Solving coupled stochastic differential equations to determine the volatility distribution of financial returns

Humberto Ribeiro de Souza, Leonardo Paulo Maia

Instituto de Fisica de Sao Carlos - USP

Objectives

Forecast stock prices with neural networks using only simulated data in the training process

Introduction

To find the financial return of a certain asset, we could use the assumption that its volatility can be interpreted as a stochastic process. One of the most known contributions on the area, the Heston Model, determines that the return/volatility SDE's can be represented by the equation:

$$dX = \mu dt + \sigma dW_1 \quad (1)$$
$$dY = \alpha(\theta - Y)dt + \kappa\sqrt{Y}dW_2$$

Where X is the stochastic variable related to the financial return, Y is somehow related to the volatility ($Y = \sigma^2$) of the asset and dW_1 and dW_2 are Wiener processes with non-zero correlation between them.

Stochastic simulation

The first goal in this project is the creation of an artificially generated dataset. We used a package that already does this process: *sdeint*, with arbitrary values for the constants of in the equation

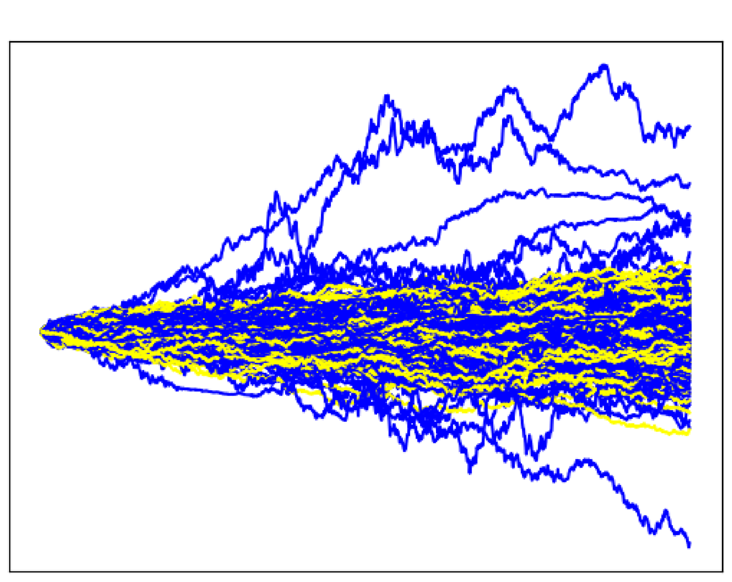
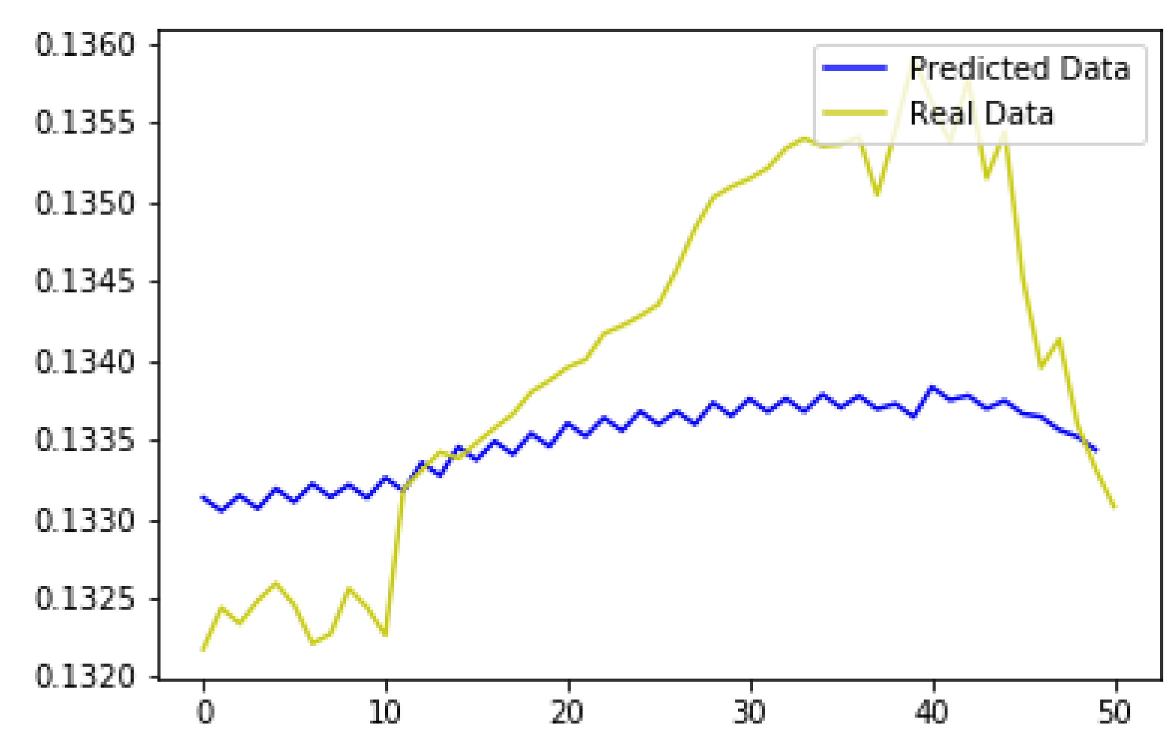


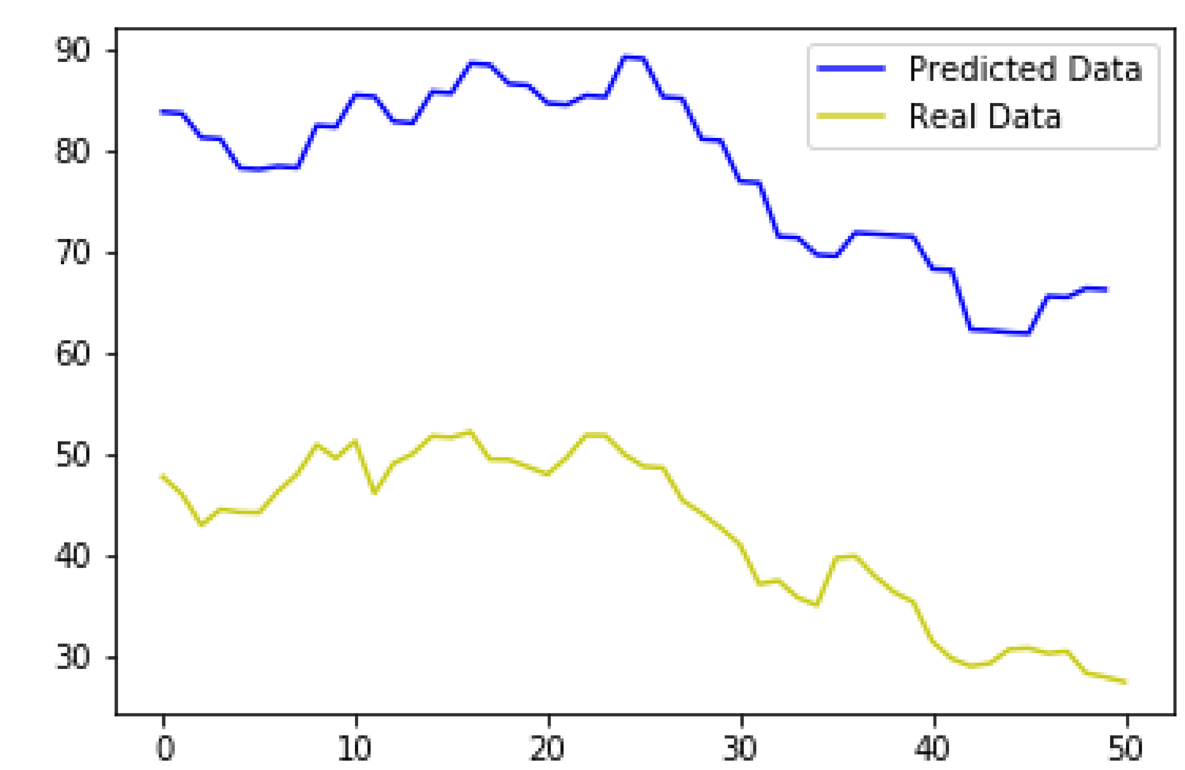
Figura: Random stochastic processes for a gaussian distribution (yellow) and for the Heston Model (blue)

Results

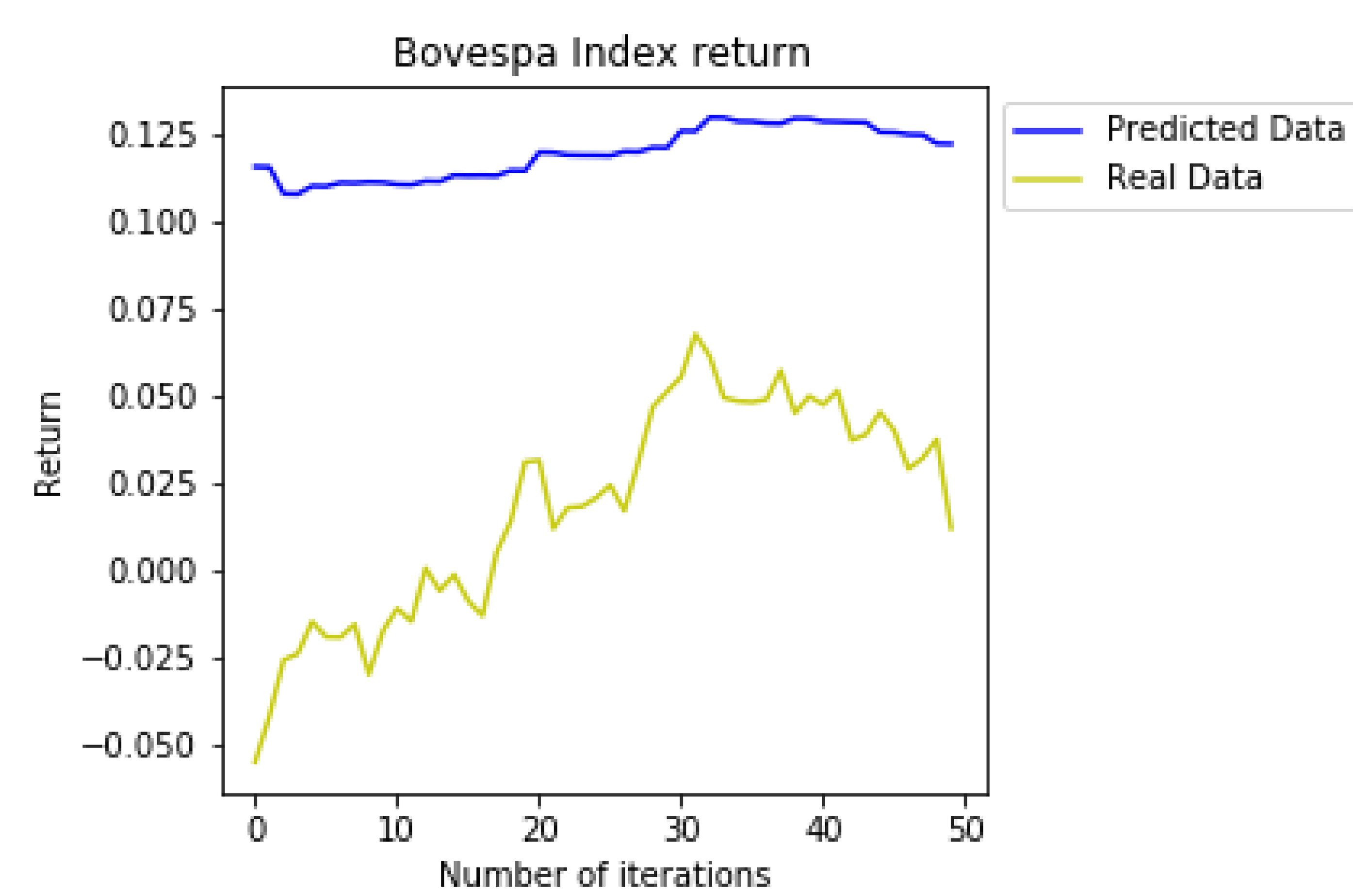
Those are the results of a the best model developed using only a simple neural network, with fully connected layers. To test the efficiency of the model, we also used it in the Bovespa Index:



(a) "Best"result in the simulated market



(b) Best result in the simulated market



(c) Bovespa Index return prediction

Figura: Prediction of returns on: simulated data (2a - 2b) and in the Bovespa Index (2c)

Conclusion

- This model seems to find smooth changes in the return, but it's ouput is out of scale;
- Using this network in a real market seems to give the same results of a simulated market

Next steps

- Use more complex architectures in the next model such as the Transformer Network;
- Use this model as a transfer learning model: use real data to re-train our network;
- Check if we get the same results using other simulated processes.

SIFSC