engineering proceedings

MDPI

Proceeding Paper

Deep Learning-Based Method for Computing Initial Margin [†]

Joel Pérez Villarino * and **Álvaro Leitao Rodríguez**

Research Group M2NICA, Department of Mathematics, CITIC, Universidade da Coruña, Campus de Elviña, 15071 A Coruña, Spain; alvaro.leitao@udc.es

* Correspondence: joel.perez.villarino@udc.es

[†] Presented at the 4th XoveTIC Conference, A Coruña, Spain, 7–8 October 2021.

Abstract: Following the guidelines of the Basel III agreement (2013), large financial institutions are forced to incorporate additional collateral, known as Initial Margin, in their transactions in OTC markets. Currently, the computation of such collateral is performed following the *Standard Initial Margin Model* (SIMM) methodology. Focusing on a portfolio consisting of an interest rate swap, we propose the use of Artificial Neural Networks (ANN) to approximate the Initial Margin value of the portfolio over its lifetime. The goal is to find an optimal configuration of structural hyperparameters, as well as to analyze the robustness of the network to variations in the model parameters and swap features.

Keywords: computational finance; collateral; initial margin; deep learning

Citation: Pérez Villarino, J.; Leitao Rodríguez, Á. Deep Learning-Based Method for Computing Initial Margin. *Eng. Proc.* **2021**, *7*, 41. https://doi.org/10.3390/engproc2021007041

Published: 19 October 2021

1. Introduction

Due to the financial crisis experienced in 2008, the G8 World Council promoted the regulation of stricter actions for *over-the-counter* (OTC) derivatives market, especially to reduce the counterparty credit risk. Among the mandated measures is the progressive implementation of an additional type of collateral, known as Initial Margin (IM), with the aim of acting as a "cushion" against pronounced changes in the value of the portfolio contracts.

For the IM calculation, it is standard market practice to follow the Standard Initial Margin Model (SIMM) methodology [1], promoted by International Swaps and Derivatives Association (ISDA), which only requires the sensitivities of the portfolio as input data. When the goal is to know this amount over the whole life of the portfolio, the SIMM simulation becomes challenging due to the heavy computational burden coming from nested Monte Carlo simulations and the high-dimensional nature of the problem [2].

Among the existing alternatives to brute-force simulation, there are approaches based on Deep Learning algorithms, as [2]. We aim to implement a supervised neural network for computing the IM over the considered portfolio's life, with special attention to its structure's design. In this regard, we limit our work to portfolios consisting of a single product, a vanilla interest rate swap.

2. Materials and Methods

As a Deep Learning model for the task of computing the IM, we propose to use a self-normalizing neural network (SNN) [3], adding a single unit output layer (since the IM is a scalar quantity) with a ReLu activation function and He normal kernel initialization strategy [4]. We impose that all hidden layers have the same number of units, and such hyperparameters are fixed in the later results.

A supervised training is carried out. Unlike the usual methodology, where features associated with the scenario ω and time step j tuple are considered as a single input data for training, $\boldsymbol{x}_j^{\omega}$, with the corresponding target y_j^{ω}; we propose to use the entire scenario as input data, $\boldsymbol{x}^{\omega}$, with the corresponding target vector $\boldsymbol{y}^{\omega}$. We believe that this incorporates additional information to the training, allowing the learning of intrinsic features that can improve it.

The interest rate swap portfolio's dataset is produced synthetically, on the fly, from the simulation of several interest rate scenarios under the Hull–White dynamic [5]. We establish that it is necessary to know the following quantities throughout the life of the portfolio: the swap value; the two weeks, 1 month, 3 month and 6 month cash rates; the swap par rates for the following vertices: 1 year, 2 years, 3 years, 5 years, 10 years, 15 years, 20 years, and 30 years (as input features of the model); and the IM value (as model's target), for which is necessary to know the swap sensitivities in relation to the rates mentioned above.

The methodology recommended by ISDA, termed as PV01, is chosen for the production of swap sensitivities. It consists of calculating the impact of small changes in the swap rates used to construct the zero curve.

The SIMM methodology [6], is followed for the production of IM. Based on the assumptions of working in a single currency unit and exclusively with a portfolio consisting of a swap, the following formula is obtained for the SIMM:

$$SIMM = \max\left(1, \sqrt{\frac{|\sum_k s_k|}{CT}}\right)\sqrt{\sum_k (RW_k s_k)^2 + \sum_k \sum_{l \neq k} \rho_{k,l}(RW_k s_k)(RW_l s_l)}, \quad (1)$$

where s_k, RW_k are the net sensitivity and the risk weight for the rate tenor k; $\rho_{k,l}$ is the tenor correlation and CT is the concentration threshold for the given currency. RW_k, $\rho_{k,l}$ and CT are parameters given by ISDA.

3. Results

First of all, we study the optimal choice of structural hyperparameters of our proposed neural network (depth and width). Finally, we present some experiments related to training robustness as a function of Hull–White simulation parameters and swap features (A summary of the results obtained is presented. The extended version can be found in [7]).

3.1. Numerical Experiments to Set Structural Hyperparameters

For the test in this subsection, a 1-year fixed, 6-months floating at-the-money swap with 10-year maturity is considered. We establish the theoretical values $a = 0.1$, $\sigma = 0.5\%$ for the Hull–White parameters and we choose the market forward rate, $f(0,t)$, obtained from all Eurozone governments bonds on 28 January 2021 (Source: European Central Bank (ECB)). A dataset with 5000 scenarios and 199 time steps is produced. In all tests, we use 4000 scenarios for training and 1000 for validation.

With respect to our neural network, we worked with the stochastic gradient descent optimizer and the following training hyperparameters: a bath size of 256, a learning rate of 0.001, and 1000 epochs.

3.1.1. The Depth Test

We set the total number of units to 512, which will be distributed, by means of integer division, over the following number of hidden layers: 1, 2, 3, 4, 6, 8, 10, 12, and 16. We present the results from 10 training trials due to the stochasticity of the optimization algorithm.

We can observe in Figure 1 that a moderate number of hidden layers (between 3 and 6) tend to offer a better performance than the model with two hidden layers, theoretically the one with the highest capacity. We set the number of hidden layers in our network to 4. It presents the best performance on the trials considered, with shorter execution time than its direct competitors.

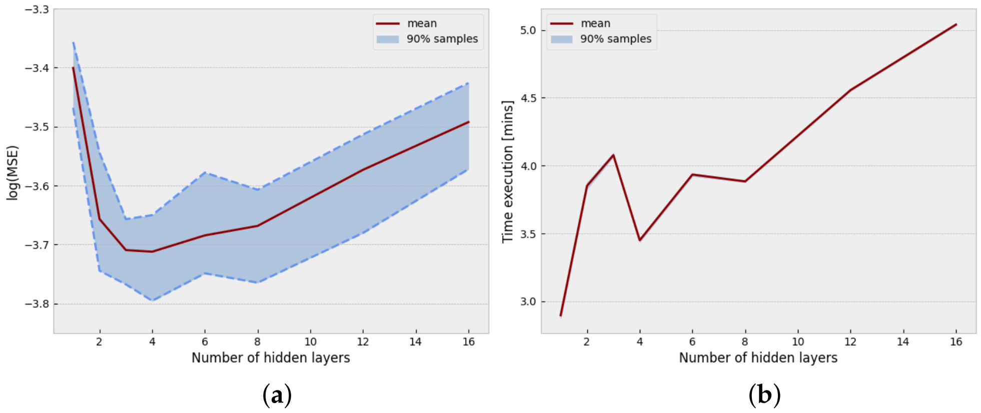

Figure 1. Results obtained for the depth test. (**a**) convergence of the MSE training set with respect to the number of hidden layers; (**b**) execution time according to the number of hidden layers.

3.1.2. The Width Test

We set the number of hidden layers to 4 and we consider the following numbers of units: 1, 2, 4, 8, 16, 24, 32, 48, 64, 96, and 128. All other specifications remain unchanged.

The test shows that, as the number of neurons per layer increases, the network performance increases, as well as the execution time required. In order to achieve a balance between network efficiency and training time, we choose to select 48 units per hidden layer.

3.2. Numerical Experiments on Network Robustness

In this subsection we used the Adam optimizer with a learning rate of 10^{-4}.

On the one hand, it has been tested how the model training responds to market situations different from the reference configuration. In general, similar results are obtained, although in situations of stressed volatility the so-called zero-inflated data problem appears. On the other hand, the influence of the swap features is analyzed. Roughly speaking, it will be necessary to have a trained model for each maturity considered, but it is feasible to use the model trained for a given frequency payments on swaps at different frequencies.

4. Conclusions and Future Research

We have found that the proposed Deep Learning model provides good approximations of IM trajectories for the simplified portfolio considered. It shows an excellent performance on our main study dataset. It is maintained for higher volatility environments. We also concluded that it is feasible to use the same model as an IM computation engine for swaps with different payment structures. However, this is not possible for different maturities. It is necessary to have a model for each case.

Future research related to this work should be focused on the scalability of the model to other interest rate products; building a model for the IM computation of other ISDA product classes, such as equity or commodity; and developing a similar neural network-based methodology to compute the IM for a real portfolio, consisting of many contracts from different product classes and driven by multiple risk factors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

References

1. ISDA. *Key Trends in the Size and Composition of OTC Derivatives Markets in the First Half of 2020*; ISDA: New York, NY, USA, 2020.
2. Ma, X.; Spinner, S.; Venditti, A.; Li, Z.; Tang, S. Initial Margin Simulation with Deep Learning. *SSRN Electron. J.* **2019**. [CrossRef]

3. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 971–980.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV '15), Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
5. Hull, J.; White, A. Numerical Procedures for Implementing Term Structure Models I. *J. Deriv.* **1994**, *2*, 7–16. [CrossRef]
6. ISDA. *ISDA SIMM Methodology*, version 2.0; ISDA: New York, NY, USA, 2017.
7. Villarino, J.P. Deep Learning-Based Method for Computing Initial Margin. Master's Thesis, University of A Coruña, A Coruña, Spain, 2021.