A Time Series Forecasting Method Using DBN and Adam Optimization

Takashi Kuremoto[1(✉)], Masafumi Furuya[2], Shingo Mabu[2], and Kunikazu Kobayashi[3]

[1] Nippon Institute of Technology, Saitama 345-8501, Japan
kuremoto.takashi@nit.ac.jp
[2] Yamaguchi University, Yamaguchi 755-8611, Japan
[3] Aichi Prefectural University, Aich 480-1342, Japan

Abstract. Deep Belief Net (DBN) was applied to the field of time series forecasting in our early works. In this paper, we propose to adopt Adaptive Moment Estimation (Adam) optimization method to the fine-tuning process of DBN instead of the conventional Error Back-Propagation (BP) method. Meta parameters, such as the number of layers of Restricted Boltzmann Machine (RBM), the number of units in each layer, the learning rate, are optimized by Random Search (RS) or Particle Swarm Optimization (PSO). Comparison experiments showed the priority of the proposed method in both cases of a benchmark dataset CATS which is an artificial time series data used in competitions for long-term forecasting, and Lorenz chaos for short-term forecasting in the sense not only prediction precision but also learning performance.

Keywords: Time series forecasting · Deep learning · Deep Belief Net · Error Back-Propagation · Adam learning optimization

1 Introduction

The study of time series forecasting benefits to many fields, such as the prediction of electricity consumption, stock prices, population, amount of rainfall, and so on. Generally, there are two kinds of theories of time series forecasting: linear models, and non-linear models. The former includes Auto-Regressive (AR), Moving Average (MA), and a combination of them ARIMA. For the effect to financial and economic fields, the proposer of Auto-Regressive Conditional Heteroskedasticity (ARCH) [1], R. Engle was awarded by Nobel Memorial Prize in Economic Sciences in 2003. The later, non-linear methods, usually utilize artificial neural networks such as Multi-Layered Perceptron (MLP), Radial Basis Function Net (RBFN), and deep learning methods [2–6].

In our previous works [3–6], Deep Belief Net (DBN) [7], a well-known deep learning model, was firstly applied to the time series forecasting. And a hybrid model with DBN and ARIMA was also proposed to improve the prediction precision [8, 9]. The hybrid model was a combination of Artificial Neural Networks (ANN) and linear models which is inspired by the theory of G.P. Zhang [10].

Y. Kambayashi et al. (Eds.): AICON 2022, LNICST 477, pp. 95–106, 2023.
https://doi.org/10.1007/978-3-031-29126-5_8

Generally, error Back-Propagation (BP) [11], is used as the training method (optimization) of ANNs. Meanwhile, recently, Adaptive Moment Estimation (Adam) [12], an advanced gradient descent algorithm of BP, is widely utilized in the training of deep neural networks. The concept of Adam is to adopt the first-order momentum, i.e., the past gradient, and the second-order momentum, i.e., the absolute gradient, into the update process of parameters. By considering the average gradient, Adam overcomes the local extremum problem in the high dimensional parameter space, and tackles non-stationary objectives.

In this study, Adam is firstly adopted to the fine-tuning process of DBN instead of the conventional BP optimization method. Benchmark dataset CATS [13, 14], an artificial time series data utilized in time series forecasting competition, and a chaotic time series given by Lorenz chaos which is a famous chaotic theory for its butterfly attractor, were used in the comparison experiment. In both experiments, DBN with Adam showed its priority to the conventional BP method in the fine-tuning process.

2 DBN for Time Series Forecasting

The original Deep Belief Net [7] was proposed for dimension reduction and image classification. It is a kind of deep auto-encoder which composed by multiple Restricted Boltzmann Machines (RBMs). For time series forecasting, the part of decoder of DBN is replaced by a feedforward ANN, Multi-Layered Perceptron (MLP) in our previous works [5, 6]. The structure of the DBN is shown in Fig. 1.

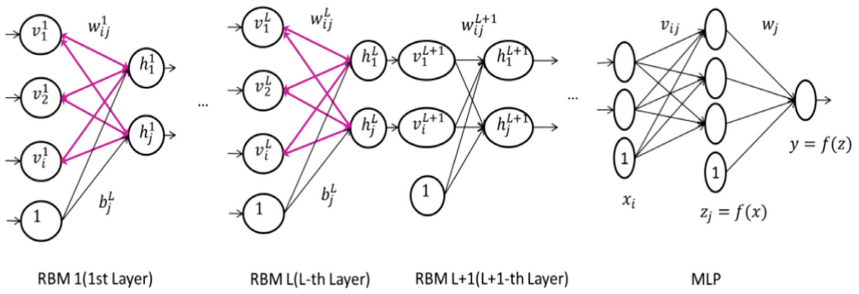


Fig. 1. A structure of a DBN composed by RBMs and MLP [6, 8, 9].

2.1 RBM and Its Learning Rule

Restricted Boltzmann Machine (RBM) [7] is a kind of Hopfield neural network but with 2 layers. Units in the visible layer connect to the units in the hidden layer with different weights. The outputs of units v_i, h_j are binary, i.e., 0 or 1, except the initial value of visible units is given by the input data. The probabilities of 1 of a visible unit and a hidden unit are according to the following.

$$p(h_j = 1|v) = \frac{1}{1 + exp(-b_j - \sum_{i=1}^{n} w_{ji} v_i)} \tag{1}$$

$$p(v_i = 1|h) = \frac{1}{1 + exp(-b_i - \sum_{j=1}^{m} w_{ij} h_j)} \tag{2}$$

Here b_i, b_j, w_{ij} are the biases and the weights of units. The learning rules of RBM are given as follows.

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_{\text{data}} - < v_i h_j >_{\text{model}}) \tag{3}$$

$$\Delta b_i = \varepsilon(< v_i > - < \tilde{v}_i >) \tag{4}$$

$$\Delta b_j = \varepsilon(< h_j > - < \tilde{h}_j >) \tag{5}$$

where $0 < \varepsilon < 1$ is a learning rate, $p_{ij} =< v_i h_j >_{\text{data}}$, $p'_{ij} < v_i h_j >_{\text{model}}$, $< v_i >$, $< h_j >$ indicate the first Gibbs sampling ($k = 0$) and $< \tilde{v}_i >$, $< \tilde{h}_j >$ are the expectations after the kth Gibbs sampling, and it also works when $k = 1$.

2.2 MLP and Its Learning Rule

A feedforward neural network Multi-Layered Perceptron (MLP) [11] inspired the second Artificial Intelligence (AI) boom in 1980s (see Fig. 1). The input x_i ($i = 1, 2, ...n$) is fired by the unit z_j with connection weight v_{ji} in a hidden layer by an activation function, and also the output $y = f(z)$ is given by the function and connection weights w_j ($j = 1, 2, ...K$) as follows.

$$y = f(z) = \frac{1}{1 + exp(-\sum_{j=1}^{K+1} w_j z_j)} \tag{6}$$

$$f(z_j) = \frac{1}{1 + exp(-\sum_{i=1}^{n+1} v_{ji} x_i)} \tag{7}$$

where biases $x_{n+1} = 1.0$, $z_{K+1} = 1.0$.

Error Back-Propagation (BP) [11] serves as the learning rule of MLP as follows.

$$\Delta w_j = -\varepsilon(y - \tilde{y}) y(1 - y) z_j \tag{8}$$

$$\Delta v_{ji} = -\varepsilon(y - \tilde{y}) y(1 - y) w_j z_j (1 - z_j) x_i \tag{9}$$

where $0 < \varepsilon < 1$ is the learning rate, $\tilde{y}$ is the teacher signal, i.e., the value of training sample.

Meanwhile, because the BP method is sensitive to the noise and easy to convergence to the local minimum, it is modified by Adam (adaptive moment) proposed by Kingma and Ba in 2014 [12].

$$\Delta\theta_t = \frac{\widehat{m}_t}{\varepsilon + \sqrt{\hat{v}_t}} \tag{10}$$

$$\widehat{m}_t = \frac{\beta_1^t m_{t-1}}{1 - \beta_1^t} + g_t \tag{11}$$

$$\hat{v}_t = \frac{\beta_2^t v_{t-1}}{1 - \beta_2^t} + {g_t}^2 \tag{12}$$

$$g_t = \nabla_\theta E_t(\theta_{t-1}) \tag{13}$$

where $\theta = (v_{ji}, w_j)$ is the parameter to be modified, $0 < \varepsilon, \beta_1^t, \beta_2^t < 1$ are hyper parameters and given by empirical scalar values. $E_t(\theta_{t-1})$ is the loss function, e.g., the mean squared error between the output of the network and the teacher signal.

Although Adam is the major optimization method of deep learning recently, it is not adopted to the fine-tuning of DBN for time series forecasting as we know. In study, it is proposed that Eqs. (10–13) replace Eqs. (6–9) for Eq. (3–5), e.g., the learning rules in fine-tuning process of DBN are given by Adam instead of the BP method.

2.3 Meta Parameter Optimization

To design the structure of the ANNs, the evolutional algorithm of swarm intelligence, i.e., the Particle Swarm Optimization (PSO) or the heuristic algorithm Random Search (RS) [15], are more effective than the empirical methods such as grid search algorithm [16]. In this study, PSO and RS are adopted to optimize the meta parameters of DBN, i.e., the number of RBMs, the number of units in each RBM, the number of units of MLP, the learning rate of RBMs, and the learning rates. Detail algorithms can be found in [16], and they are omitted here.

3 Experiments and Analysis

To investigate the performance of DBN with Adam optimization algorithm, comparison experiments of time series forecasting were carried out. A benchmark dataset CATS [13, 14] (see Fig. 2), which is an artificial time series dataset utilized in time series forecasting competition, and a chaotic time series of Lorenz chaos (see Fig. 6), were used in the experiments.

3.1 Benchmark CATS

CATS time series data is an artificial benchmark data for forecasting competition with ANN methods [13, 14]. This artificial time series is given with 5,000 data, among which 100 are missed (hidden by competition the organizers) (see Fig. 2). The missed data exist in 5 blocks:

– elements 981 to 1,000
– elements 1,981 to 2,000
– elements 2,981 to 3,000
– elements 3,981 to 4,000
– elements 4,981 to 5,000

The mean square error E_1 is used as the prediction precision in the competition, and it is computed by the 100 missing data and their predicted values as following:

$$E_1 = \{\sum_{t=981}^{1000}(y_t - \bar{y}_t)^2 + \sum_{t=1981}^{2000}(y_t - \bar{y}_t)^2 + \sum_{t=2981}^{3000}(y_t - \bar{y}_t)^2 + \sum_{t=3981}^{4000}(y_t - \bar{y}_t)^2 + \sum_{t=4981}^{5000}(y_t - \bar{y}_t)^2)\}/100 \tag{14}$$

where $\bar{y}_t$ is the long-term prediction result of the missed data.

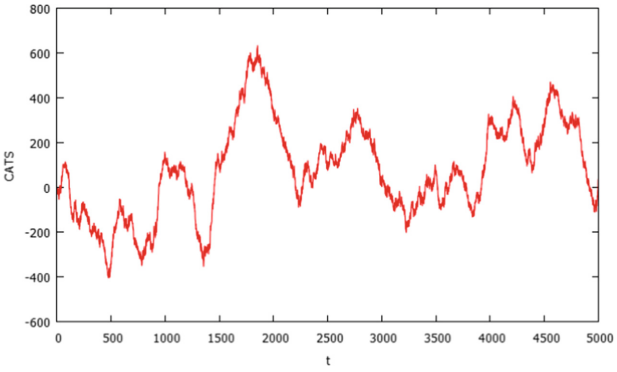


Fig. 2. A benchmark dataset CATS [13, 14].

3.2 Results and Analysis of CATS Forecasting

The meta parameter space searched by heuristic algorithms, i.e., Particle Swarm Optimization (PSO) and Random Search (RS) has 5 dimensions: the number of RBMs in DBN, the number of units of each RBM, the number of units in hidden layer of MLP, the learning rate of RBMs, the learning rate of MLP. The exploration ranges of these meta parameters are shown in Table 1.

The iteration of exploration of PSO and RS was set by convergence of evaluation functions or limitations of 2,000 in pre-training (RBM), and 10,000 in the fine-tuning (MLP). Additionally, the exploration finished when the forecasting error (mean squared error between the real data and the output of DBN) of validation data increased than the last time.

Table 1. Meta parameter ranges of exploration by PSO and RS.

Dimension	Range
The number of RBMs	0–3
The number of units in each RBM	2–20
The number of units in hidden layer of MLP	2–20
The learning rate of RBMs (pre-training)	10^{-1}–10^{-5}
The learning rate of MLP (fine-tuning)	10^{-1}–10^{-5}

Table 2. The comparison of long-term prediction precision by E_1 measurement between different methods using CATS data [13, 14].

Method	E_1
DBN (Adam + RS) (proposed)	*134.04*
DBN (Adam + PSO) (proposed)	*148.24*
DBN (BP + RS) [5]	155.53
DBN (BP + PSO) [5]	155.65
DBN(SGA) (reinforcement learning)[6]	170
DBN(BP) + ARIMA [8] [9]	244
DBN(BP)[6]	257
Kalman Smoother (The best of IJCNN '04) [14]	408
DBN[3] [4] (2 RBMs)	1215
MLP[2]	1245
A hierarchical Bayesian Learning Scheme for Autoregressive Neural Networks (The worst of IJCNN '04) [14]	1247
ARIMA[2]	1715
ARIMA + MLP(BP)[8] [9]	2153
ARIMA + DBN (BP)[8] [9]	2266

The forecasting precisions of different ANN and hybrid methods are shown in Table 2. It can be confirmed that the proposed methods, DBN using Adam fine-tuning algorithm with RS or PSO, ranked on the top of all methods. The learning curves of the proposed method (Adam adopted) and the conventional method (BP) are shown in Fig. 3 (the case of the 1st block of CATS). The convergence of loss (MSE) in Adam showed faster and smaller than the case of BP in both PSO and RS algorithms.

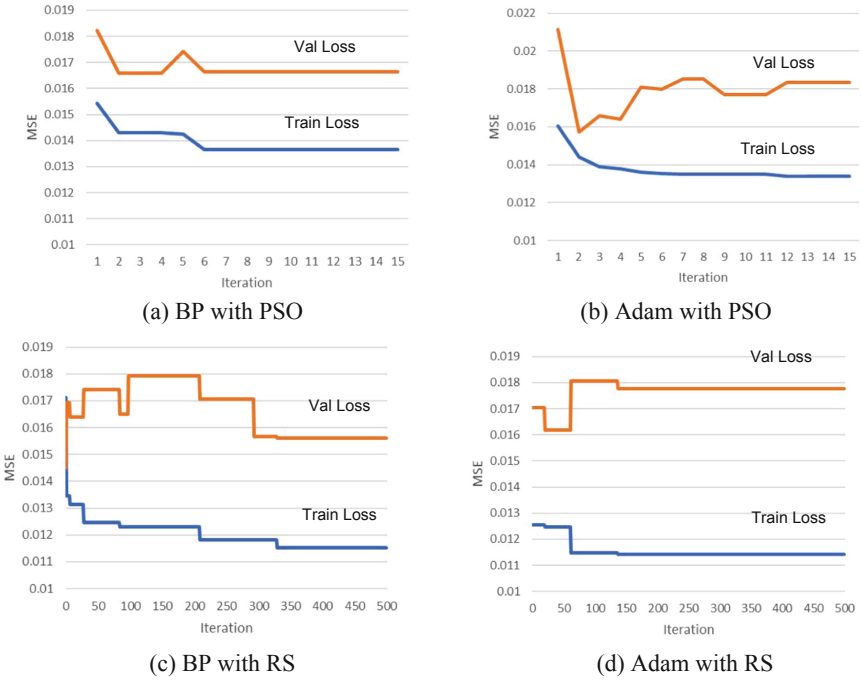
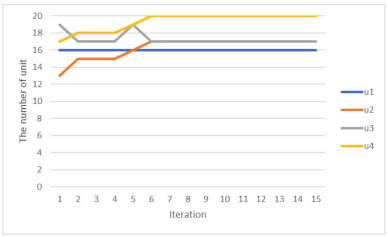


Fig. 3. The convergence of loss (MSE) of DBN in different fine-tuning processes (PSO and RS) and optimization algorithms (BP and Adam) using CATS data 1st block.

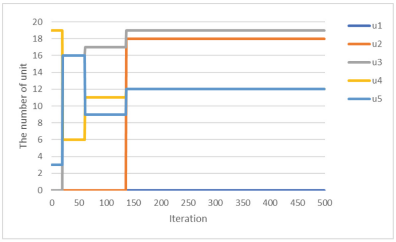
The change of the number of units in each RBM according to the different exploration algorithms, PSO and RS, is shown in Fig. 4. The iteration time of PSO ended at 15, and 500 for RS. Both exploration results showed that 2 RBMs were the best structure of DBN for the 1st block of CATS.

The change of the learning rates of different RBMs (pre-training) and MLP (fine-tuning) is shown in Fig. 5. The convergence of the learning rates were not obtained in each case of BP and Adam with PSO or RS.

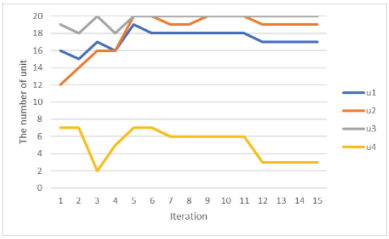
The exploration results of meta parameters for the 1st block data of CATS are described in Table 3.



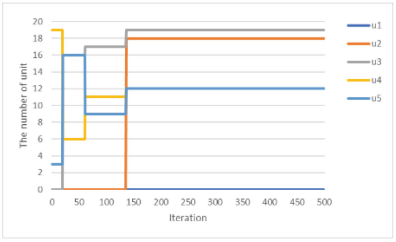
(a) PSO in BP



(b) RS in BP

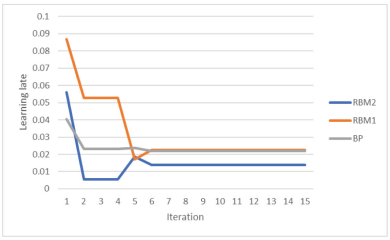


(c) PSO in Adam

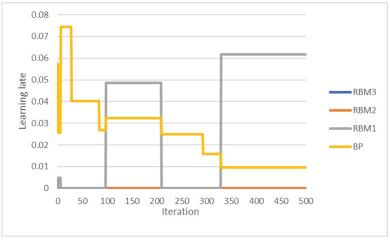


(d) RS in Adam

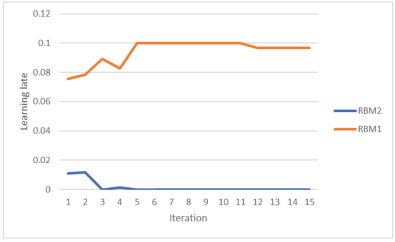
Fig. 4. Then change of number of units in RBM layers in different fine-tuning methods and optimization algorithms (in the case of CATS data 1st block).



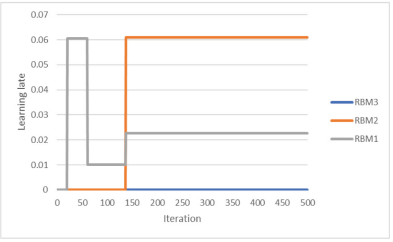
(a) PSO in BP



(b) RS in BP



(c) PSO in Adam



(d) RS in Adam

Fig. 5. The change of the learning rates in different fine-tuning methods and optimization algorithms (the case of CATS data 1st block).

Table 3. Meta parameters of DBN optimized by PSO and RS for the CATS data (Block 1)

	Adam + PSO	BP + PSO	Adam + RS	BP + RS
The number of RBMs	2	2	2	1
Learning rates of RBMs	0.0001, 0.09679	0.01392, 0.02266	0.0609, 0.0227	0.0617
Structure of DBN (the number of neurons in each layer)	17-19-20-3-1	16-17-17-20-1	18-19-12-12-1	17-5-9-1
Learning rate of MLP	Variable	0.02170	Variable	0.00951

3.3 Chaotic Time Series Data

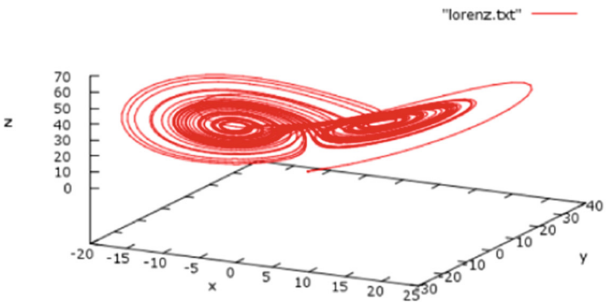
Chaotic time series are difficult to be predicted in the case of long-term forecasting [5]. Here, we used Lorenz chaos to compare the performance of DBNs with different fine-tuning methods in the case of short-term forecasting (one-ahead forecasting). Lorenz chaos is given by 3-D differential equations as follows.

$$\begin{cases} \frac{dx}{dt} = -\sigma \cdot x + \sigma \cdot y \\ \frac{dy}{dt} = -x \cdot z + r \cdot x - y \\ \frac{dz}{dt} = x \cdot y - b \cdot z \end{cases} \tag{15}$$

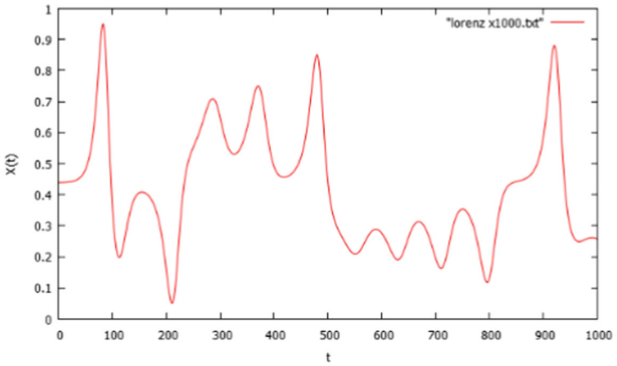
where parameters are given by $\sigma = 10,\ b = 28,\ r = \frac{8}{3},\ \Delta t = 0.01$ in the experiment. The attract of Lorenz chaos, a butterfly aspect, and the time series of x-axis are shown in Fig. 6.

3.4 Results and Analysis of Chaotic Time Series Forecasting

The exploration results of meta parameters for Lorenz chaotic time series by PSO and RS in different fine-tuning methods (Adam and BP) are described in Table 4. Adam learning rules resulted deeper structure of DBN than PSO, especially in the case of RS. The convergence of loss (MSE) of DBN in different fine-tuning processes (BP and Adam) and optimization algorithms (PSO and RS) using the time series data of Lorenz chaos (1 to 1000 in x-axis) is shown in Fig. 7. And finally, the precisions of different forecasting methods are compared by Table 5. The best method for this time series forecasting was Adam with PSO, which yielded the lowest loss 1.68×10^{-5}.



(a) The strange attractor of Lorenz chaos

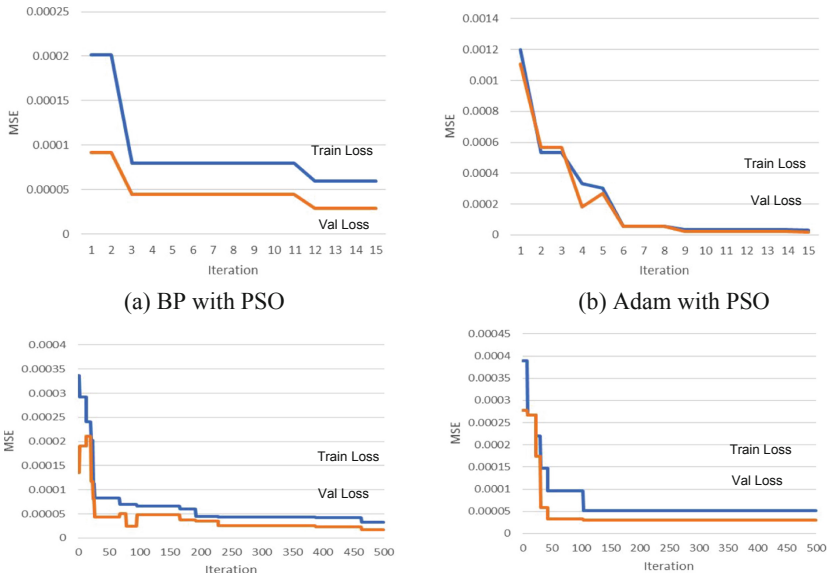


(b) Time series data in the X-axis of Lorenz chaos

Fig. 6. Lorenz chaos used in the short-term (one-ahead) prediction experiment.

Table 4. Meta parameters of DBN optimized by PSO and RS for the Lorenz chaos (x-axis).

	Adam + PSO	BP + PSO	Adam + RS	BP + RS
The number of RBMs	2	2	3	1
Learning rates of RBMs	0.01626, 0.00001	0.0949, 0.04120	0.08818, 0.02499, 0.03891	0.0659
Structure of DBN (the number of neurons in each layer)	20-20-7-2-1	6-6-10-10-1	3-9-13-12-12-2-1	20-19-10-1
Learning rate of MLP	Variable	0.0302	Variable	0.0820



(a) BP with PSO (b) Adam with PSO

(c) BP with RS (d) Adam with RS

Fig. 7. The convergence of loss (MSE) of DBN in different fine-tuning processes (BP and Adam) and optimization algorithms (PSO and RS) using the time series data of Lorenz chaos (1 to 1000 in x-axis).

Table 5. Precisions (MSE) of different DBNs (upper: training error; lower: test error).

Exploration	BP ($\times 10^{-5}$)	Adam ($\times 10^{-5}$)
RS	**3.32**	5.19
	1.70	3.03
PSO	5.95	**3.23**
	2.86	***1.68***

4 Conclusions

An improved gradient descent method Adam was firstly adopted to the fine-tuning process of the Deep Belief Net (DBN) for time series forecasting in this study. The effectiveness of the novel optimization algorithm showed its priority not only for the benchmark dataset CATS which was a long-term forecasting given by five blocks of artificial data, but also for the chaotic time series data which was a short-term forecasting (one-ahead) problem. As the optimizer Adam has been improved to be Nadam, AdaSecant, AMSGrad, AdaBound, etc., new challenges are remained in the future works.

Acknowledgement. This work was supported by JSPS KAKENHI Grant No. 22H03709, and No. 22K12152.

References

1. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica **50**(4), 987–1007 (1982)
2. Kuremoto, T., Obayashi, M., Kobayashi, K.: Neural forecasting systems. In: Weber, C., Elshaw, M., Mayer, N.M. (eds.) Reinforcement Learning, Theory and Applications, Chapter 1, pp. 1–20, INTECH (2008)
3. Kuremoto, T., Kimura, S., Kobayashi, K., Obayashi, M.: Time series forecasting using restricted Boltzmann machine. In: Huang, D.-S., Gupta, P., Zhang, X., Premaratne, P. (eds.) ICIC 2012. CCIS, vol. 304, pp. 17–22. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31837-5_3
4. Kuremoto, T., Kimura, S., Kobayashi, K., Obayashi, M.: Time series forecasting using a deep belief network with restricted Boltzmann machines. Neurocomputing **137**(5), 47–56 (2014)
5. Kuremoto, T., Obayashi, M., Kobayashi, K., Hirata, T., Mabu, S.: Forecast chaotic time series data by DBNs. In: Proceedings of the 7th International Congress on Image and Signal Processing (CISP 2014), pp. 1304–1309 (2014)
6. Hirata, T., Kuremoto, T., Obayashi, M., Mabu, S., Kobayashi, K.: Forecasting real time series data using deep belief net and reinforcement learning. J. Robotics Netw. Artif. Life **4**(4), 260–264 (2018)
7. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
8. Hirata, T., Kuremoto, T., Obayashi, M., Mabu, S., Kobayashi, K.: Time series prediction using DBN and ARIMA. In: International Conference on Computer Application Technologies (CCATS 2015), pp. 24–29. Matsue, Japan (2015)
9. Hirata, T., Kuremoto, T., Obayashi, M., Mabu, S., Kobayashi, K.: A novel approach to time series forecasting using deep learning and linear model. IEEJ Trans. Electron. Inf. Syst. **136**(3), 348–356 (2016)
10. Zhang, G.P.: Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing **50**, 159–175 (2003)
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Lendasse, A., Oja, E., Simula, O., Verleysen, M.: Time series prediction competition: the CATS benchmark. In: Proceedings of International Joint Conference on Neural Networks (IJCNN 2004), pp. 1615–1620 (2004)
14. Lendasse, A., Oja, E., Simula, O., Verleysen, M.: Time series prediction competition: the CATS benchmark. Neurocomputing **70**(13–15), 2325–2329 (2007)
15. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(2), 281–305 (2012)
16. Kuremoto, T., Hirata, T., Obayashi, M., Kobayashi, K., Mabu, S.: Search heuristics for the optimization of DBN for time series forecasting. In: Iba, H., Noman, N. (eds.) Deep Neural Evolution. NCS, pp. 131–152. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-3685-4_5