Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/45/20241107

The application and investigation of dropout layer in the generator of GAN in stock prediction

Zhengyuan Li

The Department of Material Science and Technology, Jilin University, Changchun, 130000, China

lizy1620@mails.jlu.edu.cn

Abstract. Recent years have witnessed rapid advancements in neural network capabilities. Among the cutting-edge techniques emerging in this field is the Generative Adversarial Network (GAN).The GAN framework comprises two competing neural networks: one aims to introduce noise and deceive its counterpart, while the other hones its skills on genuine data, guiding the first on enhancing the realism of its noise. After extensive training, the goal is for the former network to produce data so convincing that the latter cannot differentiate between authentic and fabricated. The study's objective is to harness this formidable technique to model time series data, with a primary focus on stock market dynamics. A GAN proficient in handling the complexities of time series data, notably unpredictable realms like the stock market, has vast potential applications. While GANs have proven adept at navigating the intricacies of time series data, they aren't without challenges, notably the issue of overfitting. In this study, it will address this limitation by integrating a dropout layer into the generator. Such advancements in GAN could revolutionize the finance sector, offering improved risk evaluations for investments. Furthermore, GANs might hold the key to creating synthetic duplicates of confidential data, ensuring data veracity without jeopardizing confidentiality. In an era where vast amounts of data are locked away due to privacy concerns, the ability to generate precise, synthetic datasets could truly be groundbreaking.

Keywords: Machine Learning, GAN, Dropout, Stock Prediction.

1. Introduction

In recent times, there has been a rapid and remarkable advancement in the capabilities of neural networks. One of the most recent techniques with these networks is Generative Adversarial Networks (GANs). In this GAN architecture, two neural networks are pitted against each other, one trying to deceive the other with noise, while the other trains on real data and responds with information on how to make that noise more realistic. Through numerous iterative cycles, the objective is to generate data that is indistinguishable from real data according to the discerning network [1]. In the time series data prediction including the stock market data, the application of GAN is rarely studied. A GAN capable of performing well with time series data, especially chaotic data like that of the market, would be highly useful in many other fields. One of them is finance, where it is possible to better predict investment-related risks, but another application could be the effective anonymization of sensitive and private data.

DOI: 10.54254/2755-2721/45/20241107

Much data today is not shared due to confidentiality, so being able to generate accurate synthetic versions without loss of information would be helpful.

While deep learning techniques have heralded remarkable advancements across a variety of sectors due to their unparalleled prowess in data handling [2-4], their infiltration into the financial domain, especially in realms like stock price forecasting, portfolio optimization, financial data processing, and strategic trade formulation, has been notably significant [5]. A predominant focus within this surge of interest in financial studies is the prediction of stock market movements. Despite the increasing emphasis on machine learning, Artificial Intelligence (AI), GANs, and AI-driven stock predictions in scholarly discourse, a palpable void exists: the integration of GANs for stock predictions remains conspicuously underexplored.

The primary objective of this analysis is to shed light on daily data interactions from the S&P 500 Index, coupled with insights from a plethora of stocks spanning diverse trading timelines, to envisage daily closing prices. Preliminary findings intimate that this GAN-centric approach holds promise, outshining several extant machine and deep learning paradigms in predictive efficacy. It is, therefore, posited that a more profound exploration in this direction can not only furnish invaluable insights to refine stock prediction techniques but also offer a groundbreaking methodology to tackle the intrinsic challenges inherent in stock market analyses.

This restructuring positions the convergence of GANs and stock prediction as a clear research gap, while also highlighting the potential promise shown by some preliminary findings. In this study, the Dropout layers will be incorporated within the generator architecture to effectively counter the potential issue of overfitting. Dropout introduces a randomized deactivation of a proportion of neurons during the training process, thereby compelling the network to develop increased resilience and adaptability to different data patterns [6]. In parallel, this study meticulously fine-tuned the regularization strength, embarking on an empirical exploration of diverse values. This deliberate adjustment aims to find an optimal equilibrium that effectively marries the critical task of generating authentic and true-to-life data with the essential goal of mitigating overfitting, thus fostering a more robust and well-balanced generator model.

2. Method

2.1. Data preparation

The data was collected from the S&P 500 companies, 20 historical days, 5 days ahead predictions, 50k epochs, and various levels of percentage threshold, by getting an alphavantage api. This will allow to download the stock data using their API. To better predict the rise or fall of stocks, the threshold is set to 0, in order to simply predict the stock's upward or downward movement

2.2. Models

Xgboost: XGBoost is a powerful and popular machine learning algorithm that falls under the category of gradient boosting methods [7, 8]. It was developed to tackle regression, classification, and ranking problems and has gained widespread adoption in both academia and industry due to its high performance and scalability.The main idea behind XGBoost is to combine the predictions of multiple weak learners, typically decision trees, into a strong learner that is capable of making accurate predictions. It does this by iteratively training decision trees on the errors made by the previous ensemble of trees, effectively learning from the mistakes of its predecessors and refining the model at each step.

GAN: GAN is a new framework which trains two models like a zero-sum game [9, 10]. Utilizing GANs in trading can open up a multitude of applications. Specifically, GANs can be employed to generate synthetic market data, predict market trends, optimize trading strategies, and detect anomalies in trading patterns. The motivation for utilizing GANs arises from their inherent capacity to comprehend and replicate the distribution of real-world market data. Additionally, these networks possess the capability to automatically identify valuable data features, offering promising opportunities for enhancing trading strategies. When applying GANs for trading applications, the initial step involves

DOI: 10.54254/2755-2721/45/20241107

data preparation, encompassing the collection of historical market data. Following this, the GAN model is trained using this dataset. Once the training is complete, the model can be applied for purposes like prediction and strategy optimization. The final step involves a thorough evaluation of the GAN's outcomes, including assessing its prediction accuracy and the efficacy of the trading strategies it suggests.

2.3. Baseline model

2.3.1. Generator

- The generators used are made of three combinations of a multilayer perceptron and activation layers. The sigmoid function is used as activation function.

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

σ(x): This is the notation for the sigmoid function applied to an input
x:The result is a value between 0 and 1.

- To avoid overfitting, L2 regularization is used, and the regularization term is defined as:

$$\lambda_G \frac{1}{Q} \sum_{i=1}^{Q} ||\theta_G^{(i)}||_2^2 \tag{2}$$

λ_G :The regularization coefficient. Determines regularization strength. A higher. value places a stronger penalty on larger weights, with 0 meaning no regularization. Set it carefully to balance between underfitting and overfitting.

$\frac{1}{Q}$:Averaging factor. Q is the total number of model parameters (weights). This term computes the average of the squared weights.

- And in the case, the loss function is the sigmoid cross entropy loss, and the total loss consists of loss produced by the fake data and the regularization loss, and is defined as:

$$G_{loss} = -\frac{1}{N} \sum_{i=1}^{N} \log(\sigma(D(G(z^{(i)}))) + \lambda_G \frac{1}{Q} \sum_{i=1}^{Q} ||\theta_G^{(i)}||_2^2 \tag{3}$$

- And the optimizer used is adam optimizers, defined as

$$\theta_t = \theta_{t-1} - \alpha \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}+\epsilon}} \tag{4}$$

θ_t:it is gradient
$\widehat{m_t}$: the first moment (the mean)
$\widehat{v_t}$:the second raw moment (the uncentered variance)
α:the step size

- Incorporate dropout into the generator.In neural computation, there's a regularization technique called Dropout. During the learning phase, it operates by arbitrarily "turning off" or zeroing out a fraction of input units. Such a method aids in preventing the system from over-adapting to the training samples. The dropout is defined as:

$$z = \frac{1}{1-p} \times mask \times z \tag{5}$$

mask: it is a binary vector
p: it is the dropout probability
z: represents the output of a neural network layer before applying dropout.

2.3.2. Discriminator. In the discriminator, three convolutional neural networks combined with two multilayer perceptrons and activation layers are used. The usual structure consists of a convolutional layer, a ReLU activation layer, and a dropout layer.

DOI: 10.54254/2755-2721/45/20241107

This structure is repeated three times, followed by two multilayer perceptron layers. Finally, a sigmoid function activates the hidden layers.

- Real loss:

$$D_{lossreal} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(1 - \sigma(D\left(x_{real}^{(i)}\right)\right) \tag{6}$$

$D_{lossreal}$: Represents the discriminator's loss on real data.
N: it is the total number of data samples in a batch.
σ:Denotes the sigmoid activation function, which squashes its input into the range [0, 1].
$D\left(x_{real}^{(i)}\right)$: it is the discriminator's output for the real data sample.

- Fake loss:

$$D_{lossfake} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(1 - \sigma(D(G(z^{(i)})\right) \tag{7}$$

$D_{lossfake}$: Represents the discriminator's loss concerning the fake or generated data.
$z^{(i)}$: the noise sample input to the generator

- Regularization:

$$\lambda_D \frac{1}{P}\sum_{i=1}^{P} ||\theta_D^{(i)}||_2^2$$

λ_D: The regularization coefficient for the discriminator.
P: denotes the total number of parameters (or weights) in the discriminator.
$||\theta_D^{(i)}||_2^2$:This represents the squared L2 norm (Euclidean norm) of the i^{th}parameter (weight) in the discriminator.

- Total loss

$$D_{loss} = D_{lossreal} + D_{lossfake} + \lambda_D \frac{1}{P}\sum_{i=1}^{P} ||\theta_D^{(i)}||_2^2$$

2.3.3. xg-boost. This study first employ a discriminator to understand and distinguish between real and fake data. The goal of this discriminator is to identify the key features of the data that distinguish real instances from fake ones. Once this discriminator has been trained and is effective at distinguishing between real and fake data, the features it has learned are exported for utilization as inputs to the prediction model. The choice for the prediction tool is the XGBoost model, as it is an optimized gradient-boosting algorithm well-suited for handling large data sets efficiently, providing both flexibility and scalability. The XGBoost model operates in an iterative manner, with each step trying to improve upon the results of the previous one. Specifically, it creates a new model in each iteration to predict the prediction errors (or residuals) of the previous model, thereby gradually improving the accuracy of its predictions. Thus, the system combines the feature understanding capability of the discriminator with the powerful predictive performance of XGBoost, resulting in highly accurate and reliable prediction outcomes.

3. Results and discussion

Analyzing the results shown in Figure 1 and Figure 2, a consistent trend emerges: as the training progresses and the number of epochs increases, there's a noticeable and steady decline in the generator's loss. This indicates that the model is learning and improving its predictions with each successive iteration. The conclusive evidence from the final confusion matrix further bolsters this observation. Specifically, the model excels in its ability to predict declining stocks with remarkable accuracy, marking it as a potentially indispensable tool for financial analysts and investors alike. This performance

DOI: 10.54254/2755-2721/45/20241107

metric not only offers a valuable point of reference for subsequent research endeavors but also underscores the model's efficacy.

One of the perennial challenges in the realm of machine learning, particularly when dealing with intricate datasets like stock prices, is overfitting. It's the situation where a model might perform exceptionally well on training data but fails to generalize its predictions on new, unseen data. Addressing overfitting is paramount to ensure the reliability and real-world applicability of any model. In this context, this model showcases a distinct advantage. Through iterative refinements and by leveraging appropriate regularization techniques, it effectively mitigates the risk of overfitting, ensuring that its predictions remain robust and reliable across diverse datasets.

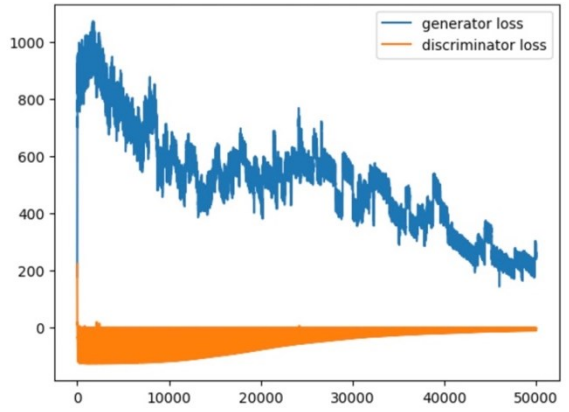

Figure 1. Generator and discriminator loss.

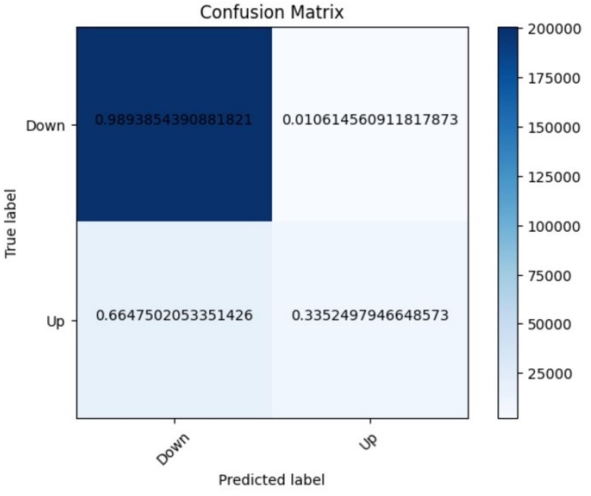

Figure 2. Confusion matrix of final prediction.

4. Conclusion

This study conducts a thorough analysis of daily data gathered from the S&P 500 Index and a diverse set of stocks across various time frames. The primary objective is to enhance the accuracy of predicting daily closing prices. Upon assessment, the predictive model exhibits strong performance, particularly in the prediction of declining stocks. This level of efficacy distinctly surpasses that of several contemporary computer learning models prevalent in the financial analytics domain. To improve the generation process and mitigate overfitting, a Dropout layer has been introduced into the GAN's generator. Training a GAN can be challenging, and employing techniques like Dropout and L2 regularization can lead to more stable and improved outcomes.

DOI: 10.54254/2755-2721/45/20241107

In this study, the stock trends of S&P 500 companies for the upcoming five days were successfully predicted, particularly in the case of forecasting declines. Nonetheless, it was discerned that the model's proficiency in predicting stock price surges necessitates further refinement. This avenue remains an area of potential exploration and enhancement in the ongoing research endeavors.

References

[1] Hirano M Sakaji H and Izumi K 2022 Policy gradient stock gan for realistic discrete order data generation in financial markets arXiv preprint arXiv:2204.13338
[2] Zhang K et al 2019 Stock market prediction based on generative adversarial network Procedia computer science 147 400-406
[3] Qiu Y et al 2022 Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training Biomedical Signal Processing and Control 72 103323
[4] Monil P et al 2020 Customer segmentation using machine learning. International Journal for Research in Applied Science and Engineering Technology (IJRASET) 8(6) 2104-2108
[5] Hamedinia H Raei R Bajalan S and Rouhani S 2022 Analysis of Stock Market Manipulation using Generative Adversarial Nets and Denoising Auto-Encode Models Advances in Mathematical Finance and Applications 7(1) 149-167
[6] Srivastava N et al 2014 Dropout: a simple way to prevent neural networks from overfitting The journal of machine learning research 15(1) 1929-1958
[7] Chen T et al 2015 Xgboost: extreme gradient boosting R package version 0.4-2 1(4) 1-4
[8] Chen T and Guestrin C 2016 Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining pp 785-794
[9] Borji A 2019 Pros and cons of gan evaluation measures Computer vision and image understanding 179, 41-65
[10] Creswell A et al 2018 Generative adversarial networks: An overview IEEE signal processing magazine 35(1) 53-65