LSTM

Robin Yadav

1 Introduction

Long Short Term Memory (LSTM) is a type of Recurrent Neural Network (RNN), and is used for learning, processing, and classifying sequential data [1]. Learning to store information or data over long period of time intervals via recurrent backprogation takes very long time. Hence, gradient gradually vanishes as they propagate to earlier time steps, this is a substantial task for it to train the data [2], [3]. This can be explained using Figure 1 and 2.

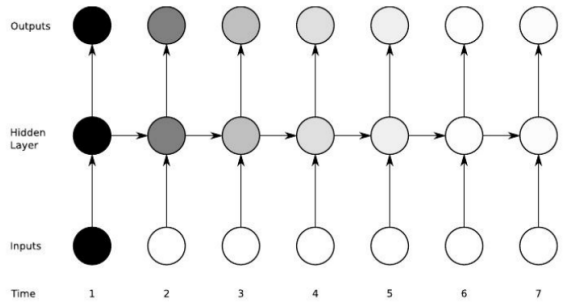

Figure 1: **Vanishing gradient problem for RNNs.** The sensitivity increases as the network backpropagates through in time. The darker the shade, the greater the sensitivity.

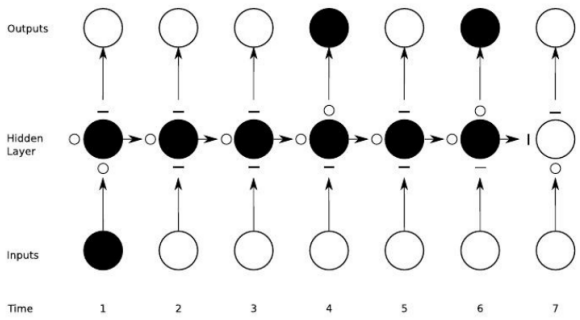

Figure 2: **Preservation of gradient information by LSTM.** The sensitivity of the output layer can be switch on and off.

And also, for the non-convex problem, the solutions confuse between local minimum and global minimum. To overcome these problem, LSTM has been introduced as RNN languages modelling learning algorithm based on the feedforward architecture [2]. By implementing LSTM language modelling concept, the time series prediction for the HF spectrum can be possible [4]. Sequential data (time series and ordered data structures) is widely used in natural language processing and speech recognition. For example, the probability of a time series x_1^N can be decomposed based on Markov assumption as in Equation (1)

$$p(x_1^T) = \prod_{t=1}^{T} p(w_t)|p(w_{t-n+1}^{t-1}) \tag{1}$$

such that only the most recent $(n-1)$ preceding words (time series) are considered for predicting the current word (time) x_t [2]. In simple word, task is to use currently available points in the time series to predict the future point. The target for the network is the difference between the values $x_{(t+p)}$ of the time series p steps ahead and the current value x_t.

1.1 Methodology

LSTM is based on RNN, therefore, the basic structure of RNN is explained first and then LSTM structure is explained referencing RNN. RNN memorize information from previous data with feedback loops inside it, which helps to keep data information over time as shown in Figure 3 [5].

It has an arrow pointing to itself, indicating that the data inside block "A" will be recursively used. Once expanded, its structure is equivalent to the chain shown in the right-hand

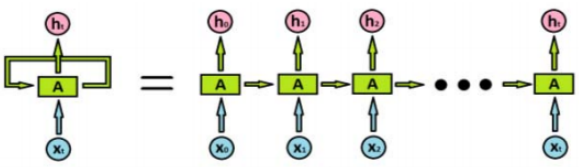

Figure 3: Basic structure of RNN unit [5]

side of Figure 3. The problem arises when long term dependencies of the previous data or sample in the datasets. And LSTM memorize the information for the long period of time , which is important in many applications such as time prediction of the HF spectrum[5]. The basic structure of the RNN and LSTM are similar as shown in Figure 4 and Figure 5 respectively.

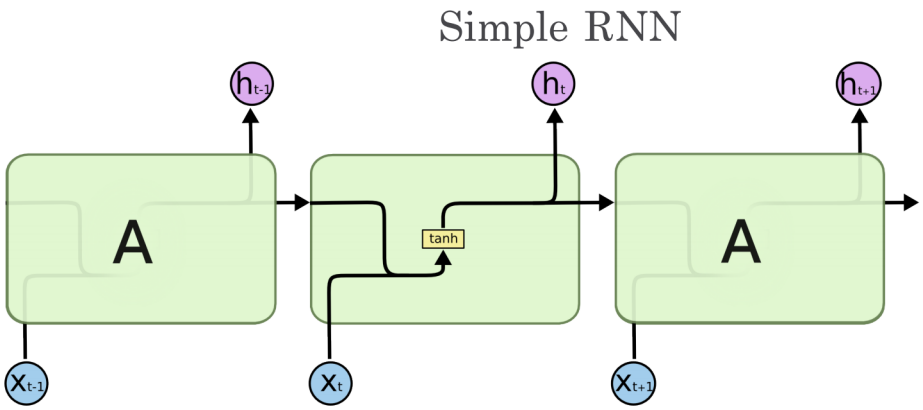

Figure 4: RNN's Cell [5]

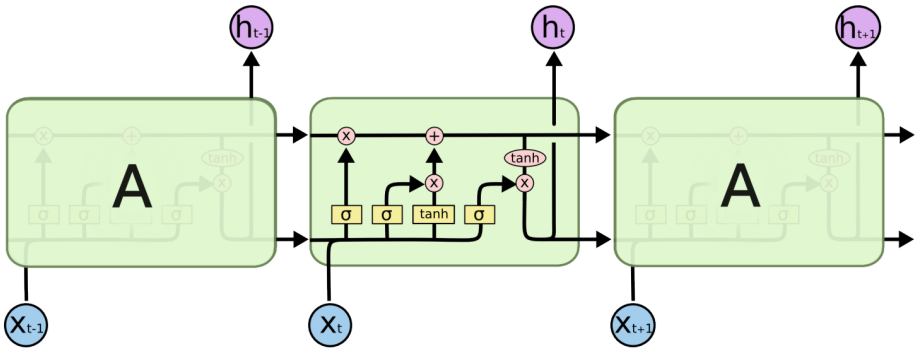

Figure 5: LSTM's Memory Cell [5]

The difference between RNN and LSTM are: RNN cell has only one tanh layer while LSTM cell has four layers: forget gate layer, store gate layer, new cell state layer, output layer, and previous cell state as shown in Figure 5 [5].

The forget layer is responsible for deciding what information to retain from the previous cell state, and what information is to be forgotten or removed [2],[5].

$$f_t = \sigma(W_t.[h_{t-1}, x_t] + b_f) \tag{2}$$

The store gate layer has an input gate using which we calculate another variable called new candidate values. The new candidate values are information which seem relevant are added to the cell state [5].

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{4}$$

The new cell state layer calculates the new cell state by updating the information from last cell. And the new cell state is calculated using the information acquired from the previous two layers [5].

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

The output layer makes use of all this information gathered over the last three layers to produce an output.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

Also, the cell state at the top of the Figure 5 starting with $c(t-1)$ runs horizontally as it keeps the information integrity from long period of time with some minor linear attractions [5].

References

[1] Schmidhuber J. Gers F.A., Eck D. *Time Series Predictable Through Time-Window Approaches.* Springer-Verlag London Limited 2002, London, United Kingdom, 2002.

[2] M. Sundermeyer, H. Ney, and R. Schlüter. From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):517–529, March 2015.

[3] Cs 224d: Deep learning for nlp from standford university.

[4] Haris Haralambous and Harris Papadopoulos. 24-h hf spectral occupancy characteristics and neural network modeling over northern europe. *IEEE Transactions on Electromagnetic Compatibility*, 59(6):1817–1825, 2017.

[5] D. Dong, Z. Sheng, and T. Yang. Wind power prediction based on recurrent neural network with long short-term memory units. In *2018 International Conference on Renewable Energy and Power Engineering (REPE)*, pages 34–38, Nov 2018.