Quant GANs: Deep Generation of Financial Time Series

Machine Learning for Finance

Laure Michaud - Tim Houdayé

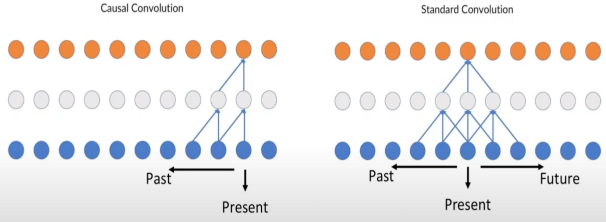
September 18, 2022

ENSAE
IP PARIS

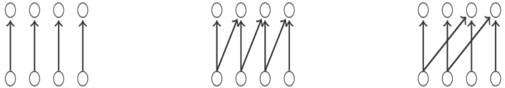
Sommaire

Dilated Causal Convolutional layer

- *Causal Convolution* are convolutions where the output only depends on the past and the present values not the future values.



- In order to construct *Temporal Convolutional Networks*, we will use the dilated ones where K denoted the kernel size and D the dilatation factors.



(a) $K = D = 1$ (b) $K = 2, D = 1$ (c) $K = D = 2$

Advantage of the architecture

- Possibility to look back without increasing the number of weights. Indeed, as we want to deal with long-range dependencies, increasing the filter could be a solution to capture more information but it will require more weights. This is why a dilatation factor is introduced.

- One of the drawbacks of this structure is that we cannot see values between x_{t-D} and x_t. To overcome this issue, the authors stack convolutions with different dilatation factors which will lead to the *Vanilla TCNs*. As for the *MLP*, *Vanilla TCNs* are obtained by composition of dilated causal convolution layers with activation functions.

- The article gives for the first time a rigorous mathematical definition of the *TCNs* which constitutes an important step in deep learning.

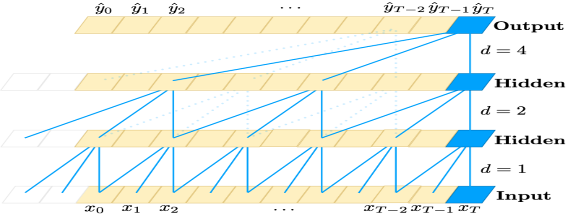
A rigorous definition of the TCNs for the first time

Temporal Convolutional Networks

A temporal convolutional network with L hidden layers is a function $f : \mathbb{R}^{N_O \times T_O} \to \mathbb{R}^{N_{L+1} \times T_L}$ such that:

$$f(X, \theta) = w \circ \psi_L \circ ... \circ \psi_1(X)$$

where : $\psi_l : \mathbb{R}^{N_{l-1} \times T_{l-1}} \to \mathbb{R}^{N_l \times T_l}$ is a block module (lipschitz function), $w : \mathbb{R}^{N_L \times T_L} \to \mathbb{R}^{N_{L+1} \times T_L}$ a 1×1 convolutional layer and $T_0, L, N_0, ..., N_{L+1}, S_l \in \mathbb{N}$ such that $T_l := T_{l-1} - S_l$ leading to $T_L - T_0 = \sum_{l=1}^{L} S_l$.



Advantages/Disadvantages of TCNs

Advantages

- **Parallelism**
- **Flexible RFS**: a *TCN* can change its RFS in multiple ways. For example, stacking more dilated causal convolutional layers, using larger dilatation factors, or increasing the filter size. Easy to adapt to different domains.

Disadvantages

- **Data storage during the evaluation**
- **Potential parameter change for a transfer of domain**

The Generative Adversarial Networks

While using *GANs*, $(\mathbb{R}^{N_Z}, \mathcal{B}(\mathbb{R}^{N_Z}))$ and $(\mathbb{R}^{N_X}, \mathcal{B}(\mathbb{R}^{N_X}))$ are called the **latent** and the **data measure space**, respectively. The variable Z represented the **noise prior** and X the **target random variable**.

Generator

Let g : $\mathbb{R}^{N_Z} \times \Theta^{(g)} \to \mathbb{R}^{N_X}$ be a network with a parameter space $\Theta^{(g)}$. The random variable X, defined as follows is called the **generated random variable**. The network g is called the **generator**, and X_θ the **generated random variable with parameter** θ.

$$X : \Omega \times \Theta^{(g)} \to \mathbb{R}^{N_X}$$
$$(\omega, \theta) \mapsto g_\theta(Z(\omega))$$

Discriminator

Let d : $\mathbb{R}^{N_X} \times \Theta^{(d)} \to \mathbb{R}$ be a network with parameters $\eta \in \Theta^{(d)}$ and $\sigma : \mathbb{R} \mapsto$ [0,1] : $x \mapsto \frac{1}{1+e^{-x}}$ be sigmoid function. A function d : $\mathbb{R}^{N_X} \times \Theta^{(d)} \mapsto$ [0,1] defined by d : $(x, \eta) \mapsto \sigma \circ d_\eta(x)$ is called a **discriminator**.

The GAN objective

The goal of the generator g is to generate a sample (X) such that the discriminator is unable to distinguish whether the sample were generated or sampled from the target. Therefore, we obtain the following *GAN objective* :

$$\min_{\theta \in \Theta^{(g)}} \max_{\eta \in \Theta^{(d)}} \mathcal{L}(\theta, \eta)$$

where :

$$\mathcal{L}(\theta, \eta) := \mathbb{E}[log(d_\eta(X))] + \mathbb{E}[log(1 - d_\eta(g_\theta(Z)))] = \mathbb{E}[log(d_\eta(X))] + \mathbb{E}[log(1 - d_\eta(\tilde{X}_\theta))].$$

- Pros: efficient replication of a random variable distribution - No need of an explicit density function - flexibility.
- Cons: hard to know when stopping the training of the generator and discriminator - no explicit representation of the generator's density - hard to invert generated model to get back to the latent variables.

Numerical Application

Replication of the log return of the S&P 500 from 05/01/2009 to 31/12/2018 using :

- **Models**: Pure TCN, C-SVNN and GARCH(1,1)
- **Evaluation metrics**: Earth mover distance, DY metrics, ACF score, Leveraged effect score
- **Objective**: Capture all the stylized effects of logarithmic returns and try to generate "fake" returns whose characteristics are closest to the real ones.

The models

Pure TCN

They modeled the log returns directly using a TCN with a RFS $T^{(g)} = 127$ as the generator function. The return process is given by :

$$R_{t,\theta} = g_\theta(Z_{t-(T^{(g)}-1):t})$$

for the three-dimensional noise prior $Z_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, \mathbb{I})$, ($N_Z$=3), and a RFS $T^{(g)} = 127$ as the generator function.

Constrained log return NP C-SVNN with drift

$$R_{t,\theta} = \sigma_{t,\theta}\epsilon_{t,\theta} + \mu_{t,\theta}$$

with volatility NP $\sigma_{t,\theta}$, drift NP $\mu_{t,\theta}$, and innovation NP $\epsilon_{t,\theta}$ constrained to being i.i.d. $\mathcal{N}$(0,1)-distributed. The latent process is still $Z_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, \mathbb{I})$ for $N_Z = 3$. The innovation process takes the form $\epsilon_{t,\theta} = Z_{t,1}$ for any $t \in \mathbb{Z}$.

The models

GARCH(1,1) with constant drift

This model is well known to model the returns of financial time series. The model is defined as follows :

$$
\begin{aligned}
R_{t,\theta} &= \xi_t + \mu \\
\xi_t &= \sigma_t \epsilon_t \\
\sigma_t^2 &= \omega + \alpha \xi_{t-1}^2 + \beta \omega_{t-1}^2 \\
\epsilon_t &\overset{\text{i.i.d}}{\sim} \mathcal{N}(0, 1)
\end{aligned}
$$

for $\mu \in \mathbb{R}$, $\omega > 0$, and $\alpha, \beta \in$[0,1] such that $\alpha + \beta < 1$, and the parameters vector $\theta = (\omega, \alpha, \beta, \mu)$.

Results

The authors gave the following results regarding their numerical application.

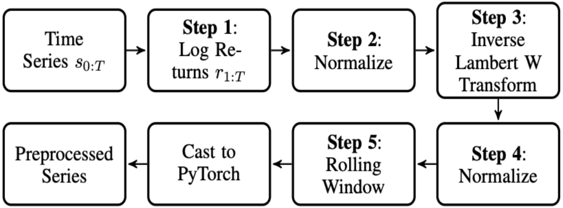
- The C-SVNN model gives results that are above the 2 others regarding the evaluation metrics we give.
- The pure TCN model is slightly comparable to the C-SVNN as their results are close
- The GARCH(1,1) model is the worst one. The model achieves to well capture the autocorrelation but fails to capture the exactness and the leverage effect.

Our implementation

Packages used

- Classical packages : numpy, pandas, random, matplotlib
- Collect of data from Yahoo Finance : yfinance
- Statistical treatments : scipy, statsmodel
- NN packages : torch

We follow the preprocessing steps mentioned in the article and the architecture of NN described in the Annex B.



Our results : Lambert transforms

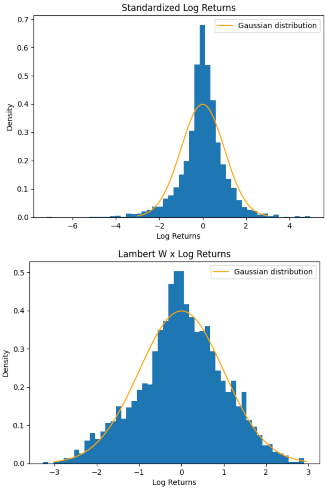
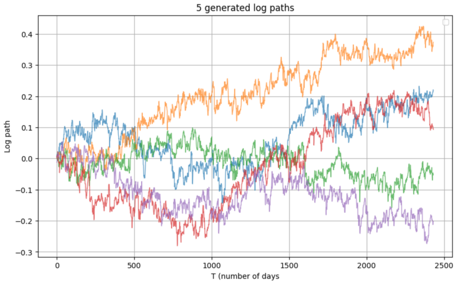


Figure: Log returns and gaussian density function

Our results : generated log path



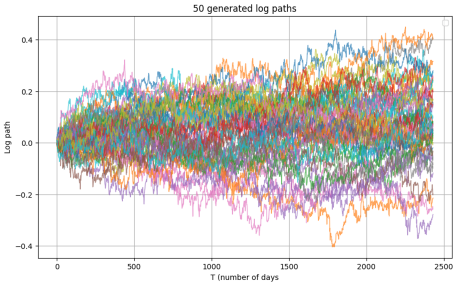


Figure: Generated logpaths (left : 5 paths - right : 50 paths)

Our results : densities of S&P500

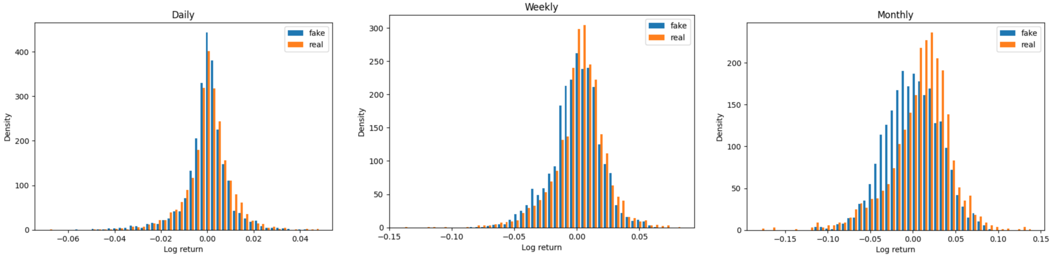


Figure: Comparison of generated and historical densities of the S&P500

Conclusion

- *GAN* variation which utilizes temporal convolutional networks (*TCNs*) aiming at capturing long-range dependencies like volatility clusters.
- One of the first solid work applying *GANs* to financial time series generations.
- The objective was to approximate a realistic asset price simulator by using a neural network, data-driven concept.
- Competitive results, which can be used to approximate financial time series.
- Lack of a unified metric to quantify the goodness of fit of two datasets, but overall the findings were a solid step in developing data-driven models in finance.

References

BAI, S., ZICO, J., KORN, R., AND KOLTUN, V.
An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.
arXiv:1803.01271 (2018).

GEORG, M.
Lambert W random variables - a new family of generalized skewed distributions with applications to risk estimation.
The Annals of Applied Statistics (2010).

GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y.
Generative adversarial nets.
http://papers.nips.cc/paper/ 5423-generative-adversarial-nets.pdf (2014).

WIESE, M., KNOBLOCH, R., KORN, R., AND KRETSCHLMER, P.
Quant GANs: Deep Generation of Financial Time Series.