Forward and Backpropagation in an LSTM

Jaime Sancho Molero

December 6, 2022

Contents

1 Review of LSTM structure

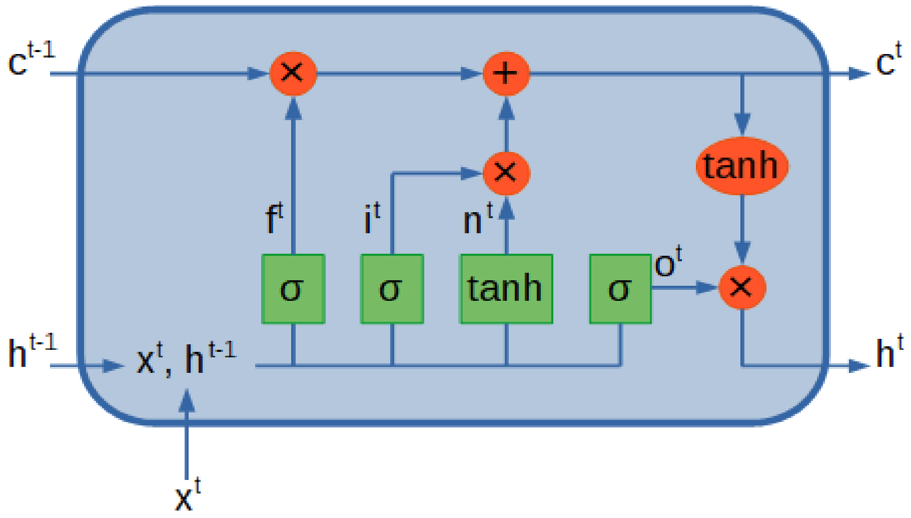

Figure 1: LSTM at time step t

In the figure 1 we can see the diagram of the LSTM block.

Instead of a unit that simply applies an elementwise nonlinearty to the affine transformation of inputs and recurrent units, LSTM recurrent networks have "LSTM cells" that have an internal recurrence (a self-loop) in addition to the outer recurrence of the RNN. Each cell has the same inputs and outputs as an ordinary recurrent neural network, but has more parameters and a system of gating units that control the flow of information. The latter affirmation is very complicated to understand. The fact that it works is practically a "mystery" for most of the Deep Learning practicioners. The important thing is that appliying an optimizer algorithm such as Gradient Descent, the recurrent neural network itself will decide which information is important and which is not.

2 Forward Propagation

Notation. Although in the figure 1 the candidate cell (n^t) is refered with an n, i will refer to it with a g.

2.1 Gates equations

Let's start with the gates equations. Looking at the figure 1 is pretty straightforward to derive the equations for forward propation.

- **Forget gate**

$$f_i^{(t)} = \sigma(\sum_j W_{ij}^f x_j^{(t)} + \sum_j R_{ij}^f h_j^{(t-1)} + b_i^f) \tag{1}$$

- **Input gate**

$$i_i^{(t)} = \sigma(\sum_j W_{ij}^i x_j^{(t)} + \sum_j R_{ij}^i h_j^{(t-1)} + b_i^i) \tag{2}$$

- **Output gate**

$$o_i^{(t)} = \sigma(\sum_j W_{ij}^o x_j^{(t)} + \sum_j R_{ij}^o h_j^{(t-1)} + b_i^o) \tag{3}$$

- **Candidate gate**

$$g_i^{(t)} = \tanh(\sum_j W_{ij}^g x_j^{(t)} + \sum_j R_{ij}^g h_j^{(t-1)} + b_i^g) \tag{4}$$

2.2 Hidden, Cell and Prediction equations

Once we have compute the gate equations, we can get the rest of the needed equations.

- **Cell State**

$$c_i^{(t)} = g_i^{(t)} i_i^{(t)} + c_i^{(t-1)} f_i^{(t)} \tag{5}$$

- **Hidden State**

$$h_i^{(t)} = o_i^{(t)} \tanh(c_i^{(t)}) \tag{6}$$

- **Prediction**

$$y_i^{(t)} = F\big(\sum_i W_{ij}^y h_i^{(t)} + b_i^y\big) \tag{7}$$

We have everything we need to derive the backward propagation equations.

3 Backward propagation

This is not an explanation of how backpropagation works. It is only the derivation of the equations in case you want to know how to get them or want to implement an LSTM from scratch in some programming language. The latter is the reason why i derived them myself since i have not encountered any paper with a good derivation.

The objective of backpropagation is to obtain the relations:

$$\delta W_\theta, \delta R_\theta, \delta b_\theta \qquad\qquad \theta = \mathbf{f}, \mathbf{g}, \mathbf{i}, \mathbf{o}$$

3.1 $\delta \mathbf{y^{(t)}}$

$$\delta y_k^{(t)} \equiv \frac{\partial \mathcal{L}}{\partial y_k^{(t)}} = \frac{\partial G}{\partial y_k^{(t)}}$$

for some function G.

3.2 $\delta \mathbf{h^{(t)}}$

Looking at the figure 1 we can see that the derivative of the loss respect the hidden state at time step t will have more than one term because changing $\mathbf{h^{(t)}}$ also affect the gates of $t{+}1$.

$$\delta h_k^{(t)} \equiv \frac{\partial \mathcal{L}}{\partial h_k^{(t)}} = \frac{\partial \mathcal{L}}{\partial h_k^{(t)}} + \sum_j \left(\sum_\theta \frac{\partial \mathcal{L}}{\partial \theta_j^{(t+1)}} \frac{\partial \theta_j^{(t+1)}}{\partial h_k^{(t)}} \right) \tag{8}$$

for $\theta = f, g, i, o$.

Even though we do not yet have these quantities at time step $t+1$ (or t), we will get them later in this section, so do not worry about that. The important thing to notice is that when we change the hidden state at time step t, this change will affect the gates at time step $t+1$ given the relation at equations (1), (2), (3), (4).

Since all gates has the same structure, we can derive the second term in equation (8) in a general form.

$$\frac{\partial \theta_j^{(t+1)}}{\partial h_k^{(t)}} = \frac{df_\theta}{dx}|_{x^{(t+1)}} \frac{\partial}{\partial h_k^{(t)}} \big(\sum_\alpha R_{j\alpha}^\theta h_\alpha^{(t)} \big) = \frac{df_\theta}{dx}|_{x^{(t+1)}} R_{jk}^\theta \tag{9}$$

The term $df/d\theta$ is evaluated at the argument of the function θ at time step $t+1$. This function f_θ will be the sigmoid function for $\theta = f, o, i$ and the hyperbolic tangent function for $\theta = g$. So, to obtain $df/d\theta$ you would have to derive the corresponding function and evaluating it with the argument of the function itself at time step $t+1$. Introducing this last equation in equation 8 we get

$$\delta h_k^{(t)} = \frac{\partial \mathcal{L}}{\partial h_k^{(t)}} + \sum_j \left(\sum_\theta \frac{\partial \mathcal{L}}{\partial \theta_j^{(t+1)}} \frac{df_\theta}{dx}|_{x^{(t+1)}} R_{jk}^\theta \right) \tag{10}$$

Let's derive the first term of the equation (8.)

$$\frac{\partial \mathcal{L}}{\partial h_k^{(t)}} = \sum_j \frac{\partial \mathcal{L}}{\partial y_j^{(t)}} \frac{\partial y_j^{(t)}}{\partial h_k^{(t)}} \tag{11}$$

Using equation (7)

$$\frac{\partial \mathcal{L}}{\partial h_k^{(t)}} = \sum_j \frac{\partial \mathcal{L}}{\partial y_j^{(t)}} \frac{\partial}{\partial h_k^{(t)}} F\left(\sum_i W_{ji}^y h_i^{(t)} \right) = \sum_j \delta y_j^{(t)} W_{jk} \frac{dF_j}{dx}|_{x^{(t)}} \tag{12}$$

Putting everything together

$$\delta h_k^{(t)} = \sum_j \delta y_j^{(t)} W_{jk} \frac{dF_j}{dx}|_{x^{(t)}} + \sum_\theta \left(\sum_j \frac{\partial \mathcal{L}}{\partial \theta_j^{(t+1)}} \frac{df_\theta}{dx}|_{x^{(t+1)}} R_{jk}^\theta \right) \tag{13}$$

We can put this equation in matrix form as follows

$$\delta \mathbf{h}^{(\mathbf{t})} = (\mathbf{W}^y)^T \left(\delta \mathbf{y} \odot \frac{df_y}{dx}|_{x^{(t)}} \right) + \sum_{\theta=\mathbf{f},\mathbf{g},\mathbf{i},\mathbf{o}} (\mathbf{R}^\theta)^T \left(\delta\theta^{(t+1)} \odot \frac{df_\theta}{dx}|_{x^{(t+1)}} \right) \tag{14}$$

The functions df_y/dx and df_θ/dx have to be evaluated at, respectively:

$$x^{(t)} \equiv (W^y)h^{(t)} + b_y$$
$$x^{(t+1)} \equiv (W^\theta)x^{(t+1)} + R^\theta h^{(t)} + b^\theta$$

where $x^{(t+1)}$ is the input at time step $t+1$ and $h^{(t)}$ is the hidden state at time step t.

3.3 $\delta \mathbf{c}^{(\mathbf{t})}$

Same case as before. When we change $c^{(t)}$ we will affect the cost at time step t and the cost at time step $t+1$, because $c^{(t+1)}$ depends on $c^{(t)}$.

$$\delta c_k^{(t)} \equiv \frac{\partial \mathcal{L}}{\partial c_k^{(t)}} = \frac{\partial \mathcal{L}}{\partial h_k^{(t)}} \frac{\partial h_k^{(t)}}{\partial c_k^{(t)}} + \frac{\partial \mathcal{L}}{\partial c_k^{(t+1)}} \frac{\partial c_k^{(t+1)}}{\partial c_k^{(t)}} \tag{15}$$

In matrix form

$$\delta \mathbf{c}^{(\mathbf{t})} = \delta \mathbf{h}^{(\mathbf{t})} \odot \mathbf{o}^{(\mathbf{t})} \odot \frac{d \tanh}{dx}(\mathbf{c}^{(\mathbf{t})}) + \delta \mathbf{c}^{(\mathbf{t+1})} \odot \mathbf{f}^{(\mathbf{t+1})} \tag{16}$$

3.4 $\delta\theta^{(t)}$

This is the last step before being able to compute the gradients of the parameters of interest.

- Output gate

$$\delta o_k^{(t)} = \frac{\partial \mathcal{L}}{\partial h_k^{(t)}} \frac{\partial h_k^{(t)}}{\partial o_k^{(t)}} = \delta h_k^{(t)} \tanh(c_k^{(t)})$$

  In matrix form

$$\delta \mathbf{o}^{(\mathbf{t})} = \delta \mathbf{h}^{(\mathbf{t})} \odot \tanh(\mathbf{c}^{(\mathbf{t})}) \tag{17}$$

- Forget gate

$$\delta f_k^{(t)} = \delta c_k^{(t)} c_k^{(t-1)}$$

  In matrix form

$$\delta \mathbf{f}^{(\mathbf{t})} = \delta \mathbf{c}^{(\mathbf{t})} \odot \mathbf{c}^{(\mathbf{t-1})} \tag{18}$$

- Input gate
  Following the same process

$$\delta \mathbf{i}^{(\mathbf{t})} = \delta \mathbf{c}^{(\mathbf{t})} \odot \mathbf{g}^{(\mathbf{t})} \tag{19}$$

- Candidate gate

$$\delta \mathbf{g}^{(\mathbf{t})} = \delta \mathbf{c}^{(\mathbf{t})} \odot \mathbf{i}^{(\mathbf{t})} \tag{20}$$

3.5 $\delta \mathbf{W}^\theta, \delta \mathbf{R}^\theta, \delta \mathbf{b}^\theta$

Now we have everything we need to compute these gradients. Once we are finished we will be able to use these equations to apply an optimizer algorithm.

Every change of these parameters at any time step will affect the loss. That is why we have to sum over all time steps in order to compute these gradients. Let's start with the weights and bias of the prediction: $\mathbf{W}^y, \mathbf{b}^y$.

3.5.1 $\delta \mathbf{W}^y, \delta \mathbf{b}^y$

$$\delta W_{jk}^y \equiv \frac{\partial \mathcal{L}}{\partial W_{jk}^y} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial y_i^{(t)}} \frac{\partial y_i^{(t)}}{\partial W_{jk}^y} = \sum_t \sum_i \delta y_i^{(t)} \frac{\partial}{\partial W_{jk}^y} \big(\sum_\alpha W_{i\alpha} h_\alpha^{(t)} \big) \frac{df_y}{dx}|_{x^{(t)}}$$

$$\delta W_{jk}^y = \sum_t \delta y_j^{(t)} h_k^{(t)} \frac{df_y}{dx}|_{x^{(t)}}$$

We can put this last equation in matrix form

$$\delta \mathbf{W}^y = \sum_t \big(\delta \mathbf{y}^{(t)} \odot \frac{df_y}{dx}|_{x^{(t)}} \big) (\mathbf{h}^{(t)})^T \tag{21}$$

Repeting the same procedure for $\delta \mathbf{b}^y$:

$$\delta \mathbf{b}^{\mathbf{y}} = \sum_t \big(\delta \mathbf{y}^{(t)} \odot \frac{df_y}{dx}|_{x^{(t)}} \big) \tag{22}$$

3.5.2 $\delta \mathbf{W}^g, \delta \mathbf{R}^g, \delta \mathbf{b}^g$

$$\delta W_{jk}^g \equiv \frac{\partial \mathcal{L}}{\partial W_{jk}^g} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial g_i^{(t)}} \frac{\partial g_i^{(t)}}{\partial W_{jk}^g} = \sum_t \delta g_i^{(t)} \frac{d \tanh}{dx} (x_i^{(t)}) x_k^{(t)}$$

$$\delta R_{jk}^g \equiv \frac{\partial \mathcal{L}}{\partial R_{jk}^g} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial g_i^{(t)}} \frac{\partial g_i^{(t)}}{\partial R_{jk}^g} = \sum_t \delta g_i^{(t)} \frac{d \tanh}{dx} (x_i^{(t)}) h_k^{(t-1)}$$

$$\delta b_j^g \equiv \frac{\partial \mathcal{L}}{\partial b_j^g} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial g_i^{(t)}} \frac{\partial g_i^{(t)}}{\partial b_j^g} = \sum_t \delta g_i^{(t)} \frac{d \tanh}{dx} (x_i^{(t)})$$

where $x_i^{(t)}$ is the input of the gate function at time step t.

3.5.3 Rest of gradients

Since the rest of the gates has the same structure, we can do it in terms of an arbitrary parameter θ.

$$\delta W^\theta_{jk} \equiv \frac{\partial \mathcal{L}}{\partial W^\theta_{jk}} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial \theta_i^{(t)}} \frac{\partial \theta_i^{(t)}}{\partial W^g_{jk}} = \sum_t \delta\theta_i^{(t)} \frac{d\sigma}{dx}(x_i^{(t)}) x_k^{(t)}$$

$$\delta R^\theta_{jk} \equiv \frac{\partial \mathcal{L}}{\partial R^\theta_{jk}} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial \theta_i^{(t)}} \frac{\partial \theta_i^{(t)}}{\partial R^\theta_{jk}} = \sum_t \delta\theta_i^{(t)} \frac{d\sigma}{dx}(x_i^{(t)}) h_k^{(t-1)}$$

$$\delta b^\theta_j \equiv \frac{\partial \mathcal{L}}{\partial b^\theta_j} = \sum_t \sum_i \frac{\partial \mathcal{L}}{\partial g_i^{(t)}} \frac{\partial g_i^{(t)}}{\partial b^\theta_j} = \sum_t \delta\theta_i^{(t)} \frac{d\sigma}{dx}(x_i^{(t)})$$

In matrix form

$$\delta \mathbf{W}^\theta = \sum_t \big(\delta\theta^{(t)} \odot \frac{df_\theta}{dx}(x^{(t)})\big)(\mathbf{x}^{(t)})^T \tag{23}$$

$$\delta \mathbf{R}^\theta = \sum_t \big(\delta\theta^{(t)} \odot \frac{df_\theta}{dx}(x^{(t)})\big)(\mathbf{h}^{(t-1)})^T \tag{24}$$

$$\delta \mathbf{b}^\theta = \sum_t \big(\delta\theta^{(t)} \odot \frac{df_\theta}{dx}(x^{(t)})\big) \tag{25}$$

4 Review

In this section we gather all the necessary equations to apply backpropagation.

$$\delta\mathbf{h^{(t)}} = (\mathbf{W}^y)^T\left(\delta\mathbf{y} \odot \frac{df_y}{dx}|_{x^{(t)}}\right) + \sum_{\theta=\mathbf{f},\mathbf{g},\mathbf{i},\mathbf{o}} (\mathbf{R}^\theta)^T\left(\delta\theta^{(t+1)} \odot \frac{df_\theta}{dx}|_{x^{(t+1)}}\right)$$

$$\delta\mathbf{c^{(t)}} = \delta\mathbf{h^{(t)}} \odot \mathbf{o^{(t)}} \odot \tanh\prime(\mathbf{c^{(t)}}) + \delta\mathbf{c^{(t+1)}} \odot \mathbf{f^{(t+1)}}$$

$$\delta\mathbf{o^{(t)}} = \delta\mathbf{h^{(t)}} \odot \tanh(\mathbf{c^{(t)}})$$

$$\delta\mathbf{f^{(t)}} = \delta\mathbf{c^{(t)}} \odot \mathbf{c^{(t-1)}}$$

$$\delta\mathbf{i^{(t)}} = \delta\mathbf{c^{(t)}} \odot \mathbf{g^{(t)}}$$

$$\delta\mathbf{g^{(t)}} = \delta\mathbf{c^{(t)}} \odot \mathbf{i^{(t)}}$$

$$\delta\mathbf{W}^y = \sum_t \left(\delta\mathbf{y}^{(t)} \odot \frac{df_y}{dx}|_{x^{(t)}}\right)(\mathbf{h}^{(t)})^T$$

$$\delta\mathbf{b^y} = \sum_t \left(\delta\mathbf{y}^{(t)} \odot \frac{df_y}{dx}|_{x^{(t)}}\right)$$

$$\delta\mathbf{W}^\theta = \sum_t \left(\delta\theta^{(t)} \odot \frac{df_\theta}{dx}(x^{(t)})\right)(\mathbf{x}^{(t)})^T$$

$$\delta\mathbf{R}^\theta = \sum_t \left(\delta\theta^{(t)} \odot \frac{df_\theta}{dx}(x^{(t)})\right)(\mathbf{h}^{(t-1)})^T$$

$$\delta\mathbf{b}^\theta = \sum_t \left(\delta\theta^{(t)} \odot \frac{df_\theta}{dx}(x^{(t)})\right)$$

4.1 Final Comments

If $t+1 >$ "length of sequence", then, any term in that time step will be zero. The same thing ocurr if any parameter/function is evaluated at time t < 0.

Another thing to have in mind is that in practice the matrices R are not usually used. The common practice is to stack the input vector at time step t: $x^{(t)}$ and the hidden state at time step $t-1$: $h^{(t-1)}$ into a single column vector: $[h^{(t-1)}, x^{(t)}]$ and applying backpropagation with this notation.

You can watch my own implementation of an LSTM in my github page:
https://github.com/jimysancho, where i apply it to some problems in order to see its performance.