Supplementary information RNN

Idriss Gouigah

December, 2024

1 The LSTM unit

An LSTM unit is composed of a cell and three gates : an input gate, an output gate, and a forget gate. The computation made by the unit are as follows :

- $f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$ forget gate computation.
- $i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_o)$ input gate computation.
- $o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$ output gate computation.
- $c'_t = \sigma_c(W_{c'} x_t + U_{c'} h_{t-1} + b_{c'})$ cell intermediate value.
- $c_t = f_t \odot c_{t-1} + i_t \odot c'_t$ update of cell value.
- $h_t = o_t \odot \sigma_h(c_t)$ update of hidden state value. We will assume that σ_h is the identity so we can rewrite : $h_t = o_t \odot c_t$

Where :

- x_t the input at time t, of dimension d.
- h_t the hidden state at time t, of dimension h.
- c_t the cell state at time t, of dimension h.
- For each gate, $W_{gate} \in \mathbb{R}^{h \times d}$, $U_{gate} \in \mathbb{R}^{h \times h}$ and $b_{gate} \in \mathbb{R}^h$ are the gate's parameters.
- σ_g, σ_c are respectively the sigmoid and tanh activation function.
- $\odot$ is the element-wise product.

The LSTM unit is initialized with a hidden state and a cell state equal to zero. When input with a sequence of data, the unit goes through the data in sequential order and updates its hidden state and cell to reflect the time dependency. The output is also a sequence which has the same length as the input sequence, representing the hidden state at each time step.

2 Backpropagation through time (BPTT)

The model we are about to implement introduces a dependency through time. Since we are dealing with sequential data, the previous input of the sequence will influence the current processed input. This means that the same input could produce a different output, depending on the previous inputs. This is important to keep in mind because it means that backpropagation will not only consider the current gradient, but also gradients from previous computation. We are thus going to perform a BackPropagation Through Time (BPTT).

As a consequence of the dependence through time and the chain rule, gradients need to be accumulated over timesteps. Suppose we have a sequence of T timesteps and want to backpropagate the gradient through them. The total loss of the sequence is :

$$\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}_t$$

So the total gradient of the loss with respect to a parameter W will be :

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial W}$$

For recurrent models like LSTMs, parameters W influence the loss $\mathcal{L}_t$ not just directly at timestep t, but also indirectly through their effect on earlier timesteps. This happens because the hidden state h_t, which depends on W, is passed forward through time and influences future states. So it is not straightforward to compute $\frac{\partial \mathcal{L}_t}{\partial W}$. However, using the formulas that we will develop bellow thanks to the chain rule, it will become simple.

Formula of the gradient with respect to the hidden state. The gradient of the loss with respect to the hidden state h_t at time t involves two terms :

- The direct contribution from the current time step:

$$\frac{\partial \mathcal{L}_t}{\partial h_t}$$

- The indirect contribution of future timesteps $t+1, t+2 \dots T$:

$$\sum_{k=t+1}^{T} \frac{\partial \mathcal{L}_k}{\partial h_k} \frac{\partial h_k}{\partial h_t}$$

So if we write the total gradient of the loss with respect to h_t as δ_{h_t}, we have :

$$\frac{\partial \mathcal{L}}{\partial h_t} = \delta_{h_t} = \frac{\partial \mathcal{L}_t}{\partial h_t} + \sum_{k=t+1}^{T} \frac{\partial \mathcal{L}_k}{\partial h_k} \frac{\partial h_k}{\partial h_t} \tag{1}$$

Which we can rewrite :

$$\delta_{h_t} = \frac{\partial \mathcal{L}_t}{\partial h_t} + \delta_{t+1} \cdot \frac{\partial h_{t+1}}{\partial h_t}$$

Formula of the gradient with respect to the cell. In the computation of the hidden state, the cell c_t contributes in two places :

- Through the computation of $h_t = o_t \odot c_t$.
- Through future timestep via c_{t+1}.

Thus there is a similar formula involving δ_{h_t} for $\frac{\partial \mathcal{L}}{\partial c_t} = \delta_{c_t}$:

$$\frac{\partial \mathcal{L}}{\partial c_t} = \delta_{c_t} = \delta_{h_t} \cdot \frac{\partial h_t}{\partial c_t} + \delta_{c_{t+1}} \frac{\partial c_{t+1}}{\partial c_t} \tag{2}$$

Formula of the gradient with respect to the input. For the LSTM unit, the input x_t at time t contributes in a few places :

- The direct contribution via the activation of the gates f_t, i_t, c'_t, o_t at time t.
- The indirect contribution on future states of the LSTM via the recurrent connexion.

This time however, the indirect contribution will fully be expressed in δ_{h_t} and δ_{c_t} as all operations performed on h_t and c_t imply x_t.

In other words, for each gate we can directly compute $\frac{\partial \mathcal{L}}{\partial gate_t} = \delta_{gate_t}$ and then compute (chain rule):

$$\frac{\partial \mathcal{L}}{\partial x_t} = \delta_{x_t} = \sum_{gate} \frac{\partial \mathcal{L}}{\partial gate_t} \cdot \frac{\partial gate_t}{\partial x_t} = \sum_{gate} \delta_{gate_t} \cdot \frac{\partial gate_t}{\partial x_t} \tag{3}$$

Algorithm for BPTT We now have all the formulas needed for BPTT, let's take a look at the algorithm.

- The input is the values $\frac{\partial \mathcal{L}_t}{\partial h_t}$ for $t = 1 \dots T$.
- We start from timestep T and go backward through time.
- We compute δ_{h_T} using the formula (1) with $\delta_{h_{T+1}} = 0$.
- We compute δ_{c_T} using the formula (2) with $\delta_{c_{T+1}} = 0$ and the formula from LSTM computation.
- We can now compute δ_{gate_t} for each gate using the formulas for LSTM computation.
- We can now go backward through each gate and compute δ_{x_t} as well as $\frac{\partial h_{T+1}}{\partial h_T}$ and $\frac{\partial c_{T+1}}{\partial c_T}$. Again, this uses the formulas of LSTM computation and the chain rule.
- During the backward of each step we can also compute the gradient with respect to the parameters for the current time step. The gradients will be summed over the time steps. This uses the chain rule and the formulas for LSTM computation.
- We now have everything we need from this time step and can go to the previous time step.