LSTM Method Discussion Report

Lawrence Owusu, Jordan Sturtz, and Swetha Chittam

I. Summary

LONG Short Term Memory (LSTM) is a powerful machine learning method for learning tasks involving sequences with long-term dependencies. LSTM is a variant of recurrent neural networks with a specialized architecture designed to solve the vanishing gradient problem that prevents deep neural networks from learning long-term dependencies. The original authors of LSTM, Hochreiter and Schmidhuber, designed the first variant of LSTM in 1997 that introduced the idea of creating a unit in the neural network architecture that enforces constant error backflow [1]. To solve the problem of conflicting weight signals, Hochreiter and Schmidhuber then added two additional features to their model architecture: an input gate and an output gate [1]. Later, other authors would develop this architecture further to add a "forget" gate, which is the variant of the LSTM architecture we explore in this report [2]. These gates use a sigmoid layer together with pointwise multiplication to allow the network to filter signals.

II. Recurrent Neural Networks

Recurrent neural networks are neural networks that feature feedback connections. These feedback connections typically occur when the output at time step t is combined, usually concatenated, with the input for time step $t+1$ (See Fig. 1). LSTM is a recurrent NN. Each memory block is recurrently connected to the next memory block, which each memory block assigned to a specific timestep in the input sequence.

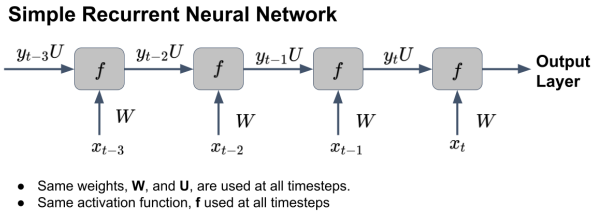

Fig. 1. Typical Simple RNN Structure

III. The LSTM Architecture

Characteristic of the LSTM architecture is a set of chained together cells called "memory blocks". The number of memory blocks in the chain equals the length of the timesteps in our target dataset (Fig 2). Each memory cell has three gating units (forget, input and output gates) which conditionally regulate how information flows into and out of the memory block (Fig 3). Intuitively, the "forget" gate can be viewed as a step where irrelevant information from the hidden state is first "forgotten" before being passed to the input gate, which constructs the new cell state. Before that cell state can output its value to the next memory cell, it is filtered again through an output gate that decides what values to keep internal to the memory cell (i.e. the hidden state) and what to output to the next memory cell or final output layer.

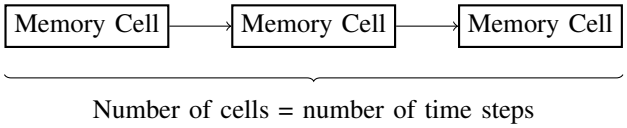

Fig. 2. Illustration of recurrently chained memory cells

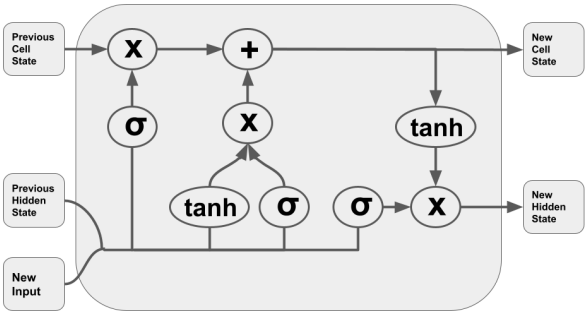

Fig. 3. LSTM architecture inside a memory cell

A. The Forget Gate

The forget gate decides what information to discard from the previous hidden state given the new input data and previous hidden state. What inputs to ignore is learned in a sigmoid-activated neural network that is then point multipled by the previous cell state, which effectively produces a "gate" that filters the previous cell state. The results are then passed on to the next step. The output of the forget gate can be expressed as $f_t = \sigma(W_f h_{t-1} + U_f X_t + b_f)$, where W_f, U_f are weight matrices and b_f is a bias vector [3, p. 555].

B. The Input Gate

The input gate in the LSTM is responsible for deciding which new information to update the cell state given the new input data and the previous hidden state. The new cell state update can be expressed with the following equations [3, p. 555]:

$$\begin{aligned} C_t &= f_t \otimes C_{t-1} + i_t \bar{C} \\ i_t &= \sigma(W_t h_{t-1} + U_i X_t + b_i) \\ \bar{C} &= tanh(W_c h_{t-1} + U_c X_t + b_c) \end{aligned}$$

The authors are graduate students at NCA&T

W_i, U_i are weight matrices and b_i is a bias vector. C_t is the new cell state, and the above equation represents this new state as a pointwise addition of the previous cell state after running through the forget gate, $f_t C_{t-1}$, and the result of the current input gate, $i_t \bar{C}$. The equation for i_t is the scale factor applied to the candidate that could be added to the internal state, represented by $\bar{C}$.

C. The Output Gate

Once the cell state has been updated, the output gate decides which components of the cell state to keep "internal" to the memory cell. The output gate step can be expressed by the following equations [3, p. 555]:

$$h_t = o_t \otimes tanh(C_t)$$
$$o_t = \sigma(W_o h_{t-1})$$

The current cell state, C_t is passed to a hyperbolic tangent function, which is then pointwise multiplied by the output gate, o_t. The output function is again another sigmoid squashing function that acts as a filter for the output to output the new hidden state.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[2] F. Gers, J. Schmidhuber, and F. Cummins. "Learning to forget: Continual prediction with LSTM." *Neural computation*, vol. 12, no. 10, 2451-2471, 2000.

[3] O Calin. *Deep learning architectures: A mathematical approach*. Switzerland: Springer, 2020. pp. 553-547.