Construction and Optimization of a Prediction Model for Shenzhen Component Index Price Changes Based on a BP Neural Network

Yuexuan Che*

School of Computer and Big Data, Heilongjiang University, Harbin, Heilongjiang Province, 150080, China

Abstract. In the increasingly complex and dynamic global financial markets, the Shenzhen Component Index (Shenzhen Index), as one of the core indices of China's securities market, has garnered extensive attention from both investors and researchers. Price swings in the Shenzhen Index are due to many different complex reasons making it quite difficult to predict prices accurately. With the growth of AI technology, the Backpropagation Neural Network (BP Neural Network), known for its strong non-linear matching and adaptive qualities, is becoming a more significant tool used in predicting market movements. This research is aimed at improving and building a BP Neural Network model, trying to increase how well it can predict Shenzhen Index price changes. In this paper, efforts are made to optimize the structure and parameters of the BP Neural Network model to offer more precise guidance to investors and fill certain blanks in the existing literature. It is thus represented from the results that the use of the BP neural network is very effective for the leading price fluctuation prediction of the Shenzhen Index. This paper offers a theoretical and practical guide which may be beneficial for subsequent applications within the area of financial time series prediction.

1 Introduction

The complexity and volatility of financial markets have increased significantly in a global market, which is mainly due to the uncertainty coming from the fluctuation of time series data including currency exchange rates and stock index values that mostly affect investor behaviors [1]. Although traditional forecasting methods like the AutoRegressive Moving Average model (ARMA) and Generalized AutoRegressive Conditional Heteroskedasticity model (GARCH), are successful in linear prediction, they perform poorly when dealing with intricate nonlinear links inherent to financial markets [2]. In recent years, Backpropagation (BP) neural network, as a result of its powerful nonlinear fitting power, self-learning and self-adaptive features, has played an increasingly important role on financial prediction. One of them is a key stock index in China, the Shenzhen Index, which has practical significance for financial risk warnings and investment strategies optimization [3]. Due to the limited research

* Corresponding author: Che.Yuexuan@fayettetech.edu

in this domain, the price fluctuations of Shenzhen Index will be forecasted in the BP neural network model in this paper, and provide important data support and decision-making guidance for investors.

Financial forecasting, one of the principal areas where BP Neural Networks has been extensively used internationally and nationally as a significant body of research. Furthermore, using BP Neural Networks, Fang and successfully predicted the movement of the Shanghai Stock Exchange Index with rapid convergence and high prediction precision [4]. Zhang et al. fused Integrated Bacterial Foraging Optimization Algorithm (IBCO) and BP Neural Networks to form a reliable Standard & Poor's 500 Index (S&P 500) prediction model, it can be used in two scenes of forecasting the S&P 500 Index, one is short-term prediction, another long-term forecasting mode [5]. Liu et al. enhanced the accuracy of stock market price forecasting by combining a BP Neural Network and a Genetic Algorithm (GA) modified based on Cordero, Herrerab, [6]. Wu et al. Generally, it is reported that BP Neural Networks perform relatively better in dealing with the nonlinear fluctuations within the financial markets [7]. Additionally, Vui et al. employed BP Neural Networks from a hybrid trading system to predict stock prices, which further confirmed the availability of our method [8]; Peng et al. optimized BP Neural Networks with Genetic Algorithm, which increased the prediction accuracy [9]. Li et al. Before the deep learning era, Zhang et al. significantly improved the accuracy of a BP Neural Network for nonlinear data [10], and their refined learning algorithm allowed them to predict trustworthy stock markets accomplished by using it. Zhang and Shen proposed the use of Particle Swarm Optimization (PSO) algorithm to optimize BP Neural Networks in order to obtain more effective forecasting results for financial markets [11]. Qiu et al. proved that the BP Neural Networks can well deal with complicated variations in the financial market, also has a stronger robustness and adaptability [12].

However, the current related work on forecasting Shenzhen Index pricing fluctuations is rare. To this end, this paper optimizes the model of the BP Neural Network. The primary objective of the research is to forecast the Shenzhen Index using a framework of BP Neural Network. This fine-tuning of the neural network architecture for better performance of these predictions will involve changing the number of neurons in the hidden layers and the use of historical data in developing a reliable model. That said, relevant research aims to construct a model that can predict price fluctuation in Shenzhen Index with high precision. It would make much more sense for investors and market regulators.

The rest of this paper is organized as follows: Chapter 1 gives an introduction to the background and related work, the paper illustrates some necessary theory about BP Neural Networks model and introduces datasets for Shenzhen Index in Chapter 2; In chapter 3, the paper shows our experiment results and analyze them in detail; Chapter 4 concludes the paper and outlooks future work.

2 Research Methodology

2.1 BP Neural Network

This research employs the BP Neural Network, a multilayer feedforward neural network frequently utilized for time series forecasting and classification problems. The backpropagation of an error mechanism is essentially based on gradually adjusting the weights and biases inside the network for the purpose of improving the predictive accuracy of a model. A BP neural network is basically made up of three critical components, including the input layer, hidden layer, and output layer. In the framework represented in Fig.1, the input layer will receive the input data, mostly from the extracted feature value of the financial time series data. The hidden layer would be the most crucial ingredient of a generic neural network, where nonlinear mappings and processing of the input to extract useful features from it are done. The information synthesized in the hidden layer is summed up at the output layer to present a predicted output.

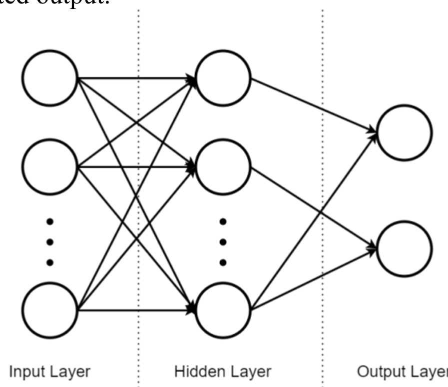

Fig. 1. Three-layer BP Neural Network(Picture credit : Original)

The operational framework of a BP neural network contains mainly two processes: forward propagation and backward propagation. During the process of forward propagation, information moves step by step from input via hidden layers to the output layer, thus enabling the mapping of forecasted values. Each node in these layers applies an activation function to bring in nonlinearities, hence the ability of the network to handle complex relationships. Only after forward propagation has taken place does the predicted result face the original value for error calculation. The quintessential process of BP neural network is the backpropagation phase, whereby weights and biases across the entire network will be regulated step by step in every layer by sending the error from the output layer backward to the input layer. Because it uses a gradient descent algorithm to minimize the loss by iteratively updating weights in such a way as to reduce the error. On one hand, backward propagation effectively enhances the accuracy of the predictions. On the other hand, it enhances learning and generalization of the model. In this respect, related code has been developed in the study with the use of MATLAB. Equipped with superior matrix operation capability, MATLAB is capable of handling large data efficiently and fast for training. Besides, it includes the Deep Learning Toolbox, which consists of functions and applications meant for designing, training, and simulating deep neural networks and traditional neural networks. It provides various tools to study, train, and simulate deep neural networks and conventional neural networks. It presents convenience in training, including but not limited to a BP neural network and Convolutional Neural Network (CNN); besides, it provides model evaluation and optimization. However, the implementation of BP neural networks is not confined to this toolbox alone, but any other programming language such as Python can be used as well.

The key factor to the good performance of BP neural network is to design the structure reasonably and choose proper parameters. This is because the number of hidden layers and nodes, the choice of the activation function, and setting a learning rate can substantially impact the training effectiveness of a network. In this paper, the three-layer network structure with one input layer, one hidden layer, and one output layer is used after several experimental adjustments. Besides, the number of neurons which is set in the hidden layer has been optimized for better performance to fit non-linear relationships, hence enabling high accuracy in prediction. In short, due to the backpropagation mechanism, the BP neural network performs very well in processing complex nonlinear financial data. Through iterative training and weight adjustment, the model eventually completes high-precision predictions of price fluctuations in the Shenzhen Index.

2.2 Data Source

The historical data pertaining to the Shenzhen Index utilized in this research was obtained from the Investing website. The Shenzhen Index's historical fluctuations are comprehensively represented by the dataset, which includes key indicators such as the daily opening price, closing price, highest price, lowest price, and trading volume. Data reliability is a cornerstone of this research. As a globally recognized financial data platform, Investing provides extensive market data from around the world, widely used in financial analysis and investment decision-making, offering high levels of authority and accuracy. This study employed essential preprocessing techniques, such as outlier removal and missing value management, to guarantee data completeness and dependability. The data spans the past ten years, a period selected to ensure that the prediction results are representative. Table 1 shows a sample of the data utilized for training and testing, while Fig.2 illustrates the historical volatility trend of the Shenzhen Index. The training and predictive efficacy of the BP neural network model is established on a firm foundation by the acquisition and preprocessing of this data, which guarantees its quality and reliability.

Table 1. Historical Data of the Shenzhen Component Index (Sample).

Date	Close	Open	High	Low	Volume	Change(%)
2024.8.23	8,181.92	8,134.40	8,207.59	8,133.57	8.83B	0.24%
2024.8.22	8,162.17	8,234.28	8,247.95	8,151.63	9.77B	-0.82%
2024.8.21	8,229.75	8,221.35	8,297.10	8,213.09	9.25B	-0.28%
2024.8.20	8,252.87	8,363.87	8,367.38	8,233.07	9.93B	-1.24%

Fig. 2. Changes in the Closing Price of the Shenzhen Component Index Over Time(Picture credit : Original)

2.3 Network Structure Diagram

This neural network for this study has an input layer, one hidden layer, and an output layer. The input layer of this neural network contains five variables, describing the characteristics of the input data: opening price, highest price, lowest price, closing price, and volume of trade. These five features capture the key fluctuation information of the stock market within one trading day, which are also common core data for financial market analysis and price prediction. Data transferred from the input layer to the concealed layer for processing.

This hidden layer can contain up to 50 neurons in this research. Each neuron in the hidden layer performs a linear transformation using weights and biases. An activation function then follows to inject nonlinearities into the processing so that complex inputs can be handled with greater capability by the model.

After nonlinear processing in this hidden layer, the signal reaches the output layer. The output layer contains only one neuron, which is supposed to yield the final result of the prediction. In training, the network would automatically readjust the weights and biases to minimize the error in this prediction so that it aligned with the goal value as best as possible. The complex time series data is coped with effectively by this structure of network. Therefore, the Shenzhen Index price fluctuation forecast can find its theoretical basis and practical reference here.

Fig.3 shows the particular structure of the network. The architecture strikes a good balance between model complexity and generalization skills. Since this model includes an appropriate number of neurons in its hidden layer, it can keep good performance in predicting nonlinear financial data without overfitting and underfitting problems.

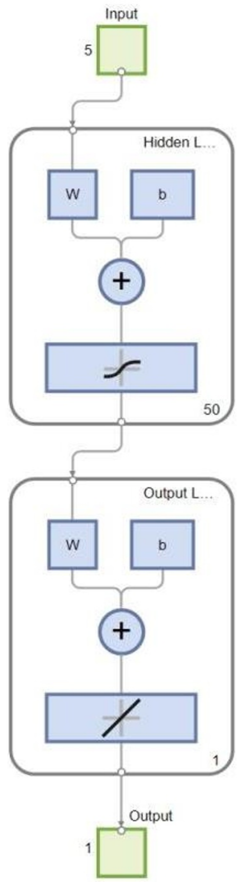

Fig. 3. Network Structure Diagram(Picture credit : Original)

3 Results and Analysis

3.1 Mean Squared Error

As a means of financial forecasting models, the Mean Squared Error 0(MSE) is considered one of the critical measures for evaluating the accuracy of a prediction. The MSE is a measure of dispersion that describes the difference between the observed value and the value that is expected; it is calculated as the average of the squared differences between them. A smaller MSE gives evidence of the model's higher efficacy in making better predictions.

It has, however, been realized in this investigation that throughout this training, the prediction error of the BP neural network gradually kept on decreasing. The best performance of the model was on the 6th iteration on the validation set; the MSE fell as low as 0.0002312, meaning at this point, the model fitted optimally according to the validation data. When the number of training iterations was increased, the error began to stabilize; this stabilized error rate means that the model had converged with a good generalization capability. The average error with a general decreasing trend shows the powerful capability for learning in financial time series data of the BP neural network. Initially, the error was high, but as the training progressed, the model kept on changing its weights and biases continuously to reduce the error at a fast speed until it reached its desired state.

Fig.4 shows curves about MSE variation of training, validation, and testing. The rapid decline at the beginning of training reflects that the model is well learned. Besides, the least validation error shows the best performance on the validation dataset, so the reliability and accuracy of the model.

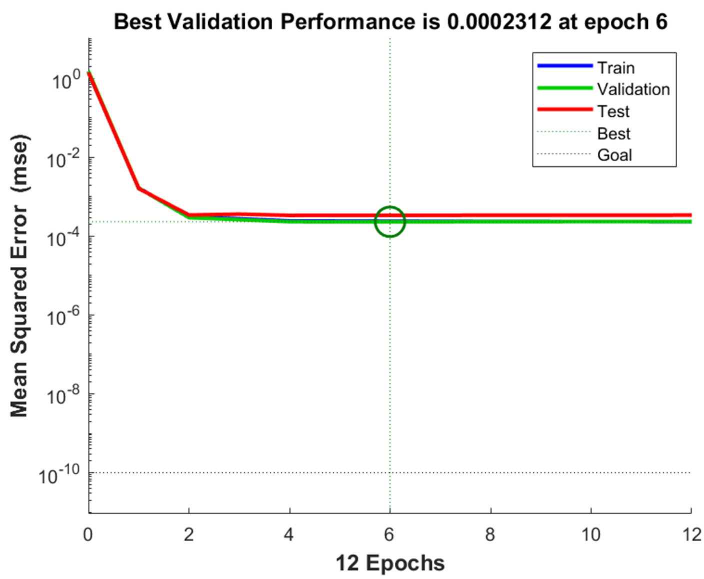

Fig. 4. MSE Changes During Neural Network Training (Picture credit : Original)

3.2 Parameter Changes During Neural Network Training

During the training process of the BP neural network, the changed model parameters are very useful and important to observe the progress of learning and ultimately converging. The most common parameters for each running network include gradient, Mu value, and the number of validations. These indices make it possible to visualize the effectiveness of model training and optimization.

The gradient reflects the magnitude of weight adjustments during backpropagation of error. In the early training phase, the gradient value is big, and this indicates that the network is in a rapid learning process where weights are adjusted with big magnitudes for minimization of the errors; with iterations of training as time goes by, the gradient value decreases gradually to smooth out-this is indicative that an optimum value of error has been reached from the model and the weight adjustment is reduced in magnitude to enter into the fine-tuning phase.

Mu is a tunable parameter of the BP neural network, which influences the learning rates. The Mu value increased rapidly within a very short period at the beginning of model training; it means that the network dynamically adjusts the learning rate to improve optimization efficiency. It stays at the high magnitude for a longer period while training will help the network adaptively change the learning rate. These variations in Mu value show how the network adopts a strategy for learning rate adjustment at different stages of training.

The quantity of the validation checks is one of the vital parameters that judge the model's performance concerning the validation dataset. The increase in validation checks signifies that the model is improving sequentially in prediction efficacy on the validation set. In the initial rounds of training, there will be infrequent validation checks. Once the model continuously trains and its parameters get optimized, the performance on the validation set increases; hence, increasing the number of validation checks.

Fig.5 shows the variation of gradient, Mu update and number of validation checks during training. It can be seen from curves in this figure that at iteration 12, all parameters leveled off, which means that the network converged and became stable.

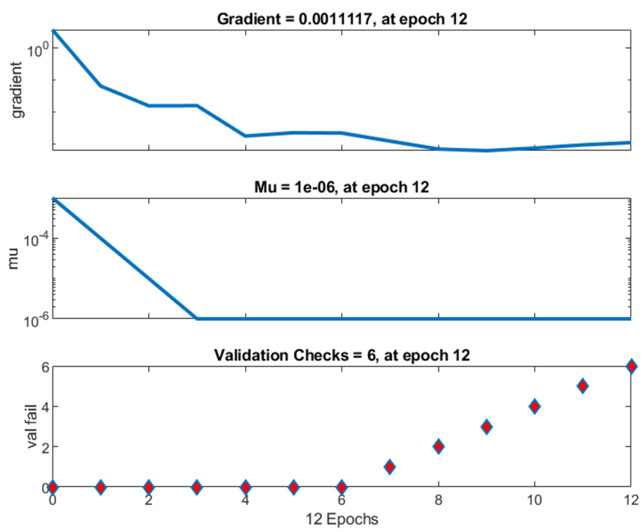

Fig. 5. Parameter Changes During Neural Network Training (Picture credit : Original)

3.3 Regression Analysis

Within the training procedure, some of the most important indicators refer to changes in these parameters, which can be used to check the progress of learning and eventually the convergence of this model. Such common parameters are, for example, the gradient, the Mu value, and finally the validation checks. With such indicators, it is good to visualize how the training and optimization of the model is effective.

Regression analysis is one of the major procedures in neural network prediction models. It makes use of a statistic, known as the coefficient of determination, symbolized as R^2, which looks at the closeness between the model's predictions and the actual data. The statistic R^2 falls within the range from 0 to 1, and values closer to 1 reflect a high level of precision in the forecasts, and also guarantee good agreement between the model forecasts and target values. Below is the regression plot analysis of the three phases: training, validation, and testing, including overall regression, focusing on the relationship between projected vs. actual values. The capability of the model as a predictive one would be effectively evaluated by comparing the fit of the regression line to the target line.

As seen in Fig.6, the R values for training, validation, testing, and overall regression are as follows: R_training = 0.99656, R_validation = 0.99683, R_testing = 0.99469, and R_overall = 0.99643. It can be seen from the result that there has been a fine consistency among the predicted and actual values of the training dataset, validation dataset, and test dataset. This developed BP neural network model had an excellent predictive performance with an R^2 value close to 1.

These high values of R^2 show how incredibly well the model performs, catching the price fluctuation of the Shenzhen Index. The model fitted well not only with the historical data but also generalized well on both the validation and test sets. The regression curves in the following Fig.6 further confirm this point: the forecasted points are highly coincident with the target line, and there exists a high coincidence between the fitted and actual values. Based on this, the R^2 analysis for confirming the improved performance of the proposed financial time series forecasting model using the BP neural network provides a theoretical and empirical foundation for further practical applications.

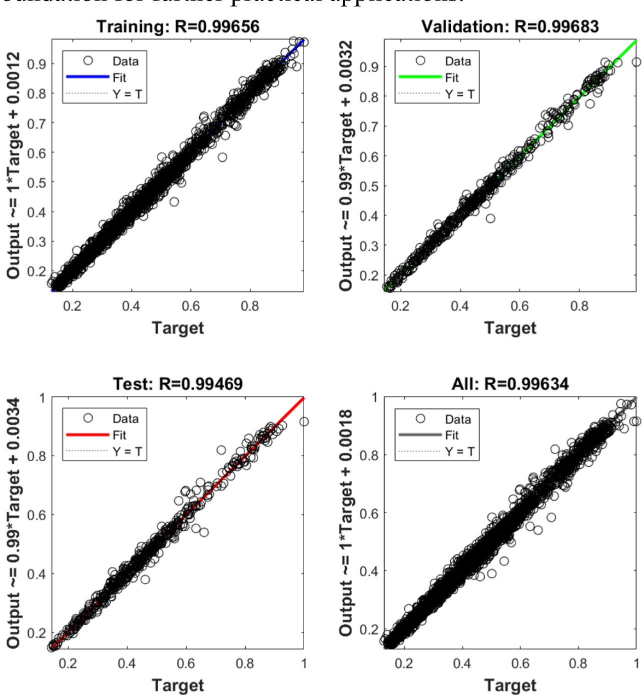

Fig. 6. Regression Analysis Chart for the BP Neural Network Model (Picture credit : Original)

3.4 Comparison of Predicted and Actual Values

Comparing predicted values with actual ones is a necessary method for testing the performance of a model. Only by such comparison can one visually estimate the ability of a model to encapsulate fluctuation tendencies of the Shenzhen Index against predictions provided by the BP neural network model. Although there are small differences in certain data, the whole prediction result coincides with the actual trend. It proves that the BP neural network can predict the price fluctuation of Shenzhen Index with a high precision, and has significant actual financial decision-making.

As seen in Fig.7, the blue curve represents the actual numbers, whilst the red one indicates the forecasted values. These two curves nearly entirely coincide, illustrating the model's forecasts for the Shenzhen Index with remarkable precision. The significant degree of fit indicates that the BP neural network is proficient at managing intricate financial time series data and demonstrates robust generalization abilities.

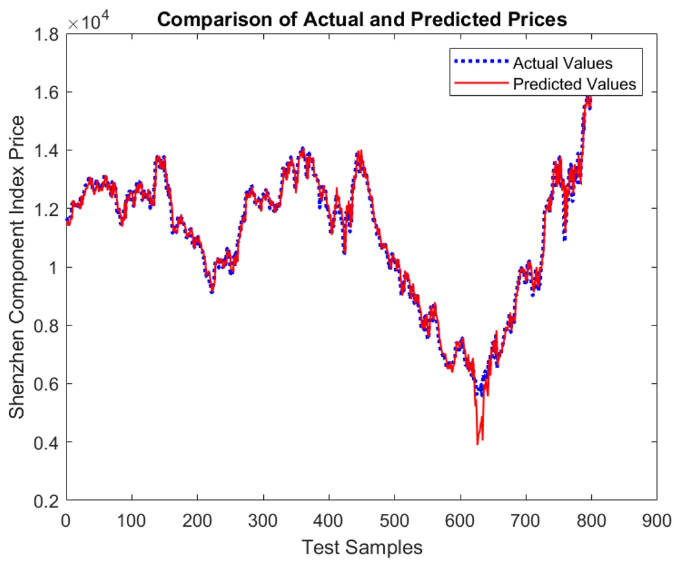

Fig. 7. Comparison of Actual Values and Predicted Values from the Optimal Network(Picture credit : Original)

3.5 Performance Optimization

The quantity of neurons in the buried layer of neural network models significantly influences the model's prediction efficacy. This study exclusively examines BP neural networks including a single hidden layer, as a three-layer BP neural network may approximate any nonlinear function. The quantity of neurons in the hidden layer considerably influences the network's efficacy. Modifying the quantity of concealed neurons can significantly influence the model's fitting and generalization abilities. Experiments are often conducted to ascertain the optimal number of neurons by evaluating the effect of varying neuron counts on the model's performance.

Fig.8 depicts the correlation between the number of neurons in the buried layer and the associated relative error. As the number of neurons increases, the relative error remains relatively stable when the number of neurons is small, particularly between 10 and 20 neurons, where the error hovers around 0.02. Within this period, the error of the model is relatively small, which means that the performance of the model is fairly good in prediction. With further increase up to a certain point, the error starts showing big variations, peaking when the number of neurons approaches 40, hence meaning that an excessive number of neurons increases the model complexity due to overfitting, impacting negatively on generalization capability.

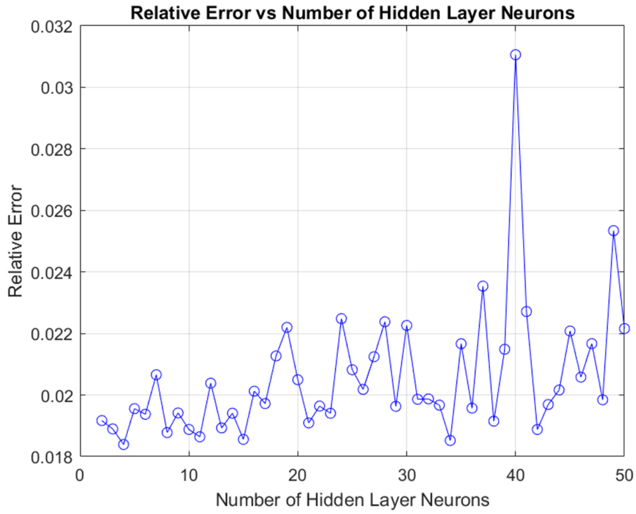

Fig. 8. Relationship Between Number of Hidden Layer Neurons and Relative Error(Picture credit : Original)

Fig.9 presents the relationship between the number of hidden neurons and R^2. It is obvious that the R^2 increases first and then drops as neurons grow. Generally speaking, too few or too many neurons result in poor model performance, while the best prediction can get with an optimal number of neurons. It starts to go up where the neurons vary from 10 to 30; though the R^2 value will fluctuate quite much, it shows that the fitting performance of the model improves slowly in this section. While approaching 30 neurons, the R^2 value reaches a high level, indicating that the fitting of the model at this place is relatively good.

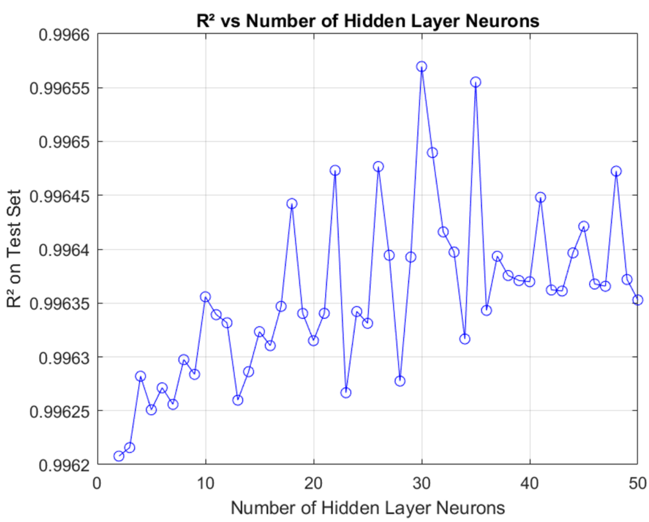

Fig. 9. Relationship Between Number of Hidden Layer Neurons and R^2(Picture credit : Original)

The number of neurons in the hidden layer clearly impacts how well the model predicts, which is seen when there are from 10 to 30 neurons, with errors being low while R^2 increases bit by bit, meaning the model does well. But then, when nearing 30 neurons, R^2 stays high, but error starts going up and down more noticeably, showing that overfitting could be a problem. Because of that, choosing the right number of neurons, not too many, not too few, becomes important in keeping the model accurate but also generalizable. From the data, 30 neurons appear as a good setup since R^2 peaks and error drops at that point. This points to a higher predictive capability of the model being at play.

4 Conclusion

This research uses a BP neural network model to predict the price fluctuation of the Shenzhen Index with a high degree of accuracy. Because it is one of the main indexes in China's stock market, the change in price for the Shenzhen Index is highly representative of complex and extremely nonlinear characteristics. By working out the structural optimization in the neural network architecture and painstakingly adjusting certain parameters-the number of hidden layer neurons-nonlinear features of the price change in this index were also grasped rather well by this model. On the training, validation and testing data sets, all performed quite well, which the R^2 in regression analysis is close to 1. That means the fitted line of actual historical data of Shenzhen Index can be fitted accurately by using the BP neural network, and the forecasted values are very close to the actual market trend.

During training, with the increase in training times, the model MSE decreased steadily and eventually converged. Therefore, the convergence and effectiveness of the BP neural network were proved. This research showed that the number of hidden neurons greatly influences the prediction performance of a single-hidden-layer neural network. Proper selection of the number of neurons could enhance the model's effectiveness in reducing the risk of overfitting or underfitting.

Though the predictive results are generally promising, some deviations in the line of data still existed; this could be due to the noise in the data itself or some other external factors that might not have been accounted for. Further research may use additional algorithms in order to obtain better performance from the model. Besides, a more expansive financial scope could include further financial indicators and data of other markets, which would provide wide applicability of the model in a broader set of financial markets for effective predictions.

In conclusion, this study shows that the use of a BP neural network in predicting fluctuations in the Shenzhen Index price comes with immense advantages. This can facilitate theoretical support and practical reference for future related works in the financial time series forecast. More complex data sources will be used to optimize the model structure for even more difficult financial prediction tasks in further work.

References

1. Z.F. Cao, Prediction of Gold Futures Prices Based on Hybrid Deep Learning. Comput. Program. Techn. Maint. 08, 118-121+125 (2024).
2. L.Q. Pu, H.M. Ma, Research on Carbon Emission Trading Price Forecast in China Based on LSTM Model. China Price 07, 29-34 (2024).
3. J.Y. Sun, G.Y. Bing, A New Method for Copper Futures Price Prediction Based on Multi-source Data Fusion. Oper. Res. Manag. 1-8 (2024).
4. B. Fang, S. Ma, Application of BP Neural Network in Stock Market Prediction. Int. Symp. Neural Netw. 5553, 1082-1088 (2009).

5. Y. Zhang, L. Wu, Stock Market Prediction of S&P 500 via Combination of Improved BCO Approach and BP Neural Network. Expert Syst. Appl. 36, 8849-8854 (2009).
6. P. Liu, W. Peng, Stock Price Forecasting Using BP Neural Network Optimized by Genetic Algorithm. J. Syst. Sci. Complex. 30, 895-905 (2015).
7. Y. Wu, J. Zhang, Comparative Study of BP Neural Networks and Other Models in Stock Market Prediction. Int. Conf. Inf. Sci. Manag. Eng., 284-288 (2014).
8. L.K. Vui, L.S. Lim, Time Series Stock Price Prediction Using BP Neural Network with Hybridized Trading System. Int. Conf. IT Convergence Secur. 225-231 (2013).
9. Z. Peng, W. Liu, Stock Market Prediction Using BP Neural Network Optimized by Genetic Algorithm. Procedia Comput. Sci. 122, 187-194 (2015).
10. W. Li, M. Qiu, Enhanced BP Neural Network for Financial Market Prediction. J. Financ. Eng. 36, 10050-10058 (2016).
11. L. Zhang, C. Shen, Stock Price Prediction Using BP Neural Network Optimized by PSO Algorithm. J. Financ. Eng. 36, 10050-10058 (2019).
12. M. Qiu, X. Song, Improving the Accuracy of Stock Market Prediction Using BP Neural Network and Particle Swarm Optimization. Expert Syst. Appl. 36, 10050-10058 (2016).