A Novel Study on the Fine-Tuning of Deep Belief Net for Time Series Forecasting

Takashi Kuremoto*, Masahumi Furuya, Shingo Mabu
Yamaguchi University, Yamaguchi, Japan

Abstract: A Deep Belief Net (DBN) with stacked Restricted Boltzmann Machines (RBMs) and a Multi-Layer Perceptron (MLP) was proposed for time series forecasting in our previous works. In this study, the fine-tuning method of DBN is modified by Adaptive moment estimation (Adam) instead of the conventional error Back-Propagation (BP). Experiment using a benchmark data CATS and chaotic time series showed the efficiencies of different learning methods.

Key-Words: *deep belief net (DBN), restricted Boltzmann machine (RBM), fine-tuning, time series forecasting*

1. Introduction

The study of time series forecasting has a long history since 1940s. Classically, linear models such as Autoregressive (AR), Moving Average (MA) and autoregressive integrated moving average (ARIMA) (Box-Jenkins 1976), and Autoregressive Conditional Heteroskedasticity (ARCH) [1] of R. Engle, who is the winner of 2003 Nobel Memorial Prize in Economic Sciences, are well known. Meanwhile, after the error Back-Propagation (BP) [2] was invented for the optimization of a feed-forward Artificial Neural Network (ANN), Multi-Layered Perceptron (MLP), there have been more than 6,000 publications of time series forecasting using ANNs until 2007 [3].

Deep Learning (DL) methods, which improved the BP by pre-training between the adjacent layers of ANNs [4], triggered the 3rd boom of the Artificial Intelligence (AI) research since the middle of 2000s. In our previous study, DL was firstly applied to the field of time series forecasting [5]-[9]. In [6] and [7], Hinton & Salakhutdinov's deep belief net (DBN), which is composed by Multiple Restricted Boltzmann machine (RBM) - a bi-directional recurrent neural network, was used to compare the accuracy of prediction of long-term forecasting a benchmark dataset and one-ahead forecasting of chaotic time series. The structure of the DBN, i.e., the number of RBMs and the number of units in each layer of RBMs was optimized by the Particle Swarm Optimization (PSO), and the modification of the parameters of the DBN such as the weights of connections of units, and the biases of each unit of RBM was given by the pre-training of RBMs with the gradient of network energy and the fine-tuning of whole DBN using the training samples and BP method, i.e., the gradient of Mean Squared Error (MSE) of the output of the DBN. Experiment results showed that the DBN was superior to the conventional MLP not only for the long-term forecasting but also one-ahead forecasting. In [8] and [9], the DBN was improved by composing the multiple RBMs and one MLP, which output continuous values more reasonable for time series forecasting.

The modification of parameters of ANNs is not limited to BP, but also approached by Evolutional Algorithms (EA), e.g., Genetic Algorithm (GA), Simulated Annealing (SA), PSO, Genetic Programming (GP), etc. Furthermore, Reinforcement

Received: 2020/12/01, Accepted: 2021/01/09
*Corresponding author: Takashi Kuremoto
E-mail address:wu@yamaguchi-u.ac.jp

Learning (RL), an error-trials learning method different from supervised learning, is also widely utilized to tuning of the parameters of ANNs [10]-[17]. R. J. Williams firstly proposed a RL algorithm REINFORCE to adjust the connection weights of units in 1992 [10]. Kimura & Kobayashi proposed a RL algorithm Stochastic Gradient Ascent (SGA) for the learning of continuous actions [11]. Sutton et al.'s RL method in [12] is well-known and applied to function approximation and system control. V. Mnih et al. successes on a development of game software using a DL model Convolutional Neural Network (CNN) and BP pre-training as well as RL for fine-tuning [13]. This success was developed to AlphaGo which is very famous for winning professional human players of GO [16]. In our previous works, SGA was adopted to a feed-forward ANN Radial Basis Function Network (RBFN) for time series forecasting and the proposed model showed its priority comparing to the case of BP learning and other forecasting methods [14]. In [15] and [17], we successfully used SGA as the fine-tuning method of DBN instead of conventional BP, especially for the case of real data forecasting.

In this paper, we adopt Adaptive moment estimation (Adam) [18] to the fine-tuning of DBN [7]-[9] for time series for casting. Adam is an improved BP learning method. It utilizes not only the gradient of Mean Squared Error (MSE) between the output of the model and teacher signals, but also the 1st moment (the change of parameters in one step before of modification) and the 2nd moment (the change of the modification between the two steps before and one step before). The prediction performance of the DBN with Adam is confirmed by the experiment using benchmark dataset CATS [19] [20] which was utilized in the time series forecasting competition with ANN methods.

2. DBN and its Fine-Tuning Methods

A Deep Belief Net (DBN) composed by multiple Restricted Boltzmann Machines (RBMs) was used for dimensionality reduction and image processing in the original study [4]. To apply DBN to time series forecasting, which needs continuous values of output, we combined RBMs with MLP [8] [9] [15] [17]. In this section, the structures of DBN and RBM, and the pre-training learning rule are introduced at first, then, fine-tuning methods including BP and Adam are described. The optimization of structure of DBN utilizes Random Search (RS) proposed by Bergstra & Bengio in 2012 [21] and Particle Swarm Optimization (PSO). The details of these optimization methods can be found in [22].

2.1 The Structure of DBN for Time Series Forecasting

DBN for time series forecasting is composed by stacked RBMs and a MLP as shown in Fig.1.

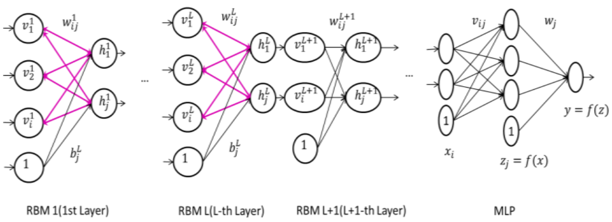


Fig.1 A structure of DBN for the time series.

Restricted Boltzmann Machine (RBM) [4] is a kind of Recurrent Neural Network (RNN) which has two layers: visual layer and hidden layer as shown in the left of Fig.1. ith unit v_i in visual layer connects to jth unit h_j with a symmetric weight w_{ij}. Units in RBM have binary values 0 or 1, which are given by a sigmoid probability distribution including parameters of biases b_i, b_j.

$$p(h_j = 1 \mid v) = \frac{1}{1 + \exp(-b_j - \sum_i v_i w_{ij})} \tag{1}$$

$$p(v_i = 1 | h) = \frac{1}{1+\exp(-b_i - \sum_j h_j w_{ij})} \quad (2)$$

$$P(v | h) = \prod_j P(v_j | h) \quad (3)$$

$$P(h | v) = \prod_i P(h_i | v) \quad (4)$$

The network energy of RBM is as Eq. (5), and the pre-training learning rules Eqs. (6)-(8) are given by the gradient of the network energy.

$$E(v,h) = -\sum_i b_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (5)$$

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_{data} - < v_i h_j >_{model}) \quad (6)$$

$$\Delta b_i = \varepsilon(< v_i > - < \tilde{v}_i >) \quad (7)$$

$$\Delta b_j = \varepsilon(< h_j > - < \tilde{h}_j >) \quad (8)$$

where $0 < \varepsilon < 1$ is a learning rate, $p_{ij} =< v_i h_j >_{\mathbf{data}}, p_{ij}^{'} =< v_i h_j >_{\mathbf{model}}$ are the distributions of Gibbs sampling (sampling time *k*=0), $< \tilde{v}_i >, < \tilde{h}_j >$ are the distributions of re-sampling (*k*=1) in Contrastive Divergence (CD) algorithm.

2.2 The Structure of MLP and the Fine-Tuning Methods

The structure of Multi-Layered Perceptron (MLP) is shown in the right of Fig.1. In MLP, units connect to all units in the next layer but no connections between units in the same layer [2]. Data in multiple dimensions are input to the *i*th unit x_i ($i = 1,2,...n$) and it is input to the *j*th unit z_j in the hidden layer with a connection weight w_{ji}=w_{ij}. The output of MLP is given as follows.

$$y = f(z) = \frac{1}{1+\exp(-\sum_{j=1}^{K+1} w_j z_j)} \quad (9)$$

$$f(z_j) = \frac{1}{1+\exp(-\sum_{i=1}^{n+1} v_{ji} x_i)} \quad (10)$$

where n is the dimensionality of input, K is the number of units in hidden layer, $x_{n+1} = 1.0, z_{K+1} = 1.0$ are biases of hidden units and output units, respectively. $v_{j(n+1)}, w_{K+1}$ are the connection weights between units.

Parameters in Eqs. (9) and (10) are adjusted by using the mean squared error (MSE) between the output of MLP and teacher data. Using gradient method, BP learning rules are derived as follows.

$$\Delta w_j = -\varepsilon(y - \tilde{y}) y (1-y) z_j \quad (11)$$

$$\Delta v_{ji} = -\varepsilon(y - \tilde{y}) y (1-y) w_j z_j (1 - z_j) x_i \quad (12)$$

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

In the case of Adam [18], parameters $\mathbf{\theta} = (v_{ji}, w_j)$ of a neural network are modified by follows.

$$\Delta \mathbf{\theta}_t = \frac{\hat{m}_t}{\varepsilon + \sqrt{\hat{v}_t}} \quad (13)$$

$$\hat{m}_t = \frac{\beta_1^t m_{t-1}}{1-\beta_1^t} + g_t \quad (14)$$

$$\hat{v}_t = \frac{\beta_2^t v_{t-1}}{1-\beta_2^t} + {g_t}^2 \quad (15)$$

$$g_t = \nabla_\theta l_t(\theta_{t-1}) \quad (16)$$

where , $0 < \varepsilon, \beta_1^t, \beta_2^t < 1$ are hyper-parameters, originally $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 10^{-8}$ and $l_t(\theta_{t-1})$ is a stochastic objective function with parameter , usually MSE or cross entropy function, $m_0 = v_0 = 0$ [18].

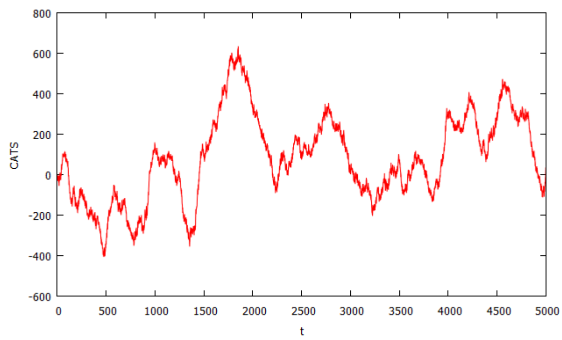


Fig.2 Time series data of CATS [19] [20]

3. Experiments and Results

3.1 Time Series Data

3.1.1 Benchmark Dataset CATS

To confirm the performance of Adam for DBN, a benchmark dataset CATS [19] [20] was utilized in the time series forecasting experiments. CATS is a kind of artificial time series data which includes 5 block data (as shown in Fig.2). There are 1,000 data (integer value) in each block, and the last 20 values, e.g., from 981 to 1,000, are hidden for the evaluation of long-term forecasting methods. The evaluation measurements are given by two mean squared errors E_1 and E_1 as follows.

$$E_1 = \frac{\sum_{t=981}^{1000}(y_t-\hat{y}_t)^2}{100} + \frac{\sum_{t=1981}^{2000}(y_t-\hat{y}_t)^2}{100} + \frac{\sum_{t=2981}^{3000}(y_t-\hat{y}_t)^2}{100} + \frac{\sum_{t=3981}^{4000}(y_t-\hat{y}_t)^2}{100} + \frac{\sum_{t=4981}^{5000}(y_t-\hat{y}_t)^2}{100} \quad (17)$$

$$E_2 = \frac{\sum_{t=981}^{1000}(y_t-\hat{y}_t)^2}{80} + \frac{\sum_{t=1981}^{2000}(y_t-\hat{y}_t)^2}{80} + \frac{\sum_{t=2981}^{3000}(y_t-\hat{y}_t)^2}{80} + \frac{\sum_{t=3981}^{4000}(y_t-\hat{y}_t)^2}{80} \quad (18)$$

3.1.2 Lorenz Chaos

Lorenz chaos [24] is very famous for its "butterfly attractor" as shown in Fig.3. The attractor is given by a 3-dimension differential equation as follows.

$$\begin{cases} \dfrac{dx}{dt} = -\sigma x + \sigma y \\ \dfrac{dy}{dt} = -xz + rx - y \\ \dfrac{dz}{dt} = xy - bz \end{cases} \quad (19)$$

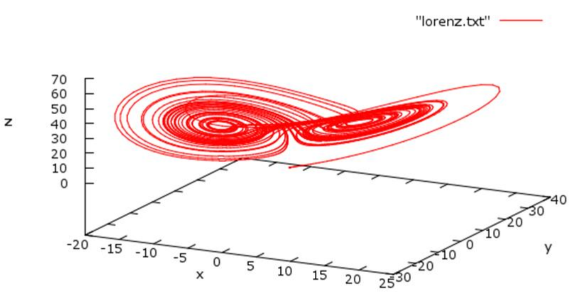


Fig.3 Lorenz chaos given by Eq. (19) where $\sigma = 10.0, b = 28.0, r = 8/3, \Delta t = 0.01$.

To examine the effectiveness of the DBN with Adam for nonlinear phenomena, the time series data of x-dimension of Lorenz chaos were used in the one-ahead forecasting experiment. 1,000 data of x-dimension of Lorenz chaos are normalized to (0.0, 1.0) and plotted in Fig.4. Data 1-600 were used as training samples, 601-800 were validation data, and 801-1000 as test data.

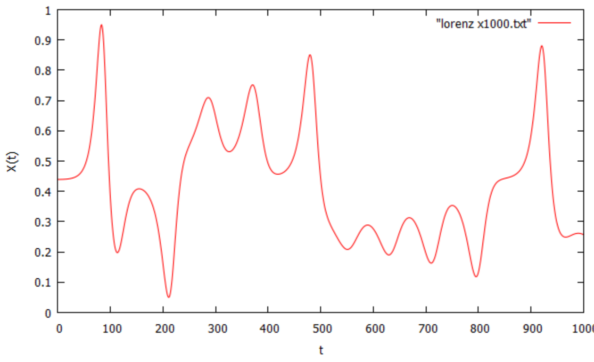


Fig.4 Chaotic time series used in the experiment.

3.2 Parameters Used in Experiments

For the optimal structure of DBM are decided by PSO or RS as described in Section 2, the exploration ranges for the hyper-parameters such as the number of RBMs in DBN, the number of units in each RBM's hidden layers, learning rate of RBMs, and learning rate of Adam or BP were set as shown in Table 1.

The limitations of exploration times were set to be 15 for PSO and 500 for RS empirically. The limitation of training time of RBM was the maximum 2,000 times or the network energy converged (the change of average value of 50 episodes less than 0.05). For MLP, it was 10,000 times limitation or the error between forecasted value and validation data increased.

Table 1. Exploration rage of PSO and RS for optimization of DBN

Dimension	Exploration Range of PSO and RS
The number of RBMs	0-3
The number of units in RBM's hidden layers	2-20
Learning rate of RBMs	10^{-1}-10^{-5}
Learning rate of MLP	10^{-1}-10^{-5}

3.3 Experiment Results and Discussions

3.3.1 Results of CATS dataset

The parameter exploration results for block No. 1 of CATS are shown in Table 2. By PSO exploration, the numbers of RBMs in DBN were 2, both cases of Adam and BP learning methods. Meanwhile, the learning rates and the numbers of neurons in each layer were different. The long-term forecasting errors (E_1) of different methods are shown in Table 3. DBM with Adam fine-tuning showed its priority not only in the case of PSO but also RS. The DBN with Adam and RS model had the highest prediction accuracy comparing to the conventional methods listed in the Table 3.

Table 2 Parameters of DBN used for the CATS data(A case of Block 1)

	Adam + PSO	BP + PSO	Adam + RS	BP + RS
The number of RBMs	2	2	2	1
Learning rate of RBM	0.0001, 0.0968	0.01392, 0.02266	0.0609, 0.0227	0.062
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	0.001	0.02170	0.001	0.00951

Table 3 The comparison of long-term prediction error (E_1) between different methods using CATS dataset [19] [20]

Method	E_1
DBN (Adam + RS) (proposed)	**134.04**
DBN (Adam + PSO) (proposed)	**148.24**
DBN (BP + RS) [7]	155.53
DBN (BP + PSO) [7]	155.65
DBN(SGA) [15] [17]	170
DBN(BP)+ARIMA [8] [9]	244
DBN (BP + PSO) [5] [6]	257
Kalman Smoother(The best of IJCNN '04) [20]	408
DBN (2 RBMs) [4]	1215
MLP [2]	1245
A hierarchical Bayesian Learning Scheme for Autoregressive Neural Networks (The worst of IJCNN '04) [20]	1247
ARIMA [23]	1715
ARIMA+MLP(BP) [8] [9]	2153
ARIMA+DBN (BP) [8] [9]	2266

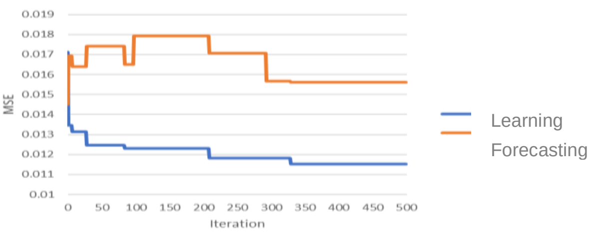


Fig.5 The change of the learning error (MSE) during fine tuning by BP method (CATS data [1-980])

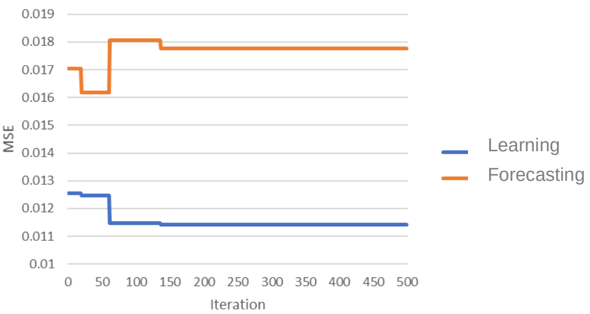
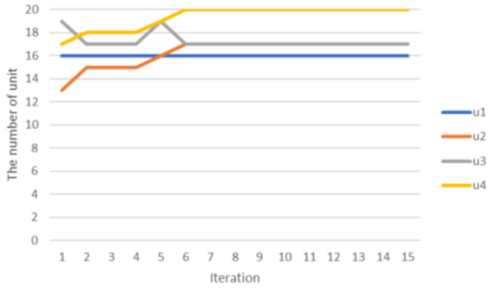
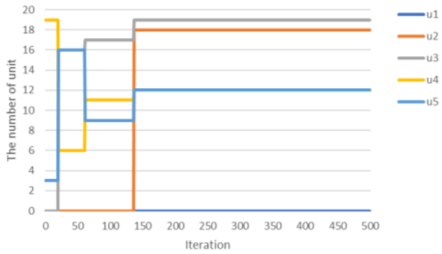


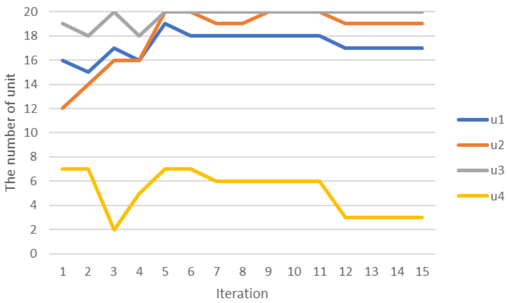
Fig.6 The change of the learning error (MSE) during fine tuning by Adam method (CATS data [1-980])



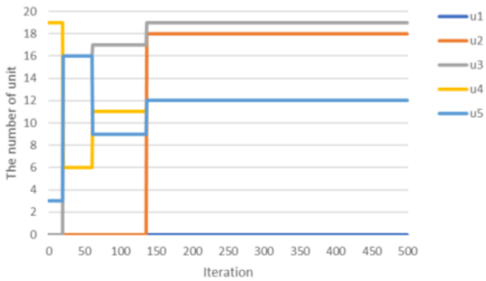
(a) The case of PSO for BP



(b) The case of RS for BP

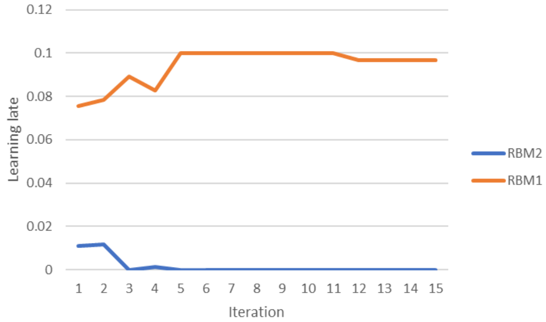


(c) The case of PSO for Adam

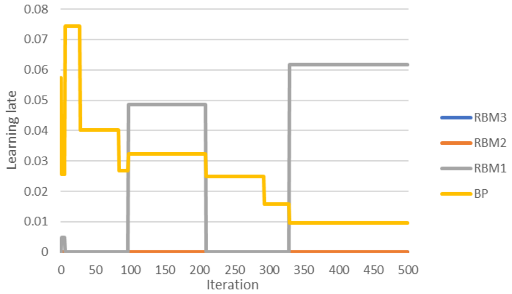


(d) The case of RS for Adam

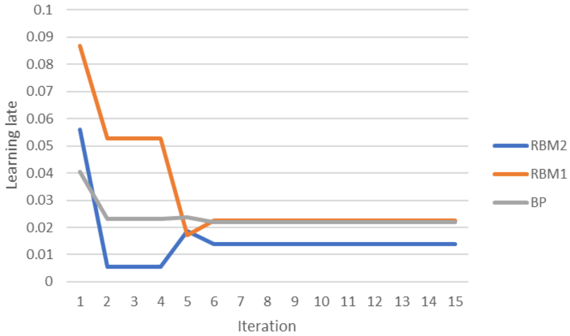
Fig.7 The change of the number of hidden neurons of RBMs by PSO and RS explorations (CATS data [1-980])



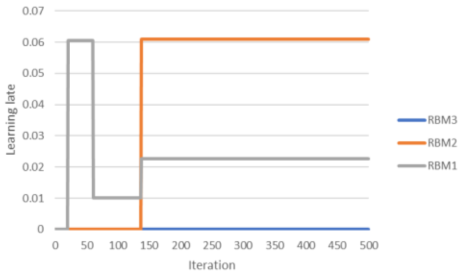
(a) The case of PSO for BP



(b) The case of RS for BP



(c) The case of PSO for Adam



(d) The case of RS for Adam

Fig.8 The change of the learning rates by PSO and RS explorations (CATS data [1-980])

To confirm the learning performance of BP and Adam, the changes of learning error (MSE) and forecasting error of validation data (MSE) are shown in Fig.5 and Fig.6. The learning errors (one-ahead forecasting) reduced according to the increase of iterations of the BP and Adam learning algorithms and for unknown validation data, forecasting errors (one-ahead forecasting) showed unstable during iteration of learning. The training errors of BP and Adam learning converged to 0.0115 and 0.0114, meanwhile, the forecasting error of BP learning was converged to 0.0156 (Fig.5), lower than Adam's 0.0179 (Fig.6).

To confirm the performance of PSO and RS optimization, the change of the number of hidden units is shown in Fig.7. The optimal number of hidden units of different methods is shown in Table 2.

The change of the learning rates of RBMs and MLP are shown in Fig.8 and the optimal learning rates are shown in Table 2. Note that learning rates in Adam algorithm were fixed to 0.001 as same as in [18].

Details of learning error and forecasting error of 5 blocks of CATS data by different models are shown in Table 4 (one-ahead forecasting results). The best results were presented in bold font. The optimal parameters exploited by PSO and RS of block No. 2 to block No.5 are omitted here.

The long-term forecasting errors are shown in Table 4. RS exploited DBN with Adam showed its priority not only in the sense of E_1 but also E_2.

3.3.2 Results of Lorenz chaos

The hyper-parameters exploited by RS and PSO are shown in Table 6 and Table 7. It can be found that structures of DBNs are different according to the different exploitation and different learning methods. Training errors and forecasting errors (one-ahead forecasting) of different methods for the chaotic time series are shown in Table 8. DBN with PSO and Adam showed its superiority among 4 DBN models. Note that chaotic time series are sensitive to its initial value and parameters in model, it is impossible to perform the long-term forecasting.

Table 4 Learning errors (MSEs) (upper) and forecasting errors (MSEs) (under) of different blocks of CATS data by 4 kinds of models: PSO with BP, PSO with Adam, RS with BP, and RS with Adam.

Data	BP PSO/RS	Adam PSO/RS
cats1000	1.37/1.15 1.66/**1.56**	1.34/**1.14** 1.84/1.78
cats2000	1.64/**1.18** 1.79/1.78	1.64/1.20 **1.64**/1.79
cats3000	1.57/**0.90** 2.17/**1.17**	1.56/1.01 2.05/1.61
cats4000	1.30/1.04 1.45/2.05	1.29/**0.86** 1.55/**1.27**
cats5000	1.66/**1.02** **4.11**/4.82	1.65/1.21 4.21/4.57

Unit: x10^{-2}

Table 5 Long-term forecasting errors of CATS dataset by 4 kinds of models: PSO with BP, PSO with Adam, RS with BP, and RS with Adam.

Error	BP PSO/RS	Adam PSO/RS
E_1	155.65/155.53	148.24/**134.04**
E_2	128.13/132.65	122.81/**112.22**

Table 6 Hyper-parameters decided by RS for Lorenz chaos.

Method	Number of RBMs	Number of units in hidden layers	Learning rate
BP	1	20－19－10－1	RBM: 0.065878 BP: 0.081975
Adam	3	3－9－13－12－12－2－1	RBM1: 0.088181 RBM2: 0.024990 RBM3: 0.038914

Table 7 Hyper-parameters decided by PSO for Lorenz chaos.

Method	Number of RBMs	Number of units in hidden layers	Learning rate
BP	2	6－6－10－10－1	RBM1: 0.094890 RBM2: 0.041967 BP: 0.030188
Adam	2	20-20-7-2-1	RBM1: 0.016258 RBM2: 0.000010

Table 8 Training errors (MSEs) (upper) and one-ahead forecasting errors (MSEs) (under) by different methods for Lorenz chaos.

	MSE (one-ahead forecasting)	
	BP	Adam
RS	3.32 1.70	5.19 3.03
PSO	5.95 2.86	**3.23** **1.68**

Unit: $x10^{-5}$

4. Conclusion

Adam, a deep learning optimization method which is widely used for Convolutional Neural Networks (CNNs), was adopted to the fine-tuning of parameters of Deep Belief Net (DBN) for time series forecasting in this paper. Comparing to the conventional fine-tuning methods such as error Back-Propagation (BP) and the Reinforcement Learning (RL), the proposed DBN with Adam showed its superiority in both cases of benchmark dataset CATS and the chaotic time series. The performance of the proposed method in the case of real data forecasting needs to be confirmed in the future. Additionally, there are some advanced Adam such as AdaSecant，AMSGrad，AdaBound, and so on are able to adopted to the fine-tuning process of DBN.

Acknowledgment:

A part of this study was supported by JSPS KAKENHI Grant No. 19K12120.

References:

[1] R. Engle: “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation”, Econometrica, Vol. 50, No. 4, 1982, pp. 987-1008

[2] D. E., Rumelhart, G. E., Hinton, G. E. & R. J. Williams: “Learning Representation by Back-Propagating Errors”, Nature, Vol. 232, No. 9,1986, pp. 533-536.

[3] T. Kuremoto, M. Obayashi, K. Kobayashi: “Neural Forecasting Systems”, Reinforcement Learning, Theory and Applications (ed. C. Weber, M. Elshaw & N. M. Mayer), 2008, Chapter 1, pp. 1-20, INTECH.

[4] G.E. Hinton, R.R. Salakhutdinov: “Reducing the Dimensionality of Data with Neural Networks”, Science, Vol.313,2006, pp. 504-507.

[5] Kuremoto, T., Kimura, S., Kobayashi, K., and Obayashi M.: Time Series Forecasting Using Restricted Boltzmann Machine. Proceedings of the 8th International Conference on Intelligent Computing (ICIC 2012), Communications in Computer and Information Science (CCIS), Vol. 304,2012, pp.17-22, Springer-Verlag

[6] T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi: “Time Series Forecasting using a Deep Belief Network with Restricted Boltzmann Machines”, Neurocomputing, Vol.137, No.5,2014, pp. 47–56.

[7] T. Kuremoto, T. Hirata, M. Obayashi, S. Mabu, K. Kobayashi “Forecast Chaotic Time Series Data by DBNs”, Proceedings of the 7th International Congress on Image and

Signal Processing (CISP 2014), 2014, pp. 1304-1309.

[8] T. Hirata, T. Kuremoto, M. Obayashi, S. Mabu, K. Kobayashi: "Time Series Prediction Using DBN and ARIMA", International Conference on Computer Application Technologies (CCATS 2015), Matsue, Japan, 2015, pp. 24-29.

[9] T. Hirata, T. Kuremoto, M. Obayashi, S. Mabu, K. Kobayashi: "A Novel Approach to Time Series Forecasting using Deep Learning and Linear Model", IEEJ Transaction on EIS, Vol. 136, No. 3, 2016, pp. 248-356.

[10] R.J. Williams: "Simple Statistical Gradient Following for Connectionist Reinforcement Learning", Machine Learning, Vol. 8, 1992, pp. 229-256.

[11] H. Kimura, S. Kobayashi: "Reinforcement Learning for Continuous Action using Stochastic Gradient Ascent", Proceedings of 5th Intelligent Autonomous Systems, 1998, pp. 288-295.

[12] R.S. Sutton, D. Mc Allester, S. Singh, Y. Mansour: "Policy Gradient Methods for Reinforcement Learning with Function Approximation", Advances in Neural Information Processing Systems, 2000, pp. 1057-1063.

[13] V. Mnih, et al.: "Human-Level Control Through Deep Reinforcement Learning", Nature, Vol. 518, 2015, pp. 529-533.

[14] T. Kuremoto, K. Ko, M. Obayashi, S. Mabu, K. Kobayashi: "Neural Networks using Reinforcement Learning and their Applications to Time Series Forecasting", Proceedings of 3rd International Conference on Automatic Control, Softcomputing, and Human-Machine Interaction (WSEAS/ASME'15), 2015, pp. 69-74

[15] Hirata, T., Kuremoto, T., Obayashi, M., Mabu, S., Kobayashi, K.: Deep Belief Network using Reinforcement Learning and its Applications to Time Series Forecasting, In Proc. Intern. Conf. Neural Inform. Processing, (ICONIP'16), Lecture Notes in Computer Science (LNCS), Springer, Vol. 9949, 2016, pp. 30-37.

[16] Silver, D., et al.: "Mastering the game of Go without human knowledge", Nature, 550, 2017, pp. 354-359

[17] Hirata, T., Kuremoto, T., Obayashi, M., Mabu, S., Kobayashi, K.: "Forecasting real time series data using deep belief net and reinforcement learning", Journal of Robotics, Networking and Artificial Life, Vol. 4, No.4, 2018, pp. 260-264

[18] D. P. Kingma, J. Lei Ba: "Adam: A method for stochastic optimization", https://arxiv.org/pdf/1412.6980.pdf (2015)

[19] A. Lendasse, E. Oja, O. Simula, M. Verleysen: "Time Series Prediction Competition: The CATS Benchmark", Proceedings of International Joint Conference on Neural Networks (IJCNN'04),2004, pp. 1615-1620.

[20] A. Lendasse, E. Oja, O. Simula, M. Verleysen: "Time Series Prediction Competition: The CATS Benchmark", Neurocomputing, Vol. 70, 2007, pp. 2325-2329

[21] J. Bergstra, Y. Bengio: "Random Search for Hyper-Parameter Optimization", Journal of Machine Learning Research, Vol. 13, 2012, pp. 281-305

[22] Kuremoto, T., Hirata, T., Obayashi, M., Kobayashi, K., Mabu, S., Search Heuristics for the Optimization of DBN for Time Series Forecasting, In Deep Neural Evolution, Natural Computing Series (eds. H. Iba & N. Noman), 2020, Chapter 5, pp. 131-152, Springer Nature

[23] G.P. Zhang: "Time Series Forecasting using a Hybrid ARIMA and Neural Network Model", Neurocomputing, Vol.50,2003, pp. 159–175.

[24] E. N. Lorenz:" Deterministic Nonperiodic Flow", Journal of the atmosphere Sciences, Vol. 20,1963, pp. 130-141