Long Short-Term Memory (LSTM) Networks: A Mathematical Overview

Shirin Amiraslani

August 8, 2024

Abstract

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that are capable of learning long-term dependencies. This document provides an introduction to the key calculations and equations that define LSTM networks, including the gate mechanisms, cell state, and hidden state updates.

1 Introduction

LSTM networks are designed to avoid the long-term dependency problem, which is a common issue in traditional RNNs. The LSTM architecture introduces a memory cell and gating mechanisms that control the flow of information, allowing the network to retain important information over extended time intervals.

2 LSTM Cell Structure

An LSTM cell is composed of several key components:

- **Cell State** (C_t): The memory of the network.
- **Hidden State** (h_t): The output of the LSTM cell.
- **Forget Gate** (f_t): Determines what information to discard from the cell state.
- **Input Gate** (i_t): Decides which new information to store in the cell state.
- **Candidate Memory Cell** ($\tilde{C}_t$): The new memory content to be added.
- **Output Gate** (o_t): Controls the output of the cell.

3 Equations of LSTM

The LSTM cell is governed by the following equations:

3.1 Forget Gate

The forget gate determines which part of the previous cell state C_{t-1} should be forgotten:

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

where W_f and b_f are the weight matrix and bias for the forget gate, respectively, h_{t-1} is the previous hidden state, and x_t is the input at the current time step.

3.2 Input Gate

The input gate controls which parts of the new input x_t should be used to update the cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

where W_i and b_i are the weight matrix and bias for the input gate, respectively.

3.3 Candidate Memory Cell

The candidate memory cell creates a new memory content that could be added to the cell state:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

where W_C and b_C are the weight matrix and bias for the candidate memory cell.

3.4 Cell State Update

The cell state is updated by combining the forget gate, the previous cell state, the input gate, and the candidate memory cell:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4}$$

3.5 Output Gate

The output gate controls the hidden state of the LSTM, which is also the output of the cell:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

where W_o and b_o are the weight matrix and bias for the output gate.

3.6 Hidden State

The hidden state (or the output of the LSTM cell) is calculated as:

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

3.7 Prediction

Finally, prediction is based on the last hidden state calculated:

$$y_t = \sigma(W_y \cdot h_t + b_y) \tag{7}$$

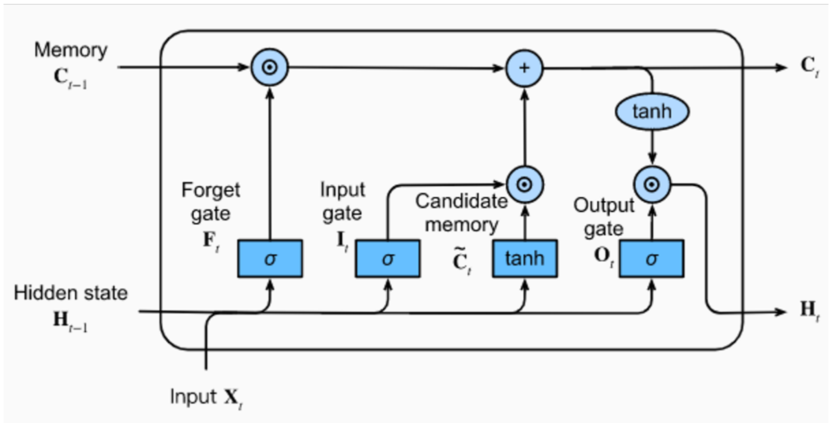

Figure 1: https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c

4 Back Propagation Through Time

Training LSTM networks involves computing gradients of the loss function with respect to the parameters. The gradients are calculated using backpropagation through time (BPTT), which is an extension of the standard backpropagation algorithm for recurrent neural networks.

5 Key LSTM Equations

Once again, let us re-write the key equations governing the forward pass of the LSTM cell:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{8}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{10}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{11}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{12}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{13}$$

6 Loss Function

Let L denote the loss function. The goal of training is to minimize L by updating the parameters W_f, W_i, W_C, W_o, b_f, b_i, b_C, and b_o using gradient descent.

7 Gradients for Backpropagation

The gradients with respect to the loss function are computed as follows:

7.1 Gradient with respect to the Hidden State

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} \cdot \frac{\partial h_{t+1}}{\partial h_t} + \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial h_t} \tag{14}$$

where y_t is the output of the network at time step t.

7.2 Gradients with respect to the Previous Hidden State

$$\begin{aligned} \frac{\partial L}{\partial h_{t-1}} = & \frac{\partial L}{\partial o_t} \cdot o_t(1 - o_t) W_{o,a} \\ & + \frac{\partial L}{\partial i_t} \cdot i_t(1 - i_t) W_{i,a} \\ & + \frac{\partial L}{\partial f_t} \cdot f_t(1 - f_t) W_{f,a} \\ & + \frac{\partial L}{\partial \tilde{C}_t} \cdot (1 - \tanh^2(\tilde{C}_t)) \cdot W_{C,a} \end{aligned} \tag{15}$$

7.3 Gradient with respect to the Output Gate

$$\frac{\partial L}{\partial o_t} = \frac{\partial L}{\partial h_t} \cdot \tanh(C_t) \tag{16}$$

7.4 Gradient with respect to the Cell State

$$\frac{\partial L}{\partial C_t} = \frac{\partial L}{\partial h_t} \cdot o_t \cdot \frac{\partial \tanh(C_t)}{\partial C_t} + \frac{\partial L}{\partial C_{t+1}} \cdot f_{t+1} \tag{17}$$

$$= \frac{\partial L}{\partial h_t} \cdot o_t \cdot (1 - tanh^2(C_t)) + \frac{\partial L}{\partial C_{t+1}} \cdot f_{t+1} \tag{18}$$

7.5 Gradient with respect to the Previous Cell State

$$\frac{\partial L}{\partial C_{t-1}} = \frac{\partial L}{\partial C_t} \cdot f_t \tag{19}$$

7.6 Gradient with respect to the Forget Gate

$$\frac{\partial L}{\partial f_t} = \frac{\partial L}{\partial C_t} \cdot C_{t-1} \tag{20}$$

7.7 Gradient with respect to the Input Gate

$$\frac{\partial L}{\partial i_t} = \frac{\partial L}{\partial C_t} \cdot \tilde{C}_t \tag{21}$$

7.8 Gradient with respect to the Candidate Cell State

$$\frac{\partial L}{\partial \tilde{C}_t} = \frac{\partial L}{\partial C_t} \cdot i_t \tag{22}$$

8 Gradients with respect to Parameters

Finally, the gradients with respect to the LSTM parameters are derived as follows:

8.1 Weights for the Forget Gate

$$\frac{\partial L}{\partial W_f} = \sum_{t=1}^{T} \frac{\partial L}{\partial c_t} \cdot C_{t-1} \cdot f_t \cdot (1 - f_t) \cdot [h_{t-1}, x_t] \tag{23}$$

8.2 Weights for the Input Gate

$$\frac{\partial L}{\partial W_i} = \sum_{t=1}^{T} \frac{\partial L}{\partial c_t} \cdot \tilde{C}_t \cdot (i_t) \cdot (1 - i_t) \cdot [h_{t-1}, x_t] \tag{24}$$

8.3 Weights for the Candidate Cell State

$$\frac{\partial L}{\partial W_C} = \sum_{t=1}^{T} \frac{\partial L}{\partial \tilde{C}_t} \cdot (1 - tanh^2(\tilde{C}_t)) \cdot [h_{t-1}, x_t] \tag{25}$$

8.4 Weights for the Output Gate

$$\frac{\partial L}{\partial W_o} = \sum_{t=1}^{T} \frac{\partial L}{\partial o_t} \cdot o_t \cdot (1 - o_t) \cdot [h_{t-1}, x_t] \tag{26}$$

9 Bias Gradients

The gradients with respect to the biases b_f, b_i, b_C, and b_o are derived similarly by summing the respective gradients over all time steps:

$$\frac{\partial L}{\partial b_f} = \sum_{t=1}^{T} \frac{\partial L}{\partial f_t} \cdot f_t \cdot (1 - f_t) \tag{27}$$

$$\frac{\partial L}{\partial b_i} = \sum_{t=1}^{T} \frac{\partial L}{\partial i_t} \cdot i_t \cdot (1 - i_t) \tag{28}$$

$$\frac{\partial L}{\partial b_C} = \sum_{t=1}^{T} \frac{\partial L}{\partial \tilde{C_t}} \cdot (1 - tanh^2(\tilde{C_t})) \tag{29}$$

$$\frac{\partial L}{\partial b_o} = \sum_{t=1}^{T} \frac{\partial L}{\partial o_t} \cdot o_t \cdot (1 - o_t) \tag{30}$$

10 Conclusion

The LSTM architecture effectively mitigates the vanishing gradient problem encountered in standard RNNs, making it well-suited for tasks involving long-term dependencies. The gating mechanisms provide control over the flow of information, enabling the network to maintain relevant information over extended sequences. The gradients derived in this document are essential for updating the LSTM parameters during training. By computing these gradients, we can apply gradient descent (or its variants) to minimize the loss function and train the LSTM network effectively.