International Journal of Studies in Business Management, Economics and Strategies
Volume 4, Issue 6, June - 2025
ISSN (E): 2949-883X
Scholarsdigest.org

USING SINGLE LAYER NEURAL NETWORKS TO PREDICT STOCK PRICES

Haider Abbas abdullah Aljanabi
Kerbala University, Collage of Administration & Economics
Haider.abbas@uokerbala.edu.iq
ORCID ID : 0000-0003-4362-9071

Abstract
The research aims to shed light on the use of advanced technology in the field of financial investment. By using single-layer neural networks to predict future stock prices. The use of this advanced model over other similar models is to determine the accuracy of this technology in order for investors to resort to using it to achieve unusual returns by investing in stocks that achieve these returns. The results of the research showed that neural networks are highly accurate compared to traditional models, and thus using this technology helps investors make investment decisions quickly and accurately. The research recommended the need to educate investors to resort to technically advanced models in order to help achieve their investment goals and returns that help and encourage increasing the pace of trading within the stock market.

Keywords: Neural networks - single layer - stock prices.

Introduction

Intelligent automation of tasks and functions has become a crucial focus of work in modern university financial programs and companies. With the technologies of today, the potential of AI can now be acted upon. Deep learning is a modernized prototype of neural network studies that has seen considerable drive from various corporations and university researchers. There has been great improvement in solving old engineering issues, such as stock prediction, processing unstructured data, and much more. Deep learning of stock data might greatly assist in stock forecasting for the future. Stock price forecasting is an important and challenging research area. Some of the main challenges include: no stocks have certain prediction processes, data tends not to follow a typical distribution, various events can impact stock returns, and the intrinsic goal to exploit the profit of a certain model. Many forecasts of stock prices have been made, but a number of stock forecasting methods are considered effective, especially backward-looking architecture. Finally, a new forecast model using a single-tier layer framework based on neural networks was developed under the condition that financial information was temporarily difficult to manage. However, data collection problems have given rise to an upsurge in the number of studies in alternate marketplaces with alternate models including investment. As evidence demonstrates, the stock market is increasingly differentiated from actual business. Data mining, which is essentially needed to investigate patterns and correlations in these abstract data, must therefore improve in its study. The complexities of present analytical methods have certainly made a significant contribution, but in soft computing, the demand has increased. Usually, the purpose is to develop new methodologies that increasingly improve performance in stock prediction. Technical analyses of data, especially time series, are now commonly used in both academics and the field of business for predictive analytics. Furthermore, the integration of AI technologies in financial programs and companies has revolutionized the way tasks and functions are carried out. The advancements in deep learning, a refined form of neural network research, have garnered immense interest from various entities, including corporations and universities. This has led to significant progress in resolving long-standing engineering challenges, such as accurately predicting stock prices and analyzing unstructured data. Deep learning methodologies applied to stock data have the potential to greatly enhance future stock forecasting endeavors. However, stock price forecasting presents its own set of complexities and obstacles. For instance, the absence of definitive prediction processes for stocks, the non-conforming nature of data distributions, the impact of various external events on stock returns, and the core objective of optimizing profit through a specific model pose significant challenges to researchers. While numerous stock price forecasts have been conducted, certain stock forecasting methods have proven to be particularly effective, notably those employing backward-looking architecture. Recently, a novel forecast model leveraging a single-tier layer framework based on neural networks was devised to overcome the temporary difficulties in managing financial information. However, the emergence of data collection problems has prompted an upsurge in studies exploring alternative marketplaces and models, including investment strategies. It is becoming increasingly evident that the stock market is diverging from real business scenarios. Therefore, the study of data mining, which is critical for uncovering patterns and correlations in these abstract datasets, must be continually enhanced. Although current analytical methods have made significant contributions to understanding the complexities of stock prediction, there is a growing demand for advancements in soft computing. The focus is shifting towards developing new methodologies that enhance the performance of stock prediction models. Technical analyses of data, particularly time series data, have become commonplace in both academic and business spheres for accurate predictive analytics. In conclusion, the integration of AI technologies, specifically through deep learning methodologies, has ushered in a new era of intelligent automation in university financial programs and corporate environments. The scope of applications for these technologies has expanded to address longstanding engineering challenges, such as stock prediction and unstructured data processing. Deep learning, when applied to stock data, holds immense potential for improving future stock forecasting endeavors. However, the intricacies of stock price forecasting, including the absence of definitive prediction processes, deviations from typical data distributions, and the influence of external events, present significant challenges. Notwithstanding these challenges, certain stock forecasting methods, particularly those utilizing backward-looking architecture, have proven to be highly effective. Recent advancements include the development of a novel forecast model that leverages a single-tier layer framework based on neural networks, overcoming temporary difficulties in managing financial information. However, data collection problems have given rise to an upsurge of research in alternative marketplaces and models, including investment strategies. Furthermore, it is increasingly evident that the stock market is diverging from real business scenarios, necessitating further enhancement of data mining techniques to unravel patterns and correlations in abstract datasets. While existing analytical methods have made valuable contributions, the demand for advancements in soft computing techniques has surged. The focus now lies in developing new methodologies that enhance the performance of stock prediction models. Technical analyses, particularly of time series data, have become widespread practices in academia and the business sector for accurate predictive analytics. The integration of AI technologies has revolutionized financial programs and companies, opening up new avenues for automation and efficiency.

1.1. Background and Motivation

Early literature in stock price prediction has focused on the application of statistical methods such as logistic regression, machine learning algorithms, technical analysis, and various technical indicators. Such methods have faced numerous obstructions, particularly the required availability of a plethora of fundamental and technical indicators to capture most financial challenges and the assumption of "market efficiency." The standard techniques have struggled with capturing the advertised "complexity" in stock prices and finances, thus were unable to identify hidden patterns. (Kurani et al., 2023)

The neural network model has been recognized as a widely relied upon model in several financial fields. Neural networks are showcased to conduct nonlinear mapping between the input and output pattern spaces by optimizing weights using training. The techniques, being flexible and innovative, inspired by the human brain's architecture, can be easily tailored to a variety of prediction problems after an informed understanding of the financial properties and modeling requirements. This has opened a new era of incorporating machine learning techniques in financial time series data. Furthermore, the rapid increase in the volume and variety of financial data being traded in the form of chatter data, textual news, tweets, images, transaction data, and real-time data presents the need for a sophisticated machine learning model in the financial market. (Ashtiani et al., 2022)

Accurate stock price predictions enable professionals to predict the long-term trends signaling when to buy or sell stock, thus generating higher annualized returns. Asset management firms can use predictions to adjust portfolio compositions and make crucial investment decisions based on current trends in the market. The following section thus motivates the utilization of alternative and innovative methodologies like single-layer neural networks, which offer further inquiry. (Chhajer et al., 2022)

1.2. Objectives of the Study

Neural networks that contain only a single layer are seldom used for the purpose of stock price prediction. However, given that more complex neural network structures are combinations of single layers, it may be beneficial to train stock price prediction systems using this elementary structure as a proof of concept. This paper serves as an introduction to stock price prediction with a focus on improving neural network performance via data pre-modeling. Sub-goals of this study are outlined as follows: 1) to develop a predictive model trained using only a single layer of a neural network; 2) to evaluate the performance of this model when utilized for stock price prediction; 3) to allow the results to provide insight into the importance of using more complex network structures to facilitate accurate stock price prediction; 4) to explore alternative data pre-processing and modeling techniques that improve classification accuracy. Each of these objectives tackles a different type of risk specific to financial prediction. The first two objectives are of interest to the financial prediction community because the predictions created with single layer networks come at little cost, and therefore carry a large potential for high rates of return. The second two objectives act to demonstrate that the out-of-sample accuracy results produced from this model are robust to changing economic conditions. By experimenting extensively with the stocks, the third objective confirms the degree to which the single layer network can classify examples into different stocks.

This paper offers the results of the above outlined study and extends the practical applications of the results. In particular, we established an experimental stock prediction system that solely utilizes a single layer neural network. This is a first attempt to consider stock price prediction directly using an elementary neural network structure. However, this elementary neural network constrains stock to pull only one target attribute from all possible outputs; that is, it is limited to function only as a classification system with stock attributes as the classification labels. Stock classification was evaluated temporarily as a result, but we suggest that for future work the system can be more extensively cross-evaluated as a market prediction system by considering the stock with the highest percentage.

2. Neural Networks in Finance

Neural networks have been popular in finance for the past few decades. In the most basic sense, a neural network is a computer system likened to the human brain, which is also a computational network. The neurons in our biological networks are connected to each other; an individual neuron will only fire and activate if its connections have computed a certain mathematical decision based upon the input it has received. This same concept lies within a computer's neural network architecture. These networks are designed to generate a non-linear mapping function from the inputs to the outputs. Because of this, they are well-suited for financial data. The basic structure of a neural network consists of nodes that are organized in layers, with an input layer matrix, a hidden layer, and an output layer. No connections occur between nodes in a single layer dimension. (Papp et al., 2021)

Neural networks may effectively model the relationship between non-linear financial data and have been used extensively for a plethora of financial tasks including portfolio optimization, asset pricing, prediction of bankruptcy, evaluation of stock risk, pricing contingent claims, market risk exposure, and credit/loan risk. Many researchers use neural networks in the financial world for stock price predictions. Some researchers used a two-layer feedforward neural network to predict stock prices and achieved a prediction accuracy of 64%. Other researchers combine neural networks with other prediction methods, as some combined artificial neural networks with a support vector machine and genetic algorithms to predict stock prices. The above-mentioned studies prove the resilience of neural networks and their ability to model complex non-linear relationships within data and predict stock price directions. (Zulqarnain et al.2020)

Traditional models and theories used in finance promote the use of linear relationships between explanatory variables to predict future stock price movements. The problem with this approach is that stock returns typically follow a non-linear and non-stationary approach. A neural network approach to stock prediction may overcome this issue, as neural networks are generally known for their ability to model complex non-linear data. As reviewed earlier, neural networks have various applications in finance, and in fact, many have used them to predict stock market data with a certain degree of success. However, to the best of our knowledge, there is limited evidence to show that they are able to work with the Australian Stock Exchange stock data. (Chhajer et al., 2022)

2.1. Overview of Neural Networks

Neural networks are made up of simple computational units (neurons) arranged in interconnected layers and executed in the sequential order of these layers. A layer is a collection of neurons that execute a single operation, while a network contains the arrangement of these layers. Each neural network needs to define an input layer, an output layer, and none, one, or multiple hidden layers in between. Here, the input layer represents the initiating layer responsible for acquiring the inputs, and the output layer offers the final outputs. If more than one hidden layer is used, they appear in between the input and output layers and execute the vast majority of the computation. Each of the connecting lines, represented as directed edges, between neurons will have a weight attached to it. (Shahvaroughi Farahani & Razavi Hajiagha, 2021)These weights establish an indication of the overall network's structure. Another feature is a concept known as a bias, which represents the value of the connection weight to eventually shift the value in the preferred direction. A neuron then applies an activation function to the incoming signals to convert the weighted sum of the input from the previous neurons and the bias into an outgoing signal via the directed edges to the subsequent neurons. So, the network can "learn" to connect pairs of events through multiple intermediate neurons with their individual transition likelihood. (Xiao et al.2020)

The sequential component, the interconnected layers, can accommodate learning via adjusting the connection weights and biases. The learning occurs when input is connected through the weighted hidden layers to produce output, and the difference to the real output is passed back to adjust the weights and biases. Neural network systems could be designed to possess small, model-like flexibility. Such features make them suitable to solve a wide array of problems, including applications in finance. It is hypothesized that a single layer would suffice to predict stock prices, thereby establishing a rationale for this problem being fully solvable by neural networks. From a technical standpoint, understanding the capacity of single-layer neural networks to capture the macroscopic economic processes could serve as a form of edge compared to similar works that either simply apply neural networks or rely on alternative and more classical methods. Understanding the technical basics of one hidden layer neural networks provides, therefore, an essential starting point. As computer programs, neural networks break down mathematics into a digital format and implement what are essentially large collections of matrix multiplications. The model and its core components are discussed in the following subsections. (Kurani et al., 2023)

2.2. Applications in Stock Price Prediction

Stock price has always been a concern for investors, and the accuracy of stock price prediction has a significant effect on financial forecasts. Neural networks are widely utilized in stock price prediction, such as the backpropagation network, the radial basis function network, the Elman network, and the Hopfield network. Some characteristic extraction algorithms are proposed based on neural networks. Intraday or foreign stock prices and future market directions are forecasted using different methods. These statistical techniques and artificial intelligence models illustrate the future price paths more accurately than those of historical stock data when various factors are controlled. (Yu & Yan, 2020)

Many studies have documented the effectiveness of various algorithms, which can outperform many linear statistical methods and many traditional time series models. The model can outperform other models for short-term trading from 1 to 5 business days. The trading signals are unusual trading on the abnormal market trends. A comprehensive review summarizing roughly two decades of financial forecasting applications of these systems is available. If the traditional method has an equal or similar forecasting performance with an artificial intelligence model, researchers utilize the traditional method in combination with curve fitting models, computational intelligence, and neural networks to illustrate the accuracy improvement based on these methods. Financial applications tend to employ more black-box techniques successfully, with the result that the systems tested cannot perform identically to expert systems. However, some limitations of neural network applications are worth exploring, such as the difficulties faced in practicing neural networks. It is also mentioned that the forecasting accuracy is significantly affected by market volatility. (Rouf et al.2021)

In summary, researchers can forecast future stock price movements in advance using the algorithms or methods mentioned above. Moreover, investors can utilize the information to invest, and financial analysts can utilize the model to predict the trend of the companies' financial statements and make previously efficient investment decisions. Therefore, the finance domain is considered broad and important in this work. The application of neural networks and fuzzy systems in the finance domain has become more and more popular. (Pang et al.2020)

3. Single Layer Neural Networks

Logically speaking, any network composed of a single layer or node can be classified as a single-layer neural network, although orthogonal norms may logically apply to distinguish between deep networks and single-layer neural networks. Drawing such a distinction means that a multi-layer neural network must be composed of two or more layers. One of the simplest and most common architectures is a network consisting of a single-layer perceptron and summation operation. A single-layer neural network carries out processing on input vector x, which is combined with connection weight vector w. Simply put, it is a linear regression model that can approximate linear functions by giving suitable weights to input variables. Unlike multilayer neural networks, single-layer neural networks only have input and output layers and no hidden layers. The output of the network is the sum of the products of the weights and the inputs, which accounts for its simplicity. (Adil et al., 2022)

One of the most well-known examples of a single-layer neural network is the single-layer perceptron. The perceptron is a model similar to biological neurons and is why it is considered the basic model of neural networks. This model has the ability to learn weights. It must be noted that the learning rule of the perceptron enables online or streaming learning, whereby the model keeps learning new data step by step perpetually. In practice, such systems are considered to be able to capture or deal with linear data. Single-layer networks have also been used for stock price prediction. The approach has limitations because it assumes an already established relationship between input and output parameters. Single-layer networks cannot capture complex relationships because they have no hidden layers to carry out complicated computations. It is best to choose a deeper model with a minimum of two hidden layers to help capture complex interactions between variables. Nonetheless, a single-layer network is efficient in training when the problem at hand is not too complicated. When working with a small dataset or a dataset with few features, such as linearly separable or linear regression problems, it may be more efficient to use a single-layer network. (Gao et al., 2020)(Ding and Qin2020)

3.1. Definition and Structure

The Artificial Neural Networks (ANNs) are mathematical models that consist of various structural layers to recognize underlying patterns. Single Layer Neural Networks (SLNN) contain a very simple structure composed of the input and output layers. We cannot call a perceptron a SLNN with k hidden units or otherwise. However, we can refer to it as a structure that does not have a hidden layer between the input layer and the output layer. It can be seen that the input layer is directly connected to the output layer. No hidden unit is available between the input and output layers. In other words, each input of xi does not have any element-wise multiplication of weight or addition of bias before being set as output in the output layer. Instead, a small number of inputs of xi are directly jumped into the output layer, even with no hidden determinant. (Xiong et al.2020)

In this very simple structure of the network, it has a significant advantage and a limitation. The advantage is real because, in the prediction process using ANN architecture, the network should learn to mimic and forecast an observed real process. With the very simple structure of SLNN, the learning part of the network focuses more on calculating the weight and the bias while creating the right prediction. The architecture seems to ignore the high complexity of a series of activities such as transformations, which are performed especially by hidden layers. On one side, the simplicity of the architecture might lead to being less precise in capturing the 'hidden' patterns that have a series of data inputs. As a result, the network might be less trustworthy in making predictions. Each of the four main keys of neural networks, which are weights, bias, transformation, and activation function, are presented in this section with some examples and mathematical formulations. (Derogar et al.2024)

3.2. Activation Functions

An activation function processes inputs and decides which signals to propagate to the next layer of neural networks. In a single-layer neural network, the activation function is, therefore, the sole non-linear term responsible for making decisions based on the input signals. The activation function is differentiable, and it assigns a non-linear transformation, which introduces non-linearity in the output since the activation of a linear function is just a scaled, translated version of the original input. Activation functions can be linear or non-linear. A linear activation function results in a linear model where no non-linearity is introduced, making the purpose of using a neural network redundant. Therefore, non-linear activation functions are commonly used in the hidden layers. The choice of the activation function primarily depends on the requirements of the output from the network. (Koçak & Şiray, 2021)(Kunc & Kléma, 2024)

The popular activation functions are Tanh, Sigmoid, Rectified Linear Unit, and Leaky Rectified Linear Unit. Algorithmically, the Tanh and Sigmoid activation functions tend to push the output in extreme directions, making data mapping a non-satisfactory task. The derivatives of the Sigmoid and Tanh functions are close to zero and tend to vanish gradients for higher values. The ReLU and Leaky ReLU functions are mathematically more straightforward and simple, and they do not saturate for positive input values. However, inputs that are less than zero will cause the ReLU activation function to return zero. This introduces dead neurons that generate false predictions during the course of training. Moreover, it is believed that the choice of the activation function for a particular task depends on the requirements of the output and the desired performance of the classification or predictive model. (Szandała, 2021)(Linka and Kuhl2023)

There have been several attempts to use different activation functions and optimize the parameters by changing the structure of deep learning models. In general, it is important to note that modifying the structure of a model for a specific task cannot be done in an ad-hoc manner, and the appropriate activation function and the structure of the network need to be decided empirically. To this end, the mechanism of deciding the activation function is a crucial factor for any prediction algorithm, especially where the performance of the trained model is of primary interest. (Yu et al., 2024)

4. Data Preprocessing

The quality of data used for modeling a neural network determines the quality of stock prediction in return. There are several preprocessing measures from reading the input data to the final modeling. The first step is to clean and organize historical stock data. Data should be thoroughly checked for missing values, outliers, and incorrect labels. Normally, the second step is to normalize and scale the data. It helps the deep learning algorithm to learn easily, training the model and having a more precise result. The third step is to handle those missing values in all time frames from two perspectives. The first perspective is directly removing the missing data; the second is to input the missing values.

In this step, the dataset should be split into the training and testing sets in the timeframe of 80% as training data and 20% as testing data. The remaining 80% would then go through some techniques for data transformations: detrending, differencing, and chain-linking. These transformations reduce the impact of the structure of a particular model. Before the data transformation, feature engineering can be done to enhance raw data to better features. Data reading should import stock time frame data into data for further processing.

The purpose of every step in the preprocessing procedure can help decrease the number of features in the dataset, reduce the closing price of a single time frame, remove and fill the missing values of data, and provide a clean dataset. Furthermore, building a model using the training data can help in a new prediction process. Without enough processing for the training data, there would be a missing value error in the prediction process. In conclusion, the first step in data preprocessing provides a clean row of training and testing data.

4.1. Data Collection

In predicting stock market prices, most researchers have built their own neural network models to predict stock prices using the same concept. They use historical data of stock prices and convert stock types to stock features using market indicators for the technical analysis method and aggregate them into a system of datasets. Before converting them into data features that could be used as system datasets, analysts and researchers need to make some strict considerations on which one is a good and credible source of market indicators and stock prices. The credibility of the source data is used to assess the credibility of stock analysis that has been conducted and the accuracy of the prediction models.

For this, large financial companies provide their open stock prices and stock data at a certain fixed time. Retailers can download this data in different formats and even use application programming interfaces to access their data solutions and data providers in real-time and historical data. The best way to describe the process of collecting datasets for stock price forecasts involves several parts, such as data sources, methods, storage and data acquisition, legal issues, challenges, spectrum data like Forex, stocks, cryptocurrencies, and revenue optimization. Information not placed on the web page or without proper API calls and access credentials cannot be collected and may contravene legally held restrictions on data access. Generally, data obtained via APIs cannot be covered with licensed databases. Moreover, collections of already aggregated financial data are scarce and chargeable, ranging from tens of dollars for Forex stock prices to millions for certain datasets. Because of the great amount of data available in different features, the main motivation for including this subdivision is web scraping and custom-made data transformations for revenue optimization. Although this data includes more redundant features, an argument can be made for the inclusion of this highly aggregated spectrum data due to its possible increased complexity for simple NLP-based models that rely on word embeddings.

4.2. Feature Engineering

Feature engineering is a critical part of improving neural networks' ability to predict stock prices. By designing meaningful features that feed information about the historical data into the network, the network's predictive accuracy should increase. The design of features is crucial to the modeling process. By nature, features should be designed to capture more than what is apparent; that is, they should aim to capture relationships between lengths that were previously unapparent. Features can be constructed from the raw data using various techniques, including dimensionality reduction and transformation. Creating and selecting the next feature is highly iterative, as a new feature can often highlight a trend or pattern not visible from previous features.

There is a significant computational impact of selecting more versus fewer features. The more features the model has to train on, the more complex the model will be, and the longer the training times will be. Model performance, in general, will degrade. Conversely, the fewer features used to train the model, the faster the training times will be, and the general model performance will increase. The optimal amount of features to train on will often be the result of testing and tuning. It was observed that with good features, the accuracy levels are high, and it is possible to accurately predict the direction and movement of individual assets. Therefore, feature engineering is required for effective ETF predictions using a single-layer neural network.

5. Model Training and Evaluation

The model training process involves iteratively fitting the model to the training data. In the case of single-layer neural networks, the weights of the model are trained using the training data and the selected loss function with optimization methods. Optimization is generally achieved using a technique called gradient descent. During training, we use validation data to avoid overfitting the model to the training data. The performance of the network on the validation set is calculated by taking model predictions and comparing them with actual values. The model parameters are updated to decrease the error. (Wang et al., 2020)
Model evaluation is an essential part of model building in the domain of machine learning. As the main aim of developing a neural network model is to predict time series accurately, the basis for evaluating the performance of a model is assessing prediction accuracy at the short-term level. The most commonly utilized performance metric to measure the mean average prediction error is the mean square error (MSE). For a better evaluation of the predictive accuracy of a model, an additional metric, the mean absolute error (MAE), is used. The accuracy of predictions made by a model trained on noisy data should be evaluated, as sequential training data is naturally generated with noise. To evaluate the robustness of models to cross-datasets, the model is also validated against multiple cross-datasets. The performance metrics, such as training MSE and validation MAE, are dependent on both the time frame and the quality of training data. The variance and trend of each performance metric may vary with the change in quality of the training data. However, on greater time horizons, the model is less sensitive to the quality of the training data. (Hodson et al.2021)

5.1. Loss Functions and Optimization

During the training process of a neural network, the network parameters are iteratively updated and shifted toward a more desired state. This desirable attribute may vary but usually includes the performance metric improvement. The primary indicator that indicates the network state is the loss functions. A regression or regression-like problem uses a typical loss function, where the targets can take continuously distributed values, such as Mean Squared Error, Mean Absolute Error, and many others. A classification problem, where the targets are sometimes called the labels that have discrete and typically disjointed values, also utilizes various types of loss functions for achieving better performance, such as Hinge, Squared Hinge, and Categorical Cross Entropy.

Optimization is the mathematical analysis of constantly changing a predicted parameter inside a neural network to update the predictions and reduce prediction mistakes with each iteration of the model. A popular optimization method is called Gradient Descent, which updates the model weights by subtracting from them the value of the derivative of the loss function multiplied by the learning rate (i.e., the bigger the learning rate value, the bigger the step size toward the optimal parameter value). Gradient Descent has practical obstacles, such as finding the right learning rate; a too-big one causes the gradients and predictions to oscillate, while a too-small one causes slower convergence; and saddle points that can exist in the curve of optimization (as they are flat regions in loss curves that can trap Gradient Descent in both directions).

In training a neural network model, the process is iteratively performed on the dataset to improve predictions. This section on basic neural network details may seem simplistic or not relevant, but a deeper comprehension would be extremely beneficial for the readers in designing a new method to predict stock prices. A brief overview of how a neural network is trained will be discussed in the subsequent subsection in every approach section. The loss function acts as the source for all the various ways to train the network iteratively. A good loss function would minimize the distances from actual prediction to the desired prediction. The basic explanation between the actual and desired values would cause any predictions close to the other to have a very small loss value, and cause predictions that are very far away from any to have a large loss value.

5.2. Performance Metrics

An important part of evaluating neural network models is the performance metrics. For regression problems, some metrics that are most commonly used are the Mean Absolute Error and the Mean Squared Error. Both metrics tell us something about how far off predictions are from the real values. The key difference, though, is that the Mean Squared Error increases the influence of larger errors, and since it squares the values, it also amplifies the impact of positive and negative errors. For both of these measures, however, it is difficult to interpret their strengths and weaknesses without also knowing the level of the target.

Another commonly used performance measure is the R-squared value, sometimes also called the "coefficient of determination." This measure shows the proportion of variance in the target variable that has been explained by the model. Looking at its value, we can then come to a conclusion about the quality of the prediction. Although using performance metrics can help evaluate models effectively, it is important to also know which metrics to use in a particular experiment. The decision of which metric to use depends on the goals of the experiment and on the type of data the experiment is assessing. In general, using visualizations to assess the model output and then choosing a metric that measures the same items shown in the visualizations will give a better insight into the model's performance.

When solving classification problems, the dominant performance metrics are confusion matrices, which measure the frequency of correct predictions compared to incorrect predictions. It is essential to use insights from the confusion matrix to assess the likelihood of the model to provide actionable predictions. Traders may prioritize precision in order to minimize losses. Therefore, metrics endpoints are selected based on key considerations stemming from the accuracy, performance, and interpretability of a given model's predictions.

References:

1. Adil, M., Ullah, R., Noor, S., & Gohar, N. (2022). Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design. Neural computing and applications.
2. Ashtiani, F., Geers, A. J., & Aflatouni, F. (2022). An on-chip photonic deep neural network for image classification. Nature.
3. Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. Decision Analytics Journal.
4. Derogar, S., Ince, C., Yatbaz, H. Y., & Ever, E. (2024). Prediction of punching shear strength of slab-column connections: A comprehensive evaluation of machine learning and deep learning based approaches. Mechanics of Advanced Materials and Structures, 31(6), 1272-1290.
5. Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. International Journal of Machine Learning and Cybernetics, 11(6), 1307-1317.
6. Gao, P., Zhang, R., & Yang, X. (2020). The application of stock index price prediction with neural network. Mathematical and Computational Applications.
7. Hodson, T. O., Over, T. M., & Foks, S. S. (2021). Mean squared error, deconstructed. Journal of Advances in Modeling Earth Systems, 13(12), e2021MS002681.
8. Koçak, Y. & Şiray, G. (2021). New activation functions for single layer feedforward neural network. Expert Systems with Applications.
9. Kunc, V. & Kléma, J. (2024). Three Decades of Activations: A Comprehensive Survey of 400 Activation Functions for Neural Networks. arXiv preprint arXiv:2402.09092.
10. Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. Annals of Data Science.
11. Linka, K., & Kuhl, E. (2023). A new family of Constitutive Artificial Neural Networks towards automated model discovery. Computer Methods in Applied Mechanics and Engineering, 403, 115731.
12. Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. The Journal of Supercomputing, 76, 2098-2118.
13. Papp, Á, Porod, W., & Csaba, G. (2021). Nanoscale neural network using non-linear spin-wave interference. Nature communications.
14. Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. (2021). Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics, 10(21), 2717.

15. Shahvaroughi Farahani, M. & Razavi Hajiagha, S. H. (2021). Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. Soft computing.
16. Soyer, M. A., Tüzün, N., Karakaş, Ö., & Berto, F. (2023). An investigation of artificial neural network structure and its effects on the estimation of the low-cycle fatigue parameters of various steels. Fatigue & Fracture of Engineering Materials & Structures, 46(8), 2929-2948.
17. Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. Bio-inspired neurocomputing.
18. Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. Annals of Data Science.
19. Xiao, S., Guo, Y., Liao, W., Deng, H., Luo, Y., Zheng, H., ... & Yu, Z. (2020). Neuronlink: An efficient chip-to-chip interconnect for large-scale neural network accelerators. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(9), 1966-1978.
20. Xiong, H., Huang, L., Yu, M., Liu, L., Zhu, F., & Shao, L. (2020, November). On the number of linear regions of convolutional neural networks. In International Conference on Machine Learning (pp. 10514-10523). PMLR.
21. Yu, K., Bao, Q., Xu, H., Cao, G., & Xia, S. (2024). … prediction algorithm based on the optimisation of the Crown Porcupine Optimisation Algorithm with an adaptive bandwidth kernel function density estimation algorithm.
22. Yu, P. & Yan, X. (2020). Stock price prediction based on deep neural networks. Neural Computing and Applications.
23. Zulqarnain, M., Ghazali, R., Ghouse, M. G., Hassim, Y. M. M., & Javid, I. (2020). Predicting financial prices of stock market using recurrent convolutional neural networks. International Journal of Intelligent Systems and Applications, 13(6), 21.