

Chaotic time series prediction with deep belief networks: an empirical evaluation

Ta Ngoc Huy Nam, Duong Tuan Anh*



Use your smartphone to scan this QR code and download this article

ABSTRACT

Chaotic time series are widespread in several real world areas such as finance, environment, meteorology, traffic flow, weather. A chaotic time series is considered as generated from the deterministic dynamics of a nonlinear system. The chaotic system is sensitive to initial conditions; points that are arbitrarily close initially become exponentially further apart with progressing time. Therefore, it is challenging to make accurate prediction in chaotic time series. The prediction using conventional statistical techniques, k-nearest-nearest neighbors algorithm, Multi-Layer-Perceptron (MLP) neural networks, Recurrent Neural Networks, Radial-Basis-Function (RBF) Networks and Support Vector Machines, do not give reliable prediction results for chaotic time series. In this paper, we investigate the use of a deep learning method, Deep Belief Network (DBN), combined with chaos theory to forecast chaotic time series. DBN should be used to forecast chaotic time series. First, the chaotic time series are analyzed by calculating the largest Lyapunov exponent, reconstructing the time series by phase-space reconstruction and determining the best embedding dimension and the best delay time. When the forecasting model is constructed, the deep belief network is used to feature learning and the neural network is used for prediction. We also compare the DBN-based method to RBF network-based method, which is the state-of-the-art method for forecasting chaotic time series. The predictive performance of the two models is examined using mean absolute error (MAE), mean squared error (MSE) and mean absolute percentage error (MAPE). Experimental results on several synthetic and real world chaotic datasets revealed that the DBN model is applicable to the prediction of chaotic time series since it achieves better performance than RBF network.

Key words: Deep Belief Network, Restricted Boltzmann Machine, chaotic time series, RBF network, forecasting

INTRODUCTION

Time series in several real world areas such as finance, environment, meteorology, and weather are characterized as chaotic in nature. A chaotic time series is generated from the deterministic dynamics of a nonlinear system (^{1,2}). The chaotic system is sensitive to initial conditions; points that are arbitrarily close initially become exponentially further apart with progressing time. Therefore, it is challenging to make accurate prediction in chaotic time series. The prediction using conventional statistical techniques, k-nearest-nearest neighbors algorithm, Multi-Layer-Perceptron (MLP) neural networks, Recurrent Neural Networks, Radial-Basis-Function (RBF) Networks and Support Vector Machines (SVMs), do not give reliable prediction results for chaotic time series.

Deep learning models, such as Deep Belief Networks (DBNs), have recently attracted the interest of many researchers in some applications on big data analysis. DBN is generative neural network model with many hidden layers, introduced by Hinton et al. ³ along with

a greedy layer-wise learning algorithm. The building block of a DBN is a probabilistic model called Restricted Boltzmann Machine (RBM). DBNs and restricted Boltzmann machines (RBMs) have already been applied successfully to solve many problems, such as classification, dimensionality reduction and image processing.

There have been several research works of applying DBNs to predict time series data in finance (^{4,5}), meteorology (^{6,7}), and industry (⁸). However, so far there have been very few works on applying DBNs in forecasting chaotic time series. Kuremoto et al. in 2014 [6] studied the application of a DBN which composes RBMs and multi-layer perceptron (MLP) to predict chaotic time series data. The hyper-parameters of the deep network were determined by particle swarm optimization (PSO) algorithm. Despite of the simple and effective structure for the proposed DBN, there exist three weaknesses in this paper. First, the paper does not make clear that the DBN model is combined with chaos theory in dealing with chaotic time series prediction. Second, the

Ho Chi Minh City University of Technology, VNU-HCM

Correspondence

Duong Tuan Anh, Ho Chi Minh City University of Technology, VNU-HCM

Email: dtanhcse@gmail.com

History

- Received: 28-8-2019
- Accepted: 26-9-2019
- Published: 04-12-2020

DOI : 10.32508/stdjet.v3iS11.571



Copyright

© VNU-HCM Press. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International license.



Cite this article : Nam T N H, Anh D T. Chaotic time series prediction with deep belief networks: an empirical evaluation. *Sci. Tech. Dev. J. – Engineering and Technology*; 3(S11):SI102-SI112.

work has tested DBN model over only two synthetic chaotic time series datasets: Lorenz and Henon Map. The lack of testing the proposed model with real world chaotic time series datasets cannot validate the high performance of DBN model in practical applications of chaotic time series prediction. Third, the work compared DBN model with only MLP model, a simple form of shallow neural networks.

In this work, we present and evaluate extensively a method of chaotic time series prediction using the DBN model which has the same structure as given in the paper⁹. However, there are three focus points in our work which makes it different from the previous work by Kuremoto et al. in 2014.

i) We combine the DBN model with the chaos theory, namely phase space reconstruction, in dealing with chaotic time series prediction.

ii) We compare the performance of DBN model to that of Radial Basis Function (RBF) network, a special kind of shallow neural networks which can bring out better forecasting precision than MLP neural network in chaotic time series prediction¹⁰.

iii) To verify the effectiveness of DBN, we experiment the performance of DBN over several benchmark chaotic time series datasets. In experiment, we use three synthetic time series datasets: Lorenz, Mackey-Class, Rossler and four real world time series datasets: Sunspots and some financial/economic datasets. The predictive performance of the two models is examined using mean absolute error (MAE), mean square error (MSE) and mean absolute percentage error (MAPE). Experimental results through three evaluation criteria revealed that DBN outperforms RBF network in most of the datasets.

The remainder of the paper is organized as follows. Section 2 provides some basic backgrounds about DBN and chaos theory. In Section 3, the method using DBN for forecasting chaotic time series is introduced. Section 4 reports the experiments to compare the prediction accuracy of the DBN method to that of the RBF network model. Finally, section 5 gives some conclusions and future work.

BACKGROUND AND RELATED WORKS

Deep Belief Network

Restricted Boltzmann Machines (RBMs) are often used to construct deeper models such as DBN. RBM is a kind of stochastic artificial neural network with two connected layers: a layer of binary visible units (v , whose states are observed) and a layer of binary hidden units (h , whose states cannot be observed). The

hidden units act as latent variables (features) that allow the RBM to model probability distribution over state vectors (see Figure 1). The hidden units are conditionally independent given visible units. Given an energy function $E(v, h)$ on the whole set of visible and hidden units, the joint probability is given by:

$$p(v, h) = \frac{e^{-E(v, h)}}{Z} \quad (1)$$

where Z is a normalization partition function, which is obtained by summing up the energy of all possible (v, h) configurations.

$$Z = \sum_{v, h} e^{-E(v, h)} \quad (2)$$

For the binary units $h_i \in \{0, 1\}$ and $v_i \in \{0, 1\}$, the energy function of the whole configuration is:

$$E(v, h) = -cv^T - bh^T - hWv^T = -\sum_{k=1}^K c_k v_k - \sum_{j=1}^J b_j h_j - \sum_{j=1}^J \sum_{k=1}^K W_{jk} v_k h_j \quad (3)$$

where W is $J \times K$ matrix of RBM weights, $c = [c_1, c_2, \dots, c_K]$ is the bias of the visible units and $b = [b_1, b_2, \dots, b_J]$ the bias of the hidden units. The marginal distribution over v is:

$$p(v) = \sum_h p(v, h) \quad (4)$$

The posterior probability of one layer given the other is easy to compute by the two following equations:

$$p(h_j = 1 | v) = \prod_j p(h_j = 1 | v) \quad (5)$$

where $p(h_j = 1 | v) = \sigma(b_j + \sum_k W_{jk} v_k)$

$$p(v_k = 1 | h) = \prod_k p(v_k = 1 | h) \quad (6)$$

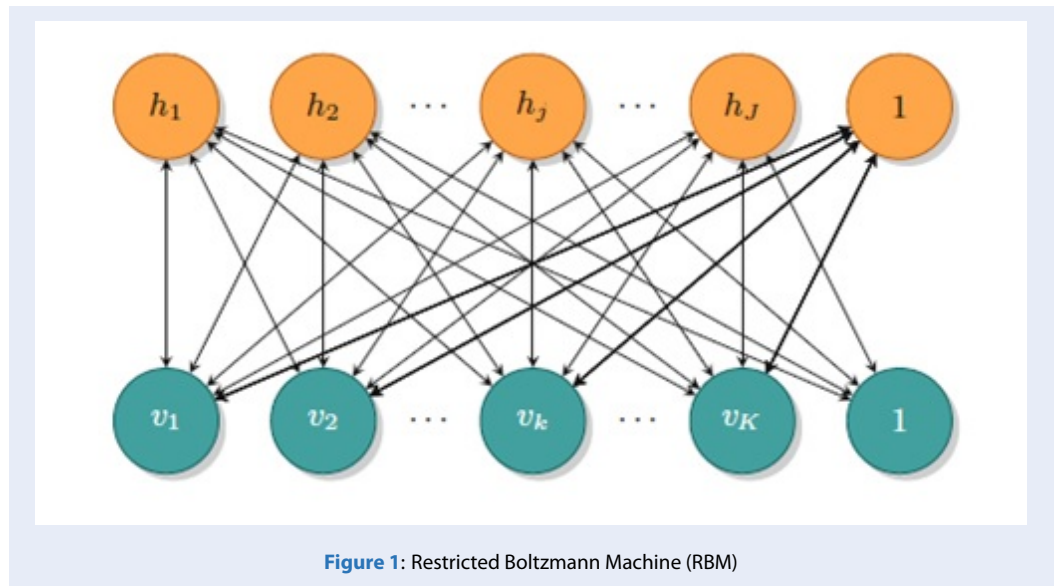
where $p(v_k = 1 | h) = \sigma(c_k + \sum_j W_{jk} h_j)$

Notice that σ is the sigmoid function. Inference of hidden factor h given the observed v can be done because h is conditionally independent given v .

A DBN is a generative model with an input layer and an output layer, separated by l layers of hidden stochastic units. This multilayer neural network can be efficiently trained by composing RBMs in such a way that the feature activations of one layer are used as the training data for the next layer.

An energy-based model of RBMs can be trained by performing gradient ascent on the log-likelihood of the training data with respect to the RBMs parameters. This gradient is difficult to compute analytically. Markov Chain Monte Carlo methods are well-suited for RBMs. One iteration of the Markov Chain works well and corresponding to the following sampling procedure:

$$v_0 \xrightarrow{p(h_0|v_0)} h_0 \xrightarrow{p(v_1|h_1)} v_1 \xrightarrow{p(h_1|v_1)} h_1$$



where the sampling operations are schematically described. Rough estimation of the gradient using the above procedure is denoted by CD-k, where CD-k represents the Contrastive Divergence algorithm³ for performing k iterations of the Markov Chain up to v_k .

Chaos Theory

Given a univariate time series x_t , where $t = 1, 2, \dots, N$, the phase space can be reconstructed using the method of delays¹. The essence in this method is that the evolution of any single variable of a system is determined by the other variables with which it interacts. Information about the relevant variables is thus implicitly contained in the history of any single variable. On the basis of this idea, an equivalent phase space can be constructed by assigning an element of the time series x_t and its successive delays as coordinates of a new vector.

$$X_t = \{x_t, x_{t+\tau}, x_{t+2\tau}, \dots, x_{t+(m-1)\tau}\}$$

where X_t are the points of phase space, t is the *delay time* and m is *embedding dimension*. The dimension m of the reconstructed phase space is considered as the sufficient dimension for recovering the object without distorting any of its topological properties, thus it may be different from the true dimension of the space where this object lies. Both the t and m parameters must be determined from the time series.

To determine a reasonable time delay t , we can apply the *mutual information* method proposed by Fraser and Swinney¹¹. To determine the minimum sufficient embedding dimension m we can apply the *false nearest neighbor* method proposed by Kennel et al.².

To check whether a time series is chaotic or not, one needs to calculate the maximal Lyapunov exponent. Rosenstein et al.¹² proposed a method to calculate the largest Lyapunov exponent from an observed time series.

Related Work

In our previous work¹⁰, we proposed an efficient method of chaotic time series prediction using Radial Basis Function (RBF) network, a special kind of shallow neural networks.

The RBF network is characterized by a set of inputs and a set of outputs. Between the inputs and outputs there is a layer of hidden units, each of which implements a *radial basis function*. Various functions have been tested as activation function for RBF network. Gaussian function is often used to activate the hidden layer. The nodes in the hidden layer operate on the distance from an applied input vector to an internal parameter vector, called a *center*. The output layer implements a weighted sum of hidden-unit outputs. The mapping function is given by:

$$p_j(X) = \sum_{i=1}^n w_{ij} \phi_i(\|X - c_i\|) \quad (7)$$

for $j = 1, 2, \dots, l$, where X is the input m -dimensional vector, $p_j(X)$ is the output of the j -th unit, w_{ij} are the output weight from the i -th hidden unit to the j -th output unit, n is the number of hidden units, ϕ_i is the i -th radial basis function at the i -th hidden node. If Gaussian function is used as radial basis function, ϕ_i is defined as follows.

$$\phi_i(\|X - c_i\|) = \exp\left(-\frac{1}{2} \left(\frac{\|X - c_i\|}{\sigma_i}\right)^2\right) \quad (8)$$

where c_i is the center and σ_i is the width of the i -th hidden unit, respectively.

RBF network is trained using unsupervised and supervised learning methods. The unsupervised method is implemented between input to hidden layers and supervised method is implemented from hidden to output layer. Clustering algorithms are capable of finding cluster centers that best represents the distribution of data in the first stage of the training. There are two alternative heuristics for finding width factors (13)

In¹⁰, we compared the performance of RBF network to that of MLP network on several real and synthetic datasets of chaotic time series. Experimental results revealed that RBF network with phase space reconstruction outperforms MLP networks in chaotic time series prediction.

DEEP LEARNING METHOD FOR CHAOTIC TIME SERIES PREDICTION: DBN

Inspired with the work by Kumemoto et al.⁹, we use the DBN model which composes one or two RBMs and MLP to forecast chaotic time series. The DBN model is used for feature learning and the MLP is for prediction. The forecasting DBN model is described in Figure 2.

As for the number of input nodes (at the first visible layer of RBM(s)) of DBN, we determine this parameter by using the embedding dimension (m) that we obtained when applying the phrase-space reconstruction method for each chaotic dataset. Each input node has one external input which represents the elements of X_t , i.e., $x_t, x_{t+\tau}, x_{t+2\tau}, \dots, x_{t+(m-1)\tau}$. That means we combine DBN model with chaos theory in chaotic time series prediction.

As for the activation function used in DBN, we use RELU function which is described by the following formula:

$$RELU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

The training algorithm for our proposed DBN consists of two stages: an unsupervised learning and a supervised learning.

The unsupervised learning stage is the Contrastive Divergence (CD) algorithm³ used for training the RBM(s). The CD algorithm progresses on a layer-by-layer basis. First, a RBM is trained directly on the input data. Hence, the neurons in the hidden layer of the RBM can capture the important features of the input data. The activations of the trained features are then used as “input data” to train a second RBM.

The supervised learning stage is the back-propagation algorithm used for training the MLP.

EXPERIMENTAL EVALUATION

In this experiment, we compare the DBN method for chaotic time series forecasting to the method using RBF network. We implemented the DBN forecasting method with Tensorflow framework (using Python language)¹⁴ and the RBF method with Microsoft Visual C#, .Net framework 4.5 and conducted the experiments on a Core i5 2.4 GHz, RAM 8GB PC.

In this study, the mean absolute error (MAE), the mean squared error (MSE) and the mean absolute percentage error (MAPE) are used as evaluation criteria. The formula for MAE, MSE and MAPE are given as follows:

$$\begin{aligned} MAE &= \frac{1}{n} \sum_{t=1}^n |\hat{y}_t - y_t| \\ MSE &= \frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2 \\ MAPE &= \frac{1}{n} \sum_{t=1}^n \frac{|\hat{y}_t - y_t|}{y_t} \end{aligned}$$

where n is the number of observations, y_t is the actual value in time period t , and \hat{y}_t is the forecast value for time period t .

Datasets and Parameter Setting

The main purpose of this study is to evaluate the performance of DBN in forecasting not only on synthetic but also real world chaotic time series. This follows the tradition of evaluating the proposed methods in chaotic time series prediction. Here, the tested datasets consist of 3 synthetic chaotic time series datasets and 4 real world chaotic time series datasets. All these datasets are commonly-used by the research community in chaotic time series prediction. They are described as follows.

1. This dataset is derived from the Lorenz system, given by the three differential equations:

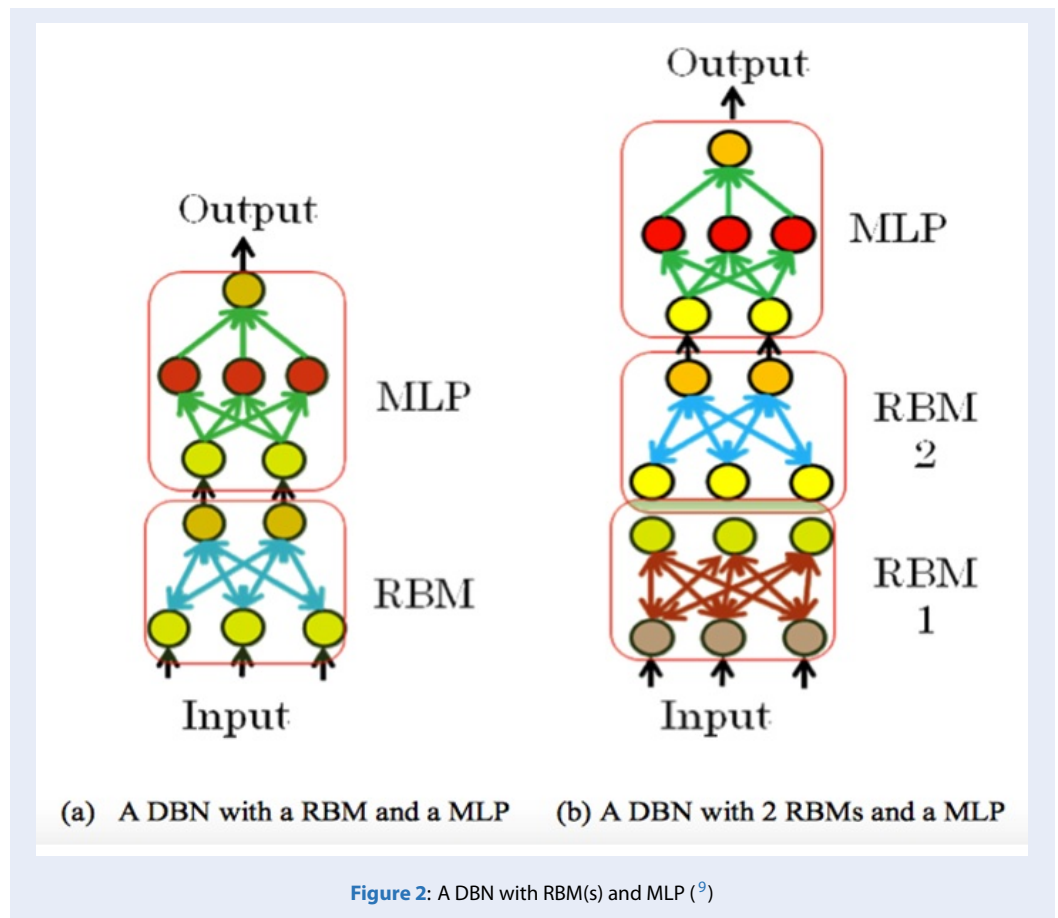
$$\begin{cases} \frac{dx}{dt} = a(y-x) \\ \frac{dz}{dt} = xy - cz \\ \frac{dy}{dt} = x(b-z) - y \end{cases} \quad (9)$$

where, $a = 10$, $b = 28$, and $c = 8/3$. This time series consists of 1000 data points.

2. This dataset is derived from the Mackey-Glass system, given by the following differential equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \quad (10)$$

where $a = 0.2$, $b = 0.1$, $c = 10$, $t = 17$ and $x_0 = 1.2$. This time series consists of 1001 data points.



3. This dataset is derived from the Rossler system, given by the three differential equations:

$$\begin{cases} \frac{dx}{dt} = -z - y \\ \frac{dz}{dt} = b + z * (x - c) \\ \frac{dy}{dt} = x + a * y \end{cases} \quad (11)$$

where $a = 0.15$, $b = 0.2$ and $c = 10$. This time series consists of 8192 data points.

4. Monthly sunspot numbers from January of 1749 to March of 1977. (This dataset from the web site: <http://sidc.oma.be>). This time series is in the field of astronomy and is a widely-used benchmark dataset in evaluation of several proposed methods for chaotic time series prediction. It consists of 305 data points.

5. Monthly CPI (Consumer Price Index) in Spain from January of 1960 to June of 2005. (This dataset from: <http://www.bde.es/bde/en/>). This time series consists of 535 data points.

6. Monthly exchange rates US dollar/British Pound (USD/GBP) from January 1981 to July of 2005. (This

dataset from the web site: <http://www.bde.es/bde/en/>). This time series consists of 295 data points.

7. Daily close prices of IBM stock from June of 1959 to June of 1960. (This dataset from the web site: <http://datamarket.com/data/set/>). This time series consists of 255 data points.

Figure 3 shows the plots of the three synthetic datasets. Figure 4 shows the plots of the four real world datasets.

With four real world datasets (Sunsports, CPI, USD/GBP, IBM) we use Lyapunov exponent to check whether each of time series is chaotic or not. The test shows that all four datasets possess the chaotic characteristics.

In this work, we estimate the embedding dimension and compute Lyapunov exponents by using the *tseriesChaos* package in the R software (website: <https://CRAN.R-project.org/package=tseriesChaos>).

In the experiment, we use RBF network in two versions: RBF with chaos theory (denoted as RBF-2) and RBF without chaos theory (denoted as RBF-1). For RBF-2, we have to determine the embedding dimen-

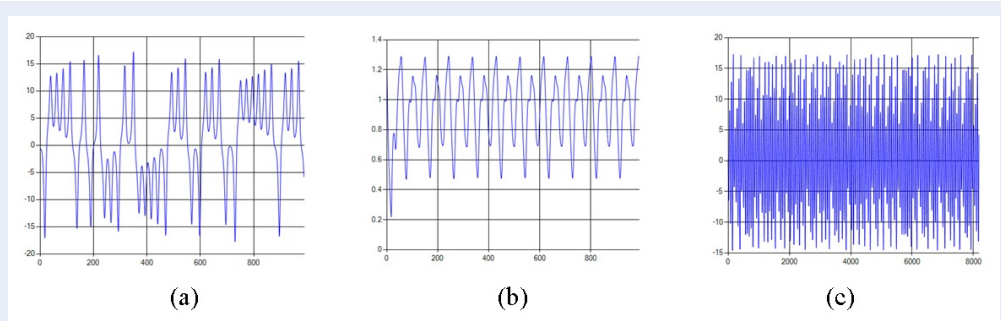


Figure 3: Three datasets: (a) Lorenz; (b) Mackey-Glass; (c) Rossler

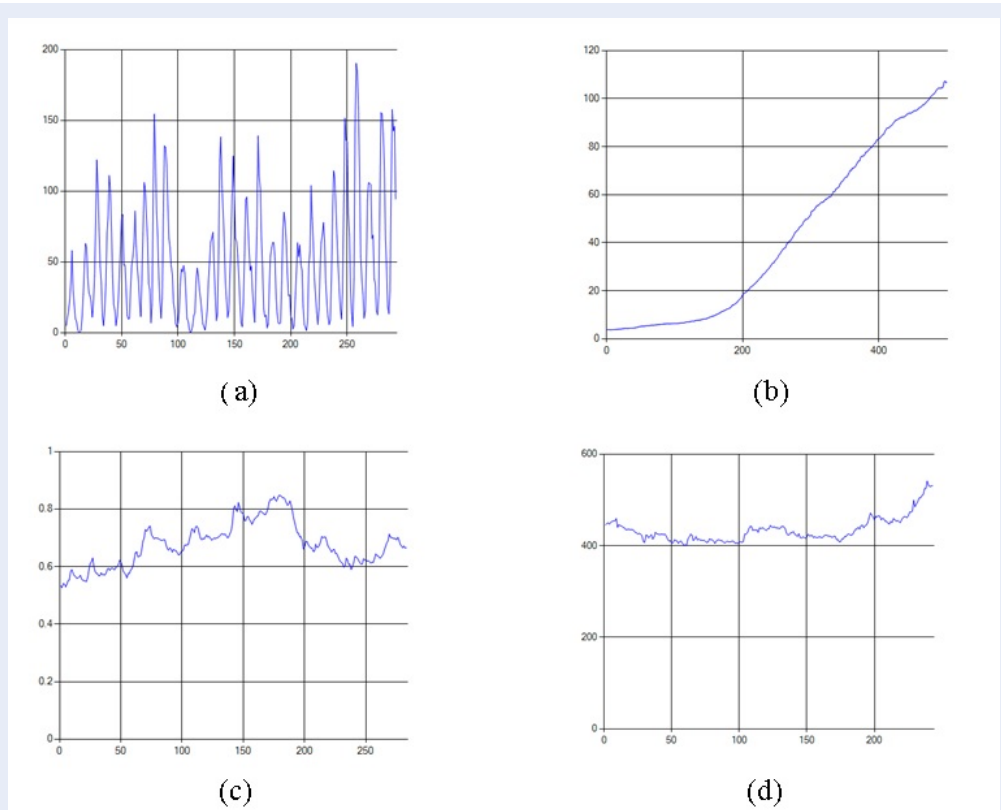


Figure 4: Four datasets (a) Sunspots; (b) CPI; (c) USD/GBP; (d) IBM Stock prices

sion m and the time delay t for each chaotic time series.

For all datasets, the RBF has the maximum number of learning iterations set to 1000. The parameter values for two versions of RBF network with all datasets are reported in Table 1. In Table 1, η_3 is the learning rate for output weights, η_2 is the learning rate for centers and η_1 is the learning rate for width factors.

The parameter values of DBN model for all datasets are reported in Table 2. As suggested in the paper⁹,

we set the number of RBMs in DBN model to 1 or 2. Except for Sunspots dataset, we use only one RBM in DBN for all the datasets and use two RBMs only for the Sunspots dataset. As for the number of input nodes (at the first visible layer of RBM(s)) of DBN, we determine this parameter by using the embedding dimension (m) that we obtained when applying the phrase-space reconstruction for each chaotic time series. That means the number of input nodes of the DBN model is the same as that of RBF network with

chaos theory (RBF-2) for each dataset. As for the number units in hidden layer of RBM and the number of units in hidden layer of MLP and all the other remaining parameters, we have to determine them through experiment.

As for the CD-k algorithm to pre-train the DBN, we set the number of iterations k to 2.

Each dataset is divided into two sets: training set and test set. The training set and test set for each of the seven datasets are given in Table 3.

Experimental Results

The experimental results on prediction accuracy are reported in Table 4. The prediction errors of DBN model are shown in column 3. The prediction errors of the two versions of RBF network method are shown in the columns 4 and 5.

From the experimental results in Table 4, we can see that the prediction errors MAE, MSE, MAPE of DBN method are always much lower than those of the two RBF methods in most of the tested datasets (5 out of 7). That means the DBN method achieves the best prediction accuracy for all seven datasets. However, for the USB/GBP dataset (length= 295) and IBM dataset (length=255), DBN still brings out lower MAE and MSE errors, but has MAPE error a little bit higher than the two other methods. We conjecture that for these two times series datasets of small size, DBN cannot manifest its strength in comparison to RBF network, a shallow form of neural networks. This observation implies that deep neural networks, such as DBNs, can exhibit their superior performance especially in working with big data rather than small size data.

The training times (in milliseconds) of the three methods over seven datasets are given in Table 5. From Table 5, we can see that the training time of DBN is much higher than those of the two versions of RBF method in most of the datasets.

DISCUSSION

In all experimented time series, RBF-2 model shows better prediction performance than RBF-1 model. This implies that by applying chaos theory (namely, phase-space reconstruction), RBF-2 model outperforms RBF-1 model which does not use chaos theory in chaotic time series prediction.

It should be noted that in this work the parameters of RBF networks (RBF1, RBF2) and the hyper-parameters of DBN model are selected only after a few trial-and-error tests on a limited number of parameter values. As for DBN model, some previous works pointed out that its hyper-parameters, such as

learning rate of RBM, the number of hidden units for MLP, learning rate of MLP, etc. have a great influence on the performance of this deep learning model⁽⁹⁾. Therefore, the hyper-parameters of DBN model should be fine-tuned to its optimum before it can be used. One of the methods to handle this problem is applying some meta-heuristics to find suitable values of hyper-parameters in DBNs for forecasting time series.

CONCLUSIONS

In this paper, we present and evaluate extensively a method using DBN in phase-space prediction of chaotic time series. We compare the performance of DBN with that of RBF network in three synthetic datasets and four real world datasets of chaotic time series. Experimental results obtained reveal that the DBN method performs better than the RBF network, a shallow form of neural networks, in most of the datasets. Experimental results also show that deep neural networks, such as DBNs, can exhibit their superior performance especially in working with big data rather than small size data.

As for future work, we intend to apply some meta-heuristics such as Particle Swarm Optimization (PSO) or Harmony Search in finding suitable values of hyper-parameters in DBNs for forecasting chaotic time series.

LIST OF ABBREVIATION USED

- MPL: Multi-Layer Perceptron
- DBN: Deep Belief Network
- SVM: Support vector machine
- RBF: Radial Basis Function
- PSO: Particle Swarm Optimization
- MAE: Mean absolute error
- MSE: Mean squared error
- MAPE: Mean absolute percentage error
- CD: Contrastive Divergence
- USD: United States Dollar
- GBP: General British Pound
- CPI: Consumer Price Index
- IBM: IBM Stock Prices

COMPETING INTEREST

The authors have declared that no competing interests exist.

AUTHOR CONTRIBUTION

Duong Tuan Anh has proposed the main ideas of the article, collected the datasets and wrote the article. Ta Ngoc Huy Nam has implemented the comparative models, conducted the experiments and reported the experimental results.

Table 1: Parameters of two versions of RBF network

Dataset	RBF-1	RBF-2
Lorenz	(5-5-0) Error threshold: 0.001 η_3 : 0.1, η_2 : 0.00001, η_1 : 0.9	(10-10-1) Error threshold: 0.001 η_3 : 0.015, η_2 : 0.00002, η_1 : 0.015 Time delay t : 7
Mackey-Glass	(10-10-1) Error threshold: 0.001 η_3 : 0.0025, η_2 : 0.0002, η_1 : 0.025	(10-10-1) Error threshold: 0.001 η_3 : 0.2, η_2 : 0.3, η_1 : 0.3 Time delay t: 17
Rosler	(10-10-1) Error threshold: 0.001 η_3 : 0.02, η_2 : 0.0001, η_1 : 0.1	(10-10-1) Error threshold: 0.001 η_3 : 0.04, η_2 : 0.02, η_1 : 0.01 Time delay t : 15
Sunspots	(4-4-1) Error threshold: 0.001 η_3 : 0.001, η_2 : 0.001, η_1 : 0.002	(5-5-1) Error threshold: 0.0001 η_3 : 0.02, η_2 : 0.001, η_1 : 0.02 Time delay t: 2
CPI	(2-2-1) Error threshold: 0.0001 η_3 : 0.0045, η_2 : 0.0035, η_1 : 0.006	(2-2-1) Error threshold: 0.0001 η_3 : 0.004, η_2 : 0.0048, η_1 : 0.0065 Time delay t: 77
USD/GBP	(4-4-1) Error threshold: 0.0001 η_3 : 0.004, η_2 : 0.003, η_1 : 0.009	(4-4-1) Error threshold: 0.0001 η_3 : 0.04, η_2 : 0.003, η_1 : 0.01 Time delay t: 17
IBM	(5-5-1) Error threshold: 0.0001 η_3 : 0.004, η_2 : 0.0001, η_1 : 0.01	(5-5-1) Error threshold: 0.0001 η_3 : 0.045, η_2 : 0.0001, η_1 : 0.009 Time delay t: 15

ACKNOWLEDGEMENT

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number C2019-20-17.

REFERENCES

1. Takens F. Detecting strange attractors in turbulence, Dynamical Systems and Turbulence, Warwick 1980, London. Springer. 1981;p. 366–381. Available from: <https://doi.org/10.1007/BFb0091924>.
2. Kennel MB, Brown R, Abarbanel HDI. Determining embedding dimension for phase-space reconstruction using a geometrical construction. PHYSICAL REVIEW A. 1992;45(6):3403–3411. PMID: 9907388. Available from: <https://doi.org/10.1103/PhysRevA.45.3403>.
3. Hinton GE, Osindero S, Teh YW. A Fast Learnig Algorithm for Deep Belief Nets. Neural Computing. 2006;18:1527–1554. PMID: 16764513. Available from: <https://doi.org/10.1162/neco.2006.18.7.1527>.
4. Chong E, Chang C, Park FC. Deep learning network for stock market analysis and prediction. Methodologies, Data Representation and Case studies, Expert Systems with Applications. 2017;83:87–205. Available from: <https://doi.org/10.1016/j.eswa.2017.04.030>.
5. Zheng J, Fu X, Zhang G. Research on exchange rate forecasting based on deep belief network, Neural Computing and Applications. Springer. 2017; Available from: <https://doi.org/10.1007/s00521-017-3039-z>.
6. Li C, Bai Y, Zeng B. Deep learning architecture for daily reservoir inflow forecasting. Water Resource Management. 2016;30:5145–5161. Available from: <https://doi.org/10.1007/s11269-016-1474-8>.
7. Cao TN, Duong HN, Tran HV, Snasel V. Predicting daily river runoff by deep belief networks. Proc of Int Conf on Information and Convergence Technology for Smart Society, Ho Chi Minh City. 2016;.
8. He W. Deep neural network based load forecast. Computing Modelling and New Technologies. 2014;18(3):258–262.
9. Kuremoto T, Obayashi M, Kobayashi K, Hirata T, Mabu S. Forecast Chaotic Time Series Data by DBNs, Proc. 7th Int. Congress on Image & Signal Processing. 2014; Available from: <https://doi.org/10.1109/CISP.2014.7003950>.
10. Truc NV, Anh DT. Chaotic Time Series Prediction Using Radial Basis Function Networks, Proc. of 4th Int. Conf. on Green Technology and Sustainable Development (GTSD2018), IEEE, November 23–24. HCMC Univ of Technology & Education, Vietnam. 2018;p. 753–758. Available from: <https://doi.org/10.1109/GTSD.2018.8595677>.
11. Fraser AM, Swinney HL. Independent Coordinates for Strange Attractors from Mutual Information. Physia D;33:1134–1986. PMID: 9896728. Available from: <https://doi.org/10.1103/>

Table 2: Parameters for DBN (RBM+MLP)

Dataset	DBN (RBM+MLP)
Lorenz	The max number of epochs for RBM: 100, Learning rate of RBM: 0.01, The max number of epochs for MLP: 1500, The number of hidden units for MLP: 5 Learning rate of MLP: 0.03
Mackey-Glass	The max number of epochs for RBM: 100, Learning rate of RBM: 0.01, The max number of epochs for MLP: 1500, The number of hidden units for MLP: 10 Learning rate of MLP: 0.03
Rosler	The max number of epochs for RBM: 20, Learning rate of RBM: 0.01, The max number of epochs for MLP: 500, The number of hidden units for MLP: 10 Learning rate of MLP: 0.03
Sunspots	The max number of epochs for RBM: 50, Learning rate of RBM: 0.05, The max number of epochs for MLP: 2500, The number of hidden units for MLP: 5 Learning rate of MLP: 0.008
CPI	The max number of epochs for RBM: 150, Learning rate of RBM: 0.01, The max number of epochs for MLP: 1200, The number of hidden units for MLP: 2 Learning rate of MLP: 0.005
USD/GBP	The max number of epochs for RBM: 400, Learning rate of RBM: 0.005, The max number of epochs for MLP: 4000, The number of hidden units for MLP: 2 Learning rate of MLP: 0.0005
IBM	The max number of epochs for RBM: 400, Learning rate of RBM: 0.01, The max number of epochs for MLP: 4000, The number of hidden units for MLP: 2 Learning rate of MLP: 0.0005

Table 3: Training set and test set for each of 7 datasets

Dataset	Length	Training set	Test set
Lorenz	1000	800	200
Mackey-Glass	1001	800	201
Rosler	8192	8000	192
Sunspots	305	288	17
CPI	535	481	54
USD/GBP	295	285	10
IBM	255	245	10

PhysRevA.33.1134.

12. Rossenstein MT, Colins JJ, Luca CJ. Reconstruction expansion as a geometry-based framework for choosing proper delay times. *Physica D*. 1994;73(1):82–98. Available from: [https://doi.org/10.1016/0167-2789\(94\)90226-7](https://doi.org/10.1016/0167-2789(94)90226-7).

13. Verleysen M, Hlavackova K. Learning in RBF Networks. *Proc of Int Conf on Neural Networks (ICNN)*. 1996;p. 199–204.

14. Pattanayak S. *Pro Deep Learning with Tensorflow*. Apress. 2017;Available from: <https://doi.org/10.1007/978-1-4842-3096-1>.

Table 4: Prediction Errors of the three methods over 7 datasets

Dataset		DBN	RBF-1	RBF-2
Lorenz	MAE	0.012	4.15	1.18
	MSE	0.000002	20.32	2.51
	MAPE	0.47%	45.4%	18.35%
Mackey-Glass	MAE	0.0047	0.14	0.01
	MSE	0.000035	0.03	0.013
	MAPE	0.78%	16.43%	11.89%
Rossler	MAE	0.000112	3.63	1.94
	MSE	0.000006	17.52	6.42
	MAPE	0.35%	137.17%	51%
Sunsports	MAE	0.0392	33.80	10.82
	MSE	0.0026	1843.37	150.61
	MAPE	18.69%	72.27%	20.49%
CPI	MAE	0.0047	2.13	1.84
	MSE	0.000036	5.49	4.46
	MAPE	0.52%	1.91%	1.66%
USD/GBP	MAE	0.0023	0.15	0.01
	MSE	0.00018	0.00027	0.0003
	MAPE	4.87%	2.12%	1.59%
IBM	MAE	0.03	14.35	5.49
	MSE	0.001608	257.647	41.46
	MAPE	3.51%	2.49%	1.06%

Table 5: Training time of the three methods on 7 datasets

Dataset	DBN	RBF-1	RBF-2
Lorenz	513000	133	4905
Mackey-Glass	151000	362	191
Rossler	395000	362	191
Sunsports	66500	1028	3363
CPI	122000	336	2189
USD/GBP	184000	36	57
IBM	219000	637	534

Dự báo dữ liệu chuỗi thời gian có tính hỗn loạn dựa vào mạng deep belief network: một đánh giá thực nghiệm

Tạ Ngọc Huy Nam, Dương Tuấn Anh*



Use your smartphone to scan this QR code and download this article

TÓM TẮT

Dữ liệu chuỗi thời gian có tính hỗn loạn xuất hiện nhiều trong các lãnh vực ứng dụng thực tế như tài chính, môi trường, khí tượng thủy văn, giao thông, thời tiết.v.v. Chuỗi thời gian có tính hỗn loạn được xem như sinh ra từ chuyển động tắt định của một hệ thống phi tuyến. Một hệ thống có tính hỗn loạn thường nhạy cảm với những điều kiện ban đầu: những điểm mà ngẫu nhiên ban đầu gần nhau lại trở nên cách xa nhau với mức độ tăng hàm mũ theo thời gian. Do đó dự báo dữ liệu chuỗi thời gian có tính hỗn loạn là một công tác đầy thách thức. Công tác dự báo sử dụng các kỹ thuật thống kê truyền thống, giải thuật k-lần cận gần nhất, mạng perceptron đa tầng (MPL), mạng nơ ron hồi quy, mạng hàm căn bản xuyên tâm (Radial-Basis-Function-RBF) và máy vector hỗ trợ, thường không đem lại kết quả dự báo đáng tin cậy đối với chuỗi thời gian có tính hỗn loạn. Trong bài báo này, chúng tôi khảo sát bằng thực nghiệm việc dự báo dữ liệu chuỗi thời gian có tính hỗn loạn bằng cách kết hợp mạng nơ ron học sâu Deep Belief Network (DBN) với lý thuyết hỗn loạn. Mạng nơ ron DBN nên được dùng để dự báo chuỗi thời gian có tính hỗn loạn. Đầu tiên, chuỗi thời gian có tính hỗn loạn được phân tích bằng cách tính số mũ Lyapunov cực đại, tái cấu trúc chuỗi thời gian dựa vào xây dựng không gian pha và xác định số chiều nhúng và độ trễ thời gian tốt nhất. Chúng tôi so sánh phương pháp đề xuất với mạng nơ ron RBF, một phương pháp hiệu quả để dự báo dữ liệu chuỗi thời gian có tính hỗn loạn. Chất lượng dự báo của hai mô hình được khảo sát dựa vào các tiêu chí đánh giá: sai số tuyệt đối trung bình (MAE), sai số bình phương trung bình (MSE) và sai số phần trăm tuyệt đối trung bình (MAPE). Kết quả thực nghiệm trên nhiều bộ dữ liệu chuỗi thời gian tổng hợp và chuỗi thời gian thực tế có tính hỗn loạn cho thấy phương pháp sử dụng mạng Deep Belief Network hữu dụng cho việc dự báo dữ liệu chuỗi thời gian có tính hỗn loạn vì phương pháp đem lại độ chính xác dự báo cao hơn mạng nơ ron RBF.

Từ khoá: Mạng Deep Belief Network, Máy Boltzmann Giới Hạn, chuỗi thời gian có tính hỗn loạn, mạng RBF, dự báo

Trường Đại học Bách Khoa,
ĐHQG-HCM

Liên hệ

Dương Tuấn Anh, Trường Đại học Bách
Khoa, ĐHQG-HCM

Email: dtanhcse@gmail.com

Lịch sử

- Ngày nhận: 28-8-2019
- Ngày chấp nhận: 26-9-2019
- Ngày đăng: 04-12-2020

DOI :10.32508/stdjet.v3iSI1.571



Bản quyền

© ĐHQG Tp.HCM. Đây là bài báo công bố mở được phát hành theo các điều khoản của the Creative Commons Attribution 4.0 International license.



Trích dẫn bài báo này: Nam T N H, Anh D T. Dự báo dữ liệu chuỗi thời gian có tính hỗn loạn dựa vào mạng deep belief network: một đánh giá thực nghiệm. *Sci. Tech. Dev. J. - Eng. Tech.*; 3(S11):SI102-SI112.