



6FKRRO RI ,QIRUPDWLRQ 6FLHQFH DQG (QJLQHULQJ /DQJKRX 8QLYHUVLW /
6FKRRO RI &RPSXWLQJ DQG ,QIRUPDWLRQ 7HFKQRORJ 8QLYHUVLW RI :ROOR
'HSDUWPHQW RI 3K VLFV ;LDPHQ 8QLYHUVLW ;LDPHQ &KLQD
6FKRRO RI 0DQDJPHQW DQG (QWHUSULVH 8QLYHUVLW RI 6RXWKHUQ 4XHHQ

Abstract

E W D W D W H H R H R E H P H W D H P P R H P R W D W R H D H
D D D R D W D D W D W W H H W R R I R D W H W D W
H D H W R D R D W I D H D H W R P R H D H E H H H R H H W R
D I D H D W H W D H D W I D H D H W R 11 P R H W D
H H R H I R R D W H W D W I R H D W D R H H H R D R W P I R
I R H D W W H R W H D R E D H D R W P H P R W D W H D R R H W
H W R D W H W D W H H R W H R W H H H 5 H D H D W D W H W
H R P D H W W R D D E D H P R H W 11 P R H D D E H W R D
H W R P D R D D D D R I W 11 P R H H H D R H H W H W D H

Received on 1 December 2015, accepted on 5 June 2016, published on 5 July 2016

Copyright © 2016 Vimalachandran *et al.*, licensed to ICST. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/

1. Introduction

only shallow neural networks but also multi-layer perceptions (MLP) have been widely deployed to time series forecasting. Later on, feed-forward neural networks and support vector machines are well developed. In the past years, deep learning, which was derived from multi-layer perceptions, has been considered as a most brain-like artificial neural network and is applied to many application fields in machine learning and data analysis. Simultaneously a lot of algorithms aiming to design an efficient artificial neural network have also been proposed and been adopted to corresponding data sets, such as “Back-Propagation”, “Extreme Learning Machine” and “Boltzmann” [1].

Most of the ancient civilizations have made use of wells, spring water and groundwater since the Bronze Age [6]. Groundwater is being considered as the most valuable natural resources in the future, and it becomes more important for many different climatic regions all over the world [7]. Now that groundwater supply is being in danger, how to make a sustainable management of groundwater resources is by far the utmost important issue for us. Groundwater use in irrigation is a vital problem in many arid and semi-arid regions especially when we take a deep look in the Heihe Region in Gansu Province of China. In this paper, we mainly focus on the Heihe Region and all the data ranges from 1986 to 2008.

As model validation is a main tool in optimization, for groundwater management there are several models being used to help optimizing groundwater usage [8]. Box-Jenkins Model (ARIMA) and Physical Based Model [11] are used for groundwater time series forecasting. In this paper, we proposed a new algorithm for conventional artificial neural network, which was different from Back Propagation, SVM and others. With its simple architecture and good performance, we have shown its experimental result in the Heihe River dataset. The rest of the paper was organized as follows. In next section, we will review the Box-Jenkins Model, ARIMA model, Physical Based Model and

*Corresponding author. Email: -LDQPLQJ_RQJXVT_HGX_DX

ANN to time series forecasting. The newly proposed algorithm is introduced in Section III. We will also report and discuss empirical results from Heihe River data sets in Section IV. Section V will conclude with these empirical results.

II. RELATED WORK

Among many approaches to time series forecasting, linear models and non-linear models are two major branches. For linear models, it is a smooth mapping from data space to feature space for ARIMA model to predict the future values from past values, which are constrained in a specific linear functions [10]. It is simple to match different data sets. However in the real world, linear models would not be able to fit and match many data sets because they include not only internal structure but also lots of different signals, such as noises. A simple addition of several ARIMA models could not solve this problem. For certain non-linear patterns observed in real problems, bilinear model, threshold autoregressive model (TAR) and auto regressive conditional heteroscedastic model (ARCH) are derived from ARIMA model.

Even though these expanded models have been proposed for non-linear time series forecasting, they are mostly based on simple linear models and their performance is not satisfying. Especially when they deal with other non-linear time series data sets, these models such as TAR, ARCH are only developed for specific non-linear time series patterns.

Since artificial neural network has been developed, it has been expected to be an excellent alternative to time series forecasting. Its non-linear modeling ability shows its power in both linear and non-linear time series forecasting. Since 1980s, ANN, especially feedforward neural network, is not only applied in time series forecasting but also other computational intelligence areas. With as many variants as multi-layer perception (MLP), recurrent neural networks (RNN) and classical feed forward neural network, currently ANN is one of the most promising tool. To deal with non-linear time series forecasting, ANN could be a better model than ARIMA models. Universal Approximation Theorem states that a feedforward neural network with one single hidden layer can approximate continuous functions when activation functions are set properly on each hidden layer neurons [12]. Several proofs have been given and further proof shows that neural networks have the potential of being universal approximation regardless of activation functions when it comes to multilayer architecture.

In [13] and [14], authors proposed using Back Propagation algorithm to train a neural network to predict the water level. The basis of BackPropagation algorithm mostly focus on the parameters and learning rate settings. Also the activation functions in the hidden layer and the output layer are important in this algorithm. In this paper, we will compare our model while the Back Propagation is our baseline method.

However [1] also discussed a few existing problems of ANN from several aspects. How to define a proper ANN structure for time series forecasting is well known for introducing the over fitting. To solve this problem we depend mostly on the operator's practical experience. The other problem is about

a suitable learning rate for training. Either too high or too low of the learning rates will always lead to different learning optimal problems. In this paper, we propose a new algorithm in ANN models and apply this model in time series forecasting in Heihe River data sets.

III. ALGORITHM

For a classic artificial neural network, its basic structure is shown as Figure. 1. Only one single hidden layer is deployed and the number of neurons is usually playing a key role in performance of every different algorithm.

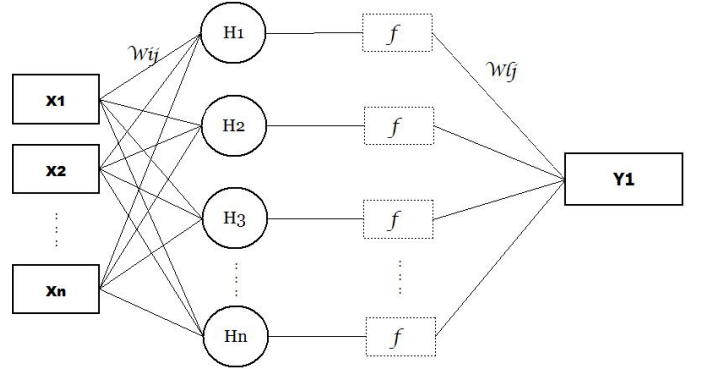


Fig. 1: Classic ANN Diagram

In Figure. 1, $\{x_i, y_i\}$ represents input and output layer units in this traditional model. For this type of feed-forward network, formulas between these three layers are shown below.

$$\bar{h}_i = \sum_1^n W_{ij} * x_i - b_i \quad (1)$$

$$h_i = f(\beta_i * \bar{h}_i) \quad (2)$$

$$y_i = \sum_1^n W_{lj} * h_j \quad (3)$$

From these formulas, there is a bunch of parameters namely $\{W_{ij}, W_{lj}, h_i, b_i, \beta_i, \varepsilon, f\}$. W_{ij} shows the connection strength between the input layer units x_i and the hidden layer units h_j . Accordingly W_{lj} shows the connection strength between the hidden layer units h_l and the output layer units y_j . Formula (1) shows how the input layer units impact the hidden layer units neurons. Every input layer units show its effect on these hidden layer neurons. And Formula (2) contains an activation function f which is considered as a reaction to these inputs. Actually these output of hidden layers neurons are defined by b_i, β_i, f . Back-propagation (BP) algorithm and support vector machine (SVM) methods are normally used in this ANN model.

In our design, we use this classic artificial neural network for our groundwater prediction model. Similar to our earlier work[15], Monte Carlo Algorithm substitutes classic algorithms such BP and SVM for parameter searching. It can be regarded as a computational algorithm based on repeating random sampling to approach an optimal result. While in [15] we have used this algorithm in classification problem, we can also adapt it to our prediction model. In our experiments it

shows powerful ability in time series learning not only in underground water prediction but also Smart Grid prediction. In this paper we will focus on underground water prediction.

Unlike that SVM searches support vector and the best fit kernel function for model, Monte Carlo Algorithm will conduct the general vector searching task and finally mapping data task, as much and precise as it can, into the model. As support vectors always demonstrate a good performance for its structure, once a suitable kernel function is found for the dataset, in this paper, Monte Carlo Algorithm is also deployed in a shallow network. With an acceptable time consumption, this structure could lead to a better prediction result.

In our original algorithm, all the parameters in this bunch, $\{\mathbf{W}_{ij}, \mathbf{W}_{lj}, \mathbf{h}_i, \mathbf{b}_i, \beta_i, \varepsilon, \mathbf{f}\}$, could be updated to suit its goal. In this paper, we simply fix parameters \mathbf{W}_{lj} which is between hidden layer and output layer. The rest of these parameters, including $\{\mathbf{W}_{ij}, \mathbf{b}_i, \beta_i, \varepsilon\}$ could be updated for one parameter each time when the cost function leads to a better performance. We herein use a different cost function from [15] to approach the final result.

In this shallow model, \mathbf{W}_{ij} and \mathbf{W}_{lj} stand for two connected layers, including input layer and hidden layer, and hidden layer and output layer. \mathbf{b}_i represents the bias of each hidden layer units. The β_i correspond to the transfer function coefficients. ε is a correcting unit for each parameters used in Monte Carlo Algorithm. In each iteration we update ε to other parameters, herein it is mainly about $\{\mathbf{W}_{ij}, \mathbf{b}_i, \beta_i\}$. When using ε to adjust other parameters to fit data relationships, each time only one random parameter is selected and being tried to update. Only if the cost function gets a not-worse result should the parameter be updated. In other words, we accept this adjustment only when the cost function Δ reduces or holds.

As shown below Formula (4) is the cost function Δ .

$$\Delta = \sqrt{\frac{1}{n} \sum_1^n (y_i - y_{i0})^2} \quad (4)$$

A main procedure of this Monte Carlo Algorithm is presented in detail below.

A. Initialization

Network requires an initial value to start working. Here we set these parameters $\mathbf{W}_{ij}, \mathbf{W}_{lj}, \mathbf{b}_i, \beta_i, \varepsilon$ to random values within a fixed interval. These intervals are selected empirically and they are critical in our analysis.

B. Update parameters

Choose one group of the parameters from $\{\mathbf{W}_{ij}, \mathbf{W}_{lj}, \mathbf{b}_i, \beta_i\}$.

Model randomly chooses one parameter from this bunch, adjusts it with the ε and calculates \mathbf{y}_i and Δ . If Δ is not becoming worse, then we accept this change and move on to next round. This step stops after a Monte Carlo Steps.

C. Repeat Monte Carlo Steps on other parameters

Change to other types of parameters from $\{\mathbf{W}_{ij}, \mathbf{W}_{lj}, \mathbf{b}_i, \beta_i\}$ after a Monte Carlo Steps.

D. Stop training

Repeatedly train this shallow neural network via B and C steps and stop when either a critical time runs out $t \geq t_0$ or cost function $\Delta \leq \Delta_0$.

In our work, this shallow neural network is applied to underground water time series prediction problem. The original underground water data has been obtained for decades [16, 17]. In [15] we used a modified cost function instead of square root value to overcome the over-fitting problem. In this paper, square root as shown in Formula (4) is used to approximate the best prediction model for underground water.

Cost function is the key factor in Monte Carlo Algorithm. It varies a lot to fit different situations for its practical results taking into the consideration of time consumption and accuracy. We have shown a powerful ability of this Monte Carlo Algorithm to fit different situations in our studies in different applications. In next section we will focus on datasets and its experiments.

IV. DATASETS AND EXPERIMENTS

Even though data is collected in a reliable method, there is a data sparsity problem due to inadequate measuring methods in the past decades. Hence we reconstruct these data in a reasonable way.

There is not any pre-processing with these data and its inner noise, which is inevitably existing in these data and is however overcome by our Monte Carlo Algorithms. We show our result without any pre-processing. And the local field theory demonstrate that noise could be used to facilitate model's learning and optimize its result as [18] and [19] had illustrated. Therefore via curating a dataset from the original data, we get a model containing enough units.

A. Dataset

In previous decades, exactly 1986-2008, the underground water data of 26 related spots had been collected. They are all located in Heihe River basin, which means these 26 related spots data could have some internal relationship. Our curated data comes from a sub group of these 26 observation spots, namely Banqiaodongliu spot, Pingchuangongcheng spot, Shandanqiao spot, Wangqizha spot, Xiaohe spot, Xingou spot, Yaruanzhangwan spot, Zhangyenongchang spot and Shajingzi spot.

In this paper, we build a dataset including eight observation feature spots and one observation label spot for a time series prediction. Overall noises exist in this dataset resulted from its data collection technology. However we still input them directly into this shallow neural network for training. It is a bit different from [15]. The cost function used in this paper and output layer setting have been changed accordingly.

B. Model

As shown in Figure. 2, this is the original shallow neural network with 8 units input, 1 units output and 200/500 hidden neural units.

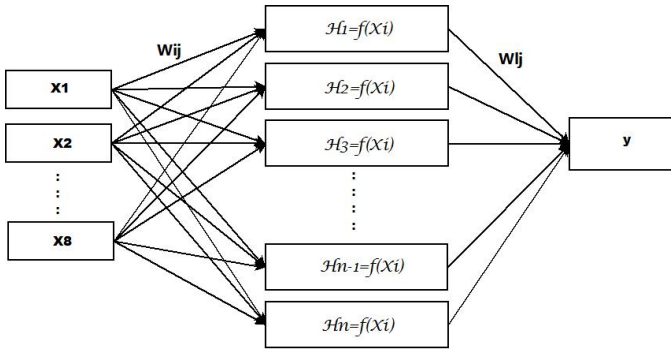


Fig. 2: Original Neural Network

C. Parameters Value Interval

It is an empirical issue for selecting suitable parameters value interval. And in our paper, $\{W_{ij}, b_i, \beta_i\}$ are chosen to be positively updated for a better cost function while W_{ij} remains the same once it is decided at initialization stage.

W_{ij} is fixed into two groups of experiments, for $\{+0.10, -0.10\}$ and $\{+0.01, -0.01\}$. Also through the change of neural transfer function, we have modified hidden neural units to observe its performance.

D. Procedure

In this subsection, the main procedure of our Monte Carlo Algorithm (see Algorithm 1) is presented.

E. Experiments

Experiments are conducted under four different settings but all achieve a good performance in their results. In our experiments, we deploy two different transfer neural functions as $x^2 + x$ and x . Below we will show the results with different settings.

As a baseline work, we chose Back Propagation algorithm as our comparison method, since most of literatures [13, 14] used BP as their method. As shown in Fig. (3) was the training result and Fig. (4) was the test result. The MSE result of the test dataset was 0.56, while in our model it was 0.113.

Algorithm 1 Monte Carlo algorithm

init supervised model: Calculate Output Layer status y_i with a pre-setting values of all parameters.

Calculate Cost Function: Calculate Cost Function value Δ_1 .

```

1: if  $\Delta_1 \geq \Delta_0$  then
2:   if  $step < Monte_{Carlo}step$  then
3:      $\varepsilon = \text{random}(+\varepsilon_0, -\varepsilon_0)$ ;
4:     Select a random  $W_{ij}$  from  $W_{ij}$  matrix;
5:      $W_{ij} = W_{ij} + \varepsilon$ ;
6:     update Output Layer status  $y_i$ ;
7:     Calculate Cost Function value  $\Delta$ ;
8:   while  $\Delta \leq \Delta_0$  :
9:     update  $W_{ij}$ ;
10:    update  $\Delta_1$ ;
11:   end if
12:
13:   if  $step < Monte_{Carlo}step$  then
14:      $\varepsilon = \text{random}(+\varepsilon_0, -\varepsilon_0)$ ;
15:     Select a random  $\beta_i$  from  $\beta_i$  matrix;
16:      $\beta_i = \beta_i + \varepsilon$ ;
17:     update Output Layer status  $y_i$ ;
18:     Calculate Cost Function value  $\Delta$ ;
19:   while  $\Delta \leq \Delta_0$  :
20:     update  $\beta_i$ ;
21:     update  $\Delta_1$ ;
22:   end if
23:
24:   if  $step < Monte_{Carlo}step$  then
25:      $\varepsilon = \text{random}(+\varepsilon_0, -\varepsilon_0)$ ;
26:     Select a random  $W_{ij}$  from  $W_{ij}$  matrix;
27:      $b_i = b_i + \varepsilon$ ;
28:     update Output Layer status  $y_i$ ;
29:     Calculate Cost Function value  $\Delta$ ;
30:   while  $\Delta \leq \Delta_0$  :
31:     update  $b_i$ ;
32:     update  $\Delta_1$ ;
33:   end if
34: end if

```

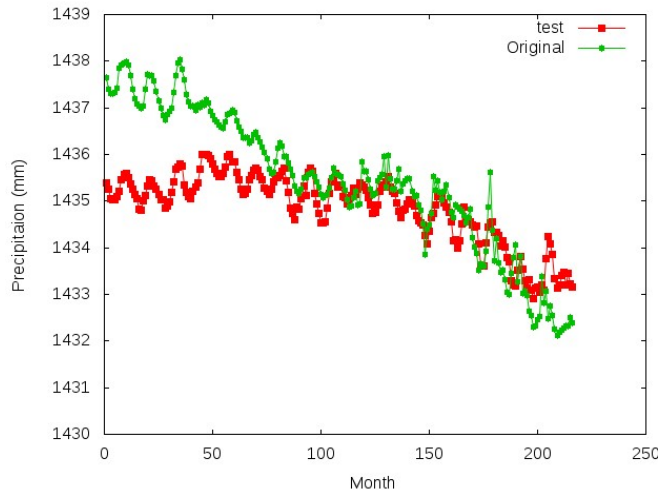


Fig. 3: Training on Back Propagation algorithm

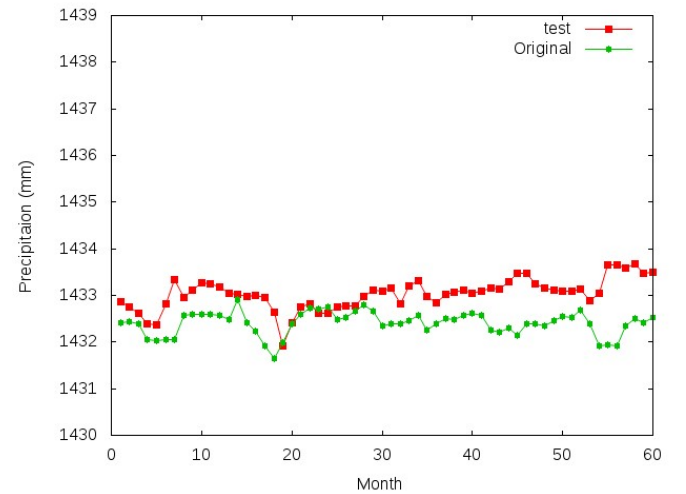


Fig. 4: Test on Back Propagation algorithm

As shown in Fig. (5) and (6), a 200 hidden neural units layer was allocated for our first experiment. In this network, a $x^2 + x$ transfer function was used and also the value of W_{lj} is $\{+0.1, -0.1\}$. It delivered a good performance of prediction.

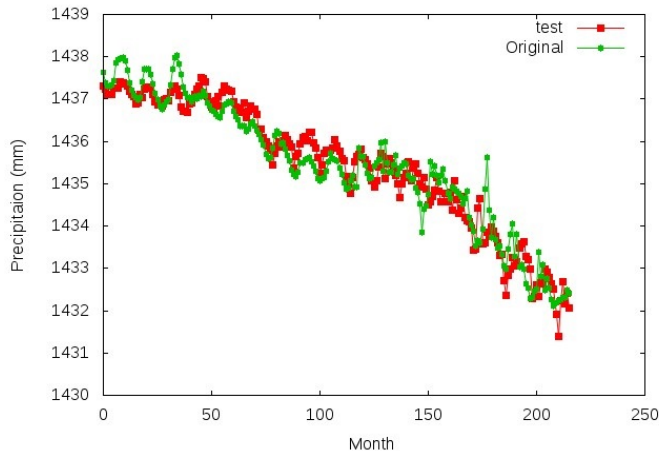


Fig. 5: Training under 200 hidden units and $x^2 + x$ function

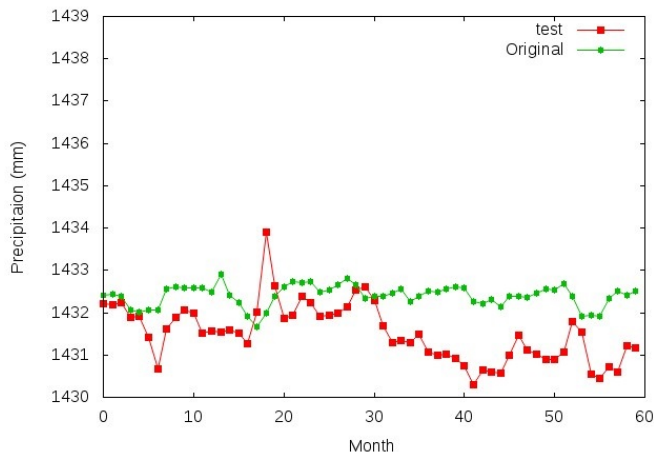


Fig. 6: Result under 200 hidden units and $x^2 + x$ function

In the next, we slightly changed the parameter of W_{lj} from $\{+0.1, -0.1\}$ to $\{+0.01, -0.01\}$, the results showed an almost same result as we can tell from Fig. (7) and (8). It demonstrated that under Monte Carlo Algorithm category, this shallow neural network had a better learning and fitting ability for understanding the data. The high scalability could be obtained via a bunch of parameters $W_{ij}, b_i, \beta_i, \varepsilon$ even when W_{ij} is fixed.

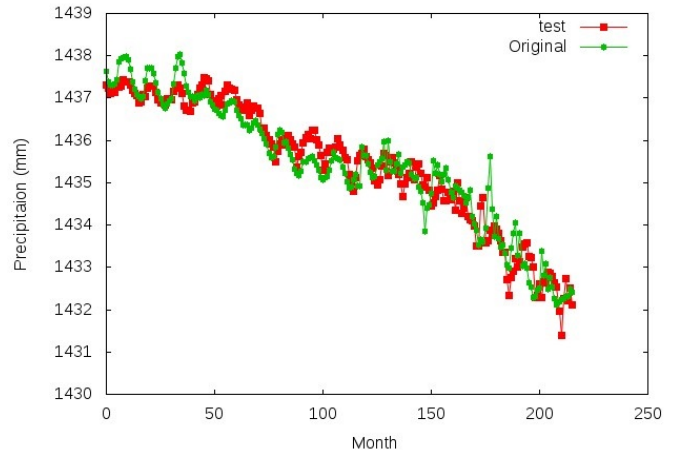


Fig. 7: Training under 200 hidden units and $x^2 + x$ function

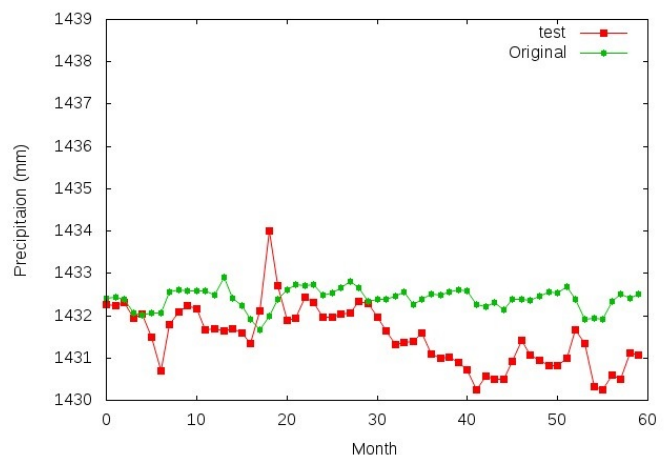


Fig. 8: Result under 200 hidden units and $x^2 + x$ function

Fig. (9) and (10) demonstrated a better performance for our prediction in the case when we changed the transfer function from $x^2 + x$ to x . The trends and values in our test have shown a better result under the construction of x transfer neural function. Furthermore, we also tried to configure a different size of hidden neural units. As shown in Fig. (11) and (12), the x transfer neural function got a better ability to fit data regardless of the hidden neural units size. In our previous test where $x^2 + x$ set as the transfer neural function, the best result came from 200 hidden neural units.

From Fig. (9) and (10), a suitable transfer neural function achieved a even better and stable result with 200 and 500 hidden neural units. It showed that searching for general vectors in this shallow neural network was much easier with the deployment of Monte Carlo Algorithm. But for its performance, it would strongly depend on its neural network and cost function for a positive updating process.

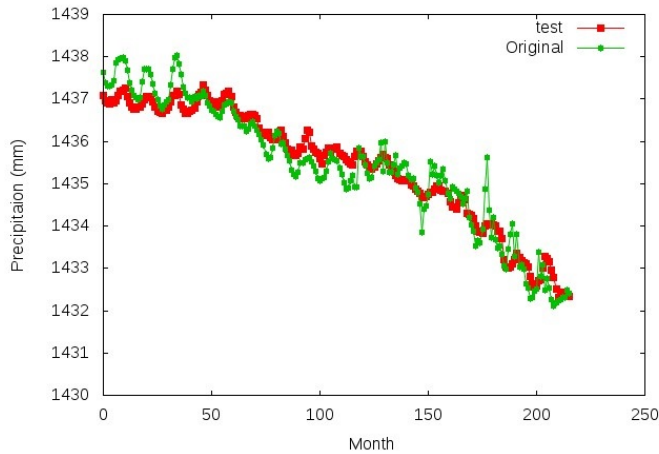


Fig. 9: Training under 200 hidden units and x function

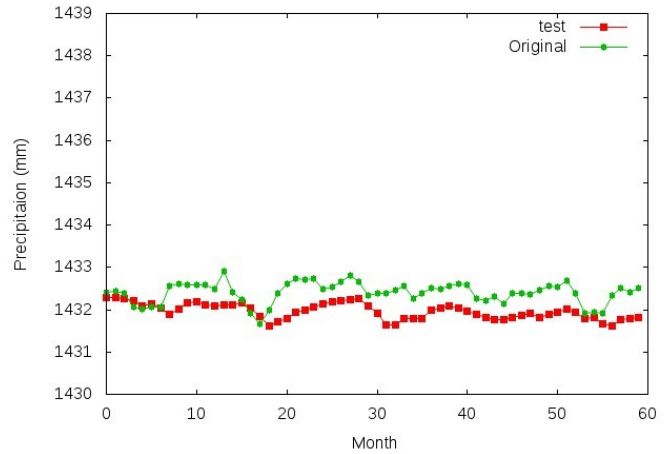


Fig. 12: Result under 500 hidden units and x function

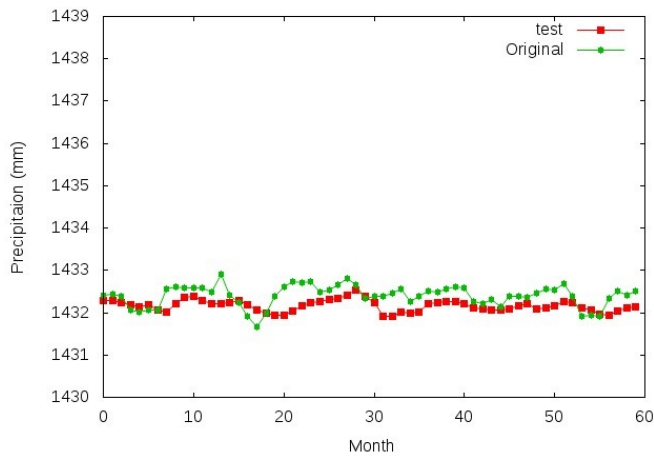


Fig. 10: Result under 200 hidden units and x function

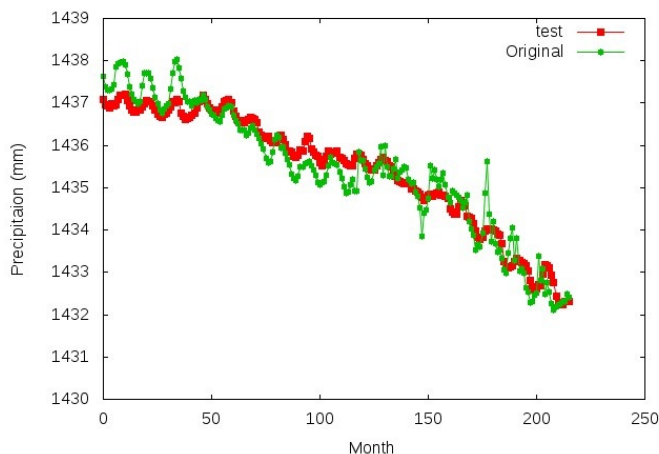


Fig. 11: Training under 500 hidden units and x function

V. CONCLUSION

Tackling water prediction problem nowadays is more important for a proper regional development strategy, especially in the arid Heihe River basin. In this paper, we have curated a dataset from two decades of collected data. A consideration of area related relationship is important in this paper. We tried to figure out the relationship between these nine observation spots selected out of the whole 26 spots. The Monte Carlo Algorithm [15] is an efficient approach as shown in our paper and we also discussed the internal settings and structure of the deployed shallow neural network.

The baseline work in our paper is Back Propagation algorithm. We achieve a better result than BP while in BP the MSE result is 0.56 and in our model it is 0.113. The results is promising in dealing with the difficulties in such problem.

In the future work, expanding this method to other data curated not only from our own data but also other researchers' available data is under consideration. Also introducing bio-inspired algorithm [20] to facilitate the training speed and performance of Monte Carlo algorithm. And a comparison between our approach and the state of art of deep learning methods will also be undertaken.

ACKNOWLEDGMENT

This work is supported by a scholarship from the China Scholarship Council while the second author pursues his PhD degree in University of Wollongong. This work was supported by National Natural Science Foundation of China under Grant No. 61402210 and 60973137, Program for New Century Excellent Talents in University under Grant No. NCET-12-0250, "Strategic Priority Research Program" of the Chinese Academy of Sciences with Grant No. XDA03030100, Gansu Sci.&Tech. Program under Grant No. 1104GKCA049, 1204GKCA061 and 1304GKCA018. The Fundamental Research Funds for the Central Universities under Grant No. lzujbky-2014-49, lzujbky-2013-k05, lzujbky-2013-43, lzujbky-2013-44, lzujbky-2014-235 and lzujbky-

2012-44, Gansu Telecom Cuiying Research Fund under Grant No. Izudxcy-2013-4, Google Research Awards and Google Faculty Award, China. This work is also facilitated by the help of University of Wollongong's University Internationalisation Committed's UIC Linkage program and Global Challenge Program's Travel Fund.

REFERENCES

- [1] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, 2014.
- [2] H. Wang, Y. Zhang, and J. Cao, "Access control management for ubiquitous computing," *Future Generation Computer Systems*, vol. 24, no. 8, pp. 870–878, 2008.
- [3] L. Wang, J. Shen, and J. Yong, "A survey on bio-inspired algorithms for web service composition," in *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*. IEEE, 2012, pp. 569–574.
- [4] M. Li, X. Sun, H. Wang, Y. Zhang, and J. Zhang, "Privacy-aware access control with trust management in web service," *World Wide Web*, vol. 14, no. 4, pp. 407–430, 2011.
- [5] I. N. Daliakopoulos, P. Coulibaly, and I. K. Tsanis, "Groundwater level forecasting using artificial neural networks," *J. Hydrol.*, vol. 309, no. 1-4, pp. 229–240, 2005.
- [6] L. W. Mays, "Groundwater Resources Sustainability: Past, Present, and Future," *Water Resour. Manag.*, vol. 27, no. 13, pp. 4409–4424, 2013.
- [7] D. K. Todd, *Ground water hydrology*. New York : Wiley, 1959, includes bibliography.
- [8] S. Mohanty, M. K. Jha, A. Kumar, and K. P. Sudheer, "Artificial neural network modeling for groundwater level forecasting in a river island of eastern India," *Water Resour. Manag.*, vol. 24, no. 9, pp. 1845–1865, 2010.
- [9] C. Hamzaçebi, "Improving artificial neural networks' performance in seasonal time series forecasting," *Inf. Sci. (Ny)*, vol. 178, no. 23, pp. 4550–4559, 2008.
- [10] G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [11] E. Coppola Jr., M. Poulton, E. Charles, J. Dustman, and F. Szidarovszky, "Application of Artificial Neural Networks to Complex Groundwater Management Problems," *Nat. Resour. Res.*, vol. 12, no. 4, pp. 303–320, 2003.
- [12] M. Baker and R. Patil, "Universal approximation theorem for interval neural networks," *Reliable Computing*, vol. 4, no. 3, pp. 235–239, 1998.
- [13] R. Bustami, N. Bessaih, C. Bong, and S. Suhaili, "Artificial neural network for precipitation and water level predictions of bedup river," *IAENG International journal of computer science*, vol. 34, no. 2, pp. 228–233, 2007.
- [14] S. Rani and F. Parekh, "Predicting reservoir water level using artificial neural network," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, pp. 14 489–14 496, 2014.
- [15] H. Chen, H. Zhao, J. Shen, R. Zhou, and Q. Zhou, "Supervised machine learning model for high dimensional gene data in colon cancer detection," in *Big Data (BigData Congress), 2015 IEEE International Congress on*, June 2015, pp. 134–141.
- [16] X. Wu, J. Zhou, H. Wang, Y. Li, and B. Zhong, "Evaluation of irrigation water use efficiency using remote sensing in the middle reach of the heihe river, in the semi-arid northwestern china," *Hydrological Processes*, vol. 29, no. 9, pp. 2243–2257, 2015.
- [17] X. H. Wen, Y. Q. Wu, L. J. E. Lee, J. P. Su, and J. Wu, "Groundwater flow modeling in the Zhangye Basin, Northwestern China," *Environ. Geol.*, vol. 53, pp. 77–84, 2007.
- [18] H. Zhao and T. Jin, "A global algorithm for training multilayer neural networks," *arXiv preprint physics*, 2006.
- [19] H. Zhao, "Designing asymmetric neural networks with associative memory," *Physical Review E*, vol. 70, no. 6, 2004.
- [20] L. Wang, J. Shen, J. Luo, and F. Dong, "An improved genetic algorithm for cost-effective data-intensive service composition," in *Semantics, Knowledge and Grids (SKG), 2013 Ninth International Conference on*. IEEE, 2013, pp. 105–112.