

Estudo de Redes Neurais Generativas aplicadas ao Mercado Acionário Brasileiro

Aluno: Caio Emmanuel Soares Rocha

Orientador: Raul Ikeda Gomes da Silva

Palavras-chave: Generative Adversarial Network, Deep Learning, Long Short-Term Memory, Rede Neural Convolutacional, Mercado Financeiro

1 Descrição do Problema e Revisão Bibliográfica

Estimar o valor do preço de uma ação na bolsa de valores é um grande e antigo desafio no mundo das finanças[1]. Tal possibilidade permitiria maior garantia de retorno e cobertura contra risco de mercado. O que levou diversos pesquisadores e instituições a investir seus recursos no desenvolvimento de técnicas para isto.

Ainda assim, estimar o preço de uma ação tem se mostrado difícil devido à complexidade e a dinâmica caótica do mercado e variáveis estocásticas não-estacionárias envolvidas[2].

Várias soluções foram desenvolvidas para lidar com este problema, no campo da econometria, alguns dos modelos mais populares são *autoregressive method (AR)*, *moving average (MA)* e *autoregressive integrated moving average (ARIMA)*[3, 4] que, explicando brevemente, inferem o preço em um momento t utilizando uma combinação linear dos preços nos instantes $\{1, 2, \dots, t - 1\}$. Entretanto, um problema desta classe de modelos é que eles partem de premissas sobre a diferença entre o preço em dois instantes (como distribuição- t ou variáveis independentes e identicamente distribuídas), que não se confirmam com dados reais na maioria das vezes.

Já no campo de *soft computing* (área que envolve inteligência artificial), algumas das principais soluções incluem a utilização de *Redes Neurais Artificiais*[5, 6] e *Support Vector Machine*[7, 8] e outras da área de *Deep Learning*[9], que têm atraído olhares devido à capacidade dessa estrutura de extrair *features* abstratas dos dados (inclusive de dados não usuais, como *tweets* ou *headlines* de jornais[10, 11]). Entretanto, essa classe de modelos têm uma série de problemas que dificultam a implementação como custo de treinamento das redes neurais, expertise necessária para lidar com as especificidades e restrições dos diferentes tipos de dados que podem ser passados por estas.

Nesse contexto surge o modelo de GAN-FD[12], abreviação para *Generative Adversarial Network - Forecast and Direction*, um novo modelo que combina as qualidades dos modelos de *Deep Learning*

mas que usa apenas dados simples na entrada que podem ser extraídos de um *software* de *trading*, o que reduz a barreira de acesso para investidores e instituições menores.

A GAN-FD tem sua estrutura baseada nas *Generative Adversarial Network (GAN)*[13, 14], uma estrutura de rede neural desenvolvida para recriar imagens a partir de uma matriz ruidosa de dados fornecida. A estrutura principal de uma *GAN* consiste de duas redes neurais adversárias que competem entre si, a primeira é a rede **Geradora (G)** e a rede **Discriminante (D)**, que podem ser vistas na Figura 1. Enquanto G tem a função de gerar amostras de forma a enganar D, a rede D tem o papel de distinguir corretamente amostras reais de amostras geradas por G.

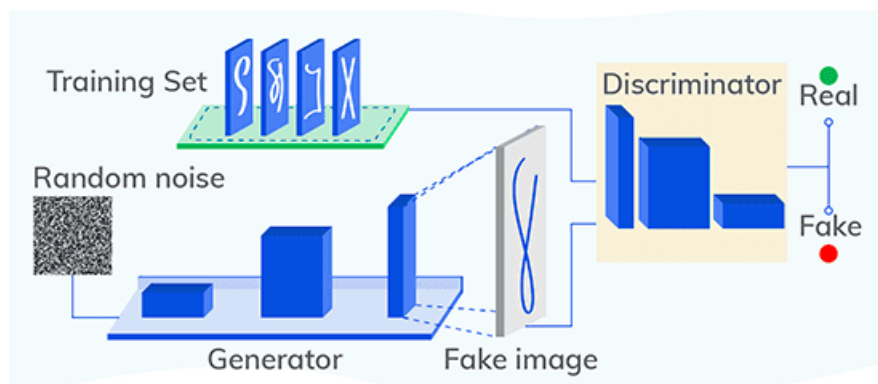


Figura 1: Arquitetura de uma GAN clássica

Para implementar uma GAN-FD precisa-se adaptar as redes neurais escolhidas para compor G e D, de modo que:

1. Consiga treinar com o tipo de dado escolhido para realizar a estimativa;
2. Diminua o erro na estimativa do preço (*forecast*) e na direção da previsão (*direction*).

A solução encontrada por *Xingyu Zhou et al.*[12] é substituir as redes neurais que são tradicionalmente usadas como geradoras por uma *LSTM*, abreviatura de *Long Short-Term Memory*[15, 16], uma arquitetura baseada em *Redes Neurais Recorrentiais (RNN)*, uma classe de redes neurais desenvolvida para tratar com sequências de dados. Ao invés das redes neurais mais tradicionais (ANN) que lidam com cada entrada da sequência de dados separadamente, as RNN iteram seu *estado*[17] a cada nova entrada da sequência e esse estado é reiniciado entre o processamento de uma sequência e a próxima, em suma, é semelhante a uma rede neural com um loop, onde o output é usado como input em uma próxima iteração. Fazendo das RNNs ideais para processamento de séries temporais (como a série de preços ou indicadores de uma ação). A LSTM foi criado com o intuito de solucionar um problema que as primeiras RNNs

possuíam: preservar informação passada entre unidades futuras da rede[17] e este é o principal motivo de usá-las.

A rede discriminante continua a ser uma *Rede Neural Convolutacional (CNN)* como na arquitetura original de uma *GAN*, que, nesse caso, recebe ambas a série de preços reais e a série de preços com o último elemento sendo o valor predito e através de operações de convolução retorna se a série falsa foi capaz de enganá-la ou não[18].

A estrutura completa da GAN-FD pode ser vista na Figura 2, onde X é a série de entrada, \hat{Y} é a série de preços com o último valor sendo o preço estimado e Y é a série de preços real.

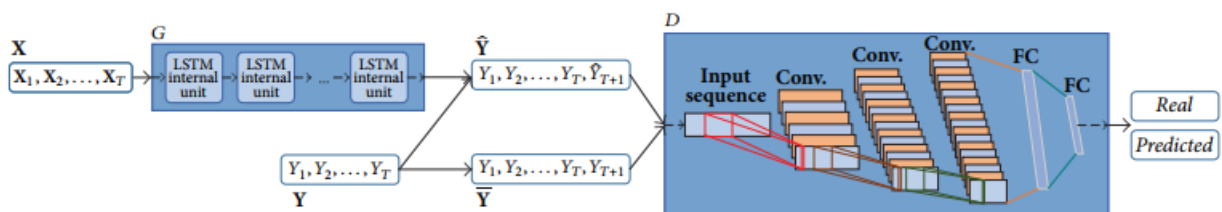


Figura 2: Arquitetura de uma GAN-FD

2 Objetivo

Este projeto tem como objetivo apresentar uma análise sobre o modelo GAN-FD aplicado ao mercado de ações com dados intradiários, restrito a ações que componham o Índice Bovespa (Ibovespa).

A análise visa comparar esta nova abordagem com modelo econométrico clássico (ARIMA) e outros modelos de *soft computing* (ANN, SVM, etc.) além de outras versões deste modelo como sutis modificações. A comparação levará em conta o erro no valor da estimativa e na direção do movimento do preço do ativo.

3 Proposta de Metodologia

Com o objetivo de estudar a implementação desse modelo no cenário nacional, será construída uma GAN-FD para ser treinada com dados do mercado acionário brasileiro. Os dados serão coletados a partir de bases históricas da Bloomberg, as ações escolhidas serão aquelas que constituem o Índice Bovespa.

A intervalo entre dois registros será de um minuto e serão coletados registros de um ano para todas as ações e indicadores.

Durante o treinamento, para cada ação, os dias serão segmentados em pares de (M, N) minutos, isso

significa M minutos de observação para estimar os próximos N minutos, caminhando também N minutos para a frente, como pode-se observar no esquema da Figura 3.

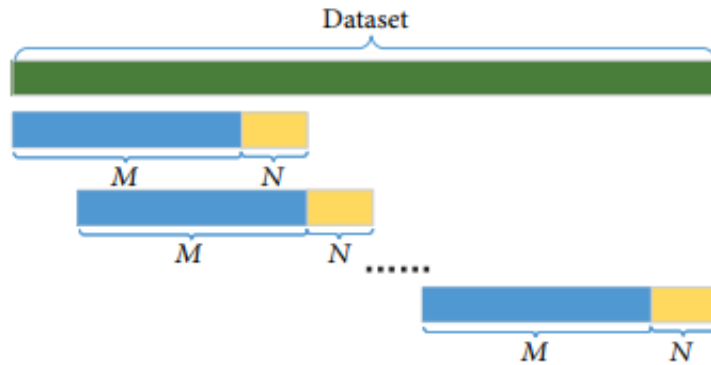


Figura 3: Esquema de segmentação de um dia da base

É importante dizer que o processo reinicia no início de todos os dias da base, isso para evitar flutuações anômalas no preço que podem ocorrer durante o fechamento e abertura do mercado.

Os pares de minuto (M,N) serão formados escolhendo M no conjunto 10,20,60 e N no conjunto 5,10,20, obtendo assim nove pares (M,N) que serão todos testados e comparados entre si.

Para treinar a rede G , será usado uma combinação de diferentes funções de *loss*[19]. Para explicar isto é importante definir X_t como sendo o conjunto de indicadores no instante t , Y_t é o preço de fechamento no mesmo instante.

A primeira função de *loss* é a *adversarial loss* de G ($L_{adv}^G(\hat{Y})$), definida pela equação (1).

$$L_{adv}^G(\hat{Y}) = L_{sce}(D(\hat{Y}), 1) \tag{1}$$

Onde \hat{Y} é o preço de fechamento estimado pela rede G ($\hat{Y}_{T+1} = G(X)$), onde \mathbf{X} é o *set* de todos os X_t ($t = 1, 2, \dots, T$) e $L_{sce}(D(\hat{Y}), 1)$ é dado pela equação (2).

$$L_{sce}(D(\hat{Y}), 1) = - \sum_i \log(\sigma(D(\hat{Y}_i))) \tag{2}$$

Entretanto, para assegurar que a rede G não somente confundiu D como estimou o preço com uma boa diferença, adiciona-se o *forecast error loss* (L_p) dado pela equação (3).

$$L_p(\bar{Y}, \hat{Y}) = \|\bar{Y} - \hat{Y}\|_p \tag{3}$$

E por último, a direção da predição é importante elemento para realizar o trading, portanto, acrescenta-

se a *direction prediction loss* (L_{dpl}) calculado conforme (4).

$$L_{dpl} = |\text{sgn}(\hat{Y}_{T+1} - Y_T) - \text{sgn}(Y_{T+1} - Y_T)| \quad (4)$$

Onde sgn é a função sinal que retorna -1 caso o valor seja negativo, 0 caso seja zero e 1 caso seja positivo.

Criando os parâmetros da combinação linear λ_{adv} , λ_p e λ_{dpl} , a função *loss* de G será aquela dada por (5).

$$L_G(X, Y) = \lambda_{adv} L_{adv}^G(\hat{Y}) + \lambda_p L_p(\bar{Y}, \hat{Y}) + \lambda_{dpl} L_{dpl} \quad (5)$$

Já a função *loss* para treinar D será apenas a *adversarial loss* de D dado por (6).

$$L_D(X, Y) = L_{adv}^D(\bar{Y}, \hat{Y}) = L_{sce}(D(\hat{Y}), 0) + L_{sce}(D(\hat{Y}), 1) \quad (6)$$

O algoritmo de treino usado será o *Stochastic Gradient Descent* (SGD)[20].

Já para validação e comparação, as métricas usadas serão a *Raiz do Erro Médio Quadrático Relativo*, ou *RMSRE* (*Root Mean Squared Relative Error*) definida pela equação (7).

$$RMSRE = \sqrt{\frac{1}{T_0} \sum_t \left(\frac{\hat{Y}_{t+1} - Y_{t+1}}{Y_{t+1}} \right)^2} \quad (7)$$

Utilizado para medir o erro relativo à distância entre a estimativa e o valor real.

E também a *Acurácia na Direção da Previsão*, *DPA* (*Direction Prediction Accuracy*) definido por (8).

$$DPA = \frac{100}{T_0} \sum_t I_t \quad (8)$$

Onde:

$$I_t = \begin{cases} 1, & (\hat{Y}_{T+1} - Y_T)(Y_{T+1} - Y_T) > 0 \\ 0 & \end{cases} \quad (9)$$

4 Resultados Esperados

Após realizado o *backtesting* serão produzidos os resultados para as métricas estabelecidas na metodologia para o modelo proposto (todas suas variantes) e os usados no benchmark. Além de validar a aplicabilidade desse modelo no mercado brasileiro.

Referências

- [1] Rami Al-Hmouz, Witold Pedrycz e Abdullah Balamash. “Description and prediction of time series: A general framework of Granular Computing”. Em: *Expert Systems with Applications* 42.10 (2015), pp. 4830–4839. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2015.01.060>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417415000834>.
- [2] A. Marszałek e T. Burczyński. “Modeling and forecasting financial time series with ordered fuzzy candlesticks”. Em: *Information Sciences* 273 (2014), pp. 144–155. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2014.03.026>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025514003107>.
- [3] J. D. Hamilton. *Time Series Analysis*. Vol. 2. Princeton University Press, 1994.
- [4] Jeffrey Marc Wooldridge. *Introductory Econometrics: A Modern Approach*. ISE - International Student Edition. South-Western, 2009. ISBN: 9780324581621.
- [5] Yakup Kara, Melek Acar Boyacioglu e Ömer Kaan Baykan. “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange”. Em: *Expert Systems with Applications* 38.5 (2011), pp. 5311–5319. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2010.10.027>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417410011711>.
- [6] A. Prastyo, D. Junaedi e M. D. Sulistiyo. “Stock price forecasting using artificial neural network: (Case Study: PT. Telkom Indonesia)”. Em: *2017 5th International Conference on Information and Communication Technology (ICoICT)*. 2017, pp. 1–6. DOI: 10.1109/ICoICT.2017.8074673.
- [7] Alaa Sheta, Sara Ahmed e Hossam Faris. “A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index”. Em: *International Journal of Advanced Research in Artificial Intelligence* 4 (jul. de 2015), pp. 55–63. DOI: 10.14569/IJARAI.2015.040710.
- [8] Wei Huang, Yoshiteru Nakamori e Shou-Yang Wang. “Forecasting stock market movement direction with support vector machine”. Em: *Computers Operations Research* 32.10 (2005). Applications of Neural Networks, pp. 2513–2522. ISSN: 0305-0548. DOI: <https://doi.org/10.1016/j.cor.2004.03.016>. URL: <https://www.sciencedirect.com/science/article/pii/S0305054804000681>.

- [9] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [10] Johan Bollen, Huina Mao e Xiaojun Zeng. “Twitter mood predicts the stock market”. Em: *Journal of Computational Science* 2.1 (2011), pp. 1–8. ISSN: 1877-7503. DOI: <https://doi.org/10.1016/j.jocs.2010.12.007>. URL: <https://www.sciencedirect.com/science/article/pii/S187775031100007X>.
- [11] László Nemes e Attila Kiss. “Prediction of stock values changes using sentiment analysis of stock news headlines”. Em: *Journal of Information and Telecommunication* 1 (2021), pp. 1–20. DOI: 10.1080/24751839.2021.1874252. eprint: <https://doi.org/10.1080/24751839.2021.1874252>. URL: <https://doi.org/10.1080/24751839.2021.1874252>.
- [12] X. Zhou et al. “Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets”. Em: *Mathematical Problems in Engineering* 2018 (2018), pp. 1–11.
- [13] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].
- [14] François Chollet. *Deep Learning with Python*. Manning, 2017. Cap. 8. ISBN: 9781617294433.
- [15] Sepp Hochreiter e Jürgen Schmidhuber. “Long Short-term Memory”. Em: *Neural computation* 9 (dez. de 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.
- [16] François Chollet. *Deep Learning with Python*. Manning, 2017. Cap. 6. ISBN: 9781617294433.
- [17] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016. Cap. 10.
- [18] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016. Cap. 9.
- [19] Katarzyna Janocha e Wojciech Marian Czarnecki. “On Loss Functions for Deep Neural Networks in Classification”. Em: *CoRR* abs/1702.05659 (2017). arXiv: 1702.05659. URL: <http://arxiv.org/abs/1702.05659>.
- [20] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly Media, Inc., 2019. Cap. 4.

Cronograma de atividades

Atividades	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
Estudo do Problema	x	x										
Coleta e Limpeza dos Dados		x	x	x								
Aplicação do Ferramental				x	x	x	x	x	x			
Análise de Resultados								x	x	x	x	
Redação do Relatório Final											x	x