Deep Learning by Long Short-Term Memory Model for Rainfall Prediction in Thailand

Wanna Wirojdanthai
Computer Science, Faculty of Science,
Chandrakasem Rajabhat University
Bangkok, Thailand
e-mail: aj.wanna252@gmail.com

Abstract— In this work, the long short-term memory (LSTM) technique in the deep learning neural network is applied to find the pattern in time series data. The LSTM model is applied to find the rainfall pattern in Thailand over 20 years. The results show that the LSTM model can predict the seasonal behavior in each year. Also, it can detect correctly the peaks in the rainy season and can capture the minimum rainfall in a summer period over six years of testing data. The average rainfall in rainy season is approximately 250 mm. The training process is done successfully without vanishing gradient problem. These results show the advantage of the LSTM model in time series forecasting for seasonal data.

Keywords-Recurrent neural network, LSTM, rainfall, time series

I. INTRODUCTION

Time series modelling arises in many fields of applications that involve the process or data relating to temporal change of variables. Time series techniques relate dependent variables with time, then they can be represented as the function in time variable. The time series analysis is important in the fields of environmental engineering, data communication, atmospheric forecasting, or water quality in a river.

To solve or predict the values in the time series problem, there are many algorithms in neural network that can be applied to solve, for example, feed forward neural network, neural network with back propagation, or recurrent neural network. Some of the recent developments include temporal characteristics into the neural network. It can be created by a spatial representation of time pattern by giving input sequences, see [1,2,3]. The neural network with time delays has been developed by [4, 5]. The combinations of these techniques can be found in [6].

The recurrent neural network is one of the an important method in deep learning technique. It can be used to predict some particular time series problems. But for the problem with very long data sequence, the vanishing gradient problem usually occurs during the training process. The next development of neural network model to overcome this training problem is called the long short term memory or LSTM model, see [7,8]. There are many functions and gates in each LSTM unit. We will describe briefly the basic concepts of these algorithms. Finally, the LSTM approach will be applied to predict the rainfall pattern during 2014-2017 in Thailand.

II. NEURAL NETWORK MODELS

A. *Feed forward neural network*

As shown in Figure 1, the input data from an input layer can flow only in one direction (one way) through the hidden layers. The final layer is the output layer. By this architecture, it has no memory in the hidden layer, so there may be a training problem to find optimal weights, [7]. Also, it cannot notice the order of value in time, it can learn only from the past, so the predicted values are usually incorrect. More details of feed forward algorithm can be found in [7].

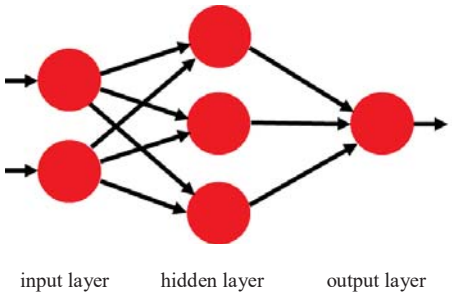

Figure 1. Feed forward neural network

B. *Back propagating neural network*

The back propagation method receives the information from the input and then check whether the output is correct. If it is incorrect enough under a given threshold, the training process is performed back to the network to find the better accurate output. Mathematically, the error variation in terms of partial derivatives with respect to the weighted values are calculated. The gradient descent method can be applied to find numerically these optimal values. Then the whole computations can be done iteratively to minimize the error functions. The concept of this method is shown in Figure 2.

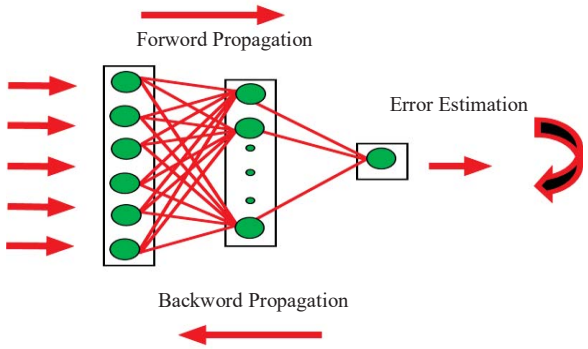

Figure 2. Back propagation architecture

C. *Recurrent Neural Networks*

Recurrent Neural Network (RNN) is one of other neural network models that can learn or find optimal weighted values from the previous values and predict the next values. This is sometimes referred to the sequential data algorithm. The applications can be any time series data, for example financial data, weather data, and speech data. The RNN can solve the dynamics of temporal data that connect contents within individual time frame. The RNN diagram is shown in Figure 3. In RNN, the information from the input nodes can flow through the hidden layer nodes, the calculations of fitted weighted values are performed iteratively until the designed errors are minimized when applying the efficiently optimization solvers. By this reason, the RNN are referred to short time memory algorithm. It can generate output values and then copy these values back into the network for better training processes. More details of the recurrent neural network algorithm can be seen in [8].

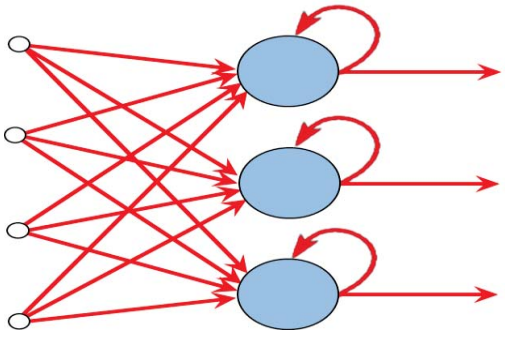

Figure 3. Recurrent neural network in view of node details.

It should be noted that the RNN can find weighted values from the current and the previous input. The calculations can be done iteratively by the gradient descent method or back propagation method. The RNN can map information from one to many, many to one or many to many whereas the feed forward algorithm can only map from one input to one output.

As shown in Figure 4, the RNN can be modelled as the neural network sequence. It can be trained iteratively after receiving information from the previous network. If we have time series data, the information can pass from the previous time stage to the next time stage. This process calculates errors for every time step. So, the computational time is very massive when the number of time step is very large. Also when the gradients of errors are very small, the training process stops. It is called as vanishing gradients problem. This situation will take the number of iterations very large. It is required the method that needs not to find error gradients. The long short term memory (LSTM) method proposed by [9, 10] can solve this problem. More details are given in the next section.

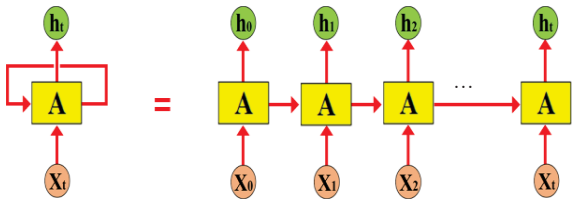

Figure 4. RNN as the sequence of neural network

D. *Long short term memory (LSTM)*

From the limitation of RNN as described previously, the Long Short-Term Memory (LSTM) network is developed to solve vanishing gradient problem. The LSTM can remember input values over a long period of time since it stores previous information in a virtual memory. So, the LSTM can read data, write data and delete information from its virtual memory through gate cells that they can decide to delete or store information. It likes a gate that can open or close. It implies that the method can learn over time.

The LSTM model is first introduced by [9]. The main concept is to solve the vanishing gradient problem as discussed in the previous section that this problem usually occurs in the traditional recurrent neural network. The neural unit in the LSTM is called as a memory cell which represents the hidden layer of deep learning neural network architecture. There are three types of gates composing of input gate, forget gate and output gate. The input gate can estimate and reset the new input. The forget gate can delete the information if it is not important. The output gate can allow or delete the information before passing the correct value to the next LSTM cell. The value returned from the gate is provided by a sigmoid function. Its range is 0 to 1. If the value is 1, it allows to do back propagation again. The vanishing gradients problem can then be solved by the LSTM due to its architecture that can keep error gradient large enough

in the iteration process. When the gradient is vanishing, the gate closes. Therefore the computational time in the training process is short.

The structure in each memory cell is shown in Figure 5. There is a recurrent edge that has many weight value in each cell state to overcome the vanishing problems. The values associated with this recurrent edge is called cell state. In the LSTM cell unit, there are many controlled units with various operators.

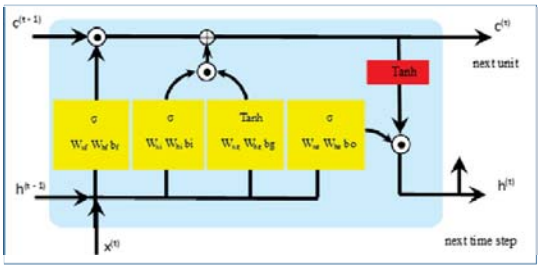

Figure 5. The structure of the LSTM unit in a hidden neural network layer.

The calculation process can be described as follows. The sequence of cell state in the time series data from the previous time step denoted by $C^{(t-1)}$ is recalculated to obtain the update cell state denoted by $C^{(t)}$. The operation is done without being multiplied directly with any weight factor.

The symbol is the standard element-wise product and the symbol ⊕ is the standard element-wise summation. The variable $x^{(t)}$ refers to the actual input data at time t, and $h^{(t-1)}$ is the result calculated at time t – 1.

There are four main blocks in the LSTM unit. These are composed of the blocks with an activation function which can either be the sigmoid function (σ) or the hyperbolic tangent (Tanh) function. There are various weight values (W with a subscript) that are associated with each block. These blocks apply the linear summation by performing matrix and vector multiplications obtained from their inputs. The computations on these units are done using the sigmoid activation functions. Finally, the output is passed through operation, . So the last block is generally called a gate unit.

Generally, in an LSTM unit, there are three different types of gates. These are the forget gate, the input gate, and the output gate. The function for each gate can be described as follows.

The forget gate (f) allows the information in the cell to reset the cell state without indefinitely output result. The forget gate can decide which information should be allowed to pass through or to drop the information value. The calculation in the forget gate can be expressed by the following formula,

$$f = \sigma(W_{xf}x^{(t)} + W_{hf}x^{(t-1)} + b_f) \quad (1)$$

where W_{xf}, W_{hf} are weight matrices and b_f is a bias vector associated with input data $x^{(t)}$ and $x^{(t-1)}$. Note that the forget gate is not original proposed in the LSTM unit.

This gate is introduced later by [10]. It can improve the performance of the general LSTM architecture. The input gate (i) and input node (g) are responsible for updating the cell state. They are computed as follows:

$$i = \sigma(W_{xi}x^{(t)} + W_{hi}x^{(t-1)} + b_i) \quad (2)$$

$$g = Tanh(w_{xg}x^{(t)} + w_{hg}x^{(t-1)} + b_g) \quad (3)$$

Similar to the forget gate, the formula is composed of weighted matrix and bias vector with different subscripts to link to between input and output in each gate.
The cell state at time t is computed as follows.

$$c^{(t)} = (c^{(t-1)} \odot f) \oplus (i \odot g) \quad (4)$$

The calculation is done by the element-wise product, and element-wise summation from the forget gate, input gate and input node. The update value in the LSTM unit is done by the output gate (o). The calculation is expressed by

$$o = \sigma(W_{xo}x^{(t)} + W_{ho}x^{(t-1)} + b_o). \quad (5)$$

Finally, the update value prepared for the next LSTM unit at time step t is that

$$h^{(t)} = o \quad Tanh(c^{(t)}) \quad (6)$$
$$h^{(t)} = o \quad Tanh(c^{(t)}) \quad (7)$$

By this complicate architecture in the LSTM model, the prediction regarding the sequence to sequence in the time series forecasting problem is efficient. The vanishing gradient problem that usually occurs in the standard recurrent neural network can be solved by the LSTM model with various gate functions to memorise or to forget the input information and cell state from the previous neural unit in deep learning model. In the next section, we will apply the LSTM model to the rainfall time series data in Thailand throughout more than 20 years. The performance can be measured by the root mean square error (RMSE).

III. *MODEL RESULTS*

Rainfall data is obtained from the Hydro and Agro Informatics Institute, Ministry of Science and Technology, Thailand (HAII). Data is collected during 1981 to 2010. The value of rainfall is averaged in millimeter unit over month. Plot of rainfall for each month is shown in Figure 6. The profile is in seasonal. It has individual peaks in rainy season and minimum values in summer.

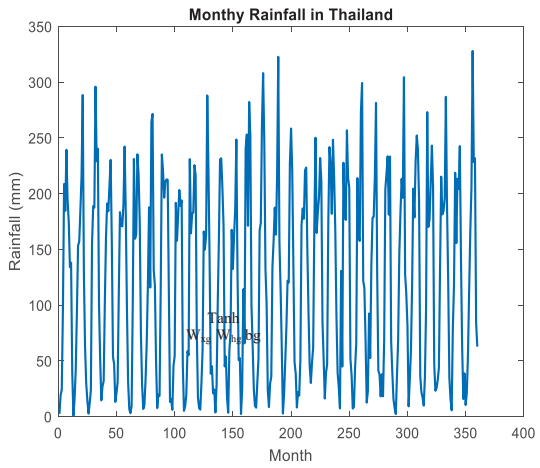

Figure 6: Plot of average monthly rainfall in Thailand during 1981-2010.

The whole data is divided into two parts which are training and testing parts with the ratio 80 and 20 percent respectively. For a better fit and prevent the diverging in the training process, we perform standardize the data to shift the data value to zero mean and unit variance. The LSTM can learn the training data and predict the value in the next time step. Table 1 shows the number of hidden units and the corresponding RMSE during training process with 250 epochs. The optimal values of hidden units is between 200 and 240 in this case.

Table 1: The number of hidden units in the LSTM and the RMSE during training process with 250 maximum iterations.

Number of hidden units	RMSE in the Training process
50	0.3714
100	0.3692
150	0.3445
200	0.3213
240	0.3261

For the application of LSTM to rainfall prediction, we construct the LSTM model with 200 hidden units in the LSTM layer. The maximum number of iterations or epochs is set to be 250. In our work, the optimal weights in the LSTM network can be estimated using the Adam optimization algorithm, see [11]. This algorithm is an extension to stochastic gradient descent approach that is widely used in many deep learning applications. The gradient threshold is set to be 1. The initial learning rate is 0.005. We then drop this rate by a factor of 0.2 after 125 epochs to keep better convergence. As shown in Figure 7, the training process start converging around 100 iterations. When we stop the process, the RMSE is below 0.4 for 250 iterations.

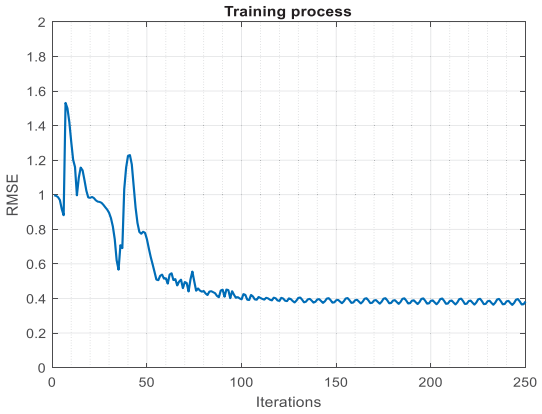

Figure 7: Root mean square error and iteration number in the LSTM training process.

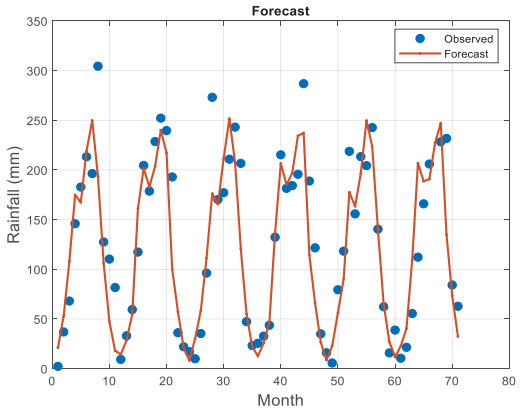

Figure 8: Plot of rainfall for testing data.

The performance of the LSTM prediction is shown in Figure 8. The scatter plot of the observed testing data is shown by circles while the model prediction is shown by a line. This data set covers the last 6 years record. There are six peaks in rainy season. The seasonal period is approximately 12 months as expected. The LSTM results are well fit to the observed data. The LSTM model can capture most important behaviors, dry in summer and wet in rainy season. The peak averaged is around 250 mm except 3 years having more than 250 mm. In normal situation, the LSTM model can capture the amount of rainfall correctly. The RMSE for this testing data is 30.59. This value is significant because we have compared with the whole testing data that includes unusual events. But, it less than 0.1 when comparing the overall results with the extreme peak with its value of 328.01.

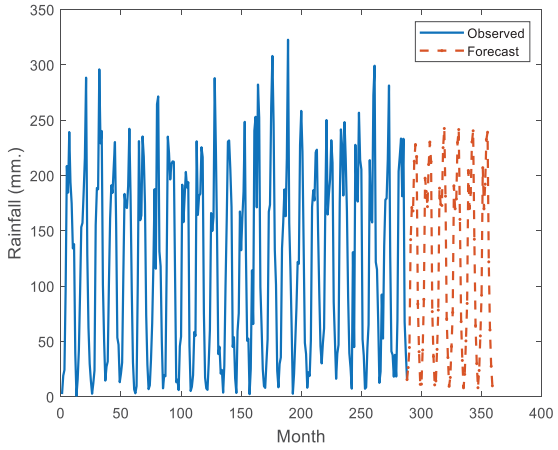

Figure 9: Plot of rainfall for observed and predicting data.

The forecasting rainfall for the last 6 years by the LSTM model is shown by a dashed line in Figure 9. Overall the model can predict the rainfall pattern continuously. It shows reasonably the maximum and minimum of rainfall in each year.

In this work, the basic concepts of neural network models have been reviewed, starting from the basic feed forward network and following by the back propagation algorithm. The purposes of our work is to apply the efficient neural network model for time series data forecasting. The recurrent neural network model is one of the candidate model to predict the sequence values for next time step in time series problem, but it is well-known that the traditional recurrent neural network has a vanishing gradient problem during the training process. The development of the LSTM becomes a powerful technique to solve the vanishing gradient problem since it has many gates that can allow or delete the information for transferring data in the next time step. The applications of the LSTM to predict the rainfall time series in Thailand over 20 years are demonstrated. The results show that the LSTM model can predict the seasonal behavior over a one-year period. Also, it can detect correctly the peaks in the rainy season and can capture the minimum rainfall in a summer period over six years of testing data. The average rainfall in rainy season is approximately 250 mm. The extension of is work should be the relationship finding between rainfall pattern and other atmospheric indexes. This work is in process and will be reported elsewhere further.

ACKNOWLEDGMENT

The author would like to thank the Hydro and Agro Informatics Institute (HAII) of Thailand for supporting rainfall data.

REFERENCES

[1] Hsu K.L., Gup H.V., Sorooshian S., "Artificial neural network modeling of rainfall-runoff process", Water Resour Res, 1995, 31(10):2517-2530.

[2] Jain S.K., Das D., Srivastava D.K. "Application of ANN for reservoir inflow prediction and operation", J. Water Resour Plan Manage, 1999, ASCE, 125(5):263-271.

[3] Coulibaly P., Anctil F., Bobee B., "Daily reservoir inflow forecasting using artificial neural networks with stopped training approach", J. Hydrol, 2000, 230:244-257.

[4] Waibel A., Hanazawa T., Hintin G., Shikano K., Lang K.J., Phoneme recognition using time delay neural networks, IEEE Trans ASSP, 1989, 37(3):328-339.

[5] Wan E.A., "Time series prediction using a connectionist network with internal delay lines, Time series prediction: forecasting the future and understanding the past. Addition-Wesley, 1993, Reading: 195-217.

[6] Coulibaly P., Anctil F., Bobee B., "Multivariate reservoir inflow forecasting using temporal neural networks", J. Hydrol Eng ASCE, 2001, 6(5):367-376.

[7] Peter A., Burgsteiner H., Maass W., "A learning rule for very simple universal approximators consisting of a single layer of perceptrons", Neural Networks, 2008, 21(5): 786-795.

[8] Schmidhuber J., "Deep Learning in Neural Networks: An Overview", Neural Networks, 2015, 61: 85–117.

[9] Hochreiter S., Schmidhuber J., "Long short-term memory", Neural Computation, 1997, 9(8): 1735-1780.

[10] Gers F., Schmidhuber J., Cummins F., "Learning to Forget:Continual Prediction with LSTM", Neural Computation, 2000, 12:2451-2471.

[11] Diederik P. K., Jimmy B. Adam, "A method for stochastic optimization", The 3rd International Conference for Learning Representations, San Diego, 2015.