

## Research Article

# Data Normalization to Accelerate Training for Linear Neural Net to Predict Tropical Cyclone Tracks

Jian Jin,<sup>1</sup> Ming Li,<sup>2</sup> and Long Jin<sup>3</sup>

<sup>1</sup>Department of Computer Science & Technology, East China Normal University, Shanghai 200241, China

<sup>2</sup>Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China

<sup>3</sup>Guangxi Climate Center, Guangxi Meteorological Bureau, Nanning 530022, China

Correspondence should be addressed to Jian Jin; [jjin@cs.ecnu.edu.cn](mailto:jjin@cs.ecnu.edu.cn)

Received 27 August 2014; Accepted 20 December 2014

Academic Editor: Cagdas Hakan Aladag

Copyright © 2015 Jian Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

When pure linear neural network (PLNN) is used to predict tropical cyclone tracks (TCTs) in South China Sea, whether the data is normalized or not greatly affects the training process. In this paper, min.-max. method and normal distribution method, instead of standard normal distribution, are applied to TCT data before modeling. We propose the experimental schemes in which, with min.-max. method, the min.-max. value pair of each variable is mapped to  $(-1, 1)$  and  $(0, 1)$ ; with normal distribution method, each variable's mean and standard deviation pair is set to  $(0, 1)$  and  $(100, 1)$ . We present the following results: (1) data scaled to the similar intervals have similar effects, no matter the use of min.-max. or normal distribution method; (2) mapping data to around 0 gains much faster training speed than mapping them to the intervals far away from 0 or using unnormalized raw data, although all of them can approach the same lower level after certain steps from their training error curves. This could be useful to decide data normalization method when PLNN is used individually.

## 1. Introduction

Numerous prediction models have been proposed to raise the forecasting precisions of the tropical cyclone tracks (TCTs) in South China Sea (SCS) in the past decades to reduce the loss from these disasters. TCT forecasting is a time series problem. There are a lot of improved time series techniques that could be applied to this field [1–4]. The approaches are getting more and more complicated, including ensembles [5–8]. The results presented here are part of a wider project on committee machines aiming to obtain higher performance through combining multiple simple experts. These experts require reasonable accuracy and diversity [9]. We use pure linear neural networks (PLNNs) as the experts and then combine their results by means of fuzzy logic. So it relies on comprehensive understanding of the PLNNs, especially PLNNs on the particular data set [10].

One of the problems we face is before modeling whether data should be normalized or not to predict TCTs in SCS and what is the effect. It has been reported in literature that

normalization could improve performance. Sola and Sevilla [11] pointed out the importance of data normalization prior to the neural network training to fasten the calculations and obtain good results in nuclear power plant application. Jayalakshmi and Santhakumaran [12] concluded that various statistical normalization techniques enhance the reliability of trained feed forward backpropagation neural networks and the performance of the diabetes data classification model using the neural networks is dependent on the normalization methods. Zhang and Sun [13] also provided a normalization method considered to be suitable for the particular data set from UCI repository. So, data normalization is important whereas the results are by no means universally applicable to the particular data. In predicting the TCTs, we usually use raw environmental data or simply map the raw data to 0-1 interval linearly. But we are not certain if that is beneficial.

In this paper, we try two commonly used normalization methods, that is, linear min.-max. method and normal distribution method, propose the experimental schemes to map the raw data variables to the intervals near 0 as well

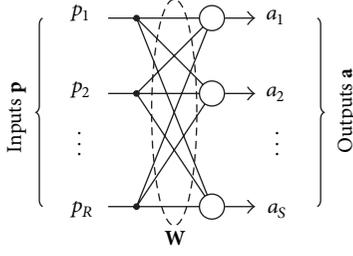


FIGURE 1: Structure of pure linear neural network.

as far away from 0, and then apply the normalized data to a standalone PLNN model. We intend to study whether data normalization really works and how it affects the PLNN model with TCT data, so that we can be more aware of how to pretreat the data and make the combining system more controllable once the PLNNs are used as ensemble members. We demonstrate that proper data normalization speeds up convergence remarkably with the TCT data. From the trends of error curves, lower levels of training error can be achieved in all situations as long as enough training iterations are provided. However, mapping the data to a proper interval makes more sense than how the data is mapped, according to the min.-max. method and normal distribution method.

## 2. Pure Linear Neural Network

The pure linear neural network in this paper is a single-layer structure model named ADALINE (adaptive linear neuron), first proposed by Widrow and Hoff in 1960, as shown in Figure 1 [14]. Each node in the structure represents a neuron, and it has its own weights  $\mathbf{w}$  and a bias  $b$  and gets the same input vector  $\mathbf{p}$ . The weighted inputs are summed with the bias to form the following net output  $n$ :

$$n = \mathbf{w}^T \cdot \mathbf{p} + b, \quad (1)$$

where  $\mathbf{w} = [w_1, w_2, \dots, w_R]^T$  and  $\mathbf{p} = [p_1, p_2, \dots, p_R]^T$ . Equation (1) could also be expressed as

$$n = \mathbf{x}^T \cdot \mathbf{z} \quad (2)$$

for simplicity, where  $\mathbf{x} = [\mathbf{w}^T, b]^T$  and  $\mathbf{z} = [\mathbf{p}^T, 1]^T$ .

The neuron uses pure linear transfer function

$$a = \text{purelin}(n) = n, \quad (3)$$

which takes the net output as its input and forms the neuron output  $a$ . The neuron is shown in Figure 2.

The learning rule could be derived by minimizing the mean square error of the training instance

$$E = e^2 = (t - a)^2, \quad (4)$$

where  $e$  is the difference between the target (or observed value)  $t$  and the network output (or prediction)  $a$ . The rule is written as

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + 2\alpha_w \cdot e \cdot \mathbf{z} \quad (5)$$

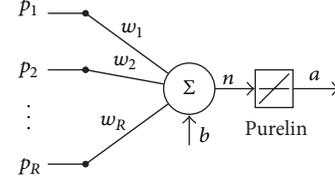


FIGURE 2: Pure linear neuron.

TABLE 1: Numbers of CLIPER variables and physical variables initially gathered for July, August, and September.

Position	July	August	September	Commonly used and chosen
Longitude	46	57	40	8
Latitude	100	62	75	11

or

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \alpha_w \cdot e \cdot \mathbf{z} \quad (6)$$

for simplicity [15], where  $\alpha_w$  is a learning rate.

In this paper, we use a MATLAB function

$$\alpha_w = \text{maxlinlr}(\mathbf{z}) = \frac{0.9999}{\max(\text{eig}(\mathbf{z} \cdot \mathbf{z}^T))}, \quad (7)$$

introduced in the Neural Network Toolbox, as the learning rate. This function takes (6) as the learning rule.

We use the traditional incremental training strategy, with which the weights and biases are updated based on each single instance, while, with the batch training, weights and biases are updated based on more than one instance [16–18].

## 3. Experiments and Results

**3.1. Data Set.** The used TCT data set in our experiment is published by China Meteorological Administration. The objects are the TCs that form in or move into the SCS area and last for at least 48 h. The TCTs were sampled every 12 h, starting from the moment when a TC moves into the sea area or when a TC developed in the area. We use the sampled 750 instances since 1960. Our object is to predict the longitude and latitude of the TC center in the next 24 h [19].

A TCT and its changes are associated with TC's intensity, accumulation and replenish of energy, and various nonlinear changes in its environmental flow field, which are referred to as variables in this paper. The variables used in our experiment include the climatology and persistence (CLIPER) factors representing changes of TC itself, such as changes in the latitude, longitude, and intensity of a TC at 12 and 24 h before prediction time. Table 1 lists the variables initially selected by using the CLIPER method with the significance level of 0.01 [19, 20].

For our case study we adapt the preceding 720 instances for training and the following 30 as independent instances for testing. Eight and 11 common predictors among July, August, and September are carefully picked from the different number

TABLE 2: Variable information of the data set.

Variable	Min.	Max.	Mean	Std.	Meaning
Lon. v1	105.3	123.0	114.7	4.2	Initial lon. (E)
Lon. v2	-39.8	23.1	-9.1	10.3	Zonal motion at -12 h
Lon. v3	-36.1	20.8	-9.3	9.7	Zonal motion at -24 h
Lon. v4	106	130	116.8	4.9	Lon. (E) at -12 h
Lon. v5	-43.9	20.8	-9.6	10.3	Zonal motion from -24 to -12 h
Lon. v6	-4.5	2.7	-1.0	1.2	Lon. difference between 0 and -12
Lon. v7	106.0	124.6	115.2	4.3	Lon. (E) at -6 h
Lon. v8	106.0	128.3	116.3	4.7	Lon. (E) at -24 h
Lon. t	101.4	128.3	112.8	4.7	Current lon. (E)
Lat. v1	10.8	23.5	18.7	2.4	Initial lat. (N)
Lat. v2	-39.8	23.1	-9.1	10.3	Zonal motion at -12 h
Lat. v3	-17.4	26.6	4.2	6.5	Meridional motion at -12 h
Lat. v4	-36.1	20.8	-9.3	9.7	Zonal motion at -24 h
Lat. v5	0.0	525.2	50.2	62.0	Squared zonal motion at -24
Lat. v6	10.0	23.7	18.2	2.4	Lat. (N) at -12 h
Lat. v7	-4.5	2.7	-1.0	1.2	Lon. difference between 0 and -12
Lat. v8	-8.2	4.8	-2.1	2.2	Lon. difference between 0 and -24 h
Lat. v9	10.3	23.8	18.5	2.4	Lat. (N) at -6 h
Lat. v10	9.6	24.0	18.0	2.5	Lat. (N) at -24 h
Lat. v11	10	60	23.5	9.9	Max. surface wind at -6 h
Lat. t	12.2	30.0	19.8	2.6	Current lat. (N)

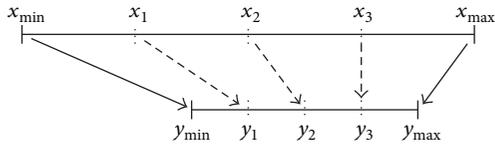


FIGURE 3: Min.-max. method of normalization.

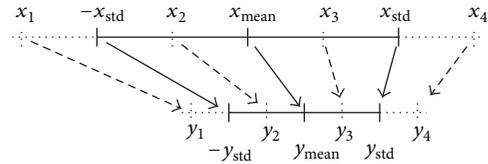


FIGURE 4: Normal distribution method of normalization.

of variables as shown in Table 1 for longitude and latitude, respectively. Some statistical information about the data set is listed in Table 2.

**3.2. Experiment Setup.** Two types of normalization methods, min.-max. method and normal distribution method, are used in our experiment. The min.-max. method linearly maps the original values to the new interval determined by the assigned min.-max. values (see Figure 3). The original minimum value  $x_{\min}$  and maximum value  $x_{\max}$  can be achieved from the statistical information of the raw data, and the new minimum value  $y_{\min}$  and maximum value  $y_{\max}$  are assigned by us, so the new value  $y$  could be calculated from original value  $x$  by

$$y = \frac{(x - x_{\min}) \cdot (y_{\max} - y_{\min})}{x_{\max} - x_{\min}} + y_{\min}, \quad (8)$$

while the normal distribution method maps the original values to the new interval according to the new mean and deviation (see Figure 4).

TABLE 3: Normalization configuration.

Conf. number	Method	Parameters
Conf. 1	Min.-max.	$y_{\min} = -1$ , $y_{\max} = 1$
Conf. 2	Min.-max.	$y_{\min} = 0$ , $y_{\max} = 1$
Conf. 3	Normal distribution	$y_{\text{mean}} = 0$ , $y_{\text{std}} = 1$
Conf. 4	Normal distribution	$y_{\text{mean}} = 100$ , $y_{\text{std}} = 1$
Conf. 5	Raw data set	

Similarly, the new value could be calculated by

$$y = \frac{(x - x_{\text{mean}}) \cdot y_{\text{std}}}{x_{\text{std}}} + y_{\text{mean}} \quad (9)$$

except the mean and deviation in place of the min.-max. values.

Four normalized data sets originated from the raw data set, as well as the raw one, are used in our experiment and are described in Table 3.

The training process on each training data set is summarized in Algorithm 1 [21], in which we have the following.

TABLE 4: Required iterations to reach decreasing MAE stages during pretraining.

Conf. number	MAE of longitude				MAE of latitude		
	1.2	1.175	1.15	1.125	1.1	1.0	0.9
Conf. 1	4	4	5	409	26	46	91
Conf. 2	107	119	144	$10^3+$	169	302	593
Conf. 3	7	8	11	$10^3+$	5	9	20
Conf. 4	$10^3+$	$10^3+$	$10^3+$	$10^3+$	$10^3+$	$10^3+$	$10^3+$
Conf. 5	811	$10^3+$	$10^3+$	$10^3+$	722	909	$10^3+$

```

(1) begin
(2)   repeat
(3)     Calculate the learning rate with (7) and the current
         training set  $\mathbf{P}_{\text{trn}}$ 
(4)     forall  $\mathbf{p} \in \mathbf{P}_{\text{trn}}$  do
(5)       Calculations of the instance  $\mathbf{p}$ 
(6)       Update the weights and biases making use of the instance  $\mathbf{p}$ 
         and the incremental learning rule (6)
(7)     until the maximum number of iterations ( $K_{\text{max}}$ ) is reached;
(8)   end

```

ALGORITHM 1: Training process on the training set  $\mathbf{P}_{\text{trn}}$ .

- (1)  $\mathbf{p} \in \mathbf{P}_{\text{trn}}$ : each instance in the training set is applied to training process one by one in turn.
- (2) *Calculations of instance* indicates necessary calculations such as forward calculating the network output, errors, and so on, as required.

The experimental process is summarized in Algorithm 2, in which Algorithm 1 is used twice, and we have the following.

- (1) *Network weights* are initialized with uniform distributed random real values ranging between 0 and 1, and the experiments on the five data sets use the same initial weights.
- (2) *Training set* is initialized with the preceding 720 out of the total 750 instances (including both longitude and latitude data and the same below).
- (3) *Testing set* is initialized with the last 30 instances.
- (4) During the pretraining, *the maximum number of iterations* ( $K_{\text{max}}$ ) in Algorithm 1 is set to 1000.
- (5) During the *repeat* statement,  $K_{\text{max}}$  is set to 20.

In Algorithm 2, we arranged a pretraining process in which the maximum number of iterations ( $K_{\text{max\_pretrain}}$ ) is much larger than that in the training process with respect to each independent instance ( $K_{\text{max\_train}}$ ). This arrangement is made with the concerns that we hope the prediction of the independent instance is made using the adequately trained model. If  $K_{\text{max\_pretrain}}$  is set too small, the model will be inadequately trained when it applies to predict the independent instances. To compensate for this,  $K_{\text{max\_train}}$  must be set larger to train the initial training set more, other than the newly added independent instance, to make the model more adequately trained. This is reasonable for the

beginning independent instances. However,  $K_{\text{max\_train}}$  aims to train the newly added independent instance based on the adequately trained model using the former training set. For the later independent instances, there is only one new instance added to the training set, so the training load is not so heavy, and the larger  $K_{\text{max\_train}}$  wastes more resources. As a result,  $K_{\text{max\_pretrain}}$  is set to be larger and  $K_{\text{max\_train}}$  is set to be smaller. Another character in Algorithm 2 is that each time only one instance is predicted and after that the predicted instance is added to the training set. So the model should be retrained before next prediction. This is in accordance with the actual facts. We use the 24-hour model to predict the track center in the next 24 hours. Once it comes true, this instance could be more valuable and should be added to the training set for training to prepare for the next 24 hours.

**3.3. Result and Analysis.** Figure 5 shows the convergence curves during the pretraining process for the five normalization configurations. The left half of the figure is for the longitude and the right one for the latitude. Both halves share the same  $x$ -coordinate, namely, iterations labeled on the top edge and numbered along the bottom edge. The  $y$ -coordinate is the mean absolute error (MAE) for all the instances in the training set, after  $k$  iterations of training. The scales in the  $y$ -coordinate are different for longitude and latitude and are numbered along the left edge of the figure. The five colored thick curves with various line styles reflect the training precision and speed. The smaller the  $y$ -value, the higher the precision, and the steeper the curve, the faster the convergence. We should make it clear that all the results, including the MAEs, are denormalized before drawing in the figures or listing in the table, and it is the same as below. As Algorithm 1 shows, 1 iteration of training means that each

```

(1) begin
(2)   Initialization begin
(3)     Network weights  $x$ 
(4)     Training set  $P_{trn}$ 
(5)     Testing set  $P_{tst}$ 
(6)   end
(7)   Pre-training on the initial training set  $P_{trn}$  using Algorithm 1
(8)   repeat
(9)     Training on the current training set  $P_{trn}$  using Algorithm 1
(10)    Test on the first ( $P_{tst}$ )
(11)    Move the first ( $P_{tst}$ ) from  $P_{tst}$  to  $P_{trn}$ 
(12)  until  $P_{tst} = \emptyset$ ;
(13) end
    
```

ALGORITHM 2: Experimental process.

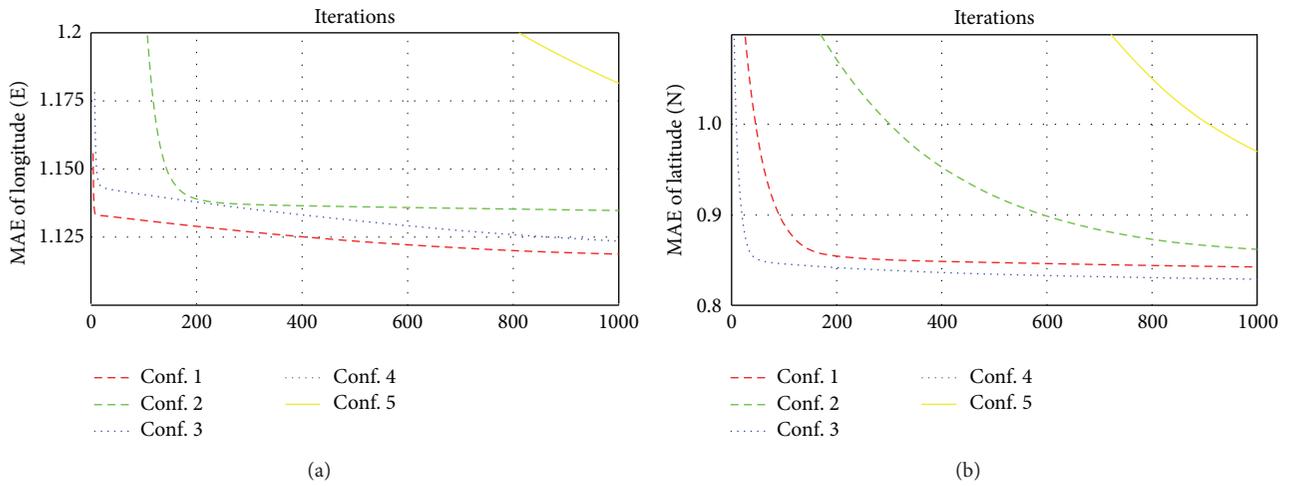


FIGURE 5: Mean absolute errors (MAEs) of the training set during pretraining.

of the instance in the training set should be applied to the learning rule to update the weights for once.

Both curves of Conf. 1 and Conf. 2 use the min.-max. method of normalization. The difference is that Conf. 1 makes the normalized data located around zero with the radius 1 while Conf. 2 around 0.5 with the radius 0.5. This minor difference makes the curve of Conf. 1 smaller and steeper than that of Conf. 2 during the pretraining, for both longitude and latitude.

What Conf. 1 and Conf. 3 have in common is that their means are 0 and radii are about 1. Their differences are uniform distribution for Conf. 1 and normal distribution for Conf. 3. The convergence curves of Conf. 1 and Conf. 3 are very close, both for longitude and latitude data. For longitude data, MAE values on the curve of Conf. 1 is smaller than on the curve of Conf. 3, while for latitude data things are the opposite. So it is hard to tell which configuration is more superior to the other.

Conf. 3 and Conf. 4 both use normal distribution, but the mean of Conf. 3 is 0, while that of Conf. 4 deviates from 0 to 100. As a result, the convergence MAE for Conf. 4 is too large and disappears out of the axis limits in Figure 5.

Conf. 5 uses the raw data, located in different intervals ranging from  $-45$  to  $525$  for different variables (see Table 2). The curve of Conf. 5 doesn't deviate so much as the curve of Conf. 4 does, although the curve of Conf. 5 is flatter than the curves of Conf. 1, Conf. 2 and Conf. 3. It implies that the initial weights ranging between 0 and 1 are possibly not so decisive for big valued data, although data from Conf. 4 could be influenced to some extent by the same initial weights with other configurations.

Table 4 lists the required iterations for PLNN to reach the same level of training errors during the pretraining process with both longitude and latitude data. It quantitatively illustrates that mapping the data to the intervals around 0 (Conf. 1 and Conf. 3) could reach the smaller training error much faster, and even if slight shift for the mean from 0 to 0.5 (Conf. 2) could extend this process, let alone giant shift from 0 to 100 (Conf. 4).

Figure 6 illustrates the fitting curves of longitude and latitude on independent instances. The left and right halves of the figure are for longitude and latitude, respectively. Both halves share the same  $x$ -coordinate indicating the indexes of the independent instances labeled on the top edge

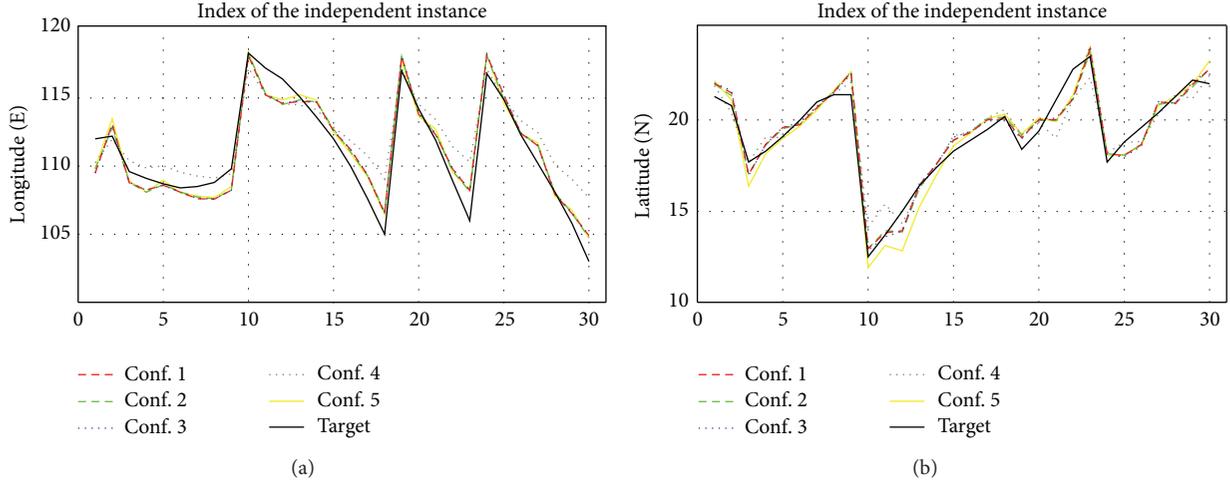


FIGURE 6: Predictions of the independent instances for 5 normalized data and the observations.

and numbered along the bottom edge. The  $y$ -coordinate is the real east longitude or north latitude value. The five prediction curves, as well as the observed one (the targets, black solid line), are drawn for comparison. Each curve connects the adjacent fitting points. It is worth noting that the corresponding training set to each independent instance includes the original training set (720 instances) and the previous used independent instances. For example, if  $x$  is 5, then the current training set is 720 plus 4, totally 724 instances. For  $y$ -value, the closer to the target point, the more accurate the prediction. For  $y$ -value of east longitude, the bigger value implies that the prediction is east to the target position and the smaller value implies west, and vice versa. For  $y$ -value of north latitude, the bigger and smaller values imply north and south to the target values, respectively.

The curves of Conf. 1, Conf. 2, and Conf. 3 are very consistent, and there is almost no difference in the figure. Conf. 5 uses the unnormalized data, and its predictions are similar to that of Conf. 1, Conf. 2, and Conf. 3. For the curve of Conf. 4, sometimes the deviations from the targets are large, and sometimes small, and show more instability. However, when combining systems, this instability could cancel out errors. So, the direction and magnitude of these instable deviations are also worth further study. From the qualitative point of view, prediction trends of all the five configurations are consistent with the target. Table 5 lists the mean absolute errors of the testing set, from the quantitative point of view.

The MAEs for longitude and latitude are calculated by

$$\text{MAE} = \frac{1}{30} \sum_{i=1}^{30} |t_i - a_i|, \quad (10)$$

in which 30 is the size of testing set;  $t_i$  and  $a_i$  are target and predictive output of the  $i$ th independent instance. The distance error (Dst., as row header in Table 5) is calculated by the common used equation

$$\text{Dst} = 110 \times \sqrt{\text{MAE}_{\text{lon}}^2 + \text{MAE}_{\text{lat}}^2}. \quad (11)$$

TABLE 5: Mean absolute errors of the testing set.

	Conf. 1	Conf. 2	Conf. 3	Conf. 4	Conf. 5
Lon. (°)	1.016	0.960	1.017	1.529	0.983
Lat. (°)	0.566	0.570	0.568	0.618	0.666
Dst. (km)	127.9	122.8	128.2	181.4	130.6

The distance errors of Conf. 1 and Conf. 3, whose means are zero, have a difference of only 0.3 km. Among the errors of Conf. 1, Conf. 2, and Conf. 3, whose means are near zero, they have a maximum difference of only 7.8 km. For such a large-scale motion as TC, these errors could be regarded as on the same level, while, for the distance error of Conf. 4, the difference is up to 56.8 km, comparing to the minimum error of Conf. 2.

#### 4. Discussion

The experiments show that data normalization does affect the performance, as we had expected. It is more likely to converge, if we use the data scaled to around zero, no matter the min.-max. method or normal distribution method. However, it is not so absolute that the data center should locate on the zero point. In our experiments, testing errors using the data scaled to (0, 1) are even smaller than that using the data scaled to (-1, 1). We consider it within the normal error tolerance range. But when the data center reaches 100 and distribution radius remains 1, the errors increase greatly. This is probably because the data located around zero are easier to cancel out errors. The farther the data center from zero and the shorter the data distribution radius, the harder the convergence.

The raw data are made up of all kinds of meteorological variables, so it is difficult to make sure that they can locate around zero. This could affect the performance. In our experiments, although the values far from zero also exist in the raw data, which is as low as -45 and high up to 525, the errors are not so large and the convergence speed is not as slow as Conf. 4. This is probably because of the diversity of

the raw data, in which some are around zero, some are not, and some are positive, while some are negative. They could also cancel out the errors to some degree.

The probing value ranges for data normalization in our experiments are not systematic, both for min.-max. method and normal distribution method. The experiments could be improved by using more data configurations to confirm the model trends. The experiments could have more different data centers and more ranges. For examples, the data centers could be 0, 1, 50, 100, 500, 1000, and so forth, and, on each data center, several radiuses on different orders of magnitude, depending on the center value, could be considered.

In our experiments, only one normalization method is used on all variables in the same data. In other words, in one experiment all the variables in the data set use the same normalization method. However, this is different from the real situations. Maybe different variables should use different methods. For example, the variable, longitude, should be scaled to an evenly distributed range, because the earth is round and the longitudes are evenly distributed on the surface, while the surface wind speed should use the normal distribution method because of its possible values.

Despite slower convergence speed and larger mean errors of Conf. 4, individual points on the fitting curve of independent instances exist with higher precision, as well as lower precision, and show more instability. However, when combining subsystems in the committee machines, which require the experts to be reasonably accurate (better than guessing) and diverse (errors are uncorrelated), some instabilities may be beneficial. The diversity of curves for Conf. 1, Conf. 2, Conf. 3 and Conf. 5 in Figure 6 is obviously not obvious. So, in the study of committee machines, how to make use of the normalization methods to control the model outputs is worthy of further exploration, not only showing some diversity, but also being limited to some degree.

## 5. Conclusions

In this work, we have proposed the experimental schemes to map the TCT data to different ranges, near to as well as far away from 0, using min.-max. and normal distribution methods. Then the normalized data were applied to the PLNN model to predict TC centers in SCS. It has been shown that both min.-max. and normal distribution methods produce similar results, as long as they map the data to similar value ranges. And we also demonstrated that mapping the data to around 0 could remarkably fasten the training speed, although, provided enough training iterations, reaching smaller error is possible in all given situations.

Future work will be devoted to initialization of the network weights, which is also supposed to greatly affect the results. Being more aware of how data normalization and initial weights of the networks influence the training process, plus the theoretically optimal learning rate provided by MATLAB Neural Network Toolbox, we can have a more comprehensive understanding of individual PLNN and can better design and control the combining system.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61203301), Major Special Funds for Guangxi Natural Science Foundation (no. 2011GXNSFE018006), and National Natural Science Foundation of China (no. 41065002). Ming Li acknowledges the supports in part by the National Natural Science Foundation of China under the Project Grants nos. 61272402, 61070214, and 60873264. Ming Li also thanks the support by the Science and Technology Commission of Shanghai Municipality under Research Grant no. 14DZ2260800. The authors are grateful to Xiao-Yan Huang for providing and refining the variable information of the data set.

## References

- [1] E. Egrioglu, U. Yolcu, C. H. Aladag, and E. Bas, "Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting," *Procedia—Social and Behavioral Sciences*, vol. 109, pp. 1094–1100, 2014.
- [2] E. Egrioglu, U. Yolcu, C. H. Aladag, and C. Kocak, "An ARMA type fuzzy time series forecasting method based on particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 935815, 12 pages, 2013.
- [3] C. H. Aladag, U. Yolcu, E. Egrioglu, and E. Bas, "Fuzzy lagged variable selection in fuzzy time series with genetic algorithms," *Applied Soft Computing Journal*, vol. 22, pp. 465–473, 2014.
- [4] E. Egrioglu, "PSO-based high order time invariant fuzzy time series method: application to stock exchange data," *Economic Modelling*, vol. 38, pp. 633–639, 2014.
- [5] Q. Wang, J. Liu, and L. Zhang, "The study on ensemble prediction of typhoon track," *Journal of the Meteorological Sciences*, vol. 32, pp. 137–144, 2012.
- [6] T. C. Lee and W. M. Leung, "Performance of multiple-model ensemble techniques in tropical cyclone track prediction," in *Proceedings of the 35th Session of the Typhoon Committee*, Chiang Mai, Thailand, November 2002.
- [7] T. C. Lee and M. S. Wong, "The use of multiple-model ensemble techniques for tropical cyclone track forecast at the Hong Kong Observatory," in *Proceedings of the WMO Commission for Basic Systems Technical Conference on Data Processing and Forecasting Systems*, Cairns, Australia, December 2002.
- [8] J. S. Goerss, "Tropical cyclone track forecasts using an ensemble of dynamical models," *Monthly Weather Review*, vol. 128, no. 4, pp. 1187–1193, 2000.
- [9] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [10] L. Jin, J. Jin, and C. Yao, "A short-term climate prediction model based on a modular fuzzy neural network," *Advances in Atmospheric Sciences*, vol. 22, no. 3, pp. 428–435, 2005.
- [11] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, pp. 1464–1468, 1997.

- [12] T. Jayalakshmi and A. Santhakumaran, "Statistical normalization and back propagation for classification," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, pp. 89–93, 2011.
- [13] Q. Zhang and S. Sun, "Weighted data normalization based on eigenvalues for artificial neural network classification," in *Neural Information Processing*, vol. 5863 of *Lecture Notes in Computer Science*, pp. 349–356, Springer, Berlin, Germany, 2009.
- [14] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *IRE WESCON Convention Record*, pp. 96–140, 1960.
- [15] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing, 1997.
- [16] N. Syed, H. Liu, and K. K. Sung, "Incremental learning with support vector machines," in *Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence*, 1999.
- [17] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 31, no. 4, pp. 497–508, 2001.
- [18] S. Baluja, *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, Carnegie Mellon University, Pittsburgh, Pa, USA, 1994.
- [19] L. Jin, X. Huang, and X. Shi, "A study on influence of predictor multicollinearity on performance of the stepwise regression prediction equation," *Acta Meteorologica Sinica*, vol. 24, no. 5, pp. 593–601, 2010.
- [20] L. Xie, "A climate-continuity model for the forecaste of the south sea cyclone patheo," *Marine Forecasts*, vol. 6, pp. 20–30, 1989.
- [21] J. Jin, *Committee machines and its applications [Ph.D. dissertation]*, East China Normal University, Shanghai, China, 2008, (Chinese).



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

