Granular Computing (2022) 7:411–420
https://doi.org/10.1007/s41066-021-00274-2

ORIGINAL PAPER

Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization

Eren Bas[1] · Erol Egrioglu[1,2] · Emine Kolemen[1]

Received: 25 April 2021 / Accepted: 17 June 2021 / Published online: 22 June 2021

Abstract

Deep artificial neural networks have been popular for time series forecasting literature in recent years. The recurrent neural networks present more suitable architectures for forecasting problems than other deep neural network types. The simplest deep recurrent neural network type is simple recurrent neural networks according to the number of employed parameters. These neural networks can be preferred to solve forecasting problems because of their simple structure if they are trained well. Unfortunately, the training of simple recurrent neural networks is problematic because of exploding or vanishing gradient problems. The contribution of this study is proposing a new training algorithm based on particle swarm optimization. The algorithm does not use gradients so it has not vanished or exploding gradient problem. The performance of the new training algorithm is compared with long short-term memory trained by the Adam algorithm and Pi-Sigma artificial neural network. In the applications, ten-time series are used to compare the performance of the methods. The ten-time series is consisting of daily observations of the Dow-Jones and Nikkei stock exchange opening prices between the years 2014 and 2018. At the end of the analysis processes, the proposed method produces more accurate forecast results than established benchmarks.

Keywords Deep learning · Recurrent neural networks · Forecasting · Particle swarm optimization

1 Introduction

Early forecasting methods were based on the probability theory and they were generally statistical methods. In recent years, machine learning methods and their hybridization with statistical methods have become popular, day by day. Machine learning methods do not need any probabilistic or statistical assumptions. Machine learning methods use nonlinear structures and soft models. Artificial neural networks are an important class of machine learning methods. Artificial neural networks can be classified into two groups as shallow and deep artificial neural networks. Deep artificial neural networks use more parameters than shallow artificial neural networks. Deep artificial neural networks use a parameter sharing approach so they process more data with fewer parameters than shallow neural networks. Deep artificial neural networks produced successful forecast results in the forecast competitions. Especially, the methods based on long short-term recurrent neural networks have top ranks in the competitions. Long short-term memory artificial neural network (LSTM-ANN) was proposed by Hochreiter and Schmidhuber (1997) to solve the vanishing and exploding gradient problem of simple recurrent artificial neural networks. Although the problems of simple recurrent artificial neural networks, the number of parameters in LSTM-ANN dramatically increase because of the various gates. Instead of using the various gates in recurrent neural networks, gradient-free algorithms can be preferred to train simple recurrent neural networks. So, forecasting problems can be solved using fewer number parameters than LSTM-ANN.

Many kinds of recurrent neural networks have been used for time series forecasting in the literature. When the literature is examined, it can be concluded that artificial

✉ Erol Egrioglu
erol.egrioglu@giresun.edu.tr; e.egrioglu@lancaster.ac.uk

[1] Department of Statistics, Faculty of Arts and Science, Giresun University, 28200 Giresun, Turkey

[2] Marketing and Forecasting Research Center Department of Management Science, Management School, Lancaster University, Lancaster, UK

intelligence optimization techniques can improve the forecasting performance of recurrent neural networks. The most used stochastic optimization methods for training recurrent neural networks are genetic algorithm (GA) and particle swarm optimization (PSO). Elman and Jordan-type recurrent neural networks are well-known ANN types in the forecasting literature. LSTM, Convolutional Neural Network (CNN), and Gated Recurrent Units (GRU) have been the most employed recurrent neural networks in recent years.

In an early study, Pham and Karaboga (1999) proposed a training algorithm based on a GA to train Elman and Jordan networks for dynamic system identification and showed that GA-based training produced better results than derivative-based algorithms. Zhang et al. (2013) presented a hybrid learning algorithm that uses the complementary advantages of two global optimization algorithms, including PSO and evolutionary algorithm for the training of an Elman-style neural network to predict past solar radiation and solar radiation from solar energy. The GA, one of the most commonly used artificial intelligence optimization algorithms, has been frequently used in the training of deep artificial neural networks in recent studies.

GA-LSTMs, an LSTM method based on GA, were proposed by Chen et al. (2018) to predict network traffic. Chung and Shin (2018) proposed the GA-LSTM approach to forecast the Korean stock price index. In Qiu et al. (2018), a prediction method using a GA combined with the recurrent neural network was proposed regarding park guidance and short-term empty parking forecasting of the information system. Stajkowski et al. (2020) developed an LSTM technique optimized with GA. Lu et al. (2020) proposed a prediction method that combines the CNN and LSTM artificial neural networks optimized by GA. Gao et al. (2020) presented a model by integrating the GRU model into a GA-based optimizer.

PSO is a very useful artificial intelligence optimization algorithm in solving numerical optimization problems. PSO has been used for the training of recurrent neural networks in recent studies. Xu et al. (2007) proposed a RNN and PSO approach to remove genetic regulatory networks from time-series gene expression data. Ma et al. (2012) presented a model based on an advanced mixed recurrent neural network model and a simpler PSO. Wang et al. (2013) proposed a new hybrid optimization algorithm that uses PSO for simultaneous structure and the parameter learning of Elman-type recurrent neural networks. Egrioglu et al. (2014) used PSO for the training of recurrent multiplicative neuron model artificial neural networks for nonlinear time series forecasting. Moalla et al. (2017) presented a mixed approach that combined LSTM and PSO. Akdeniz et al. (2018) used PSO for the training of the recurrent Pi-Sigma artificial neural network. Peng et al. (2018) proposed an LSTM-ANN model with a differential evolution algorithm for electricity price prediction. Ibrahim and El-Amary (2018) proposed a recurrent neural network trained with PSO. Yao et al. (2018) proposed a PSO-based LSTM. Kim and Cho (2019) proposed a PSO-based CNN-LSTM method. Yuan et al. (2019) introduced a hybrid model of LSTM neural network and Beta distribution function based on PSO for the forecast range of wind power. Shao et al. (2019) proposed a nickel-metal price prediction model based on PSO developed with LSTM-ANN. Qiu et al (2020) proposed a railway load volume forecast model based on PSO-LSTM. Some other artificial intelligence optimization techniques are also employed for the training of recurrent neural networks. Gundu and Simon (2021) proposed an LSTM-ANN based on the optimization of an advanced PSO forecasting of the closing price of the Indian energy exchange.

Forecasting methods which use granular computing are popular in recent literature. Many kinds of fuzzy techniques were proposed in the frame of granular computing. Chen and Hsu (2008), Chen and Wang (2010), Chen et al. (2013), Chen and Phuong (2016), Chen and Jian (2017), Zeng et al. (2019), Chen et al. (2019), Egrioglu et al. (2019), Gupta and Kumar (2019a), Bisht and Kumar (2019), Bas et al. (2019), Bas et al. (2020), Egrioglu et al. (2021), Gupta and Kumar (2019b), Chang and Yu (2019) studies use granular computing for forecasting purpose. Deep learning is an effective tool for forecasting. Fan et al. (2019), Chen et. al. (2020), Wu et al. (2021) proposed new forecasting methods based on granular computing and deep learning. Deep learning can be used as an effective tool in granular computing.

It is well known that a simple recurrent neural network will be suffered from vanishing and exploding gradient problems if a training algorithm based on gradients is employed. GRU and LSTM use gates for avoiding vanishing and exploding gradient problems but using the gates dramatically increase the number of weights in the network. The motivation of this study is to propose a new gradient-free training algorithm for the simple recurrent neural network so the network will not need for using gates like in GRU and LSTM.

In this study, a new gradient-free algorithm based on a modified particle swarm optimization method is proposed for the training of the simple deep recurrent neural network to forecast single-variable time series. The contribution of this study is proposing a new training algorithm for the simple recurrent neural network. The proposed method presents an effective modeling tool for granular computing methods. In the second section of the paper, the proposed training algorithm is introduced. In the third section, application results are given. The last section is about conclusions and discussions.

2 A new training algorithm for simple recurrent neural network

The simple recurrent neural networks are suffered from exploding and vanishing gradient problems in the training process. These problems are solved using gates in LSTM and GRU but these networks need more parameters than simple recurrent neural networks. If the simple recurrent neural network can be trained by derivate-free algorithms, the vanishing and exploding gradient problems will be handled automatically. Moreover, the performance of the simple recurrent neural networks can be better than LSTM and GRU because of needing a fewer number of parameters.

In this study, a new training algorithm is proposed based on modified particle swarm optimization. The proposed training algorithm does not need derivatives of any objective function. So, the proposed method does not have a vanishing or exploding gradient problem. The proposed training algorithm has a re-starting strategy, an efficient early stopping rule. Moreover, social and cognitive coefficients are linearly changed on iterations for the increasing convergence rate of the algorithm. Similarly, the inertia weight is linearly increased using an iterative formula in the iterations. The velocities are bounded using a *vmaps* parameter.

The new training algorithm is given step by step as follows:

Step 1. The parameters of the PSO algorithm are determined. These parameters are listed below:

$c_1^{initial}$: The starting value of the cognitive coefficient.
c_1^{final} : The ending value of the cognitive coefficient.
$c_2^{initial}$: The starting value of the social coefficient.
c_2^{final} : The ending value of the social coefficient.
$w^{initial}$: The starting value for inertia weight.
w^{final} : The ending value for inertia weight.
vmaps: The bound value for the velocities.
*limit*1 : The limit value for the re-starting strategy.
*limit*2 : The limit value for the early stopping rule.
maxitr : The maximum number of iterations.
pn: The number of particles.

The counters are initialized. The re-starting strategy counter and early stopping counter are taken as zero ($rsc = 0$, $esc = 0$).

Step 2. The initial positions and velocities are randomly generated.

The positions of the PSO are weights and biases of simple deep recurrent neural networks. The outputs of a simple deep recurrent neural network with one hidden layer are calculated with the Eqs. (1–2):

$$h_t = f(x_t S + h_{t-1} W + b_1), \tag{1}$$

$$\hat{x}_t = f\left(h_t V + b_y\right). \tag{2}$$

The total number of weights and biases are $(p + h + 2)h + 1$ because the dimensions of weights and biases are $S : p \times h$, $W : h \times h$, $b_1 : 1xh$, $V : h \times 1$ and $b_y : 1 \times 1)$.

The weights and biases are generated from a uniform distribution with 0 and 1 parameters. All velocities are generated from a uniform distribution with $-vmaps$ and $vmaps$ parameters. $P_{i,j}^{(t)}$ is the jth position of the ith particle at the tth iteration. $V_{i,j}^{(t)}$ is the jth velocity of ith particle at the tth iteration:

$$P_{i,j}^{(0)} \sim Uniform(0, 1), \tag{3}$$

$$V_{i,j}^{(0)} \sim Uniform(-vmaps, vmaps). \tag{4}$$

Step 3. For each particle, the fitness function values are calculated. The fitness function values are calculated based on the difference between the forecast and its corresponding observations using the following formula which is mean square error (MSE) given in Eq. (5):

$$MSE_j = \frac{1}{n}\sum_{t=1}^{n}(x_t - \hat{x}_t)^2, \quad j = 1, 2, \ldots, pn. \tag{5}$$

Step 4. Pbest (the memory for each particle) and gbest (the memory for the swarm) are constituted.

The Pbest is a matrix and its rows are present the best position of the particles at the current iteration. The gbest is a vector and it presents the best positions of the swarm. Moreover, the gbest is also a row of the Pbest.

Step 5. The values of the cognitive, social coefficients, and inertia weight parameters are calculated by the Eqs. (6–8):

$$w^{(t)} = (w^{initial} - w^{final})\frac{maxitr - t}{maxitr} + w^{final}, \tag{6}$$

$$c_1^{(t)} = (c_1^{final} - c_1^{initial})\frac{t}{maxitr} + c_1^{initial}, \tag{7}$$

$$c_2^{(t)} = (c_2^{initial} - c_2^{final})\frac{maxitr - t}{maxitr} + c_2^{final}. \tag{8}$$

Step 6. The new velocities and positions are calculated using the Eqs. (9–11). The r_1 and r_2 are real random numbers between 0 and 1:

$$V_{i,j}^{(t)} = w^{(t)}V_{i,j}^{(t-1)} + c_1^{(t)}r_1\left(Pbest_{i,j}^{(t)} - P_{i,j}^{(t)}\right) + c_2^{(t)}r_2\left(gbest_j^{(t)} - P_{i,j}^{(t)}\right), \tag{9}$$

$$V_{i,j}^{(t)} = \min\left(vmaps, \max\left(-vmaps, V_{i,j}^{(t)}\right)\right), \tag{10}$$

$$P_{i,j}^{(t)} = P_{i,j}^{(t-1)} + V_{i,j}^{(t)}. \tag{11}$$

Step 7. Pbest and gbest are updated.

Step 8. The re-starting strategy counter ($rsc = rsc + 1$) is increased and its value is checked. If the $rsc > limit1$ then all positions and velocities are re-generated using (3) and (4) and the rsc is taken as zero. Pbest and gbest are never changed in this step.

Step 9. The early stopping rule is checked. The esc counter is increased depending on the following condition:

$$esc = \begin{cases} esc + 1, if \dfrac{MSEbest^{(t)} - MSEbest^{(t-1)}}{MSEbest^{(t)}} < 10^{-3} \\ 0, otherwise \end{cases}. \tag{12}$$

The early stopping rule is $esc > limit2$. If the rule is satisfied, the algorithm is stopped otherwise go to Step 5.

3 Applications

In the application section, Dow Jones and Nikkei stock exchange index data sets are used. The data sets were downloaded from Yahoo Finance Website (https://finance.yahoo.com). Data sets are constituted from five daily opening prices for the years 2014–2018. Time series are solved using LSTM-ANN, Pi-Sigma artificial neural network (PSGM), and simple recurrent neural network (SRNN) artificial neural network method. The number of inputs is changed from 1 to 5 with an increment of one in all artificial neural network applications.

In the application of LSTM and SRNN, the number of hidden layer units is changed from 1 to 5 with an increment of one. Each method is applied 30 times using random initial weights. In the application, the time series is divided into three parts as training, validation and test data. Training data was used to train the artificial neural network and validation data were used to select the best configuration or parameter tuning in the architecture of the artificial neural network. The test set is used to compare the performance of the different artificial neural network methods. The length of the test and validation data sets is 20. The training, validation and test sets are chosen as block structures as in Fig. 1.

First, Dow-Jones time series data sets are solved using LSTM, PSGM and SRNN. The number of inputs and number of hidden layer neurons is determined according to validation data performance for all methods. Each method is applied 30 times for the best parameter configuration using different initial weights and the test set forecast performance is calculated using the Root mean square error (RMSE) criterion. RMSE criterion is calculated using the square root of (5) Equation. The descriptive statistics (mean, standard deviation, minimum and maximum) of RMSE values for 30 repeats are calculated and given in Table 1. Mean statistics present the most probable values of the RMSE criterion for the methods. The standard deviations show the variation of repeated solutions. The minimum statistics present the best scenario while the maximum statistics present the worst scenario. If a method is better than the others, it is expected that the method has smaller mentioned descriptive statistics. The standard deviation cannot be commented on without taking into consideration of other statistics.

When Table 1 is examined, the SRNN with the proposed learning algorithm is better than others for the years 2014, 2015 and 2018 for Dow-Jones data sets according to mean statistics. Moreover, the SRNN is the second-best method for the years 2016 and 2017. The SRNN with the proposed learning algorithm is better than other methods for all years except the year 2014 for Dow-Jones data sets according to standard deviation statistics. The SRNN is the best method for only the year 2015 according to minimum statistics. The SRNN is the best method for all years except the year 2017 according to maximum statistics for Dow-Jones data sets.

The best parameter configurations of the applied ANN methods are given in Table 2. It is observed that the best number of inputs is 5 in many cases. Moreover, the best number of hidden layer units is generally selected as 5. The number of the hidden layers is selected as 2 for the SRNN in four years of the Dow Jones data set.

The success percentages of LSTM, PSGM, and SRNN methods are given in Fig. 2. Success means that the method has the best maximum, minimum, mean and standard deviation statistics. For example, if SRNN has an %80 success percentage for maximum statistics, this means that SRNN has smaller maximum statistics than others for %80 of all years. According to Fig. 2, SRNN has %60, %80, %20, and %80 success percentages for mean, standard deviation, minimum and maximum statistics, respectively.

When Table 3 is examined, the SRNN with the proposed learning algorithm is better than other methods for the years 2014, 2016 and 2017 for the Nikkei data set according to mean statistics. Moreover, the SRNN is the second-best method for the years 2015 and 2018 for the Nikkei data set. The SRNN with the proposed learning algorithm is better than other methods for all years except the year 2017 for the Nikkei data set according to standard deviation statistics. The SRNN is the best method for only the year 2017 for Nikkei data set according to minimum statistics. The SRNN is the best method in all years for the Nikkei data set according to maximum statistics.

The best parameter configurations of the applied ANN methods are given in Table 4. It is observed that the best number of inputs is 5 in many cases. Moreover, the best

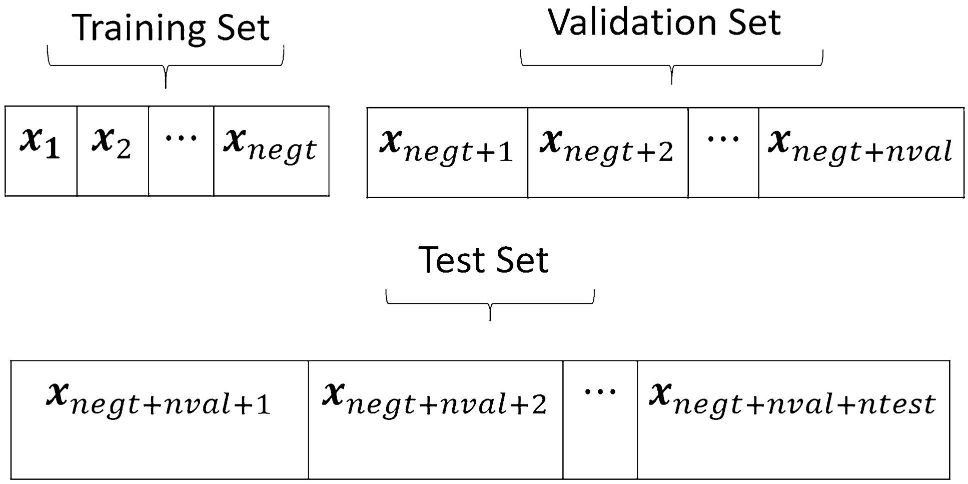

Fig. 1 Partition of time series as training, validation, and test sets

Table 1 Results for Dow-Jones data set

Year	Methods	Mean	SD	Min	Max
2014	LSTM Hochreiter and Schmidhuber (1997)	159.94	4.70	148.16	168.91
	PSGM Shin and Gosh (1991)	161.56	0.01	161.54	161.61
	SRNN	158.57	0.77	156.83	160.14
2015	LSTM Hochreiter and Schmidhuber (1997)	319.94	40.69	244.03	413.62
	PSGM Shin and Gosh (1991)	198.13	2.63	191.80	201.40
	SRNN	192.57	0.49	191.46	194.79
2016	LSTM Hochreiter and Schmidhuber (1997)	111.56	6.51	92.97	120.53
	PSGM Shin and Gosh (1991)	92.27	2.11	87.35	96.17
	SRNN	93.39	0.01	93.34	93.42
2017	LSTM Hochreiter and Schmidhuber (1997)	107.37	2.51	102.90	113.45
	PSGM Shin and Gosh (1991)	100.03	1.72	96.02	103.52
	SRNN	104.17	0.77	102.91	107.91
2018	LSTM Hochreiter and Schmidhuber (1997)	454.50	9.48	425.71	467.42
	PSGM Shin and Gosh (1991)	453.05	19.75	427.29	506.89
	SRNN	438.38	0.91	437.15	441.94

SD standard deviation, *Min* minimum, *Max* maximum

number of hidden layer units is generally selected as 4. The number of the hidden layers is selected as 2 for the SRNN in all years of the Nikkei data set.

The success percentages of LSTM, PSGM and SRNN are given in Fig. 3 for Nikkei data sets. The meaning of success percentage in Fig. 3 is the same in Fig. 2. According to Fig. 3, SRNN has %60, %80, %20 and %100 success percentages for mean, standard deviation, minimum and maximum statistics, respectively.

Table 2 The best architectures for Dow-Jones Set

Years	Methods	Number of Inputs	Number of Hidden Layer Nodes	Number of Hidden Layers
2014	LSTM Hochreiter and Schmidhuber (1997)	4	3	1
	PSGM Shin and Gosh (1991)	3	1	-
	SRNN	2	5	1
2015	LSTM Hochreiter and Schmidhuber (1997)	5	4	1
	PSGM Shin and Gosh (1991)	2	5	-
	SRNN	4	3	2
2016	LSTM Hochreiter and Schmidhuber (1997)	5	5	1
	PSGM Shin and Gosh (1991)	5	5	-
	SRNN	5	5	2
2017	LSTM Hochreiter and Schmidhuber (1997)	5	4	1
	PSGM Shin and Gosh (1991)	5	3	-
	SRNN	4	2	2
2018	LSTM Hochreiter and Schmidhuber (1997)	5	1	1
	PSGM Shin and Gosh (1991)	5	5	-
	SRNN	5	2	2

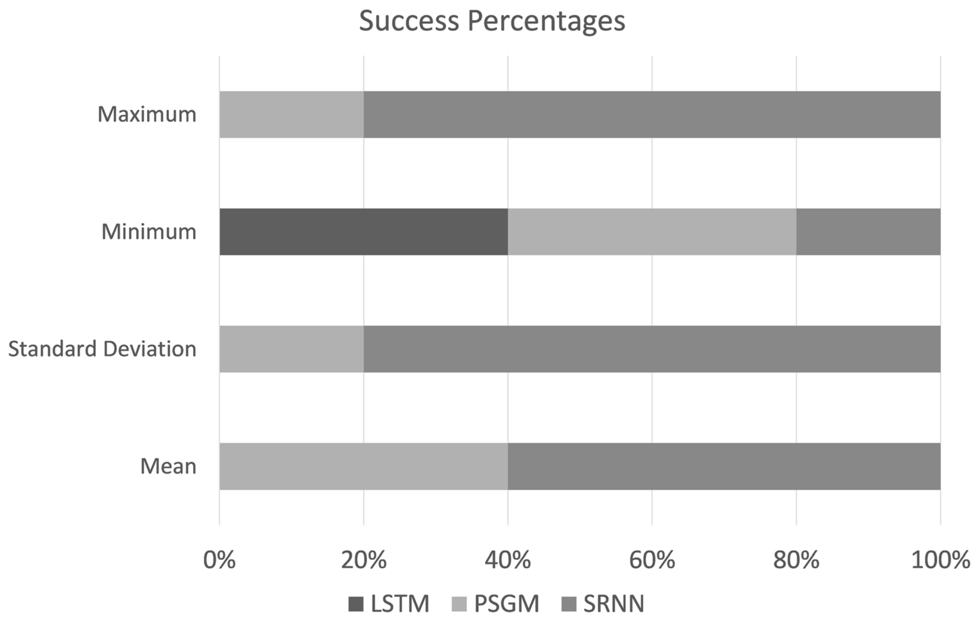

Fig. 2 Comparison of the ANNs for Dow-Jones data sets according to descriptive statistics

Table 3 Results for Nikkei Data Set

Year	Methods	Mean	SD	Min	Max
2014	LSTM	225.13	7.14	211.81	293.54
	Hochreiter and Schmidhuber (1997)				
	PSGM	226.29	6.31	218.09	244.67
	Shin and Gosh (1991)				
	SRNN	225.06	0.44	224.00	226.19
2015	LSTM	271.45	29.87	225.04	320.12
	Hochreiter and Schmidhuber (1997)				
	PSGM	219.62	1.88	215.42	224.79
	Shin and Gosh (1991)				
	SRNN	220.03	0.67	218.08	221.67
2016	LSTM	187.08	5.87	176.46	199.06
	Hochreiter and Schmidhuber (1997)				
	PSGM	149.59	2.14	144.51	153.67
	Shin and Gosh (1991)				
	SRNN	147.60	0.12	147.56	148.25
2017	LSTM	190.63	15.76	165.53	236.59
	Hochreiter and Schmidhuber (1997))				
	PSGM	157.50	3.17	148.89	165.91
	SRNN	149.59	4.90	142.09	161.80
2018	LSTM	342.50	15.42	312.71	366.38
	Hochreiter and Schmidhuber (1997)				
	PSGM	328.65	2.96	319.92	333.42
	SRNN	331.15	0.41	330.31	331.93

SD standard deviation, *Min* minimum, *Max* maximum

4 Conclusion and discussions

Deep artificial neural networks have been used to solve forecasting problems in recent years. Recurrent deep neural networks are the most preferred type of deep neural network. Simple deep recurrent neural networks are suffered from vanishing and exploding gradient problems. LSTM and GRU deep recurrent neural networks managed to solve mentioned problems using gates. Using gates means that the number of parameters should be dramatically increased.

The contribution of this paper is proposing a learning algorithm based on particle swarm optimization with some effective modifications. The proposed learning algorithm does not have vanishing or exploding gradient problems because it does not need gradients of the objective function. The performance of a deep simple recurrent neural network with the proposed learning algorithm is compared with the performance of LSTM and PSGM artificial neural networks for the stock exchange data sets. LSTM is trained by the gradient-based algorithm and this provides to compare gradient-based algorithm with PSO-based algorithm. It is shown that the forecasting performance of the proposed method is better than the others. According to mean descriptive statistics, the success rate of the proposed method is %60 for both stock exchange data sets. Moreover, the variation of the results for the proposed method is the minimum among the applied methods. The proposed method is not better than others according to minimum statistics. The best results of the proposed method are not better than the others but the results are very close to other results. It can be said that the proposed method can be used to forecast stock exchange data sets.

The limitation of the proposed method can be seen for the large networks. The PSO algorithms can have problems with a big number of hidden layers because of large-scale optimization problems. This problem can be seen for image processing problems but it will not be a problem for forecasting problems. Because forecasting problems do not need too many hidden layers.

In future studies, the architecture of the simple deep recurrent neural networks can be strengthened using different artificial neuron models and hybridization of the classical forecasting method. The obtained new deep recurrent neural networks can be trained with a simple modification of the proposed method.

Table 4 The best architectures for Nikkei Data Set

Years	Methods	Number of Inputs	Number of Hidden Layer Nodes	Number of Hidden Layers
2014	LSTM Hochreiter and Schmidhuber (1997)	2	5	1
	PSGM Shin and Gosh (1991)	4	4	-
	SRNN	3	2	2
2015	LSTM Hochreiter and Schmidhuber (1997)	3	3	1
	PSGM Shin and Gosh (1991)	3	5	-
	SRNN	5	2	2
2016	LSTM Hochreiter and Schmidhuber (1997)	1	2	1
	PSGM Shin and Gosh (1991)	5	5	-
	SRNN	4	5	2
2017	LSTM Hochreiter and Schmidhuber (1997)	5	5	1
	PSGM Shin and Gosh (1991)	5	2	-
	SRNN	4	1	2
2018	LSTM Hochreiter and Schmidhuber (1997)	2	5	1
	PSGM Shin and Gosh (1991)	2	5	-
	SRNN	1	1	2

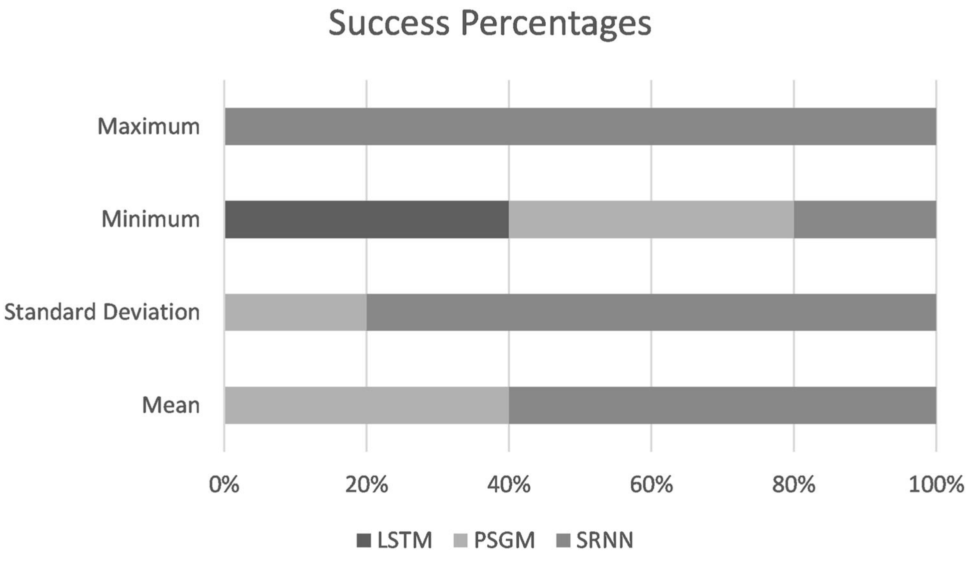

Fig. 3 Comparison of the ANNs for Nikkei data sets according to descriptive statistics

Declarations

Conflict of interest The author declares that there is no conflict of interest toward the publication of this manuscript.

References

Akdeniz E, Egrioglu E, Bas E, Yolcu U (2018) An ARMA type Pi-Sigma artificial neural network for nonlinear time series forecasting. J Artif Intell Soft Comput 8:121–132

Bas E, Egrioglu E, Yolcu U, Grosan C (2019) Type 1 fuzzy function approach based on ridge regression for forecasting. Granul Comput 4:629–637

Bas E, Yolcu U, Egrioglu E (2020) Intuitionistic fuzzy time series functions approach for time series forecasting. Granul Comput. https://doi.org/10.1007/s41066-020-00220-8

Bisht K, Kumar S (2019) Hesitant fuzzy set based computational method for financial time series forecasting. Granul Comput 4:655–669

Chang JR, Yu PY (2019) Weighted-fuzzy-relations time series for forecasting information technology maintenance cost. Granul Comput 4:687–697

Chen SM, Hsu CC (2008) A new approach for handling forecasting problems using high-order fuzzy time series. Intell Autom Soft Comput 14(1):29–43

Chen SM, Wang N (2010) Fuzzy forecasting based on fuzzy-trend logical relationship groups. IEEE Trans Syst Man Cybernetics Part B 40(5):1343–1358

Chen SM, Manalu GMT, Pan J, Liu H (2013) Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and particle swarm optimization techniques. IEEE Trans Cybernetics 43(3):1102–1117

Chen SM, Phuong BDH (2016) Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors. Knowl Based Syst 118:204–216

Chen SM, Jian WS (2017) Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups, similarity measures and PSO techniques. Inf Sci 391:65–79

Chen J, Xing H, Yang H, Xu L (2018) Network traffic prediction based on LSTM networks with genetic algorithm. In: International Conference on Signal and Information Processing, Networking and Computers, pp 411–419

Chen SM, Zou XY, Gunawan GC (2019) Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques. Inf Sci 500:127–139

Chen J, Yuan W, Cao J, Lv H (2020) Traffic-flow prediction via granular computing and stacked autoencoder. Granul Comput 5:449–459

Chung H, Shin KS (2018) Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability 10(10):3765

Egrioglu E, Yolcu U, Aladag CH, Bas E (2014) Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting. Neural Process Lett 41(2):249–258

Egrioglu E, Yolcu U, Bas E (2019) Intuitionistic high-order fuzzy time series forecasting method based on pi-sigma artificial neural networks trained by artificial bee colony. Granul Comput 4:639–654

Egrioglu E, Fildes R, Bas E (2021) Recurrent fuzzy time series functions approach for forecasting. Granul Comput. https://doi.org/10.1007/s41066-021-00257-3

Fan MH, Chen MY, Liao EC (2019) A deep learning approach for financial market prediction: utilization of Google trends and keywords. Granul Comput 6:207–216

Gao MY, Zhang N, Shen SL, Zhou A (2020) Real-time dynamic earth-pressure regulation model for shield tunneling by integrating GRU Deep learning method with GA optimization. IEEE Access 8:64310–64323

Gundu V, Simon SP (2021) PSO–LSTM for short-term forecast of heterogeneous time series electricity price signals. J Ambient Intell Hum Comput 12:2375–2385

Gupta KK, Kumar S (2019a) A novel high-order fuzzy time series forecasting method based on probabilistic fuzzy sets. Granul Comput 4:699–713

Gupta KK, Kumar S (2019b) Hesitant probabilistic fuzzy set based time series forecasting method. Granul Comput 4:739–758

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780

Ibrahim AM, El-Amary NH (2018) Particle swarm optimization trained recurrent neural network for voltage instability prediction. J Electr Syst Inf Technol 5(2):216–228

Kim TY, Cho SB (2019) Particle swarm optimization-based CNN-LSTM networks for forecasting energy consumption. In: 2019 IEEE Congr Evol Comput CEC 2019—Proc, pp 1510–1516

Lu W, Rui H, Liang C, Jiang L, Zhao S, Li K (2020) A method based on GA-CNN-LSTM for daily tourist flow prediction at scenic spots. Entropy 22(3):01–18

Ma L, Ge Y, Cao X (2012) Superheated steam temperature control based on improved recurrent neural network and simplified PSO algorithm. Appl Mech Mater 128:1065–1069

Moalla H, Elloumi W, Alimi AM (2017) H-PSO-LSTM: Hybrid LSTM trained by PSO for online handwriting identification. In: International Conference on Neural Information Processing, pp 41–50

Peng L, Liu S, Liu R, Wang L (2018) Effective long short-term memory with differential evolution algorithm for electricity price prediction. Energy 162:1301–1314

Pham DT, Karaboga D (1999) Training Elman and Jordan networks for system identification using genetic algorithms. Artif Intell Eng 13:107–117

Qiu J, Tian J, Chen H, Lu X (2018) Prediction method of parking space based on genetic algorithm and RNN. In: Pacific Rim Conference on Multimedia, pp 865–876

Qiu YY, Zhang Q, Lei M (2020) Forecasting the railway freight volume in China based on combined PSO-LSTM model. J Phys 1651(1):012029

Shao B, Li M, Zhao Y, Bian G (2019) Nickel price forecast based on the LSTM neural network optimized by the improved PSO algorithm. Mathematical Problems in Engineering 2019.

Shin Y, Gosh J (1991) The Pi-Sigma network: an efficient higher order neural network for pattern classification and function approximation. In: Proceedings of the International Joint Conference on Neural Networks, Seattle, pp 13–18

Stajkowski S, Kumar D, Samui P, Bonakdari H, Gharabaghi B (2020) Genetic-algorithm-optimized sequential model for water temperature prediction. Sustainability 12(13):5374

Wang X, Ma L, Wang B, Wang T (2013) A hybrid optimization-based recurrent neural network for real-time data prediction. Neurocomputing 120:547–559

Wu F, Yan S, Smith JS, Zhang B (2021) Deep multiple classifier fusion for traffic scene recognition. Granular Computing 6:217–228

Xu R, Wunsch DC II, Frank R (2007) Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE/ACM Trans Comput Biol Bioinf 4(4):681–692

Yao Y, Han L, Wang J (2018) LSTM-PSO: Long Short-Term Memory ship motion prediction based on particle swarm optimization. In: 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), pp 1–5

Yuan X, Chen C, Jiang M, Yuan Y (2019) Prediction interval of wind power using parameter optimized Beta distribution-based LSTM model. Applied Soft Computing 82:105550

Zhang N, Behera PK, Williams C (2013) Solar radiation prediction based on particle swarm optimization and evolutionary algorithm using recurrent neural networks. In: 2013 IEEE International Systems Conference (SysCon), pp 280–286.

Zeng S, Chen SM, Teng MO (2019) Fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm and artificial bee colony algorithm. Inf Sci 484:350–366

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.