Monte Carlo Learning Data Sets and Neural Networks with Applications in Quantitative Finance 1.0

Lokman A. ABBAS TURKI *

February 28, 2024

1 Introduction

Since the advent of von Neumann's architecture, simulation has been widely articulated around the use of libraries. As a result, large companies usually find themselves with software, inherited from decades of development, suited to old/outdated cluster solutions. Changing the existing software solutions is hard because of the many evolving hardware options. The hardware uncertainty remains real even when using cloud computing since cloud providers themselves suggest using their chips, essentially for inference like TPUs by Google or Inferentia by AWS. AI allows for resolving hardware uncertainty by replacing an entire simulation code or parts of it with inference systems. The maintenance cost of these inference systems is reduced when compared to libraries that produce the same data.

This work is an attempt to frame a new Monte Carlo (MC) and neural network (NN) combination implemented on high performance architectures, especially graphics processing units. The resulting tool could be used to analyse different training methods using model-based data simulated by Monte Carlo. Our purpose is to store the trained NNs and make them reusable and scalable on many applications. The first step for cooking these NNs is to design MC data sets on which NNs are trained and compared. The purpose of establishing these data sets is to open access to generated models for data scientists who want to prove the efficiency of their methods without the need to master every aspect of the generated data. With these data sets, we would like to establish benchmarks susceptible to evolve with external contributions. These data sets are organized into three categories.

2 NN generators of standard distributions

The first category is dedicated to the simulation by NNs of common probability distributions involving an important (undefined) number of uniformly distributed random variables. We target in particular distributions simulated by acceptance-rejection methods including Poisson distribution, compound Poisson, gamma, and non-central chi-squared distribution Pagès (2018). Since these distributions are standard and can be simulated using many libraries, the data sets are less important than the NNs that generate these distributions. The data sets are essentially needed to compare the different NNs.

Note that NNs seem to have difficulties in learning unbounded distributions, hence in practice we train to labels in a certain compact range $[c, C]$ of the targeted distribution and the predicted values are all within this range.

*contributed in all data sets, contributors in sub-parts will be cited in their respective parts. This work benefited from the support of the Chair *Capital Markets Tomorrow: Modeling and Computational Issues* under the aegis of the Institut Europlace de Finance, a joint initiative of Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Université Paris Cité and Crédit Agricole CIB.

2.1 Poisson distribution

The purpose is to generate Poisson distribution $\mathcal{P}(\lambda)$ with $\lambda \in [1, 100]$ using only a few (≤ 4) uniform random variables. The expression

$$N_\lambda = \inf\{k \geq 0; U^1 ... U^{k+1} < e^{-\lambda}\}$$

is usually used for the simulation of Poisson distributed random variable N_λ with uniform ones $U^1, ..., U^{k+1}$. Neural network generation has a computational benefit when compared to MC acceptance-rejection methods when λ is large (since the number of uniforms required for the acceptance-rejection scheme is then higher), say $\lambda > 10$, with $[c, C] = [0, q^{100}_{0.999}]$, where $q^{100}_{0.999}$ is the corresponding 99.9% quantile associated to $\lambda = 100$.

2.2 Non-central chi-squared distribution

The purpose is to generate non-central chi-squared distribution Chi2(d, λ) with $d \in [0.3, 10]$ and $\lambda \in [0, 100]$ using only a few (≤ 5) uniform random variables. This has a computational benefit when compared to MC acceptance-rejection methods when d is not an integer (≥ 1) or when λ is large, say $\lambda > 10$, with $[c, C] = [0, q^{10,100}_{0.999}]$, where $q^{10,100}_{0.999}$ is the corresponding 99.9% quantile associated to $d = 10$ and $\lambda = 100$.

2.3 Compound Poisson Distribution

The purpose is to generate compound Poisson distribution involving sums on Uniform distribution $\mathcal{U}(0, 1)$ and Gaussian distribution $\mathcal{N}(0, 1)$. The time arrival of jumps has the Poisson distribution $\mathcal{P}(\lambda)$ with $\lambda \in [0.1, 100]$ using only a few (≤ 4) uniform random variables. This has a computational benefit with respect to MC acceptance-rejection methods when λ is large, say $\lambda > 10$, with $[c, C] = [0, q^{100}_{0.999}]$, where $q^{100}_{0.999}$ is the corresponding 99.9% quantile associated to $\lambda = 100$.

3 Data sets for training NN pricers

The second category gathers the data sets generated by MC simulation for a large number of financial model and product parameters (in a variety of models). Such data sets allow training an NN capable of generating the same values than the ones produced by the MC and to parametrize (as a part of the input vector) the NN with the model parameters. One can then replace heavy nested MC simulation, e.g. 10^5 price trajectories for each out of 10^9 sets of model and product parameters (where the latter includes a pricing time), with NN inference, e.g. the ability to price analytically for any model and product parameters. Each one of these MC simulations is considered to be "sufficiently" accurate (number of trajectories $\sim 1e5$). For a specific model, we obtain the data sets from a very large randomization ($\sim 1e9$ configurations) on model parameters.

As products, we start by considering vanilla options of various characteristics (strike and maturities) in the models of Sections 3.1-3.3, with fast calibration in view. Then in Section 3.4 we switch to fast pricing of exotic products.

3.1 Asset

3.1.1 B&S vol expOU (vol exponentielle d'Ornstein-Uhlenbeck)

3.1.2 Heston

3.1.3 Vol locale paramétrique (à définir)

3.1.4 vol sto paramétrique

3.2 Curve

3.2.1 HW

3.2.2 CIR

3.2.3 HW 2F

3.2.4 HW 2F vol sto

3.3 Multi-asset

3.3.1 B&S et matrice de correl

3.3.2 +Vol sto

3.4 Exotic products

3.4.1 Autocalls in B&S

4 Data sets for testing NN pricers sets

The third category gathers extensive data sets generated by MC simulation for a few model and (a bit more) product parameters, e.g. 10^7 prices for each out of $10^2 \times 10^5$ sets of model and product parameters (in a selection of the models considered at the previous point). Such data sets are primarily intended for testing the NN pricers trained in the previous part. But they can also be useful for instance for calibration experiments, to the corresponding synthetic data (playing with the parameters of a given model class, e.g. Heston, in order to retrieve the time 0 prices of selected products in an instance of the same or another model class).

4.1 Exponential Ornstein-Uhlenbeck

4.2 Heston Model

References

Pagès, G. (2018). Numerical probability. *Universitext, Springer*.