University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part B

Faculty of Engineering and Information
Sciences

2019

A research of Monte Carlo optimized neural network for electricity load forecast

Binbin Yong

Lanzhou University, yongbb14@lzu.edu.cn

Liang Huang

Lanzhou University

Fucun Li

University of Wollongong, fl626@uowmail.edu.au

Jun Shen

University of Wollongong, jshen@uow.edu.au

Xin Wang

Lanzhou University

See next page for additional authors

Follow this and additional works at: <https://ro.uow.edu.au/eispapers1>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

A research of Monte Carlo optimized neural network for electricity load forecast

Abstract

In this paper, we apply the Monte Carlo neural network (MCNN), a type of neural network optimized by Monte Carlo algorithm, to electricity load forecast. Meanwhile, deep MCNNs with one, two and three hidden layers are designed. Results have demonstrated that three-layer MCNN improves 70.35% accuracy for 7-week electricity load forecast, compared with traditional neural network. And five-layer MCNN improves 17.24% accuracy for 7-week forecast. This proves that MCNN has great potential in electricity load forecast.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Yong, B., Huang, L., Li, F., Shen, J., Wang, X. & Zhou, Q. (2019). A research of Monte Carlo optimized neural network for electricity load forecast. The Journal of Supercomputing, Online First 1-14.

Authors

Binbin Yong, Liang Huang, Fucun Li, Jun Shen, Xin Wang, and Qingguo Zhou

Noname manuscript No.
(will be inserted by the editor)

A research of monte carlo optimized neural network for electricity load forecast

Binbin Yong · Liang Huang · Fucun Li ·
Jun Shen · Xin Wang · Qingguo Zhou*

Received: date / Accepted: date

Abstract In this paper, we apply the Monte Carlo Neural Network (MCNN), a type of neural network optimized by Monte Carlo algorithm, to electricity load forecast. Meanwhile, deep MCNNs with one, two and three hidden layers are designed. Results have demonstrated that 3-layers MCNN improves 70.35% accuracy for 7 weeks electricity load forecast, compared with traditional neural network. And 5-layers MCNN improves 17.24% accuracy for 7 weeks forecast. This proves that MCNN has great potential in electricity load forecast.

Keywords Monte Carlo · neural network · electricity load forecast · deep MCNNs

1 Introduction

Electricity load forecast, as the core of electricity scheduling, plays a vital role in meeting the basic needs of modern human life. It has been widely studied in the past few decades. However, studies in the literature have shown

* Corresponding Author

Binbin Yong · Xin Wang · Qingguo Zhou

School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China

Tel.: +86-0931-8912025

Fax: +86-0931-8912025

E-mail: {yongbb14, xwang2016, zhouqg}@lzu.edu.cn

Liang Huang

School of Physical Science and Technology, Lanzhou University, Lanzhou, Gansu, China

E-mail: huangl@lzu.edu.cn

Fucun Li · Jun Shen

School of Computing and Information Technology, University of Wollongong, NSW, Australia, 2522

E-mail: fl626@uowmail.edu.au, jshen@uow.edu.au

that, as a problem of time series forecast, electricity forecast is prone to be influenced by many environmental factors, which result in lacking accuracy and stability in practice. In the past century, the electricity has gradually permeated almost everywhere. From daily life to the expansion of multinational corporations, electricity has accompanied individuals and has served for the national regeneration, which brings forth extensive research interests for electricity forecast. Meanwhile, numerous forecast models were proposed, in which back propagation neural network (BPNN) had been proved effective for electricity load forecast [7].

In fact, the application of BPNN depends on the so-called Empirical Risk Minimization (ERM) strategy [13]. In this case, the overall cost, which represents the error between actual values and forecasting values, usually has distinct trends in training samples and test samples. And BPNN could possibly perform well on training samples, while it might result in a poor performance in new situations, which is called overfitting. In order to eliminate the overfitting phenomena, many approaches were proposed [5, 8, 14]. A common technology against overfitting is the cross validation technology, which uses validation samples to select the best training model [18]. However, since the order of the data is important, cross validation might be problematic for time series forecast [1]. Dropout technology [11] is a newly proposed way to prevent BPNN from overfitting, which randomly disconnects part of connections of the network in training process to enhance the extensibility. Another strategy against overfitting is to use less hidden nodes to reduce the complexity of the BPNN, so as to control the risk of overfitting. However, fewer hidden nodes may bring in a simplistic model, which results in a less accurate result. Regularization technology, which controls the weight fluctuations to reduce the weight sensitivity of forecast model, is widely used in machine learning [9], especially BPNN, to restrain overfitting.

The cross validation and dropout technologies are fairly suitable for conditions with large samples. Unfortunately, many time series forecast problems, such as electricity forecast, may only have a few samples in practice. Hence, the structure based approaches, such as hidden nodes selection and regularization, are widely developed to improve the generalization ability. Support Vector Machine (SVM) [3] is a type of structure based approaches based on the Statistical Learning Theory (SLT) [13]. It has solid theoretical foundation and tries to find a hyperplane that separates samples with a largest margin, which is proved to be very effective for small samples [2, 10, 12]. However, in the case when the training samples are very small, SVM may select noise samples as support vectors, which ends with a poor forecasting model. Hence, some researchers tried to find models that ensemble both the advantages of BPNN and SVM.

In 2016, Zhao researched a type of neural network [19], which has a basic structure of a three-layer neural network. Meanwhile, as a mixer of neural network and SVM, the neural network is optimized by Monte Carlo algorithm (MC), and it has been successfully applied in small samples forecasts [16, 17]. In this paper, the MC optimized neural network (MCNN) is applied

into electricity load forecast, and we further study the deep MCNNs with more than one hidden layer.

The remainder of this paper is organized as follows. The details of MCNN is presented in Section 2. Section 3 introduces replication kinetics idea, and applies it into electricity load forecast. The experiments and analysis are given in Section 4. At last, Section 5 concludes our work.

2 Monte carlo neural network model

The whole structure of a 3-layer back propagation neural network (BPNN) is shown in Fig. 1 (a). It adjusts weight matrix W_1 and W_2 with back propagation algorithm, based on the ERM strategy, which trains the neural network via reducing the overall cost with gradient descent algorithm. The structure of MCNN model is shown in Fig. 1 (b). In terms of topology, MCNN is in fact equivalent to a traditional neural network. Hence, MCNN is able to fit any function with enough hidden nodes [6].

2.1 Structure of MCNN

For MCNN with one hidden layer, we assume that a neural network model consists of M input nodes, N hidden nodes and L output nodes. The outputs of hidden nodes h_n and output layer y_l are given as Eq. (1).

$$\begin{cases} h_n = f(W_\beta[n] \cdot (\sum_{i=1}^M W_1[n][i] \cdot x_i + W_b[n])) \\ y_l = \sum_{j=1}^N W_2[l][j] \cdot h_n \end{cases} \quad (1)$$

In Eq. (1), x_i denotes the i th input feature, and $W_1[n][i]$ is the weight connected between the n th hidden node and the i th input node. $W_b[n]$ denotes the bias of n th hidden node. $W_\beta[n]$ denotes the control parameter of n th hidden node, which will be discussed in detail in the next subsection. f is the activation function of hidden nodes. Similar to neural network, common activation functions of MCNN include *sigmoid* and *tanh* functions. h_n is the output of the n th hidden node. $W_2[l][j]$ is the weight connected between the l th output node and the j th hidden node, and the outputs of hidden nodes are linearly connected to the output nodes. In reality, the weight matrix W_2 is fixed after randomly initialized as -1 or 1. Hence, adjustable weight matrices of MCNN contain W_β , W_1 and W_b .

In contrast to traditional neural network, in the design of MCNN, the control parameter β is imported into the hidden nodes to reduce the design risk, as shown in Eq. (2).

$$y_l'' = \frac{d^2 y_l}{dx_i^2} = W_\beta[n]^2 \cdot W_2[l][j] \cdot W_1[n][i]^2 \cdot f''(\sum_{i=1}^M W_1[n][i] \cdot x_i + W_b[n]) \quad (2)$$

Fig. 1 The structures of BPNN and MCNN

In practice, the second derivative of output y_l with respect to feature x_i (y_l'') for MCNN is derived as Eq. (2), whose value range is controlled by ranges of W_β, W_2, W_1 and f'' . Meanwhile, W_1, W_2 and f'' are all limited in small ranges in the design of MCNN. Hence, by adjusting the range of W_β (denoted as con-

control parameter β), neural network model will keep robust for small fluctuations of the input feature vector. That is to say, control parameter β is used as a hyper-parameter of MCNN. For MCNN with more than one hidden layer, each hidden layer has the same hyper-parameter β .

As an example, $\sin$ function fitting is tested by MCNN with different hyper-parameter β . In order to fit $\sin$ function, the MCNN model is set as one input node (represent x) and one output node (represent $y = \sin(x)$). It has 100 hidden nodes with *sigmoid* activation function. Then, 20 training samples are used to train the MCNN model, and the fitting results are shown in Fig. 2. We can see that, with the decrease of β , the fitting curves are more and more rigid.

Fig. 2 Different β for $\sin$ function fitting by MCNN

In general, for regression problems, such as time series forecast, MCNN selects Eq. (3) as the cost function, in which $\hat{y}_i$ represents the i th forecasting output value and y_i is the i th actual value from the training samples. In this paper, we also import another three metrics to verify the effectiveness of MCNN model, which is Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), as defined in Eq.

(4). By using these these metrics, we can evaluate the forecasting results of the MCNN model comprehensively.

$$COST = \frac{1}{2} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (3)$$

$$\left\{ \begin{array}{l} MAPE = \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} \times 100\% \\ MAE = \frac{\sum_{i=1}^N |\hat{y}_i - y_i|}{N} \\ RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}} \end{array} \right. \quad (4)$$

The cost function (Eq. (3)) has the same trend with RMSE, which represents the discreteness of forecasting values and actual values. MAPE denotes the relative errors and MAE shows the mean error of the forecast models. Hence, these three metrics could evaluate the forecast models more comprehensively.

Specially, for electricity load forecast in this paper, it only needs one linear output node to represent the forecasting value of next time point. Hence, we will not discuss complex MCNN structures for classification and pattern recognition. More concretely, for MCNN with one hidden layer, the electricity load forecast model consists of 336 input nodes, 150 hidden nodes and 1 output nodes.

2.2 Monte Carlo algorithm

In this paper, MC algorithm is used to train the neural network. The basic idea of training MCNN is, that the change of a weight would be accepted while the overall cost decreases. Specifically, it randomly changes one weight of the weight matrices in a small deviation, and MCNN will accept the weight change while the overall cost is reduced. The flowchart of MC algorithm for training MCNN is shown in Fig. 3. Firstly, the algorithm needs to load training samples and randomly initialize the weight matrix W , which includes W_β , W_2 , W_1 and W_b . As mentioned above, W_2 is fixed after initialization. Next, the algorithm randomly selects a weight w from weight matrix W , and tries to change w in a range. Generally, the range is set as a small range, such as one tenth of the weight range, to maintain the stability of the network. The value of the small range is set as 0.1 in this paper. Then, if the overall cost decreases, the new w is accepted and the MC algorithm successfully updates its weight once. Instead, if the overall cost increases, the weight w is restored to its original value and the MC algorithm drops the update. From this perspective, MC algorithm is inefficient, compared with BP algorithm. However, modern computers are adequate for MC training algorithm, and there is also

Fig. 3 The process of Monte Carlo algorithm for training MCNN

method to optimize MC algorithm [15]. Also, BP often falls into local optimal solution, while MC is prone to avoid local optima because of its global optimization characteristic. With MC algorithm, the MCNN model will gradually converge to a stable model, which can predict robustly. Because overfitting can be suppressed by controlling parameters β , MCNN can be designed with more hidden nodes and hidden layers for small training samples, compared with traditional BPNN. For simplicity, in this paper, MCNN with one hidden layer, two hidden layers and three hidden layers are denoted as 3-MCNN, 4-MCNN and 5-MCNN, respectively.

3 Replication kinetics forecast

Most of traditional methods for electricity load forecast are based on point forecast. In this section, we will present the replication kinetics method for electricity load forecast, as shown in Fig. 4.

The idea of replication kinetics is based on the fact that a three-layer neural network with enough hidden nodes is able to describe any functions. Hence, for a dynamics system, we can collect the input-output pairs as dataset, by which we can train a neural network to represent the relationship between input and output of the dynamic system. Furthermore, the neural network could be used to forecast the behavior of the dynamics system. In fact, MCNN is very suitable for performing replication kinetics forecast, due to its robustness and stability.

Lorenz system [4] is a chaotic system described by Eq. 5. It is very sensitive for the initial values, and hence is difficult to forecast. As a test, we will use the replication kinetics principle to replicate the behavior of Lorenz system.

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (5)$$

In this system, σ , ρ and β are set as 12, 2 and 9, respectively. We select 50 continuous points in the Lorenz system as the input vector of the MCNN model, the next adjacent point is used as the output value. In this way, we move 100 steps and get 100 samples. These 100 samples are used as training samples of MCNN. Then, based on the trained MCNN model, the whole Lorenz system is evolved out.

In this paper, we focus on electricity load forecast. The electricity load data is collected from Queensland in Australia, which includes 3,024 data in 9 weeks and ranges from May 2, 2013 to July 3, 2013. The replication kinetics modeling for electricity load forecast is as follows: the first 336 data (first week) are used as the input vector, and the 337th data is used as the output, which forms the 1st sample. Then, the next 336 data (from 2nd to 337th) and the 338th data are used as the input vector and the output of the second sample. In this way, 336 training samples are constructed by using 2 week electricity load data.

Based on these 336 training samples, we train a neural network to replicate the internal variation of electricity load data, which can be used to predict the electricity load in future. In this paper, we mainly combine replication kinetics process with MCNN to forecast electricity load. Meanwhile, as a comparison, traditional BP is also used to execute replication kinetics for electricity load forecast.

4 Experiments and analysis

Fig. 4 The combining of MCNN and replication kinetics idea for modeling Lorenz system

In our experiments, we firstly test the effect of MCNN replication kinetics for reconstructing Lorenz system. Results are shown in Fig. 4. We can see that, for a chaotic system like Lorenz system, MCNN combining replication kinetics is able to evolve out the same shape. Meanwhile, from components $x(t)$, $y(t)$ and $z(t)$, we can see that MCNN can accurately predict short steps. Lorenz system is a simplified weather evolution system, which is in line with the actual weather forecast for short term prediction.

Based on this, we further test short term electricity load forecast, in which replication kinetics is executed by 3-MCNN, 4-MCNN, 5-MCNN and BPNN,

respectively. And we will compare the forecast results of these two types models based on their best network structures, which is shown in Table 1.

Table 1 Parameters of different forecast models

model	input	hidden layer	β	optimization
3-MCNN	336	400	0.01	MC
4-MCNN	336	80×80	0.03	MC
5-MCNN	336	$40 \times 40 \times 40$	0.2	MC
BPNN	336	5	*	BP

As seen in Table 1, with the increase of the number of network layers, the number of hidden nodes in each layer should be reduced and the value of control parameter β should increase. Many experiments were conducted to find the best network structures. For BPNN, it achieves the best forecast results with 5 hidden nodes, and more hidden nodes tend to worsen forecast results, which indicates that complex network structure and overfitting. On the contrary, 3-MCNN achieves its best structure with 400 hidden nodes, and more hidden nodes tend to lower forecast error. In summary, MCNN model could control overfitting by controlling parameter β . Meanwhile, more hidden nodes are used to improve the forecasting accuracy, and the forecast accuracy will keep stable when the number of hidden nodes is in excess of 400.

Table 2 Experimental results: forecast results of deep MCNN of different hidden nodes

MAPE(%)							
model	1-week	2-week	3-week	4-week	5-week	6-week	7-week
3-MCNN	3.4687	3.0894	2.9962	2.9802	3.1834	3.4532	3.6032
4-MCNN	3.0833	2.6898	2.6103	2.6216	2.7325	3.0333	3.0747
5-MCNN	2.4711	2.2312	2.4139	2.4634	2.6590	2.9726	2.9817
BPNN	3.4632	4.2318	6.3647	8.4591	10.066	11.262	12.153
MAE							
model	1-week	2-week	3-week	4-week	5-week	6-week	7-week
3-MCNN	203.20	177.70	172.56	169.23	182.02	200.51	208.61
4-MCNN	182.20	156.72	151.89	151.32	158.35	178.08	179.80
5-MCNN	147.32	130.60	140.11	141.31	153.15	173.48	173.89
BPNN	204.05	244.24	364.05	476.55	566.44	636.57	684.23
RMSE							
model	1-week	2-week	3-week	4-week	5-week	6-week	7-week
3-MCNN	250.38	220.75	215.57	211.73	229.56	261.60	271.28
4-MCNN	236.54	205.28	200.44	204.19	213.74	246.44	246.68
5-MCNN	196.15	178.19	190.90	193.06	207.18	238.79	237.26
BPNN	258.19	320.17	506.09	661.19	772.91	857.21	907.31

The detailed forecasting results by 3-MCNN, 4-MCNN, 5-MCNN and BPNN are shown in Table 2 and Fig. 5, 6, 7. Forecast curves for first week, fourth week and seventh week by four types models are shown in Fig. 5, 6 and 7. We can see that, BPNNs achieve large prediction error from the second week, while MCNNs keep stable forecast results for long 7-week forecast. 3-MCNN improves 70.35% accuracy for 7-weeks electricity load forecast. Meanwhile, with the increase of the number of layers, forecasting results of MCNN become more and more accurate. And 5-MCNN improves 17.24% accuracy for 7 weeks forecast, compared with 3-MCNN.

The MAPE, MAE and RMSE are plotted in Fig. 8. We can easily conclude that MCNN is stable for forecasting 7 weeks electricity load, and 5-MCNN has the lowest forecasting error. On the contrary, the forecast results of BPNN get worse and worse with the increase of forecast interval.

Overall, compared with traditional BPNN and point forecast, MCNN combined with replication kinetics needs less training samples (only 2 weeks). Moreover, point forecast model could only forecast the load of the next point. On the other hand, MCNN combined with replication kinetics forecasts long interval with stability.

Fig. 5 Experimental results: the forecast results of the 1st weeks by 3-MCNN, 4-MCNN, 5-MCNN and BPNN

Fig. 6 Experimental results: the forecast results of the 4th week by 3-MCNN, 4-MCNN, 5-MCNN and BPNN

Fig. 7 Experimental results: the forecast results of the 7th week by 3-MCNN, 4-MCNN, 5-MCNN and BPNN

Fig. 8 Experimental results: the MAPE, MAE and RMSE of 7 weeks forecast by 3-MCNN, 4-MCNN, 5-MCNN and BPNN

5 Conclusion

In this paper, MCNN is applied into electricity load forecast based on replication kinetics theory. Meanwhile, MCNN with one, two and three hidden layers are proposed and tested. Results show that MCNN outperforms BPNN and deeper MCNN performs better than shallow MCNN, which proves that the MC is useful for optimizing neural network, and MCNN is effective for electricity load forecast.

Acknowledgements This work was supported by National Natural Science Foundation of China under Grant No. 61402210 and 60973137, Ministry of Education - China Mobile Research Foundation under Grant No. MCM20170206, State Grid Corporation Science and Technology Project under Grant No. SGGSKY00FJJS1700302, Program for New Century Excellent Talents in University under Grant No. NCET-12-0250, Major National Project of High Resolution Earth Observation System under Grant No. 30-Y20A34-9010-15/17, Strategic Priority Research Program of the Chinese Academy of Sciences with Grant No. XDA03030100, Google Research Awards and Google Faculty Award.

Compliance with Ethical Standards

Conflict of interest All authors declare that they have no conflicts of interest regarding the publication of this manuscript.

References

1. Bergmeir C (2012) On the use of cross-validation for time series predictor evaluation. *Information Sciences* 191(9):192–213
2. Cao LJ, Tay FEH (2004) Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks* 14(6):1506–1518
3. Chen D, Tian Y, Liu X (2016) Structural nonparallel support vector machine for pattern recognition. *Pattern Recognition* 60:296–305
4. Fen MO (2017) Persistence of chaos in coupled lorenz systems. *Chaos Solitons and Fractals* 95:200–205
5. Gao Y, Kong X, Hu C, Zhang Z, Li H, Hou L (2015) Multivariate data modeling using modified kernel partial least squares. *Chemical Engineering Research & Design* 94:466–474
6. Garzon M, Botelho F (1999) Dynamical approximation by recurrent neural networks. *Neurocomputing* 29(13):25–46
7. Islam BU, Baharudin Z, Nallagownden P (2015) Short term electric load forecasting with back propagation neural network and simulated annealing genetic algorithm. *Applied Mechanics and Materials* 785:14–18
8. Li J, Qin G, Wen X, Hu N (2002) Over-fitting in neural network learning algorithms and its solving strategies. *Journal of Vibration Measurement & Diagnosis* 22(4):260–264
9. Neumaier A (1998) Solving ill-conditioned and singular linear systems: A tutorial on regularization. *Siam Review* 40(3):636–666
10. Sch C, Laptev I, Caputo B (2004) Recognizing human actions: A local svm approach. *International Conference on Pattern Recognition* 3:32–36
11. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
12. Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research* 2(1):45–66
13. Vapnik VN (1999) An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10(5):988–99
14. Wang JH, Jiang JH, Yu RQ (1996) Robust back propagation algorithm as a chemometric tool to prevent the overfitting to outliers. *Chemometrics & Intelligent Laboratory Systems* 34(1):109–115
15. Yong B, Li F, Lv Q, Shen J, Zhou Q (2017) Derivative-based acceleration of general vector machine. *Soft Computing* 10(10):1–9

16. Yong B, Shen J, Shen Z, Chen H, Wang X, Zhou Q (2017) Gvm based intuitive simulation web application for collision detection. *Neurocomputing* 279(2):63–73
17. Yong B, Xu Z, Shen J, Chen H, Tian Y, Zhou Q (2017) Neural network model with monte carlo algorithm for electricity demand forecasting in queensland. *Australasian Computer Science Week Multiconference* 47:1–7
18. Zhang Y, Yang Y (2015) Cross-validation for selecting a model selection procedure. *Journal of Econometrics* 187(1):95–112
19. Zhao H (2016) General vector machine. *arXiv preprint* DOI <http://arxiv.org/abs/1602.03950>