Atlantis Highlights in Intelligent Systems, volume 2
Proceedings of the 2021 International conference on Smart Technologies and Systems for Internet of Things (STS-IOT 2021)

Deep Learning based Optimization Model for Digital Currencies and Investment Portfolios

Jieyan Dong*

Department of Accounting, Yunnan University Dianchi College, Kunming, 650228, Yunnan, China
**Corresponding author. Email: 2426476701@qq.com*

ABSTRACT

The portfolio problem refers to investors or financial institutions that, in managing investments in all types of bonds, stocks, peripheral financial products, depending on the proportion of value held, constantly recombine to achieve stable and loss-free profits, but also to ensure that risk is minimized. Or control the risk within a certain predictable range and look for the most profitable combination of opportunities. In order to solve the portfolio problem in the digital currency market, this paper attempts model optimization based on a portfolio model based on a deep reinforcement learning framework proposed by previous authors. In this paper, based on the above model, we optimize the experience pool and avoid overfitting the model. Moreover, we perform experience repetition on the experience pool and L1 and L2 regularization on the model to prevent the model from being overfitted and thus improve the fault tolerance of the model. Finally, the Deep Reinforcement Learning algorithm is used to make direct trading decisions to fully explore and learn from the digital currency market and develop reasonable portfolio strategies.

***Keywords**: Deep Learning, Digital Currency, Portfolio Model*

1. INTRODUCTION

The latest RMB International Report shows that China's cross-border payment currency has reached a whole new high globally, approaching RMB 20 trillion, an increase of 23.4% compared to previous periods, and has become the fifth largest payment currency in the world [1]. With the rapid and steady development of our currency, the digitalization of currency in cross-border payments has given new momentum to the internationalization of our currency.

While the traditional view of currency investment is that investment is primarily about risk diversification, modern portfolio models consider the ability to solve investment risks as the primary research purpose. Based on the traditional digital currency portfolio model, explore models that can further reduce risk and guarantee predetermined returns or maximize benefits while controlling certain risks. In this way, people have the ability to control the level of risk in investments and make clearer decisions.

Deep Learning is inextricably linked to feature learning. Deep Learning trains primary feature representations into high-level feature representations and simplifies complex classification tasks through multi-layer processing [2]. The original goal is to mimic the neural network of the human brain for learning and interpreting data such as images, sounds, and texts. This paper explores the digital currency portfolio model by integrating deep learning theory to promote innovation in financial (technology) development. This paper draws on previous research and discussion on the portfolio problem, provides insight into the development history and background of the currency-based portfolio problem under deep learning conditions, and attempts to optimize the model for its shortcomings and areas for improvement.

2. DATA SELECTION AND PREPROCESSING

2.1. Data Sources

The data used in this paper comes from the major forex trading platform poloniex, one of the world's most important forex trading venues. poloniex provides historical data through the API, and users can easily access the desired data through the website, which contains about 88 freely convertible digital currencies. Table.1 Selected website data

Poloniex is the most popular exchange for digital currencies with a large following and a very user-friendly trading platform for computers and cell phones. poloniex has a large number of commercial transactions in popular currencies including Ethereum. It offers traditional cryptocurrency trading services through this platform. In particular, the services include foreign exchange trading, margin trading, and lending. The digital currency market is different from traditional financial assets. It is more decentralized and open. Its advantages are high transparency, low inflation rate, easy transferability and greater security.

Table 1. Selected website data

Features	Coins / Time	ETH	XMR	…	FCT	ETC
high	2019/1/30 16:00	0.00954	0.00548	…	0.00124	0.01
	2019/1/30 16:05	0.00482	0.00259	…	0.00241	0.01
	…					
	2019/3/28 13:05	0.06842	0.00451	…	0.00361	0.006
	2019/3/28 13:10	0.03047	0.00524	…	0.00207	0.006
close	2019/1/30 16:00	0.00647	0.00174	…	0.00285	0.01
	2019/1/30 16:05	0.00842	0.00254	…	0.00367	0.01
	…					
	2019/3/28 13:05	0.02764	0.00411	…	0.00307	0.006
	2019/3/28 13:10	0.02513	0.00319	…	0.00372	0.006
…						
volume	2019/1/30 16:00	428.90	4.77	…	3.95	0.60
	2019/1/30 16:05	195.62	1.64	…	2.03	0.60
	…					
	2019/3/28 13:05	40.28	2.91	…	3.04	0.40
	2019/3/28 13:10	18.55	7.42	…	2.34	0.40

2.2. Selection of Data

To effectively reduce the difficulty of the learning task, we usually use a method that removes feature attributes with low data relevance while ensuring that no important "relevant attributes" are missing in the process. In this work, we use the GBDT algorithm to rank and select the feature attributes of the currency, compared to the XGBOSST algorithm that directly determines the relevance values of all features in the set of feature attributes.

3. DEEP LEARNING AND REINFORCEMENT MODEL CONSTRUCTION

3.1. Model Preparation

3.1.1. Experience Pool Repetition.

The concept of experience pool repetition was first introduced by Lin in the late 20th century. The experience pool replay in the previous investment model belongs to the traditional repetition pool, and the equal probability sampling from the repetition pool is the main feature of the traditional repetition technique of the experience pool; however, the importance of different data samples for the optimization of the current algorithmic strategy in the experience repetition pool is different [3]. Simple equal probability sampling not only ignores the varying importance of data samples, but also neglects their correlation. By collecting important samples with high frequency, the learning speed of the strategy can be accelerated and the effectiveness of the algorithm can be further improved.

3.1.2. Optimization of the Experience Pool

In this paper, we present a reinforcement learning method that optimizes the experience pool sampling strategy in the digital currency portfolio optimization model. This increases the efficiency of reinforcing learning training and significantly improves the efficiency of reinforcing learning by improving the experience pool sampling strategy. The individual steps are.

Obtaining empirical data through sensor acquisition and interaction with the environment, and storing the empirical data in the empirical replay pool.

Randomly draw λ-β samples of empirical data with equal probability from the empirical playback pool, where β is a fixed value and $\lambda \geq 1$ indicates the degree of control priority sampling.

Compare the similarity between the states contained in each of the samples from the experience reproduction pool and the corresponding states of the current training strategy, and select the top β experience data samples with the highest similarity.

Train the current training strategy using the selected β experience data samples.

After a training session, determine whether the number of training steps has reached the maximum value, and if not, return to step (2). From the above scheme, it can be seen that the method improves the efficiency of reinforcement learning by improving the empirical repetitive sampling strategy. The improved empirical repetitive sampling strategy can be combined with all empirical repetitive reinforcement learning algorithms and applied to various reinforcement learning tasks. It not only improves the learning efficiency significantly, but the improved repetitive sampling strategy hardly introduces any additional computational complexity and maintains the high efficiency of training for reinforcement learning.

3.2. Network Models for Deep Learning and Portfolio Based on CNN Networks

Convolutional neural networks, or CNNs, have gained attention as a recognition algorithm that offers general advantages in many domains. CNNs are very effective in handling certain problems that have locally correlated features, such as image and speech data. In the past, these data were often processed by manually extracting features and then classifying the extracted features by a classifier. The results depend on the effectiveness of feature extraction. A convolutional neural network generally consists of three structural components, namely convolution, pooling, input layer, and output layer [4]. The input layer is used to input images or other types of data, the convolution layer and the pooling layer are used to extract feature information from the data, and the output layer is used to output the final result. The computational parameters of the CNN convolutional neural network are much smaller compared to the normal BP neural network, which is due to the local perceptual field and the sharing of weights, as well as the environmental constraints in the pooling operation [5].

3.2.1. Convolutional layer in CNN networks

The convolutional layer is unique to CNNs, and its activation function is generally

$$\mathrm{Re}\,LU(x) = \max(0, x) \tag{1}$$

The next level of the convolutional layer is the pooling layer, which is a unique part of the neural convolutional network and whose activation function does not exist. The calculus of a convolutional neural network is expressed as follows.

$$S(t) = \int x(t-a)\omega(a)da \tag{2}$$

Three-dimensional convolution is expressed as

$$s(i,j,k) = \left(X^{*}W^{*}Y\right)(i,j,k) = \sum_{m}\sum_{n}\sum_{s} x(i+m, j+n, s+k)\omega(m,n,s) \tag{3}$$

Where is the convolution kernel and X is the input value, i.e., the price matrix. In this model, X = (m, n, s) is a three-dimensional matrix, where m is the closing price, n is the lowest price, and s is the highest price. We input the data and according to the formula algorithm, we get the final result. As shown in Figure 1.

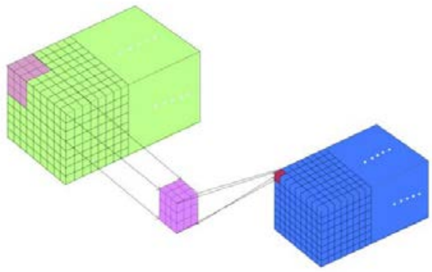

Figure 1. CNN three-dimensional convolutional network structure

3.2.2. Pooling Layer in CNN Network

One of the essential components of a CNN network is the pooling layer. The principle of the pooling layer is to compress the input sensor or matrix vector, e.g., compressing a 3X3 matrix to one element, or in the case of a 2X2 matrix, to one element. There are two common compression functions, namely.

$$f(x) = \max(X) \text{ op } f(x) = Average(X) \tag{4}$$

That is, the maximum value or the average value in the corresponding range is taken, as shown in the figure.

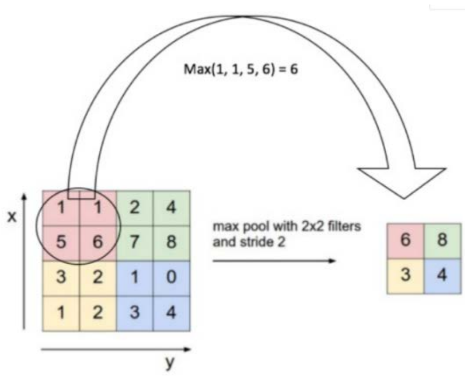

Figure 2. CNN pooling layer structure

3.2.3. CNN Network Backpropagation Algorithm

Let the pooling layer be δ^{l}, the inverse of the derivative of the previous layer be δ^{l-1}. Pooling layer we have used two compression functions max or average for pooling, in the backpropagation is the first step to first reduce the size of all submatrix elements of δ^{l} to the original to obtain the upper layer matrix δ^{l-1}, this process is usually known as upsample, so we can obtain.

$$\delta_k^{l-1} = \frac{\partial J(W,b)}{\partial a_k^{l-1}} \frac{\partial a_k^{l-1}}{\partial z_k^{l-1}} = upsample(\delta_k^{l}) \odot \sigma^{'}(z_k^{l-1}) \quad (5)$$

In this chapter, the principle and expression formula of the CNN network are presented. The whole process of the CNN network model first consists of the input data X, price matrix, first after the convolution, with the convolution operation with step size of 1, then into the pooling layer sampling, activation function ReLU, this function can effectively avoid the problem of gradient disappearance, then the second convolution, pooling into the experience set, If you insert the fund allocation weights used for the previous period of portfolio behavior into the characteristic chart, repeat the above process, you can get n feature maps of dimension 1, which is a vector of dimension 1 with the length of the currency number, where the vector is mapped to the [0,1] interval by the softmax function, then the output value is the portfolio decision made by the CNN.

4. MODEL OPTIMIZATION TO PREVENT MODEL OVERFITTING

4.1. Evaluating the merit of a model

The degree of performance of a model is judged by its data set, accuracy, recognition rate, etc. The goodness of a model depends on how good the data set is, how accurate it is, and how high the recall rate is [6]. The ability of the model to adapt to new samples is an issue that inevitably comes up in machine learning. The ability to adapt to new samples is also called generalization ability [7]. This ability indicates how good a model is, and a model with good adaptability to new samples is a good model. Table 2 shows the direct relationship between a model's generalization ability and its data set.

Table 2. Model generalization capabilities

Model	Model 1	Model 2	Model 3
Training set performance	Poor	good	good
Test Set Performance	Poor	Poor	Poor
Degree of model fit	Under fitting	over fitting	over fitting

4.2. The main methods of model optimization

The methods for remedying underfitting are broadly as follows.

Expand the hypothesis space by training with better features, adding higher order features, feature combinations, etc.

Regular term parameter lambda.

Replace new nonlinear models, such as RF, GBDT, SVM and DNN models.

The methods for solving overfitting are approximately.

Enlarge the data set, a large training database can effectively reduce the fitting.

Simplify the model, e.g., reduce the tree depth, reduce the number of features.

Bagging the learner can effectively reduce the overfitting of the model

Cross-check.

Regularization.

4.3. L1 and L2 regularization.

This paper focuses on L1 and L2 regularization methods to solve the problem of overfitting the portfolio models of previous authors. In machine learning, adding an additional term to the loss function, also called penalty term, such as $l_1 - norm$ and $l_2 - norm$, is called L1 regularization and L2 regularization, respectively. Where

L1 parameterization is.

$$\|x\|1 := \sum i = 1^n |x_i| \quad (6)$$

L1 loss function is.

$$L(w) = E_D(W) + \frac{\lambda}{n}\sum_{i=1}^{n}|w_i| \quad (7)$$

The parametrization of L2 is.

$$\|x\|2 := (\sum i = 1^n |x_i|^2)^{\frac{1}{2}} \quad (8)$$

The loss function of L2 is.

$$L(w) = E_D(W) + \frac{\lambda}{2n}\sum_{i=1}^{n}w_i^2 \quad (9)$$

Since the task of L1 regularization is to describe the energy of the signal as well as possible by a minimum number of coefficients, feature selection is performed during L1 regularization. The task of L2 regularization is to avoid overfitting the model. For the ConvLSTM network model in this work, L2 regularization yields the following equation [9].

$$s(i,j,k)=(X*W*Y)(i,k,k)=\sum_m\sum_n\sum_s x(i+m,j+n,s+k)\omega(m,n,s)+\frac{\lambda}{2n}\sum_{i=1}^{n}w_i^2 \quad (10)$$

Solving the gradient for L(w) yields.

$$w^{'}=(1-\frac{n\lambda}{n})w-\frac{\partial\sum_m\sum_n\sum_s x(i+m,j+n,s+k)\omega(m,n,s)+\frac{\lambda}{2n}\sum_{i=1}^{n}w_i^2}{\partial w} \quad (11)$$

As a result, the rate of parameter reduction becomes very slow when w tends to 0. That is, L2 regularization allows the reduction of parameters to a small range. The so-called overfitting consists in the fact that every point is considered in the fitting to the training data, so it is impossible to avoid the unusually strong variability of the fitted function, i.e. the large variance. The large variation of the function value in a given interval indicates that this function has a large K-value in that interval and only the coefficients are large enough to ensure that the function tends to fit relatively when the independent variables are constant in the given training set.

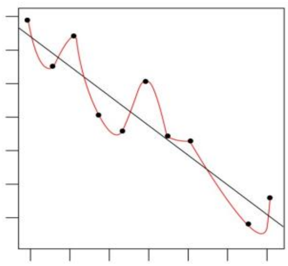

Figure 3. Fluctuation diagram of overfitting function

In general, models with small parameters can be fitted to different data sets, and if there is an offset in the large amount of currency attribute data selected in this paper, the effect on model output according to that parameter will not have much offset. This also reduces the risk of overfitting the model [8].

To support this conclusion, the following inference is made.

Assume that there is an existing model f with parameter θ and that the model obeys a normal distribution.

$$p(t|\ f,\theta,x)=N\left(f(x,\theta),\beta^{-1}\right) \quad (12)$$

The likelihood function is.

$$\ln p(\mathbf{t}|\ x,\theta,\beta)=-\frac{\beta}{2}\sum_{i=1}^{N}\left(f\left(x_n,\theta\right)-t_n\right)^2+\frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi) \quad (13)$$

Where t_n is the true sample label. The above equation is the objective function when there is no regular term. The optimal solution of θ is obtained by finding the maximum likelihood function, which is also a common loss function in machine learning. In this equation, we assume θ is a fixed value, and we assume θ is a normal distribution that obeys a mean of 0.

$$p(\theta|\ \alpha)=N\left(\mathbf{0},\alpha^{-1}\mathbf{I}\right)=\left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}}\exp\left\{-\frac{\alpha}{2}\theta^T\theta\right\} \quad (14)$$

Here, A is the parameter of this distribution and the posterior distribution of θ can be known by Bayes' la □□□□

$$p(\theta|\ \mathbf{x},\mathbf{t},\alpha,\beta)\propto p(\mathbf{t}|\ \mathbf{x},\theta,\beta)p(\theta|\ \alpha) \quad (15)$$

The left term in this equation is the model output distribution mentioned above by the quotient, and the right term is the priority distribution of θ. According to this equation, we can then obtain the optimal θ. After taking the logarithm, we obtain the objective function.

$$\frac{\beta}{2}\sum_{i=1}^{N}(f(x_n,\theta)-t_n)^2+\frac{\alpha}{2}\theta^T\theta+const' \quad (16)$$

The equation is then the regularized objective function. It can be seen that the risk of overfitting the model is effectively reduced [10].

5. CONCLUSION

An innovative idea for model optimization is proposed by combining the CNN network model with L1 and L2 regularization for the model built by the previous authors to optimize the experience pool reproduction. Then, the basic theoretical knowledge and principles of the CNN network algorithm and the ConvLSTM network algorithm are presented in detail. The underlying theories and principles of L1 and L2 regularization are presented. Through a theoretical comparison, it is shown that the ConvLSTM model can prevent overfitting after L2 regularization and generally optimize the experience pool, and that the portfolio model achieves better results after neural network regularization.

By reading numerous literature and reviewing data, we understand the whole process of portfolio modeling in Deep Learning and learn its theoretical knowledge in depth. The shortcomings of the model of previous authors have been optimized accordingly, and attempts have also been made to optimize the Deep Learning process, which can be improved. Although the optimized model has some theoretical superiority, there are still many shortcomings, and it is worthwhile to continue the in-depth study and research later. In addition, the CNN network model used in this work still needs to be improved, and a better model may appear in the future.

REFERENCES

[1] Yilin Ma, Ruizhu Han,Weizhong Wang. (2021)Portfolio optimization with return prediction using deep learning and machine learning. Expert Systems with Applications.Volume 165.

[2] Pan Wei; ,Li Jide, Li Xiaoqiang .(2020) Portfolio Learning Based on Deep Learning. Future Internet.Volume 12, Issue 11. PP 202-202.

[3] Thomas Nagler, Claudia Czado.(2016)Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. Journal of Multivariate Analysis, 15 .

[4] Lu S , Feng J , Zhang H , et al.(2018) An Estimation Method of Defect Size from MFL Image Using Visual Transformation Convolutional Neural Network. IEEE Transactions on Industrial Informatics.2018:1-1.

[5] Yarotsky D.(2017) Error bounds for approximations with deep Re LU networks. Neural Networks the Official Journal of the International Neural Network Society.94:103.

[6] Wuyu Wang, Weizi Li, Ning Zhang,Kecheng Liu.(2020) Portfolio formation with preselection using deep learning from long-term financial data. Expert Systems with Applications.Volume 143, 113042-113.

[7] Vadim Sokolov.(2017) Discussion of ‘Deep learning for finance: deep portfolios’ Applied Stochastic Models in Business and Industry.Volume 33, 16-18.

[8] Andersson Kristoffer, Oosterlee Cornelis W. (2021)Deep learning for CVA computations of large portfolios of financial derivatives Applied Mathematics and Computation.Volume 409, 2021.

[9] Yan Fan, Wolfgang Karl Härdle, Weining Wang et al.(2018) Single-Index-Based CoVaR With Very High-Dimensional Covariates Journal of Business & Economic Statistics. 36(2).

[10] Erhan Bayraktar, Ross Kravitz.(2013)Stability of exponential utility maximization with respect to market perturbations Stochastic Processes and their Applications. 123(5).