Deep Belief Network using Reinforcement Learning and its Applications to Time Series Forecasting

Takaomi HIRATA, Takashi KUREMOTO, Masanao OBAYASHI, Shingo MABU

Graduate School of Science and Engineering
Yamaguchi University
Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611
JAPAN
{v003we, wu, m.obayas, mabu}@yamaguchi-u.ac.jp

Kunikazu KOBAYASHI

School of Information Science and Technology
Aichi Prefectural University
1522-3 Ibaragabasama, Nagakute, Aichi 480-1198
JAPAN
kobayashi@ist.aichi-pu.ac.jp

Abstract. Artificial neural networks (ANNs) typified by deep learning (DL) is one of the artificial intelligence technology which is attracting the most attention of researchers recently. However, the learning algorithm used in DL is usually with the famous error-backpropagation (BP) method. In this paper, we adopt a reinforcement learning (RL) algorithm "Stochastic Gradient Ascent (SGA)" proposed by Kimura & Kobayashi into a Deep Belief Nets (DBN) with multiple restricted Boltzmann machines instead of BP learning method. A long-term prediction experiment, which used a benchmark of time series forecasting competition, was performed to verify the effectiveness of the novel DL method.

Keywords: deep learning, restricted Boltzmann machine, stochastic gradient ascent, reinforcement learning, error-backpropagation

1 Introduction

A time series is a data string to be observed in a temporal change in a certain phenomenon. For example, foreign currency exchange rate, stock prices, the amount of rainfall, the change of sunspot, etc. There is a long history of time series analysis and forecasting [1], however, the prediction study of time-series data in the real world is still on the way because of the nonlinearity and the noise affection.

Artificial neural networks (ANNs) have been widely used in pattern recognition, function approximation, nonlinear control, time series forecasting, and so on, since 1940s [2]-[11]. After a learning algorithm named error-backpropagation (BP) was proposed by Rumelhart, Hinton & William for multi-layer perceptrons (MLPs) [2],

adfa, p. 1, 2011.

ANNs had its heyday from 1980s to 1990s. As a successful application, ANNs are utilized as time series predictors today [3]-[11]. Especially, a deep belief net (DBN) composed by restricted Boltzmann machines (RBMs) [6] and Multi-Layer Perceptron (MLP) [2] was proposed in [8]-[10] recently.

Deep learning (DL), which is a kind of novel ANNs' training method, is attracting the most attention of artificial intelligent (AI) researchers. By a layer-by-layer training algorithm and stack structures, big data in the high dimensional space is able to be classified, recognized, or sparsely modeled by DL [6]. However, although there are various kinds of deep networks such as auto-encoder, deep Boltzmann machine, convolutional neural network (CNN), etc., the learning algorithm used in DL is usually with the famous BP method.

In this paper, a reinforcement learning method named stochastic gradient ascent (SGA) proposed by Kimura & Kobayashi [13] is introduced to the DBN with RBMs as the fine-tuning method instead of the conventional BP learning method. The error between the output of the DBN and the sample is used as reward/punishment to modify the weights of connections of units between different layers. Using a benchmark named CATS data used in the prediction competition [4] [5], the effectiveness of the proposed deep learning method was confirmed by the time series forecasting results.

2 DBN with BP Learning (The Conventional Method)

In [8], Kuremoto et al. firstly applied Hinton & Slakhutdinov's deep belief net (DBN) with restricted Boltzmann machines (RBMs) to the field of time series forecasting. In [9] and [10], Kuremoto, Hirata, et al. constructed a DBN with RBMs and a multi-layer perceptron (MLP) to improved the previous time series predictor with RBMs only. In this section, these conventional methods are introduced.

2.1 DBN with RBM and MLP

Restricted Boltzmann machine (RBM)

RBM is a Boltzmann machine consisting of two layers of the visible layer and the hidden layer, no connections between neurons on the same layer. It is possible to extract features to compress the high-dimensional data in low-dimensional data by performing an unsupervised learning algorithm. Each unit of the visible layer has a symmetric connection weights with respect to the unit each of the hidden layer. Coupling between the units are bi-direction. The value of the connection weights between units is the same in both directions. Unit i of the visible layer has a bias $b_{i,}$ the hidden layer b_j. All neurons in the visible layer and the hidden layer stochastically output 1 or 0, according to its probability with sigmoid function (Eq. (1) and Eq. (2)).

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

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

Using a learning rule which modifies the weights of connections, RBM network can reach a convergent state by observing its energy function:

$$E(v,h) = -\sum_i b_i v_i - -\sum_j b_j h_j - \sum_{i,j} w_{ij} v_i h_j \qquad (3)$$

Details of the learning algorithm of RBM can be found in [8].

DBN with RBM and MLP

DBN is a multi-layer neural network which usually composes multiple RBMs [6] [8]. DBN can extract the feature of features of high-dimensional data, so it is also called "deep learning", and applied to many fields such as dimensionality reduction, image compression, pattern recognition, time series forecasting, and so on.

A DBN prediction system is proposed recently by composing of plural RBMs in [8], and another DBN prediction system using RBM and MLP is proposed in [9] (see Fig. 1). However, all of these DBNs used the learning algorithm proposed in [6], i.e. RBM's unsupervised learning and BP learning, supervised learning as fine-tuning conventionally.

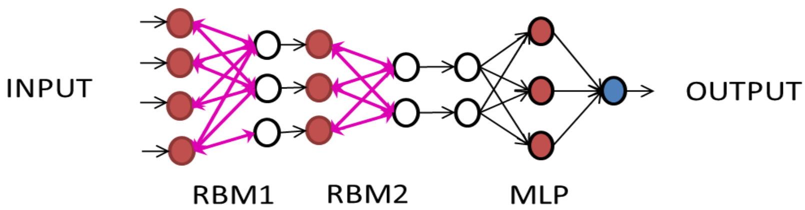

Fig. 1. A structure of DBN composed by RBM and MLP

3 DBN with SGA (The Proposed Method)

BP is a powerful learning method for the feed-forward neural network, however, it modifies the model strictly according to the teacher signal. So the robustness of the ANNs built by BP is restricted. Meanwhile, noises usually exist in the real time-series data. Therefore, we consider that prediction performance may be improved by using a reinforcement learning (RL) algorithm to modify the ANN predictors according to the rewards/punishment corresponding to a "good/bad" output. That is, if the absolute error between the output and the sample is small enough (e.g. using a threshold), then it is considered as a "good" output, and a positive reward is used to modify parameters of ANNs. In [7], MLP and a self-organized fuzzy neural network (SOFNN) with a RL algorithm called stochastic gradient ascent (SGA) [13] were shown their priority to the conventional BP learning algorithm.

Here, we intend to investigate the case when SGA is used in DBN instead of BP.

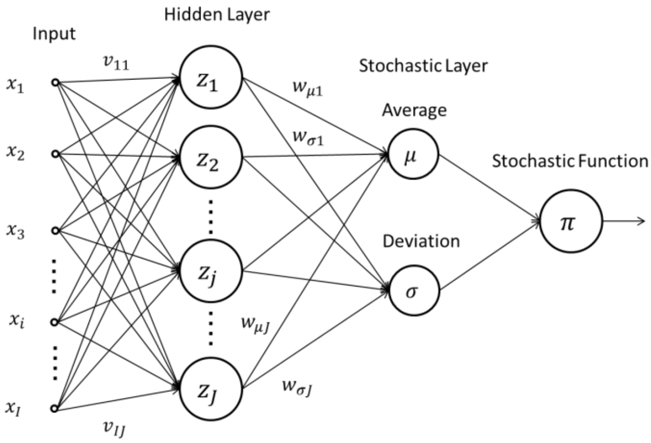

Fig. 2.The structure of MLP designed for SGA

3.1 The structure of ANNs with SGA

In Fig. 2, a MLP type ANN is designed for SGA [7]. The main difference to the conventional MLP is that the output of network is given by two units which are the parameters of a stochastic policy which is a Gaussian distribution function.SGA Learning algorithm for ANNs

The SGA algorithm is given as follows [7] [13].

1. Observe an input $x(t)$ on time *t*.
2. Predict a future data $y(t)$ as $\hat{y}(t)$ according to a probability $\pi(\hat{y}(t) \mid W, x(t)) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{(\hat{y}(t)-\mu)^2}{2\sigma^2})$ with ANN models which are constructed by parameters $W \equiv \{\mu, \sigma, w_\mu, w_\sigma, v_{ij}\}$.
3. Receive a scalar reward/punishment r_t by calculating the prediction error.

$$r_t = \begin{cases} 1 & if\ (y(t)-\hat{y}(t))^2 \le \varepsilon(t) \\ -1 & else \end{cases} \quad (4)$$

Where $\varepsilon(t) = \beta MSE(t), MSE(t) = \frac{\sum_t (y(t)-\hat{y}(t))^2}{t}$, β is a positive constant.

4. Calculate characteristic eligibility $e_i(t)$ and eligibility trace $\overline{D}_i(t)$.

$$\pi(W) \quad (5)$$

$$\overline{D}_i(t) = e(t) + \gamma \overline{D}_i(t-1) \quad (6)$$

Where $0 \le \gamma < 1$ is a discount factor, w_i denotes *i*th element of the internal variable vector *W*.

Calculate $\Delta w_i(t)$: $\Delta w_i(t) = (r_t - b)\overline{D}_i(t)$ (7)

Where b denotes the reinforcement baseline.

5. Improve the policy $\pi(\hat{y}(t) \mid x(t), W)$ by renewing its internal variable W .

$$W \leftarrow W + \alpha \Delta W \tag{8}$$

Where α is the learning rate.

6. For the next time step $t+1$, if the prediction error *MSE(t)* is converged enough then end the learning process, else, return to step 1.

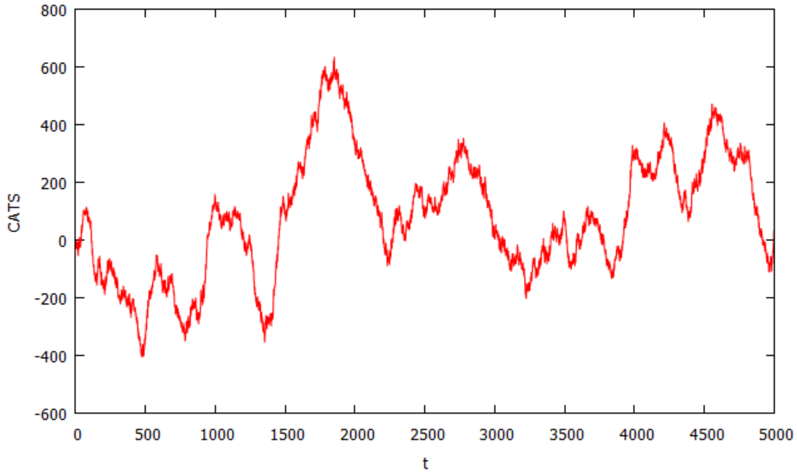

Fig. 3. Time series data of CATS

4 Prediction Experiments and Results

4.1 CATS benchmark time series data

CATS time series data is the artificial benchmark data for forecasting competition with ANN methods [4] [5].This artificial time series is given with 5,000 data, among which 100 are missed (hidden by competition the organizers). 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:

$$E_1 = \frac{1}{100}\{\sum_{t=981}^{1000}(y(t)-\hat{y}(t))^2 + \sum_{t=1981}^{2000}(y(t)-\hat{y}(t))^2 + \sum_{t=2981}^{3000}(y(t)-\hat{y}(t))^2 + \sum_{t=3981}^{4000}(y(t)-\hat{y}(t))^2 + \sum_{t=4981}^{5000}(y(t)-\hat{y}(t))^2\} \tag{9}$$

where $\hat{y}(t)$ is the long term prediction result of the missed data.

4.2 Optimization of meta parameters

The number of RBM that constitute the DBN and the number of neurons of each layer affect prediction performance seriously. In this paper, these meta parameters are optimized using random search method [12]. In the optimization of the ANN structure, random search is known to exhibit higher performance than the grid search. The meta parameters of DBN and their exploration limits are shown as following: the number of RBMs: 0-3; the number of neurons in each layers: 2-20; learning rate of each RBM: 10^{-5}-10^{-1}; learning rate of SGA : 10^{-5}-10^{-1}; discount factor: 10^{-5}-10^{-1}; constant β in Eq. (8) 0.5-2.0

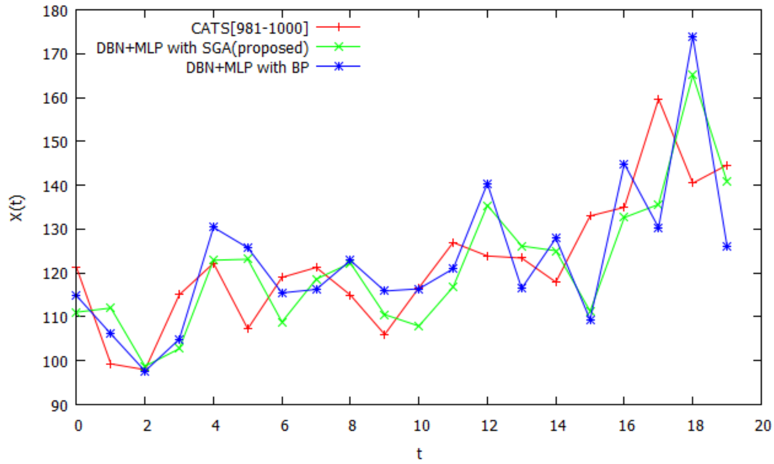

Fig. 4. The prediction results of different methods for CATS (Block 1).

Table 1. The comparison of performance between different methods using CATS data

Method	E_1
DBN(SGA) (proposed)	**170**
DBN(BP)+ARIMA (10)	244
DBN(9) (BP)	257
Kalman Smoother (The best of IJCNN '04) (5)	408
DBN(8) (2 RBMs)	1215
MLP(8)	1245
A hierarchical Bayesian Learning Scheme for Autoregressive Neural Networks (The worst of IJCNN '04) (4)	1247

4.3 Experiments Result

The prediction results of the first blocks of CATS data are shown in Fig. 4. Comparing to the conventional learning method of DBN, i.e., using Hinton's RBM unsupervised learning method [6] [8] and back-propagation (BP), the proposed method which used the reinforcement learning method SGA instead of BP showed its superiority according to the measure of the average prediction precision E_1. Additionally, the result by the proposed method achieved the top of rank of all previous studies such as MLP with BP, the best prediction of CATS competition IJCNN'04 [5], the conventional DBNs with BP [8] [9], and hybrid models [10] [11]. The details are shown in Table 1.The optimal parameters obtained by random search method are shown in Table 2.

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

	DBN with SGA (proposed)	DBN with BP [9]
The number of RBMs	3	1
Learning rate of RBM	0.048-0.055-0.026	0.042
Structure of DBN (the number of neurons in each layer)	14-14-18-19-18-2	5-11-2-1
Learning rate of SGA or BP	0.090	0.090
Discount factor γ	0.082	-
Coefficient β	1.320	-

5 Conclusion

In this paper, we proposed to use a reinforcement learning method "stochastic gradient ascent (SGA)" to realize fine-tuning of a deep belief net (DBN) composed by multiple restricted Boltzmann machines (RBMs) and multi-layer perceptron (MLP). Different from the conventional fine-tuning method using the error backpropagation (BP), the proposed method used a rough judgment of reinforcement learning concept, i.e., "good" output owns positive rewards, and "bad" one with negative reward values, to modify the network. This makes the available of a sparse model building, avoiding to the over-fitting problem which occurs by the lost function with the teacher signal.

Time series prediction was applied to verify the effectiveness of the proposed method, and comparing to the conventional methods, the DBN with SGA showed the highest prediction precision in the case of CATS benchmark data.

The future work of this study is to investigate the effectiveness to the real time series data and other nonlinear systems such as chaotic time series data.

Acknowledgment:

This work was supported by JSPS KAKENHI Grant No. 26330254 and No.25330287.

References:

1. G. E. P. Box, D. A. Pierce, Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models, *Journal of the American Statistical Association*, Vol. 65, No. 332, Dec., 1970, pp. 1509-1526.
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. M. Casdagli, Nonlinear prediction of chaotic time series, *Physica D*, Vol.35, 1981, pp.335-356 .
4. 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, pp1615-1620.
5. A. Lendasse, E. Oja, O. Simula, M. Verleysen, Time Series Prediction Competition: The CATS Benchmark, Neurocomputing, Vol. 70, 2007, pp.2325-2329.
6. G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, Vol.313, 2006, pp. 504-507.
7. T. Kuremoto, M. Obayashi, M. Kobayashi, Neural Forecasting Systems. *In Reinforcement Learning, Theory and Applications* (ed. C. Weber, M. Elshaw and N. M. Mayer), Chapter 1, pp. 1-20, 2008, INTECH.
8. 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, Aug. 2014, pp.47–56.
9. 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)*, Oct. 2014, pp. 1304-1309.
10. T. Hirata, T. Kuremoto, M. Obayashi, S. Mabu, Time Series Prediction Using DBN and ARIMA, *International Conference on Computer Application Technologies (CCATS 2015)*, Matsue, Japan, Sep. 2015, pp.24-29.
11. G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, Vol.50, 2003, pp. 159–175.
12. J. Bergstra, Y. Bengio, Random Search for Hyper-Parameter Optimization, *Journal of Machine Learning Research*, Vol. 13, 2012, pp.281-305.
13. H. Kimura, S. Kobayashi, Reinforcement learning for continuous action using stochastic gradient ascent, *Proceedings of 5th Intelligent Autonomous Systems*, 1998, pp. 288-295.