Differential Machine Learning

Jarles Andrés Marimon Hernández

Grupo de estudio: Automatic Adjoint Diferentiation
Facultad de Economía
Universidad del Rosario

2022

Training with derivatives I

- The performance of modern deep learning remains insuffycient for online application with complex transactions or trading books
- A vast number of training examples (often in the hundreds of thousands or millions) is necessary to learn accurate approximations, and even a training set of sample payoff cannot be simulated in reasonable time
- Training on noisy payoffs is prone to overffitting, and unrealistic dataset sizes are necessary even in the presence of classic regularization
- In addition, risk sensitivities converge considerably slower than values and often remain too approximate even with training sets in the hundreds of thousands of examples

Training with derivatives II

- The article proposes to resolve these problems by training ML models on datasets augmented with differentials of labels wrt inputs:

$$x^{(i)}, y^{(i)}, \frac{\partial y^{(i)}}{\partial x^{(i)}}$$

  - In Black-Scholes $x^{(i)}$ would be one possible set of values for the initial spot price, volatility, strike and expiry
  - $y^{(i)}$ would be the call price computed with these inputs (by MC or FDM since we don't know the formula)
  - $\frac{\partial y^{(i)}}{\partial x^{(i)}}$ would be the Greeks achieved with automatic adjoint differentiation (AAD).

Differential Machine Learning I

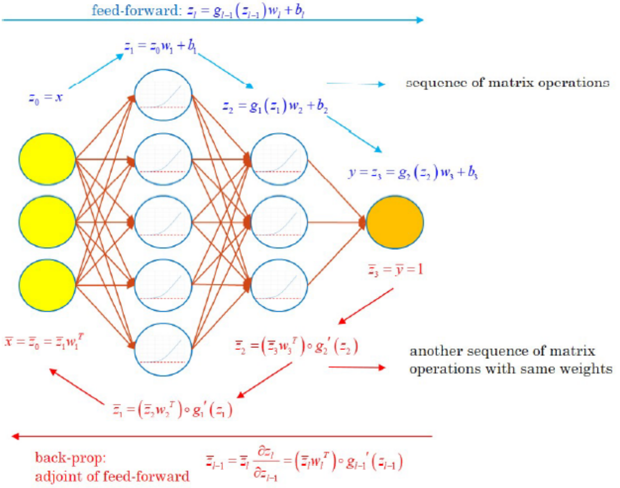

Figura 1: Feedforward neural network with backpropagation

Differential Machine Learning II

- Feedforward equations:

$$
\begin{aligned}
z_0 =& x \\
z_l =& g_{l-1}(z_{l-1}) w_l + b_l, \quad l = 1, \ldots, L \\
y =& z_L
\end{aligned}
$$

- Backpropagation equations:

$$
\begin{aligned}
\bar{z}_0 =& \bar{y} = 1 \\
\bar{z}_{l-1} =& (\bar{z}_l w_l^\top) \odot g'_{l-1}(z_{l-1}), \quad l = L, \ldots, 1 \\
\bar{x} =& \bar{z}_0
\end{aligned}
$$

with the adjoint notation $\bar{x} = \partial y / \partial x, \bar{z}_l = \partial y / \partial z_l, \bar{y} = \partial y / \partial y = 1$ and $\odot$ is the elementwise (Hadamard) product

Differential Machine Learning III

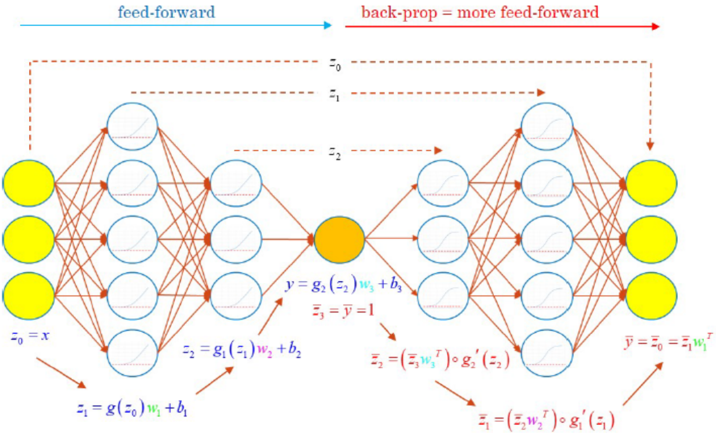

Figura 2: Twin network

Numerical result: Basket options I

Consider a basket option in a correlated Bachelier model for seven assets

$$dS_t = \sigma dW_t$$

where $S_t \in \mathbb{R}^7$ and $dW_t^j dW_t^k = \rho_{jk}$.

- The task is to learn the pricing function of a 1y call option on a basket with strike 110

Numerical result: Basket options II

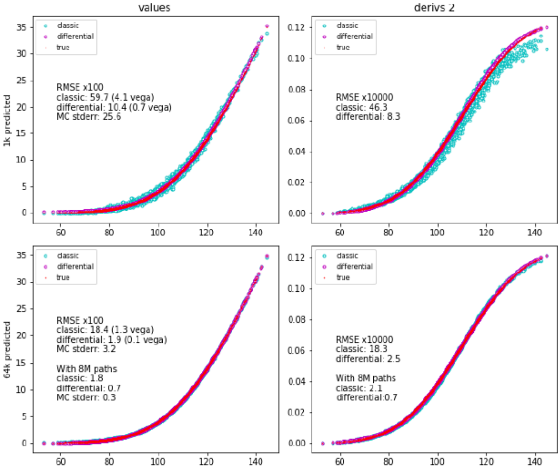

Figura 3: Basket option in Bachelier mode, dimension 7

Numerical result: Basket options III

- A differential training set takes 2-5 times longer to simulate with AAD, and it takes twice longer to train twin nets than standard ones.
- In return, we are going to see that differential ML performs up to thousandfold better on small datasets.

Codes + Slides AAD

https://github.com/differential-machine-learning
https://antoinesavine.files.wordpress.com/2021/08/readableslides-aad15min.pdf

Bibliografía I

Huge, B. and Savine, A. (2020). Differential machine learning. *arXiv preprint arXiv:2005.02347*.