{-1}\varrho^{-1})).$$

it holds that

$$\mathbb{P}\left[\left\|\widehat{f}_{\mathbf{z},\mathcal{NN}^{0,1}_{\mathbf{a}_\epsilon, R_\epsilon, \mathfrak{D}}} - u(T,\cdot)\right\|^2_{L^2([0,1]^d)} \leq \varepsilon\right] \geq 1 - \varrho.$$

A special case

Theorem C [Berner, Grohs, Jentzen (2018)]

Suppose that $\varphi(x) = \max\{\mathfrak{D} - \sum_{i=1}^{d} c_i x_i, 0\}$ (European Put Option). Then for all $\epsilon \in (0,1)$ there exists $\mathbf{a}_\epsilon \in \{d\} \times \mathbb{N}^2 \times \{1\}$ with $\mathcal{P}(\mathbf{a}) \lesssim \frac{d^2}{\epsilon^2}$ and $R_\epsilon \lesssim \frac{d^2}{\epsilon}$ such that for all $\varrho \in (0,1)$ and $\mathbf{z} = ((X^{(i)}, X^{(i)}))_{i=1}^{m}$ i.i.d. $\sim (X, Y)$ with

$$m \gtrsim d^2 \epsilon^{-4}(1 + \log(d\epsilon^{-1}\varrho^{-1})).$$

it holds that

$$\mathbb{P}\left[\left\|\widehat{f}_{\mathbf{z},\mathcal{NN}^{0,1}_{\mathbf{a}_\epsilon, R_\epsilon, \mathfrak{D}}} - u(T,\cdot)\right\|^2_{L^2([0,1]^d)} \leq \varepsilon\right] \geq 1 - \varrho.$$

Proof.

Combine Theorem A and Theorem B and note that

$$\|\cdot\|_{L^2(\mathbb{R}^d, d\mathbb{P}_X)} = \|\cdot\|_{L^2[0,1]^d}.$$

$\square$

A special case

Theorem C [Berner, Grohs, Jentzen (2018)]

Suppose that $\varphi(x) = \max\{\mathfrak{D} - \sum_{i=1}^{d} c_i x_i, 0\}$ (European Put Option). Then for all $\epsilon \in (0,1)$ there exists $\mathbf{a}_\epsilon \in \{d\} \times \mathbb{N}^2 \times \{1\}$ with $\mathcal{P}(\mathbf{a}) \lesssim \frac{d^2}{\epsilon^2}$ and $R_\epsilon \lesssim \frac{d^2}{\epsilon}$ such that for all $\varrho \in (0,1)$ and $\mathbf{z} = ((X^{(i)}, X^{(i)}))_{i=1}^{m}$ i.i.d. $\sim (X, Y)$ with

$$m \gtrsim d^2 \epsilon^{-4}(1 + \log(d\epsilon^{-1}\varrho^{-1})).$$

it holds that

$$\mathbb{P}\left[\left\|\widehat{f}_{\mathbf{z},\mathcal{NN}^{0,1}_{\mathbf{a}_\epsilon,R_\epsilon,\mathfrak{D}}} - u(T,\cdot)\right\|^2_{L^2([0,1]^d)} \leq \varepsilon\right] \geq 1 - \varrho.$$

Proof.

Combine Theorem A and Theorem B and note that

We also have a generalization estimate without curse!

$\square$

What we can really show

- μ and σ need not be affine (only well approximable by NNs). Proof becomes much more delicate...
- Theory for Semilinear PDEs (more complicated Feynman-Kac formula)
- Space-time approximation error estimates.
- φ only needs to be well approximable by NNs.