Learning the random variables: Combining Monte Carlo simulations with machine learning

Philippe von Wurstemberger
(The Chinese University of Hong Kong, Shenzhen & ETH Zurich, Switzerland)

Based on joint works with

Sebastian Becker (ETH Zurich, Switzerland),
Arnulf Jentzen (The Chinese University of Hong Kong, Shenzhen & University of Münster, Germany),
Marvin S. Müller (2Xideas Switzerland AG), and
Adrian Riekert (University of Münster, Germany)

FoCM 2023

Monday, June 12th 2023

Goal: Approximate solutions of parametric Integration problem:

$$u(x) = \mathbb{E}\big[g(x, \mathcal{W})\big], \qquad \forall\, x \in [0,1]^d,$$

where $d, \delta \in \mathbb{N}$, $u \in C([0,1]^d, \mathbb{R})$, $g \in C_b([0,1]^d \times \mathbb{R}^\delta, \mathbb{R})$, $(\Omega, \mathcal{F}, \mathbb{P})$ is a prob. space, and $\mathcal{W}\colon \Omega \to \mathbb{R}^\delta$ is a random variable.

Example application: Pricing of financial options

Reformulation as minimization problem

Let $X\colon \Omega \to [0,1]^d$ be $\mathcal{U}_{[0,1]^d}$-distributed and independent of $\mathcal{W}$. Then

(i) *there exists a unique $U \in C([0,1]^d, \mathbb{R})$ such that*

$$\inf_{v \in C([0,1]^d, \mathbb{R})} \mathbb{E}\big[|v(X) - g(X, \mathcal{W})|^2\big] = \mathbb{E}\big[|U(X) - g(X, \mathcal{W})|^2\big] \tag{1}$$

(ii) *it holds for every $x \in [0,1]^d$ that $U(x) = u(x)$.*

A first machine learning approach (Beck, Becker, Grohs, Jaafari, Jentzen 18 & Berner, Dablander, Grohs 20): Look for minimizer of (1) in a class of ANNs.

Let $L \in \mathbb{N}$, $\ell_0, \ldots, \ell_L \in \mathbb{N}$ satisfy $\ell_0 = d$ and $\ell_L = 1$, let $\mathbf{N} = \times_{n=1}^{L} (\mathbb{R}^{\ell_n \times \ell_{n-1}} \times \mathbb{R}^{\ell_n})$, $a \in C^1(\mathbb{R}, \mathbb{R})$, let $\mathcal{N} \colon \mathbf{N} \times \mathbb{R}^d \to \mathbb{R}$ satisfy for all $\phi = ((W_1, B_1), \ldots, (W_L, B_L)) \in \mathbf{N}$, $x_0 \in \mathbb{R}^{\ell_0}$, …, $x_{L-1} \in \mathbb{R}^{\ell_{L-1}}$ with $\forall\, n \in \{1, \ldots, L-1\} \colon x_n = a(W_n x_{n-1} + B_n)$ that

$$\mathcal{N}(\phi, x_0) = W_L x_{L-1} + B_L.$$

New goal: Minimize over $\phi \in \mathbf{N}$:

$$\mathbb{E}[|\mathcal{N}(\phi, X) - g(X, \mathcal{W})|^2].$$

Let $\mathrm{Loss}_{\mathcal{N}} \colon \mathbf{N} \times \mathbb{R}^d \times \mathbb{R}^\delta \to \mathbb{R}$ satisfy for all $\phi \in \mathbf{N}$, $x \in \mathbb{R}^d$, $w \in \mathbb{R}^\delta$ that

$$\mathrm{Loss}_{\mathcal{N}}(\phi, x, w) = |\mathcal{N}(\phi, x) - g(x, w)|^2,$$

let $W_{m,\mathfrak{m}} \colon \Omega \to \mathbb{R}$, $m, \mathfrak{m} \in \mathbb{N}_0$, be i.i.d. RVs with $W_{0,0}(\mathbb{P}) = \mathcal{W}(\mathbb{P})$, let $X_{m,\mathbf{m}} \colon \Omega \to [0,1]^d$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. $\mathcal{U}_{[0,1]^d}$-distributed RVs independent of $(W_{m,\mathfrak{m}})_{(m,\mathfrak{m}) \in \mathbb{N}_0^2}$, let $\mathbf{M} \in \mathbb{N}$, $(\gamma_m)_{m \in \mathbb{N}} \subseteq (0, \infty)$, and let $\Phi \colon \mathbb{N}_0 \times \Omega \to \mathbf{N}$ satisfy for all $m \in \mathbb{N}$ that

$$\Phi_m = \Phi_{m-1} - \tfrac{\gamma_m}{\mathbf{M}} \left[\textstyle\sum_{\mathbf{m}=1}^{\mathbf{M}} (\nabla_\phi \mathrm{Loss}_{\mathcal{N}})(\Phi_{m-1}, X_{m,\mathbf{m}}, W_{m,\mathbf{m}}) \right].$$

For every sufficiently large $m \in \mathbb{N}$ approximate $u \colon [0,1]^d \to \mathbb{R}$ with the (random) function

$$[0,1]^d \ni x \mapsto \mathcal{N}(\Phi_m, x) \in \mathbb{R}.$$

Learning the random variables (LRV) strategy: (Becker, Jentzen, Müller, PvW 22)
Let $\mathfrak{M} \in \mathbb{N}$, and observe that for all $x \in [0,1]^d$ we have that

$$u(x) = \mathbb{E}\big[g(x, \mathcal{W})\big] \approx \tfrac{1}{\mathfrak{M}} \big[\textstyle\sum_{\mathfrak{m}=1}^{\mathfrak{M}} g(x, W_{0,\mathfrak{m}}) \big]. \tag{2}$$

Let $\mathcal{N} \colon \mathbb{R}^{\delta\mathfrak{M}} \times [0,1]^d \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \ldots, \theta_{\delta\mathfrak{M}}) \in \mathbb{R}^{\delta\mathfrak{M}}$, $x \in [0,1]^d$ that

$$\mathcal{N}(\theta, x) = \tfrac{1}{\mathfrak{M}} \big[\textstyle\sum_{\mathfrak{m}=1}^{\mathfrak{M}} g\big(x, (\theta_{(\mathfrak{m}-1)\delta+1}, \theta_{(\mathfrak{m}-1)\delta+2}, \ldots, \theta_{(\mathfrak{m}-1)\delta+\delta})\big) \big]$$

and note that (2) suggests for all $x \in [0,1]^d$ that

$$u(x) \approx \tfrac{1}{\mathfrak{M}} \big[\textstyle\sum_{\mathfrak{m}=1}^{\mathfrak{M}} g(x, W_{0,\mathfrak{m}}) \big] = \mathcal{N}((W_{0,1}, W_{0,2}, \ldots, W_{0,\mathfrak{M}}), x).$$

New goal: Minimize over $\theta \in \mathbb{R}^{\delta\mathfrak{M}}$:

$$\mathbb{E}[|\mathcal{N}(\theta, X) - g(X, \mathcal{W})|^2].$$

Let $\mathrm{Loss}_{\mathcal{N}} \colon \mathbb{R}^{\delta\mathfrak{M}} \times \mathbb{R}^d \times \mathbb{R}^\delta \to \mathbb{R}$ satisfy for all $\theta \in \mathbb{R}^{\delta\mathfrak{M}}$, $x \in \mathbb{R}^d$, $w \in \mathbb{R}^\delta$ that

$$\mathrm{Loss}_{\mathcal{N}}(\theta, x, w) = |\mathcal{N}(\theta, x) - g(x, w)|^2,$$

and let $\Theta \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\delta\mathfrak{M}}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (W_{0,1}, \ldots, W_{0,\mathfrak{M}})$ and

$$\Theta_m = \Theta_{m-1} - \tfrac{\gamma_m}{\mathbf{M}} \big[\textstyle\sum_{\mathbf{m}=1}^{\mathbf{M}} (\nabla_\theta \mathrm{Loss}_{\mathcal{N}})(\Theta_{m-1}, X_{m,\mathbf{m}}, W_{m,\mathbf{m}}) \big].$$

For every sufficiently large $m \in \mathbb{N}$ approximate $u \colon [0,1]^d \to \mathbb{R}$ with the (random) function

$$[0,1]^d \ni x \mapsto \mathcal{N}(\Theta_m, x) \in \mathbb{R}.$$

Numerical results for European call options in the Black-Scholes model:

Parameter set (cf. $[0,1]^d$ above):

$$\mathfrak{P} = [\tfrac{1}{100}, 1] \times [90, 110] \times [90, 110] \times [-\tfrac{1}{10}, \tfrac{1}{10}] \times [\tfrac{1}{100}, \tfrac{1}{2}].$$

For all $(T, \xi, K, r, \sigma) \in \mathfrak{P}$ we have that $u(T, \xi, K, r, \sigma) \in \mathbb{R}$ corresponds to price of European call option with maturity T, initial price ξ, strike price K, drift rate r, and volatility σ.

Approximation method	Trainable parameters	L^2-approximation error	L^∞-approximation error	Training time	Time for 8192000
ANNs (2 hidden layers)	4609	0.006354	0.256444	461.10	2.28
ANNs (2 hidden layers)	17409	0.004096	0.238210	463.67	2.26
ANNs (3 hidden layers)	8769	0.002867	0.144216	511.94	2.45
ANNs (3 hidden layers)	33921	0.002131	0.113354	519.46	2.47
MC (32768 samples)	-	0.081803	1.108698	0	22.87
QMC (32768 samples)	-	0.002586	0.011917	0	21.18
LRV	8192	0.000149	0.002935	2290.65	5.85

Outlook: Generalize LRV method to other algorithms: Combine classical numerics with ML

$\Rightarrow$ Design network architectures and initializations s.th. at init.: model = algorithm

$\Rightarrow$ **Algorithmically Designed Artificial Neural Networks** (ADANNs)
(Jentzen, Riekert, PvW 23)

Example: Semilinear heat PDE

Let $T \in (0, \infty)$, $f \in C(\mathbb{R}, \mathbb{R})$, $\forall\, g \in \mathcal{I} \subseteq C([0,1], \mathbb{R})$ let $u_g \in C^{1,2}([0,T] \times [0,1], \mathbb{R})$ satisfy $\forall\, t \in [0,T], x \in [0,1]:\ u_g(0,x) = g(x)$, $u_g(t,0) = u_g(t,1) = 0$, and

$$\dot{u}_g(t,x) = (\Delta_x u_g)(t,x) + f(u_g(t,x)).$$

Goal: Approximate operator

$$\mathcal{S}\colon \mathcal{I} \to C([0,1], \mathbb{R})$$
$$g \mapsto u_g(T, \cdot).$$

Spatial finite difference discretization:

Let $N \in \mathbb{N}$, $x_1 = \frac{1}{N+1}, \dots, x_N = \frac{N}{N+1}$, $A = (1+N)^2 \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}$. Then $\forall\, g \in \mathcal{I}, t \in [0, T]$:

$$\dot{\mathbf{u}}_g(t) \approx A\mathbf{u}_g(t) + f(\mathbf{u}_g(t)),$$

where $\mathbf{u}_g(t) = (u_g(t, x_1), u_g(t, x_2), \dots, u_g(t, x_N))$.

Temporal linearly implicit discretization:

Let $M \in \mathbb{N}$, $h = T/M$. Then $\forall\, t \in [0, T-h]$:

$$\begin{aligned} \mathbf{u}_g(t+h) &\approx (1 - hA)^{-1}(\mathbf{u}_g(t) + hf(\mathbf{u}_g(t))) \\ &= \mathbf{W}_1\mathbf{u}_g(t) + \mathbf{W}_2 f(\mathbf{u}_g(t)), \end{aligned}$$

where $\mathbf{W}_1 = (1 - hA)^{-1}$, $\mathbf{W}_2 = h(1 - hA)^{-1}$.

Algorithmically designed artificial neural network (ADANN):

Let $\mathcal{N}\colon ((\mathbb{R}^{N\times N})^2)^M \times \mathcal{I} \to \mathbb{R}^N$ sat. $\forall\, W = ((W_{m,i})_{i\in\{1,2\}})_{m\in\{1,2,\dots,M\}} \in ((\mathbb{R}^{N\times N})^2)^M$, $g \in \mathcal{I}$, $U_0, U_1, \dots, U_M \in \mathbb{R}^N$ with $U_0 = (g(x_1), \dots, g(x_N))$ and

$$\forall\, m \in \{1, 2, \dots, M\}\colon \qquad U_m = W_{m,1}U_{m-1} + W_{m,2}f(U_{m-1})$$

that $\mathcal{N}(W, g) = U_M$. If N, M sufficiently large $\forall\, g \in \mathcal{I}$:

$$\mathcal{N}(\underbrace{((\mathbf{W}_1, \mathbf{W}_2), \dots, (\mathbf{W}_1, \mathbf{W}_2))}_{M\text{-times}}, g) \approx \mathbf{u}_g(T) \approx u_g(T) = \mathcal{S}(g).$$

Numerical results for a 1-dimensional Sine-Gordon type PDE

$$\left(\frac{\partial}{\partial t}u\right)(t,x) = \frac{1}{100}(\Delta_x u)(t,x) + \sin(u(t,x))$$

for $(t,x) \in [0,2] \times [0,1]$ with periodic boundary conditions.

Method	Estimated L^2-error	Trainable parameters	Training time	Time for 4096
ANN	0.014750	86390	228.48	0.003
FNO	0.004615	83455	936.53	0.020
Classical (15 timesteps)	0.005740	-	0.00	0.014
ADANN (15 timesteps)	0.000337	67500	3632.86	0.010

Recap

- Approximation of parametric integration/expectation
- LRV strategy: Problem specific ANNs (MC Neural networks)
  - Outperforms ANNs, MC, and QMC methods
  - Natural initialization for SGD
  - Needs a suitable stochastic *proposal algorithm*
- **Outlook**: General approach to combine classical numerics and deep learning
  - Algorithmically designed artificial neural networks (ADANNs)

Thank you!

Appendix

- LRV for the Lorentz equation
- Second order linearly implicit Runge-Kutta methods
- Distribution of the initial value for Sine Gordon example
- Literature

Appendix: Numerical results for a parametric stochastic Lorentz equation

$\forall\, \alpha = (\alpha_1, \alpha_2, \alpha_3), \beta = (\beta_1, \beta_2, \beta_3) \in \mathbb{R}^3$ let $u_{\alpha,\beta} \in C^{1,2}([0,\infty) \times \mathbb{R}^d, \mathbb{R})$ satisfy $\forall\, t \in [0,\infty), x \in \mathbb{R}^d\colon u_{\alpha,\beta}(0,x) = \|x\|^2$ and

$$\begin{aligned}(\tfrac{\partial u_{\alpha,\beta}}{\partial t})(t,x) &= \alpha_1(x_2 - x_1)(\tfrac{\partial u_{\alpha,\beta}}{\partial x_1})(t,x) + (\alpha_2 x_1 - x_2 - x_1 x_3)(\tfrac{\partial u_{\alpha,\beta}}{\partial x_2})(t,x) \\ &\quad + (x_1 x_2 - \alpha_3 x_3)(\tfrac{\partial u_{\alpha,\beta}}{\partial x_3})(t,x) + \textstyle\sum_{i=1}^{3} \frac{(\beta_i)^2}{2}(\frac{\partial^2 u_{\alpha,\beta}}{\partial (x_i)^2})(t,x).\end{aligned}$$

Parameter set ($(T, \alpha, \beta, x) \in \mathfrak{P}$):

$$\begin{aligned}\mathfrak{P} &= [0.01, 1] \times ([9, 11] \times [13, 15] \times [1, 2]) \\ &\quad \times [0.05, 0.25]^3 \times ([0.5, 2.5] \times [8, 10] \times [10, 12]).\end{aligned}$$

Approximation method	Trainable parameters	L^2-approximation error	L^∞-approximation error	Training time	Time for 8192000
ANNs (2 hidden layers)	18049	13.618965	377.537964	1869.14	0.65
ANNs (2 hidden layers)	68865	9.487827	106.434830	1882.02	0.66
ANNs (3 hidden layers)	34561	15.087503	200.595459	1903.12	0.63
ANNs (3 hidden layers)	134657	110.419151	313.946106	1891.98	0.63
Monte Carlo (32768 samples)	-	0.014373	0.114243	0	4.80
LRV (1024 samples)	76800	0.000958	0.015106	327.63	0.82

Appendix: Second order linearly implicit Runge-Kutta methods

Let $p_1, p_2 \in (0,\infty)$ and let $\Phi = (\Phi^h(u))_{(h,u)\in[0,\infty)\times\mathbb{R}^d} \colon [0,\infty)\times\mathbb{R}^d \to \mathbb{R}^d$ satisfy $\forall\, h \in [0,\infty)$, $U, k_1, k_2 \in \mathbb{R}^d$ with

$$k_1 = (I_d - hp_2A)^{-1}(AU + f(U)) \qquad \text{and}$$

$$k_2 = (I_d - hp_2A)^{-1}\big(A(U + h2p_1(\tfrac{1}{2} - p_2)k_1) + f(U + hp_1k_1)\big)$$

that

$$\Phi^h(U) = U + h\big[(1 - \tfrac{1}{2p_1})k_1 + (\tfrac{1}{2p_1})k_2\big].$$

Appendix: Distribution of the initial value for Sine Gordon example

As a distribution for the initial value $\mathfrak{I}\colon \Omega \to \mathcal{I}$ we use a sine expansion with decaying randomly distributed coefficients. Specifically, we have for all $x \in [0,1]$ that

$$\mathfrak{I}(x) = \sum_{n=1}^{32} \frac{5 Z_n \sin(\pi n x)}{n^2},$$

where $Z_n\colon \Omega \to \mathbb{R}$, $n \in \{1, 2, \ldots, 32\}$, are independent standard normally distributed random variables.

Appendix: Literature

- Becker, Jentzen, Müller, von Wurstemberger, *Learning the random variables in Monte Carlo simulations with stochastic gradient descent: Machine learning for parametric PDEs and financial derivative pricing. arXiv:2202.02717 (2022)*
- Riekert, Jentzen, von Wurstemberger, *Algorithmically Designed Artificial Neural Networks (ADANNs): Higher order deep operator learning for parametric partial differential equations. arXiv:2302.03286 (2023)*
- Beck, Becker, Grohs, Jaafari, Jentzen, *Solving the Kolmogorov PDE by means of deep learning. J. Sci. Comput. 88, 3 (2021)*
- Berner, Dablander, Grohs, *Numerically solving parametric families of high-dimensional Kolmogorov partial differential equations via deep learning. Advances in Neural Information Processing Systems (2020)*