1.5em

UNIVERSITY OF OXFORD

Mathematical Institute

Deep xVA solver

A neural network based counterparty credit risk management framework

CHRISTOPH REISINGER

based on work with
Athena Picarelli and Alessandro Gnoatto

CWI, 2 July 2020

Oxford Mathematics

Deep learning in finance

Machine learning, and neural networks (NN) in particular, have experienced an unprecedented surge of interest in the financial industry and academia.

On the buy ($\mathbb{P}$) side, NN have long been used to:

- forecast financial time series,
- detect trade signals,
- find optimal investment strategies (see talk by Yuying Li).

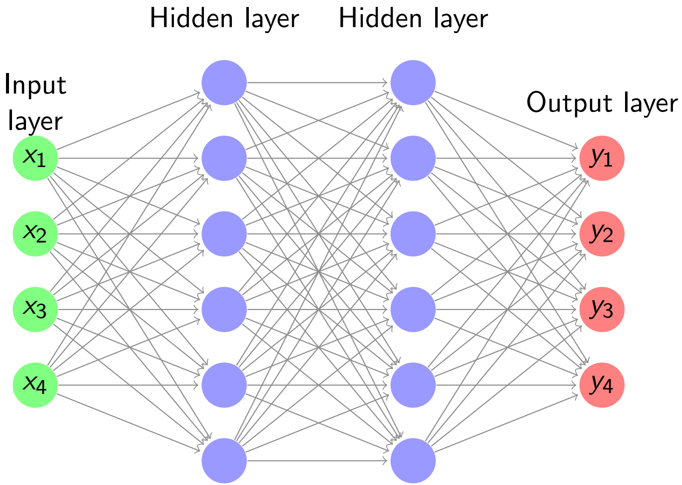
More recently also on the sell ($\mathbb{Q}$) side, to

- represent parametrised models for calibration (S. Liu, A. Borovykh);
- learn the value function in terms of spot, strike, etc;
- learn hedge ratios (S. Jain's talk).

Feedforward neural network

Two hidden layers, i.e. $\mathcal{L} = 3$, input and output dimension $d = 4$, $\nu = 6$ nodes.

UNIVERSITY OF OXFORD
Mathematical Institute



Feedforward neural network

We will consider functions $\varphi^{\varrho} : \mathbb{R}^d \mapsto \mathbb{R}^d$,

$$x \in \mathbb{R}^d \longmapsto \mathcal{A}_{\mathcal{L}} \circ \varrho \circ \mathcal{A}_{\mathcal{L}-1} \circ \ldots \circ \varrho \circ \mathcal{A}_1(x) \in \mathbb{R}^d,$$

where all $\mathcal{A}_\ell$, $\ell = 1, \ldots, \mathcal{L}$, are affine transformations

$$\mathcal{A}_1 : \mathbb{R}^d \mapsto \mathbb{R}^\nu, \quad \mathcal{A}_\ell : \mathbb{R}^\nu \mapsto \mathbb{R}^\nu, \quad \ell = 2, \ldots, \mathcal{L}-1, \quad \mathcal{A}_{\mathcal{L}} : \mathbb{R}^\nu \mapsto \mathbb{R}^d,$$

of the form $\mathcal{A}_\ell(x) := \mathcal{W}_\ell x + \beta_\ell$, $\ell = 1, \ldots, \mathcal{L}$.

The function ϱ is called *activation function*.

Regroup all parameters in a vector $\rho \in \mathbb{R}^R$.

Deep pricing and hedging

University of Oxford
Mathematical Institute

Over 150 papers since 1990s, much accelerated recently.

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Buehler et al. [2019a,b]	$\log(S)$	HR	BS-I	**CVaR**	Chronological	Simulation (BS, SV); S&P500. 5Y
Cao et al. [2019]	$S/K, \tau, \sigma_V$, underlying return	σ_I	HW	MSE	Random	S&P500. 8Y
Jang and Lee [2019]	?	C	BS-Cal, BW, KR, LSM, LV, SVJ, SVM	MAE, MAPE, MPE, MSE	?	S&P100. 9Y
Liu et al. [2019b]	$S/K, \tau$	σ_I	None	MAE, MAPE, MSE	Chronological	Simulation (BS)
Liu et al. [2019c]	$S/K, \tau, \sigma_{Cal}, r$	$(C - C_{BS-H})/K$	BS-Cal, SVJ	MAE, **MATE**, MPE, MSE	Chronological	DAX. 4Y
Karatas et al. [2019]	$S/K, \tau, r$, ?	C/K	None	MSE, R^2	Chronological	Simulation (BS, SV, VG)
Palmer [2019]	$S/K, \sigma_I\sqrt{\tau}, r$	C/K	BS-I, LSM	MAE, MAPE	Chronological	Simulation (BS)
Zheng et al. [2019]	$S/K, \tau$	σ_I	SSVI	MAPE	?	S&P500. 10Y
Ruf and Wang [2020]	$S/K, \sigma_I\sqrt{\tau}, \Delta$, $\mathcal{V}$, Vanna	HR	BS-I, HW, Linear	MSE	Chronological	Simulation (BS, SV); S&P500. 8Y; STOXX50. 3Y

Table 1: This table summarises more than 150 papers that use ANNs as a nonparametric option pricing or hedging tool. These papers are compared in terms of features (or so-called explanatory variables), outputs of the ANN, benchmark models, data partition between training and test sets, and the underlyings along with the time span of the data. The performance measures marked bold are related to evaluations along multiple periods. We refer to Tables 2–5 for a dictionary of all abbreviations used here.

Figure: Excerpt from Table 1 in: Neural Networks for Option Pricing and Hedging: A Literature Review, J. Ruf and W. Wang. J. Comput. Finance, July 2020.

Overview

What *this* talk is about:

Using deep learning for computation of valuation adjustments on portfolio level.

- Requires approximation of derivative values over scenarios in high dimensions.
  - We generate these scenarios by the "deep BSDE solver".
  - Compute simple exposures along these paths.
- Some valuation adjustments are recursive.
  - Nested application of the "deep BSDE solver".

See also talk by K. Andersson.

A derivative portfolio

Consider d risky asset values

$$S_t = (S_t^1, \ldots, S_t^d),$$

following the SDE

$$\begin{cases} \mathrm{d}S_t^m = \mu(t, S_t)\,\mathrm{d}t - \sigma(t, S_t)\,\mathrm{d}W_t^{\mathbb{Q}}, \\ S_0 = s_0, \end{cases}$$

and a portfolio of M (European, for simplicity) contingent claims,

$$\widehat{V}_t^m = \mathrm{E}^{\mathbb{Q}}\left[e^{-r(T_m - t)} g_m(S_{T_m}) | \mathcal{F}_t \right], \qquad m = 1, \ldots, M, \quad t \in [0, T].$$

Interest rate r universal and constant for time being.

FBSDEs and clean value

Then $(S, \widehat{V})$ solve the following (decoupled) FBSDE:

$$\begin{cases} \mathrm{d}S_t^m = \mu(t, S_t)\,\mathrm{d}t - \sigma(t, S_t)\,\mathrm{d}W_t^{\mathbb{Q}}, \\ -\mathrm{d}\widehat{V}_t^m = -r\widehat{V}_t^m\,\mathrm{d}t - \sum_{k=1}^{d} \widehat{Z}_t^{k,m}\,\mathrm{d}W_t^{k,\mathbb{Q}}, \\ S_0 = s_0, \\ \widehat{V}_{T_m}^m = g_m(S_{T_m}). \end{cases}$$

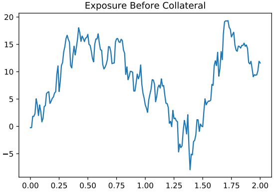
Let the "clean" portfolio value, without market frictions, be

$$\widehat{V}_t := \sum_{m=1}^{M} \widehat{V}_t^m.$$

UNIVERSITY OF OXFORD
Mathematical Institute

A bank (B) trades this portfolio with a counterparty (C).

- Default times: τ^j, for $j \in \{B, C\}$ and $\tau = \min(\tau^B, \tau^C\}$;
- Cash accounts: B_t^j, $j \in \{B, C\}$;
- Collaterals: C_t is exchanged between the parties. Assume $C_t = C(\widehat{V}_t)$ given.



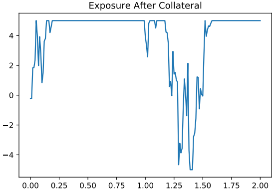


Figure: Pathwise simulation of a collateralized exposure. Left $\widehat{V}$ and right $\widehat{V} - C$, $C_t := (\widehat{V}_t - 5)^+ - (\widehat{V}_t + 5)^-$.

xVA, $x \in \{C, D, F, \ldots\}$

... account for market frictions.

- Credit Valuation Adjustment (CVA), for the risk of default of the counterparty, with intensity $\lambda^{C,\mathbb{Q}}$.
- Debt Valuation Adjustment (DVA), for the risk of the bank's default, with intensity $\lambda^{B,\mathbb{Q}}$.
- Funding Valuation Adjustment (FVA), for a differential of the interest rate r in an idealised collateral agreement, and
  - unsecured funding lending and borrowing rates: $r^{f,l}$, $r^{f,b}$;
  - interest on posted and received collateral: $r^{c,l}, r^{c,b}$.

See, e.g., Gregory (2015). The xVA challenge. Wiley.

Let $t < \tau$ (pre-default) and $\widetilde{r} = r + \lambda^{C,\mathbb{Q}} + \lambda^{B,\mathbb{Q}}$. Consider:

$$XVA_t = -CVA_t + DVA_t + FVA_t,$$

(see Biagini, Gnoatto, Oliva ('19), also Brigo, Morini, Pallavicini ('13))

$$CVA_t := B_t^{\widetilde{r}}\,\mathrm{E}^{\mathbb{Q}}\left[(1-R^C)\int_t^T \frac{1}{B_u^{\widetilde{r}}}\left(\widehat{V}_u - C_u\right)^{-}\lambda_u^{C,\mathbb{Q}}\,\mathrm{d}u \,\middle|\, \mathcal{F}_t\right],$$

$$DVA_t := B_t^{\widetilde{r}}\,\mathrm{E}^{\mathbb{Q}}\left[(1-R^B)\int_t^T \frac{1}{B_u^{\widetilde{r}}}\left(\widehat{V}_u - C_u\right)^{+}\lambda_u^{B,\mathbb{Q}}\,\mathrm{d}u \,\middle|\, \mathcal{F}_t\right],$$

$$FVA_t := B_t^{\widetilde{r}}\mathrm{E}^{\mathbb{Q}}\left[\int_t^T \frac{(r_u^{f,l} - r_u)\left(\widehat{V}_u - XVA_u - C_u\right)^{+}}{B_u^{\widetilde{r}}}\,\mathrm{d}u \,\middle|\, \mathcal{F}_t\right]$$
$$-B_t^{\widetilde{r}}\,\mathrm{E}^{\mathbb{Q}}\left[\int_t^T \frac{(r_u^{f,b} - r_u)\left(\widehat{V}_u - XVA_u - C_u\right)^{-}}{B_u^{\widetilde{r}}}\,\mathrm{d}u \,\middle|\, \mathcal{F}_t\right].$$

xVA computation

Require the (numerical) solution of BSDEs in possibly high dimension:

- for the exposures $\widehat{V}_t^m$, $m = 1, \ldots, M$;
- for the xVA itself.

Selected approaches:

- Albanese, Caenazzo, Crépey ('17): nested MC simulations;
- Shöftner ('08): "Longstaff-Schwartz" type Monte Carlo;
- de Graaf, Feng, Kandhai, Oosterlee ('14): hybrid PDE-MC scheme;
- de Graaf, Kandhai, R ('18), Chen, Christara ('20): PDE representation;
- Albanese, Crépey, Hoskinson, Saadeddine ('19), Henry-Labordére ('17): BSDEs.

Need the derivative values in the "mark-to-market cube".

Our method relies on the deep BSDE solver by E, Han, and Jentzen ('17).

Replace

$$X_t = x + \int_0^t b(s, X_s)\, \mathrm{d}s + \int_0^t a(s, X_s)^\top \, \mathrm{d}W_s^{\mathbb{Q}}, \quad x \in \mathbb{R}^d,$$

$$Y_t = \vartheta(X_T) + \int_t^T h(s, X_s, Y_s, Z_s)\, \mathrm{d}s + \int_t^T Z_s^\top \mathrm{d}W_s^{\mathbb{Q}}, \quad t \in [0, T],$$

by

$$\underset{y,\ \rho=(\rho_n)_{n\in 0,\dots,N-1}}{\text{minimise}} \ \mathbb{E}\left[\left|\vartheta(\widetilde{X}_N) - \widetilde{Y}_N^{y,\rho}\right|^2\right]$$

subject to

$$\widetilde{X}_{n+1} = \widetilde{X}_n + b(t_n, \widetilde{X}_n)\Delta t + a(t_n, \widetilde{X}_n)\Delta W_n, \qquad \widetilde{X}_0 = x,$$

$$\widetilde{Y}_{n+1} = \widetilde{Y}_n + h(t_n, \widetilde{X}_n, \widetilde{Y}_n, \widetilde{Z}_n)\Delta t + \widetilde{Z}_n^\top \Delta W_n, \qquad \widetilde{Y}_0 = y,$$

where $\widetilde{Z}_n = \phi_n^{\rho_n}(X_n)$ a NN parametrised by ρ_n.

The following error bound holds:

$$\sup_{t\in[0,T]} \mathbb{E}|Y_t - \widetilde{Y}_t^{y,\rho}|^2 + \int_0^T \mathbb{E}|Z_t - \widetilde{Z}_t^{\rho}|^2 \mathrm{d}t \leq C\left(\Delta t + \mathbb{E}\left[\left(\vartheta(\widetilde{X}_N) - \widetilde{Y}_N^{\rho,\xi}\right)^2\right]\right).$$

The last term can be estimated *a posteriori*.

It depends on the expression power of the NN and, in practice, the optimiser.

Deep BSDE solver

Mathematical Institute

Set parameters: N, L ;

▷ *time steps N , L paths for Monte Carlo loop*

Fix architecture of ANN;

▷ *intrinsically defines the dimension of parameters ρ*

Deep BSDE solver for exposure computation (N,L)

Simulate L paths $(S_n^{(\ell)})_{n=0,\dots,N}$, $\ell = 1, \dots, L$ of the forward dynamics;

Define the neural networks $(\varphi_n^{\rho})_{n=1,\dots,N}$;

for $m = 1, \dots, M$ do

minimize over ξ and ρ $\quad \frac{1}{L}\sum_{\ell=1}^{L}\left(g_m(S_N^{(\ell)}) - \mathcal{V}_N^{m,\rho,\xi,(\ell)}\right)^2$

subject to

$$\begin{cases} \mathcal{V}_{n+1}^{m,\rho,\xi,(\ell)} = \mathcal{V}_n^{m,\rho,\xi,(\ell)} + r_n \mathcal{V}_n^{m,\rho,\xi,(\ell)}\Delta t + (\mathcal{Z}_n^{\rho,(\ell)})^\top \Delta W_n^{(\ell)}, \\ \mathcal{V}_0^{m,\rho,\xi,(\ell)} = \xi \\ \mathcal{Z}_n^{\rho,(\ell)} = \varphi_n^{\rho}(S_n^{(\ell)}). \end{cases}$$

Save the optimizer $(\bar{\xi}^m, \bar{\rho}^m)$.

end

end

Algorithm 1: Deep algorithm for exposure simulation

Deep algorithm for non-recursive adjustments computation

UNIVERSITY OF OXFORD
Mathematical Institute

CVA and DVA can be written as $\mathrm{E}^{\mathbb{Q}}\left[\int_t^T \Phi(u, \widehat{V}_u)\mathrm{d}u \middle| \mathcal{F}_t\right].$

Set parameters: N, L, P ;

▷ *time steps* N *, paths for inner* (L) *and outer* (P) *Monte Carlo loop*

Fix architecture of ANN;

▷ *intrinsically defines the dimension of parameters* ρ

Apply Algorithm 1.

Simulate $(\mathcal{V}_n^{m,(p)})_{n=0\dots N, p=1\dots P}$ with $\xi = \bar{\xi}^m$, $\rho = \bar{\rho}^m$, $m = 1, \dots, M$

Define $\mathcal{V}_n^{(p)} = \sum_{m=1}^M \mathcal{V}_n^{m,(p)}$ for $n = 0, \dots, N$, $p = 1, \dots, P$;

Compute the adjustment $$\frac{1}{P}\sum_{i=1}^{P}\left(\sum_{n=0}^{N}\eta_n\Phi(\mathcal{V}_n^{(p)})\right)$$

where η_n are weights of the used quadrature form.

Algorithm 2: Deep algorithm for non-recursive adjustments

First: Single stock with Black-Scholes dynamics

$$\mathrm{d}S_t = rS_t\,\mathrm{d}t + \sigma S_t\,\mathrm{d}W_t^{\mathbb{Q}}, \qquad S_0 = s_0,$$

and a contingent claim defined by

$$\widehat{V}_t = \mathrm{E}\left[e^{-r(T-t)}g(S_T)|\mathcal{F}_t\right].$$

The expected discounted positive and negative exposure of $\widehat{V}$ are:

$$DEPE(s) = \mathrm{E}^{\mathbb{Q}}\left[e^{-r(s-t)}\left(\widehat{V}_s\right)^+\middle|\mathcal{F}_t\right],$$

$$DENE(s) = -\mathrm{E}^{\mathbb{Q}}\left[e^{-r(s-t)}\left(\widehat{V}_s\right)^-\middle|\mathcal{F}_t\right].$$

A forward on S: $g(S_T) = S_T - s_0$

Mathematical Institute

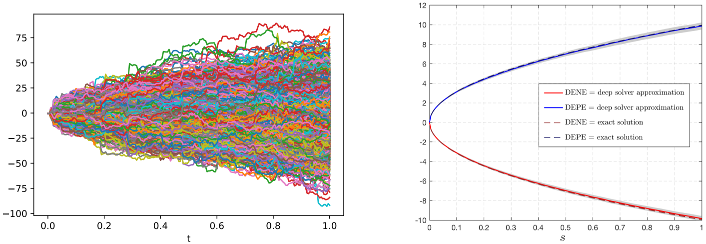
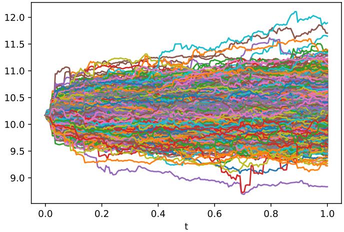


Figure: Exposure (l) and DEPE, DENE (r); $\sigma = 0.25$, $r = 0.01$, $s_0 = 100$, $T = 1$.

Parameters used: outer MC paths $P = 2048$, inner MC paths $L = 64$, internal layers $\mathcal{L} - 1 = 2$, $\nu = d + 20 = 21$, $\mathcal{I} = 4000$ (SGD iterations), time steps $N = 200$.

A European call: $g(S_T) = \left(S_T - s_0\right)^+$



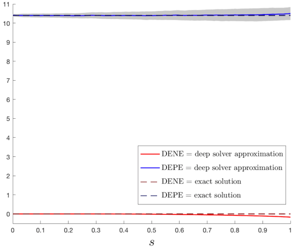


Figure: Exposure (l) and DEPE and DENE (r); $\sigma = 0.25$, $r = 0.01$, $s_0 = 100$, $T = 1$.

Parameters used: outer MC paths $P = 2048$, inner MC paths $L = 64$, internal layers $\mathcal{L} - 1 = 2$, $\nu = d + 20 = 21$, $\mathcal{I} = 4000$, time steps $N = 200$.

DEPE$_0$=10.407 (DEPE$_T$ = 10.496; exact: 10.404); $DENE_0 = 0$ (DENE$_T$ = -0.169)

A basket call: $g(S_T^1, \ldots, S_T^d) = \left(\sum_{i=1}^d S_T^i - K\right)^+$, $d = 100$.

UNIVERSITY OF OXFORD
Mathematical Institute

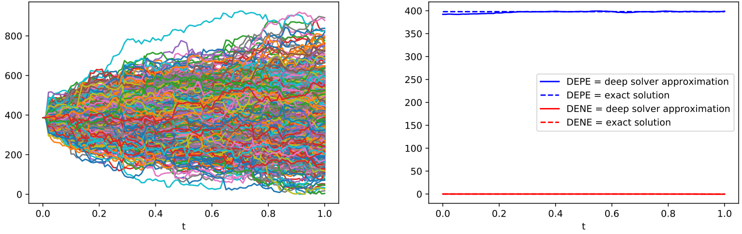


Figure: Exposure (l) and DEPE and DENE (r) for similar parameters.

Parameters used: outer MC paths $P = 1024$, inner MC paths $L = 64$, internal layers $\mathcal{L} - 1 = 2$, $\nu = d + 10 = 110$, $\mathcal{I} = 4000$, time steps $N = 100$.

The price varies over t between 393.02 and 400.82.

A time 0 MC estimate with 10^6 paths is 398.08, with C. I. [397.61, 398.56].

A recursive value adjustment: FVA

Let $t < \tau$ (pre-default). Consider now the funding term:

$$XVA_t = -CVA_t + DVA_t + FVA_t,$$

where

$$CVA_t := B_t^{\tilde{r}}\, \mathrm{E}^{\mathbb{Q}}\left[(1 - R^C) \int_t^T \frac{1}{B_u^{\tilde{r}}} \left(\widehat{V}_u - C_u\right)^- \lambda_u^{C,\mathbb{Q}}\, \mathrm{d}u \middle| \mathcal{F}_t\right],$$

$$DVA_t := B_t^{\tilde{r}}\, \mathrm{E}^{\mathbb{Q}}\left[(1 - R^B) \int_t^T \frac{1}{B_u^{\tilde{r}}} \left(\widehat{V}_u - C_u\right)^+ \lambda_u^{B,\mathbb{Q}}\, \mathrm{d}u \middle| \mathcal{F}_t\right],$$

$$FVA_t := B_t^{\tilde{r}} \mathrm{E}^{\mathbb{Q}}\left[\int_t^T \frac{(r_u^{f,l} - r_u)\left(\widehat{V}_u - XVA_u - C_u\right)^+}{B_u^{\tilde{r}}}\, \mathrm{d}u \middle| \mathcal{F}_t\right]$$

$$- B_t^{\tilde{r}}\, \mathrm{E}^{\mathbb{Q}}\left[\int_t^T \frac{(r_u^{f,b} - r_u)\left(\widehat{V}_u - XVA_u - C_u\right)^-}{B_u^{\tilde{r}}}\, \mathrm{d}u \middle| \mathcal{F}_t\right].$$

UNIVERSITY OF OXFORD

Mathematical Institute

BSDE model

The following BSDE representation also holds:

$$\begin{cases} -\mathrm{d}XVA_t = f\left(\widehat{V}_t, XVA_t\right)\,\mathrm{d}t - \sum_{k=1}^{d} Z_t^k\,\mathrm{d}W_t^{k,\mathbb{Q}}, \\ XVA_T = 0, \end{cases}$$

where

$$\begin{aligned} & f\left(\widehat{V}_t, XVA_t\right) := \\ & -(1-R^C)\left(\widehat{V}_t - C_t\right)^{-}\lambda_t^{C,\mathbb{Q}} \\ & +(1-R^B)\left(\widehat{V}_t - C_t\right)^{+}\lambda_t^{B,\mathbb{Q}} \\ & +(r_t^{f,l}-r_t)\left(\widehat{V}_t - XVA_t - C_t\right)^{+} - (r_t^{f,b}-r_t)\left(\widehat{V}_t - XVA_t - C_t\right)^{-}. \end{aligned}$$

Set parameters;
Fix architecture of ANNs;
Apply Algorithm 1
Simulate $(\mathcal{V}_n^{m,(p)})_{n=0\ldots N, p=1\ldots P}$ with $\xi = \bar{\xi}^m$, $\rho = \bar{\rho}^m$, $m = 1, \ldots, M$
Define $\mathcal{V}_n^{(p)} = \sum_{m=1}^M \mathcal{V}_n^{m,(p)}$ for $n = 0, \ldots, N$, $p = 1, \ldots, P$;
Deep BSDE solver for adjustment computation (N,P):

Define the neural networks $(\psi_n^\zeta)_{n=1,\ldots,N}$;

$$\text{minimize over } \gamma \text{ and } \zeta, \quad \frac{1}{P}\sum_{p=1}^{P}\left(\mathcal{X}_N^{\zeta,\gamma,(p)}\right)^2$$

$$\text{subject to } \begin{cases} \mathcal{X}_{n+1}^{\zeta,\gamma,(p)} = \mathcal{X}_n^{\zeta,\gamma,(p)} - f(\mathcal{V}_n^{(p)}, \mathcal{X}_n^{\zeta,(p)})\Delta t + (\mathcal{Z}_n^{\zeta,(p)})^\top \Delta W_n^{(p)}, \\ \mathcal{X}_0^{\zeta,\gamma,(p)} = \gamma, \\ \mathcal{Z}_n^{\zeta,(p)} = \psi_n^\zeta(\mathcal{V}_n^{(p)}). \end{cases}$$

end

Algorithm 3: Deep algorithm for xVA simulation

Error analysis Deep xVA Solver

Gnoatto, Picarelli, and R (2020)

UNIVERSITY OF OXFORD
Mathematical Institute

Consider

- $(\widetilde{XVA}^{\zeta,\gamma}, \widetilde{Z}^{\zeta,\gamma})$ numerical approximations to the xVA BSDE;
    - (here ζ and γ are the fitted NN parameters and initial value)
- $\widehat{V}^{\rho,y}$ the numerical approximation to the portfolio;
    - (here ρ and y are the fitted NN parameters and initial value)
- (XVA, Z) and V the true solutions.

Then

$$\sup_{t\in[0,T]} \mathbb{E}\left[\left|XVA_t - \widetilde{XVA}_t^{\zeta,\gamma}\right|^2\right] + \mathbb{E}\left[\int_0^T |Z_t - \widetilde{Z}_t^{\zeta,\gamma}|^2 \mathrm{d}t\right]$$
$$\leq K\left(\Delta t + \mathbb{E}\left[|\widehat{V}_T - \widehat{V}_T^{\rho,\xi}|^2\right] + \mathbb{E}\left[|\widetilde{XVA}_T^{\zeta,\gamma}|^2\right]\right).$$

Forward – FVA

In this stylised test, we

- ignore CVA and DVA, i.e., set $\tau^C = \tau^B = +\infty$;
- assume fully uncollateralized, i.e. $C_t \equiv 0$;
- set $r^{c,b} = r^{c,l} = r = 0.02$, and $r^{f,b} = r^{f,l} = 0.04$.

Then $V_t = \widehat{V}_t - \overline{FVA}_t$, where

$$\overline{FVA}_t = B_t^r \mathbb{E}^{\mathbb{Q}} \left[\int_t^T (B_u^r)^{-1} (r_u^f - r_u) \left(\widehat{V}_u - \overline{FVA}_u \right) \mathrm{d}u \middle| \mathcal{F}_t \right].$$

An analytic solution is available: $\overline{FVA}_0^{\text{exact}} = 0.0392$.

The deep xVA solver gives $\overline{FVA}_0 = 0.0395$.

($N = 100$, $L = 64$ and $P = 2048$ with 2 hidden layers with $d + 20 = 21$ nodes)

Conclusions

- BSDEs and DNNs suitable for portfolio risk management.
- Preliminary results encouraging.

Further extensions possible to

- collateral simulation: a by-product of the xVA solution;
- xVA hedging: note that the Delta is our primitive variable;
- further adjustments (e.g., capital): additional recursive steps;
- multiple counterparties: coupled system of BSDEs;
- more complex models, contracts, etc.

A. Gnoatto, A. Picarelli, C. Reisinger. *Deep xVA solver – A neural network based counterparty credit risk management framework*, arXiv:2005.02633.

`https://github.com/AlessandroGnoatto/Deep-xVA-Solver`