A NEW DENDRITIC CELL-BASED DEEP ARTIFICIAL NEURAL NETWORK WITH DIFFERENTIAL EVOLUTION ALGORITHM USING DE/BEST/1 MUTATION STRATEGY

Erol Egrioglu – Eren Bas

Abstract

Long short-term memory neural networks and gated recurrent unit neural networks are popular deep neural network models based on long short-term memory and gated recurrent unit cells respectively. In this study, a dendritic cell structure is used and a deep neural network using this cell structure is proposed. The training of this deep neural network is carried out with the differential evolution algorithm that uses de/best/1 mutation strategy and thus the exploding gradient problem in long short-term memory neural networks and gated recurrent unit neural networks is avoided. The forecasting performance of this proposed new deep neural network is evaluated on the Hang Seng Index stock exchange time series. This proposed deep neural network method is compared with some deep and shallow neural network methods. As a result of the analyses, it is concluded that the proposed deep neural network method produces better forecasting results than the other methods compared.

Key words: forecasting, dendritic cell, deep artificial neural network, differential evolution algorithm

JEL Code: C53, C61,

Introduction

Although classical forecasting methods are initially used to solve time series forecasting problems, these methods are based on linear models. Artificial neural network methods have been used as an alternative to these methods in the analysis of both linear and nonlinear time series. Unlike classical forecasting methods, artificial neural network methods can solve complex forecasting problems with its data-driven approach feature. At the same time, the fact that classical forecasting methods are based on some assumptions has led to the use of artificial neural network methods, which do not require preliminary assumptions, as an alternative method. One of the most frequently used artificial neural network methods in the literature is the multilayer perceptron artificial neural networks (MLP-ANN) proposed by Rumelhart et al.

(1986). However, some of the important problems of this network are that this method has only an additive aggregation function and has the problem of determining the number of hidden layer units. In addition, as the number of hidden layer units increases in MLP-ANN, the structure of the network becomes quite complex, which increases the computation time of the network and causes the computations performed in its training to become unsolvable. Unlike MLP-ANN, the single multiplicative neuron model artificial neural networks (SMNM-ANN) proposed by Yadav et al. (2007) have been frequently used by researchers due to the fact that they do not require the hidden layer identification problem and have a multiplicative aggregation function.

The Pi-Sigma artificial neural network (PS-ANN) proposed by Shin and Gosh (1991) is an artificial neural network with both additive and multiplicative aggregation functions. Another artificial neural network model, the dendritic neuron model artificial neural network (DNM-ANN), is a multilayer feed-forward artificial neural network in which the output of the network is obtained through four different layers. Unlike these shallow neural networks, there are many deep neural network methods used in the forecasting literature. Deep neural network models have many more hidden layers than shallow artificial neural networks. Deep neural networks have a longer training process due to the fact that they have more layers and connections compared to shallow neural networks, but they use the method of sharing weights by all neurons with the parameter sharing feature. Many different deep artificial neural network methods have been frequently used in forecasting problems. Bilgili and Pinar (2023) used a deep-learning technique based on a long short-term memory neural network (LSTM) to forecast gross electricity consumption in Türkiye. Abumohsen et al. (2023) used LSTM, gated recurrent units and recurrent neural networks for forecasting electrical load. Li and Yang (2023) proposed a temperature prediction model based on the hybrid seasonal autoregressive integrated moving average) and LSTM model. Jhong et al. (2024) used genetic algorithm in LSTM for flash flood forecasting. Zhang et al. (2024) proposed a hybrid approach of wavelet transform, autoregressive integrated moving average and LSTM model for the share price index futures forecasting.

In this study, a deep artificial neural network method based on a dendritic cell named Deep dendritic artificial neural network (DeepDenT) proposed by Egrioglu and Bas (2024) is used. The training of this deep neural network method is performed with the differential evolution algorithm (DEA) based on the de/best/1 mutation strategy different from Egrioglu and Bas (2024), thus avoiding the exploding gradient problem, which is an important problem of a deep neural network method.

1 Differential Evolution Algorithm

DEA is a population-based artificial intelligence optimization algorithm proposed by Storn and Price (1997). The optimization process with DEA starts with a randomly generated population. This randomly generated population consists of chromosomes corresponding to the solutions for the algorithm and chromosomes consist of genes corresponding to the decision variables. Then DEA operators are applied for each chromosome in the population. These operators are mutation and crossover operators respectively. A DEA method is referred to by the mutation operator it uses. In a DEA method using a classical mutation operator, a chromosome called the chromosome of interest is first identified. The chromosome of interest is a name given to each chromosome to be mutated in the population, starting with one. Then, three chromosomes other than the chromosome of interest are randomly selected. The difference of the first two of these three randomly selected chromosomes is taken. This chromosome is called the difference vector. This difference vector is multiplied by a scaling factor F, which is usually taken as 0.8 in the literature, and a chromosome called the weighted difference vector is obtained. Finally, this weighted difference vector is summed with a randomly selected third chromosome to obtain a sum vector. This concludes the mutation process. This mutation process is called the de/rand/1 mutation strategy in the literature.

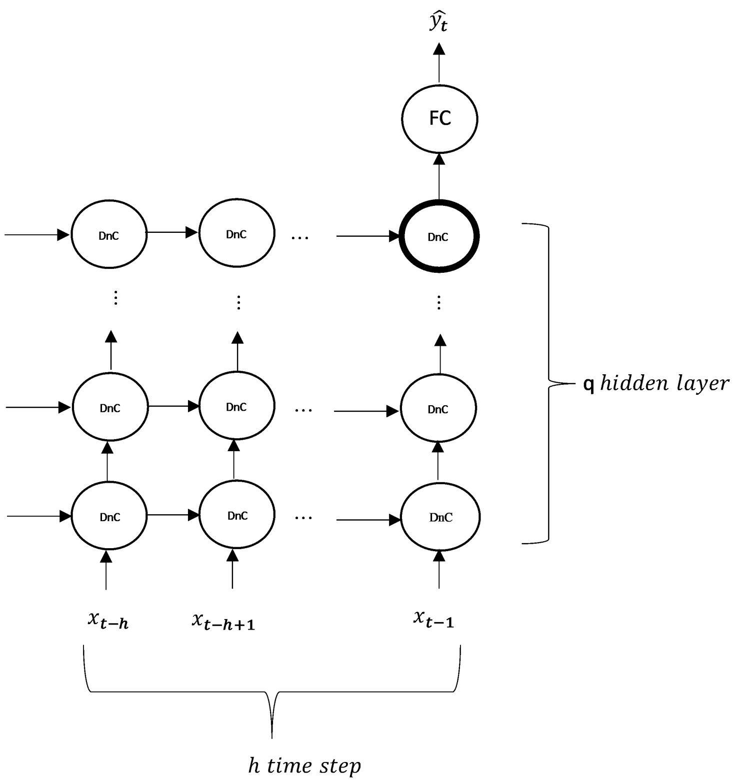
When applying this mutation strategy, only randomly selected chromosomes are considered. Another mutation operator strategy, the de/best/1 mutation strategy, differs from the de/rand/1 mutation strategy in that the best chromosome in the population is included in the mutation process. Here, the best chromosome is the chromosome that has the best fitness value according to the objective function. When applying the de/best/1 mutation strategy, two different chromosomes are randomly selected from outside the relevant chromosome. These two different chromosomes are again differenced and this difference vector is multiplied by the scaling factor. This weighted difference vector is then added to the chromosome with the best fitness value in the population and the mutation process is completed. A candidate chromosome is then obtained by comparing the mutated chromosome with the chromosome of interest. When obtaining a candidate chromosome, each gene of the mutated chromosome and the chromosome of interest must be evaluated separately. First, a crossover rate (cr) is determined. Then a random number between 0 and 1 is generated using a uniform distribution. If this random number is smaller than the crossover rate, the gene is removed from the mutated chromosome, if the random number is larger than the crossover rate, the gene is removed from the related chromosome and the candidate chromosome is created and the fitness value of the candidate chromosome is calculated. Finally, the fitness values of the candidate chromosome and the interested chromosome are compared. According to the objective function, the chromosome with the best fitness value among these chromosomes is included in the population.

2 The Proposed Method

In this study, a de/best/1 mutation strategy based on DEA is used to train the DeepDenT artificial neural network proposed by Egrioglu and Bas (2024) and shown in Fig. 1. The algorithm of the proposed method is given in the following algorithm.

Fig. 1: The architecture of DeepDenT

"Source: Egrioglu and Bas (2024) "



Algorithm. The proposed method

Step 1. Determine the parameters of the proposed method

cn: the total number of chromosomes in a population

cr: crossover ratio

$maxitr$: maximum number of iterations

h: the number of time steps

q: the number of hidden layers

m: the number of dendrites

p: the number of inputs or features

Step 2. Generate the initial population

Each gene of this proposed deep neural network method consists of the weights and biases of the neural network. Weights and biases are derived from the intervals $U[0,1]$.

Step 3. Calculate the fitness value for each chromosome

The fitness value for each chromosome is computed based on Equation (1). The fitness function is the mean squared error (MSE) based on the one-step forecasting performance. In Equation (1), $ntrain$ is the number of training set.

$$MSE_j^t = \frac{1}{ntrain}\sum_{t=1}^{ntrain}(y_t - \hat{y}_t)^2, j = 1,2,\dots,cn \qquad (1)$$

Step 4. The iteration counter is incremented by $t = t + 1$.

Step 5. Apply mutation operator

The mutation operator is implemented for all chromosomes in the population using the de/best/1 mutation strategy.

Step 6. Apply cross over operator

The cross over operator is implemented for all chromosomes in the population and the population is updated by comparing the candidate chromosomes.

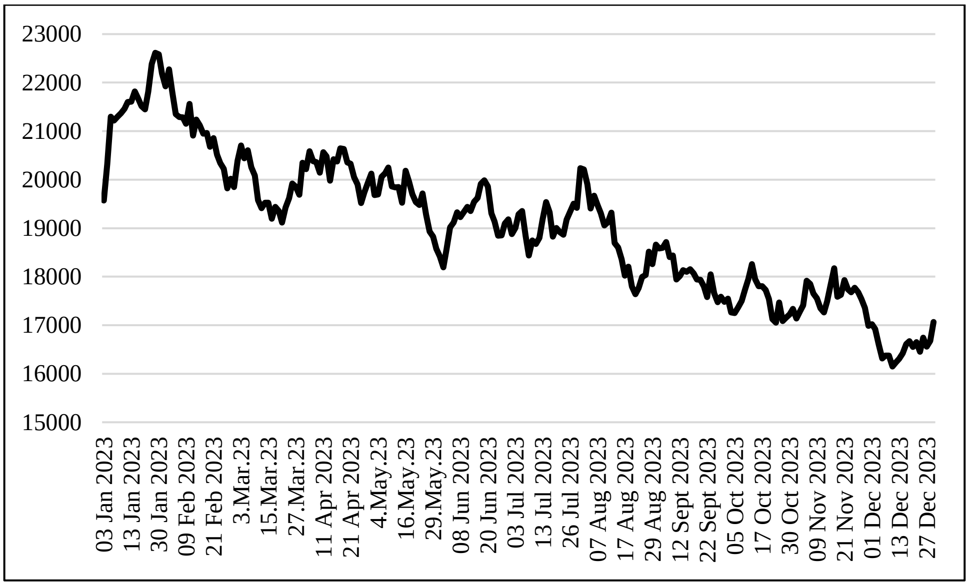
Step 7. If $t > maxitr$, the algorithm is terminated

3 Application

The analysis performance of the proposed method is evaluated over the Hang Seng Index (HKSE) time series observed daily in 2023, the graph of which is given in Fig.2. The analysis performance of the proposed method is compared with MLP-ANN proposed by Rumelhart et al. (1986), single multiplicative neuron model artificial neural networks (SMNM-ANN) proposed by Yadav et al. (2007), Pi-Sigma artificial neural network (PS-ANN) proposed by Shin and Ghosh (1991) and LSTM proposed by Hochreiter and Schmidhuber (1997).

Fig. 2: The graph of HKSE time series

"Source: Own research"



The comparison of the methods is based on the root mean square error (RMSE) criterion given by Equation (2). In the analysis phase, the HKSE time series is divided into training, validation and test sets. The length of the validation and test set for the relevant time series is set as thirty. The analysis results obtained from each method are given in Tab. 1. These analysis results are calculated for the test set of the HKSE time series. For each method, the optimal parameters are derived from the validation set by increasing the number of inputs (p), the number of hidden layers (m) and the number of hidden layer units (h) between 1 and 5. Each method is run thirty times with different random starts with the optimised parameters and thirty different RMSE values are calculated for the test set of each time series. At last, the mean, median, standard deviation, interquartile range, minimum and maximum values of the thirty various RMSE values are computed. Tab. 1 shows the statistics of the RMSE values computed over the test set of the HKSE time series and the the optimum parameters from each method.

$$RMSE = \sqrt{\frac{1}{ntest}\sum_{t=1}^{ntest}(x_t - \widehat{x_t})^2} \qquad (2)$$

Tab. 1: The RMSE statistics each method

"Source: Own research"

Methods	Mean	Median	Standard Deviation	Interquartile Range	Minimum	Maximum	p	h	m
PS-ANN	336,23	329,94	12,83	17,38	326,32	356,87	1	-	3
LSTM	224,47	223,58	3,15	4,76	219,22	230,58	5	2	1
SMNM-ANN	222,77	222,55	2,87	2,66	219,38	234,34	3	-	-
MLP-ANN	222,55	221,78	2,95	2,04	218,40	228,92	3	-	2
Proposed Method	220,83	220,72	1,22	0,97	218,71	223,40	3	2	3

When the analysis results given in Tab. 1 are analysed, it is seen that the proposed method is the most successful method in all other statistics except the minimum statistic. Although the MLP-ANN method is the most successful method for the minimum statistic, since both the standard deviation and the maximum statistic of this method are higher than the proposed method, the proposed method is a reliable method compared to the MLP-ANN method.

Conclusion

In this study, a deep artificial neural network based on dendritic cell is used and the training of this deep artificial neural network is performed with DEA based on de/best/1 mutation strategy. The forecasting performance of the proposed method is evaluated over the HKSE time series and the proposed method produces better forecasting results than various deep and shallow artificial neural networks.

In future studies, different optimisation algorithms can be used in the training of the DeepDenT artificial neural network method or the DeepDenT method can be transformed into a robust method that can be used in case of outliers in the data set.

Acknowledgment

This study was funded by the Scientific and Technological Research Council of Turkey (TUBITAK) ARDEB 1001 Grant No 122F133.

References

Abumohsen, M., Owda, A. Y., & Owda, M. (2023). Electrical load forecasting using LSTM, GRU, and RNN algorithms. Energies, 16(5), 2283.

Bilgili, M., & Pinar, E. (2023). Gross electricity consumption forecasting using LSTM and SARIMA approaches: A case study of Türkiye. *Energy*, 284, 128575.

Egrioglu, E., & Bas, E. (2024). A new deep neural network for forecasting: Deep dendritic artificial neural network. *Artificial Intelligence Review*, 57(7), 171.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Jhong, Y. D., Chen, C. S., Jhong, B. C., Tsai, C. H., & Yang, S. Y. (2024). Optimization of LSTM parameters for flash flood forecasting using genetic algorithm. *Water Resources Management*, 38(3), 1141-1164.

Li, G., & Yang, N. (2023). A hybrid SARIMA-LSTM model for air temperature forecasting. *Advanced Theory and Simulations*, 6(2), 2200502.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.

Shin, Y., & Ghosh, J. (1991, July). The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation. *In IJCNN-91-Seattle international joint conference on neural networks* (Vol. 1, pp. 13-18). IEEE.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341-359.

Yadav, R. N., Kalra, P. K., & John, J. (2007). Time series prediction with single multiplicative neuron model. *Applied Soft Computing*, 7(4), 1157-1163.

Zhang, J., Liu, H., Bai, W., & Li, X. (2024). A hybrid approach of wavelet transform, ARIMA and LSTM model for the share price index futures forecasting. *The North American Journal of Economics and Finance*, 69, 102022.

Contact

Erol Egrioglu
Giresun University
Department of Statistics, Faculty of Arts and Science, Giresun University, 28200, Giresun, Türkiye
erol.egrioglu@giresun.edu.tr

Eren Bas
Giresun University
Department of Statistics, Faculty of Arts and Science, Giresun University, 28200, Giresun, Türkiye
eren.bas@giresun.edu.tr