Forecasting Rainfall with Recurrent Neural Network for irrigation equipment

Junyu Zhou, Yubin Zhang *, Si Tian, Shangding Lai
Ningbo University of Finance and Economics, Ningbo, China
Lzlgdx08-2@163.com

Abstract. The irrigation decision-making system based on Knowledge-based Engineering (KBE) can accurately predict water requirements and realize smart irrigation. Recurrent neural network (RNN) model have recently showed state-of-the-art performance in this system. This paper deals with the problem of long-term rainfall forecasting based on this network which predicts target rainfalls based on contextual information. A novel recurrent neural network with long short term memory (LSTM) is put for model sequence process for forecasting rainfall. Back-propagation through time (BPTT) algorithm is described for updating recurrent network’s weights. Extensive empirical comparison with three networks, Feed-forward neural network (FNN), Wavelet neural network (WNN) and Auto-regressive Integrated Moving Average (ARIMA), are also provided at various numbers of parameters and configurations. Simulation results demonstrate that the recurrent model with LSTM, trained by the suggested methods, outperforms the others networks.

1. Introduction
Forecasting future events is a key task in many practical irrigation situations. One of these situations is the forecasting of rainfall. Rainfall prediction is an important factor in agricultural countries like China which is an agricultural country. Due to highly complex interactions and the contribution of various meteorological parameters, rainfall forecasting is a severe task. Statistical rainfall modeling is to predict the next rainfall in given context, thus it can be dealt with sequential data prediction problem. Many techniques [1-4] to model such as ANN have achieved impressive accuracy improvements. However, these models are limited to fixed-length contexts.

RNN is used for sequence processing [5]. RNN have several properties that make it an attractive choice for sequence data. It is able to incorporate contextual information from past inputs or future inputs, which allow it to instantiate a wide range of sequence-sequence maps. Despite its attractive qualities, training conventional RNN with the gradient-based BPTT technique is difficult due to the vanishing gradient and exploding gradient problems [6]. LSTM is an RNN architecture specifically designed to overcome these problems in practice. The model learns itself from the data how to represent memory. This allows for instance efficient representation of patterns with variable length.

The purpose of the paper is to demonstrate the power of RNN using LSTM architecture by applying them to the task of predicting the next rainfall in a stream of rainfall. Owing to the large time-scale, we are based on 36-days ahead meteorological predictions. Three networks are employed to produce rainfall forecasts to account for the precision of the model. The experimental results show that accurate forecasts are obtained, with the recurrent forecast model exhibiting superior performance with regard to the other networks.
2. Methodology
A recurrent neural network is a natural generalization of feed-forward neural network to sequences. The difference is that it not only operate on an input space but also on an internal state space – a trace of what already has been processed by the network. The idea behind RNN is to make use of sequential information. Unlike feed-forward neural network, a simple RNN has activation feedback which embodies short-term memory [7-10]. By using this property, information can cycle inside the network for arbitrarily long time [5]. A state layer is updated not only with the external input of the network but also with activation from the previous forward propagation.

The standard RNN [11][12] is formalized as follows: Given a sequence of input vectors ($x_1, \ldots, x_T$), the RNN computes a sequence of hidden states ($h_1, \ldots, h_T$) and a sequence of outputs ($y_1, \ldots, y_T$) by iterating the following equations for $t = 1$ to $T$:

$$h_t = \tanh(W_h x_t + W_{hh} h_{t-1} + b_h)$$

(1)

$$y_t = W_{oh} h_t + b_o$$

(2)

Where $W_h$ is the input-to-hidden weight matrix, $W_{hh}$ is the hidden-to-hidden weight matrix, $W_{oh}$ is the hidden-to-output weight matrix, and the vectors $b_h$ and $b_o$ are the biases. The input layer is a concatenation of $h_{t-1}$ and $x_t$, where $h_{t-1}$ is a real-valued vector. $x_t$ is the input rainfall at time $t$.

3. The experiments

3.1. Data set
The dekad precipitation data of the Yulin station of China are used. The data have been obtained from the China Meteorological Data Sharing Service (CMDSS) [13]. The data is 14 years with an observation period between 2000 and 2013.

The data contain year, month, dekad and precipitation (mm). The data was set into training and test sets, containing the data from 2000 to 2011 and the data from 2012 to 2013 respectively. The data sets’ statistics are showed in Table 1. In this table, Mean, Std, Min, Max, Median denote the over all mean, standard deviation, minimum, maximum, median, respectively. It can be seen that the precipitation data show scattered distribution.

| Data set | Mean (mm) | Std (mm) | Min (mm) | Max (mm) | Median (mm) |
|----------|-----------|----------|----------|----------|-------------|
| Training | 11.00     | 21.34    | 0        | 226      | 2           |
| Test     | 17.54     | 33.35    | 0        | 203      | 4           |
| Entire   | 11.84     | 23.29    | 0        | 226      | 3           |

3.2. Training details
Networks are trained in several epochs, in which all data from training set are sequentially presented. Size of hidden layer should reflect amount of training data and usually be 30-500 hidden units. Based
on our experiments, we used 500 units. To train the network, weights are initialized to small values (random Gaussian noise with zero mean and 0.1 variance), starting learning rate is 0.001. As usual, the full error gradient was calculated using BPTT for each sequence and weight updates were carried out at the end of each sequence. After there is again no significant improvement, training is finished. Convergence is usually achieved after 10-20 epochs.

For comparison, additional models had been trained. To establish a fair comparison basis, the structure of the competing networks is selected so that they contain approximately the same number of parameters as the recurrent networks. The hidden layer contains five neurons using sigmoid activation functions.

3.3. Evaluation criteria
As a measure to evaluate the performance of the model, three measures were used: RMSE, MAE, Nash-Sutcliffe coefficient (NS)[20]. The RMSE, MAE and NS are expressed as:

\[
RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (p_t - y_t)^2}
\]

(3)

\[
MAE = \frac{1}{n} \sum_{t=1}^{n} |p_t - y_t|
\]

(4)

\[
NS = 1 - \frac{\sum_{t=1}^{n} (p_t - y_t)^2}{\sum_{i=1}^{n} (y_i - y_{i-1})^2}
\]

(5)

In the formulate, \(p_t\) is the forecast value and \(y_t\) is the corresponding target value. The MAE is the average of the absolute error of the prediction over all input patterns[21]. The NS is particularly suited to time-series because it compares the model errors to a naive model, which is the difference between the current and previous observations. The lower the errors and the higher the NS and \(R^2\), the better the model.

4. Results
Figure 2 and Figure 3 show the best results for the training and testing data respectively. The green line is the true values and the blue line is the predicted values. From figures, it can be seen that the observation and the forecasts are very close, and the trends are very high similarity in the curves.

Three input combinations, lag-9, lag-18 and lag-36, the preceding precipitations are evaluated to estimate current precipitation value. In all cases, the output is the next prediction. The evaluation index of RNN model in training and test period are given in Table 2.
Table 2 The evaluation index of RNN applications for Yulin Station.

| Inputs | RMSE | MAE | NS  |
|--------|------|-----|-----|
| Training | Lag-9 | Lag-18 | Lag-36 |
| RMSE   | 22.51 | 19.61 | 12.70 |
| MAE    | 15.30 | 13.89 | 8.74  |
| NS     | 0.81  | 0.86  | 0.93  |
| Testing | Lag-9 | Lag-18 | Lag-36 |
| RMSE   | 20.00 | 13.02 | 10.40 |
| MAE    | 7.29  | 4.68  | 3.10  |
| NS     | 0.73  | 0.82  | 0.89  |

As depicted in table 2, there is a clear relationship between the input data and the target data. The larger lags of the input data, the greater the accuracy of the model becomes.

For the purpose of comparison, precipitation estimation has been carried out by FNN and WNN, as shown in Figure 4-7 respectively. To establish a fair comparison basis, the structure of the competing networks is selected so that they contain approximately the same number of parameters as the recurrent networks. Under these conditions, the models are evaluated in terms of representation power and their capabilities to produce efficient multistage forecasts.

The table below shows the summary of the performance for all the four networks.
Table 3 summarises the performance of the different network architectures.

| Network | RMSE (Training/Testing) | MAE (Training/Testing) | NS (Training/Testing) |
|---------|-------------------------|------------------------|-----------------------|
| LSTM    | 12.70/10.40             | 8.74/3.10              | 0.93/0.89             |
| FNN     | 14.04/13.84             | 9.49/3.91              | 0.86/0.81             |
| WNN     | 21.21/15.22             | 11.56/6.24             | 0.82/0.78             |

It is obvious that the recurrent forecast models exhibit consistently better performance compared to the other models from the various performance criteria viewpoints. Contrary to other networks, recurrent networks can be sensitive, and be adapted to past inputs.

5. Conclusion
In this paper, a recurrent network with long short term memory architecture was designed and implemented for forecasting precipitation. The network use modified back-propagation with short-term memory filters, represented by input lags and with feedback loops. This network type is especially well suited to this task, where context is vitally important. As opposed to previous approaches, it does not suffer from conceptual problems of standard recurrent neural network training. The RNN model was trained applying to different input combinations of precipitation data of Yulin Station of China. The results demonstrated that the recurrent network outperform other models and more robust in long-term forecasting. LSTM architecture is much faster to train than other approaches, and also slightly more accurate. Also, the RNN model was compared with three models. The comparison results indicated that the RNN performed better than the FNN, WNN and ARIMA models in forecasting precipitations.

For future work, we would like to apply LSTM combined with hidden Markov models to form a hybrid sequence system which likely leads to even greater forecasting accuracy.

Acknowledgments
The authors wish to thank the other team-mates for useful comments and discussions. We would also like to thank the anonymous reviewers for their comments. The work was partially supported by project of Science and technology project of Zhejiang provincial water resources bureau under Grant Nos. RB1807, and Ningbo Science and Technology under Grant Nos 2019C10103, 2017C10039, 2017C10031.

References
[1] Yubin Zhang, Zhengying Wei, Lei Zhang. Applicability evaluation of different algorithms for daily reference evapotranspiration model in KBE system[J]. International Journal of Computational Science and Engineering, 2019, 18(4) 631-674
[2] Nirmala M., Sundaram S.M. (2010). A seasonal ARIMA model for forecasting monthly rainfall in Tamilnadu. National Journal on Advances in Building Sciences and Mechanics, Vol. 1, No. 2.
[3] Yubin Zhang, Zhengying Wei, Lei Zhang, Qinyin Lin, Jun Du, Improved Online Sequential Extreme Learning Machine for Simulation of Daily Reference Evapotranspiration, Tecnología y Ciencias del Agua[J]. 2017, 8(02):127-140.
[4] Ozgur Kisi, Mesut Cimen (2012). Precipitation forecasting by using wavelet-support vector machine conjunction model. Engineering Applications of Artificial Intelligence, 25, 783–792.
[5] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Černocký, Sanjeev Khudanpur (2010). Recurrent Neural Network Based Language Model. In Eleventh Annual Conference of the International Speech Communication Association, 1045-1048.
[6] Hasim Sak, Andrew Senior, François Beaufays (2014). Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition.
[7] Alex Graves, Navdeep Jaitly (2014). Towards End-to-End Speech Recognition with Recurrent Neural Networks. Proceedings of the 31st International Conference on Machine Learning, Beijing, China, JMLR: W&CP volume 32.

[8] Shujie Liu, Nan Yang, Mu Li and Ming Zhou (2014). A Recursive Recurrent Neural Network for Statistical Machine Translation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 1491–1500.

[9] Yubin Zhang, Zhengying Wei, Qiyin Lin, Lei Zhang, Jing Xu. MBD of grey prediction fuzzy-PID irrigation control technology[J]. Desalination and Water Treatment, 2018, 110: 328-336.

[10] Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research, 3, 115–143.

[11] Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber (2001). A field guide to dynamical recurrent neural networks. IEEE Press.

[12] Herbert Jaeger (2013). A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach.

[13] Zhang Y B, Wei Z Y, Zhang L, et al. (2016). Simulation of Daily Reference Evapotranspiration Based on Least Squares Support Vector Machine Optimized by Coupled Simulated Annealing. Water Saving Irrigation, 10: 133-138.