Sit down, Shakespeare!

replacing the old bard with recurrent neural networks

Summary

...

Deep Learning in Data Science
DD2424
Spring 2023

Single-layer LSTM Backpropagation

Forward pass

The forward pass is defined as follows,

$$
\begin{aligned}
\boldsymbol{i}_t &= \sigma(W_i \boldsymbol{h}_{t-1} + U_i \boldsymbol{x}_t) \\
\boldsymbol{f}_t &= \sigma(W_f \boldsymbol{h}_{t-1} + U_f \boldsymbol{x}_t) \\
\boldsymbol{e}_t &= \sigma(W_e \boldsymbol{h}_{t-1} + U_e \boldsymbol{x}_t) \\
\tilde{\boldsymbol{c}}_t &= \tanh(W_c \boldsymbol{h}_{t-1} + U_c \boldsymbol{x}_t) \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \bullet \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \bullet \tilde{\boldsymbol{c}}_t \\
\boldsymbol{h}_t &= \boldsymbol{e}_t \bullet \tanh(\boldsymbol{c}_t) \\
\boldsymbol{o}_t &= V \boldsymbol{h}_t \\
\boldsymbol{p}_t &= \text{softmax}(\boldsymbol{o}_t)
\end{aligned}
$$

where $\bullet$ denotes element by element multiplication and $\boldsymbol{e}_t$ denotes the output/exposure gate.

Backward pass

In order to find the analytical gradients and compute the backward pass, we employ the chain rule. First, we consider the gradient of the cross-entropy loss w.r.t. the output for the final time step. We define

$$\frac{\partial L}{\partial \boldsymbol{o}_t} = -(\boldsymbol{y}_t - \boldsymbol{p}_t)^T, \quad \forall\, t = 1, 2, \ldots, T$$

and denote $\boldsymbol{g}_t := \frac{\partial L}{\partial \boldsymbol{o}_t}$. Then, in order to compute the gradients of the individual weights, we need to first find define the gradients with respect to the hidden activation and the memory cell as these have to be computed through time. Hence, we first consider the partial derivative of the loss with respect to the hidden units and the memory cell, i.e. h_t and c_t for $t = 1, 2, \ldots, T$, and consider two cases: $t = T$ and $t = 1, 2, \ldots, T-1$.

I - the case of $t = T$

For this case, the gradient computation is straightforward by employing the chain rule. Specifically, for the hidden unit we have that

$$\frac{\partial L}{\partial \boldsymbol{h}_t} = \frac{\partial L}{\partial \boldsymbol{o}_t} \frac{\partial \boldsymbol{o}_t}{\partial \boldsymbol{h}_t} = \boldsymbol{g}_t V$$

and for the memory cell we simply get

$$\frac{\partial L}{\partial \boldsymbol{c}_t} = \frac{\partial L}{\partial \boldsymbol{o}_t} \frac{\partial \boldsymbol{o}_t}{\partial \boldsymbol{h}_t} \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_t} = \boldsymbol{g}_t V \boldsymbol{e}_t (1 - \tanh^2(\boldsymbol{c}_t))$$

II - the case of $t < T$

In order to perform the backward pass for earlier time steps in the sequence, we need to consider how the information propagates forward, i.e. the hidden unit at $t-1$ passes through the hidden unit at t. Similarly, the memory cell at $t-1$ passes through the memory cell at t. For notational convenience, we define the activations for the gates as $\boldsymbol{a}_{\cdot,t}$, i.e. such that e.g.

$$\boldsymbol{i}_t := \sigma(\boldsymbol{a}_{i,t})$$

Hence, for the hidden units and the memory cell, we have that

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{h}_t} &= \frac{\partial L}{\partial \boldsymbol{o}_t}\frac{\partial \boldsymbol{o}_t}{\partial \boldsymbol{h}_t} + \frac{\partial L}{\partial \boldsymbol{o}_{t+1}}\frac{\partial \boldsymbol{o}_{t+1}}{\partial \boldsymbol{h}_t} \\
&= \boldsymbol{g}_t V + \frac{\partial L}{\partial \boldsymbol{i}_{t+1}}\frac{\partial \boldsymbol{i}_{t+1}}{\partial \boldsymbol{h}_t} + \frac{\partial L}{\partial \boldsymbol{f}_{t+1}}\frac{\partial \boldsymbol{f}_{t+1}}{\partial \boldsymbol{h}_t} + \frac{\partial L}{\partial \boldsymbol{e}_{t+1}}\frac{\partial \boldsymbol{e}_{t+1}}{\partial \boldsymbol{h}_t} + \frac{\partial L}{\partial \tilde{\boldsymbol{c}}_{t+1}}\frac{\partial \tilde{\boldsymbol{c}}_{t+1}}{\partial \boldsymbol{h}_t} \\
&= \boldsymbol{g}_t V + \frac{\partial L}{\partial \boldsymbol{i}_{t+1}}\frac{\partial \boldsymbol{i}_{t+1}}{\partial \boldsymbol{a}_{i,t+1}} W_i + \frac{\partial L}{\partial \boldsymbol{f}_{t+1}}\frac{\partial \boldsymbol{f}_{t+1}}{\partial \boldsymbol{a}_{f,t+1}} W_f + \frac{\partial L}{\partial \boldsymbol{e}_{t+1}}\frac{\partial \boldsymbol{e}_{t+1}}{\partial \boldsymbol{a}_{et+1}} W_e + \frac{\partial L}{\partial \tilde{\boldsymbol{c}}_{t+1}}\frac{\partial \tilde{\boldsymbol{c}}_{t+1}}{\partial \boldsymbol{a}_{c,t+1}} W_c
\end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{c}_t} &= \frac{\partial L}{\partial \boldsymbol{o}_t}\frac{\partial \boldsymbol{o}_t}{\partial \boldsymbol{h}_t}\frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_t} + \frac{\partial L}{\partial \boldsymbol{o}_{t+1}}\frac{\partial \boldsymbol{o}_{t+1}}{\partial \boldsymbol{h}_{t+1}}\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{c}_{t+1}}\frac{\partial \boldsymbol{c}_{t+1}}{\partial \boldsymbol{c}_t} \\
&= \boldsymbol{g}_t V \boldsymbol{e}_t (1 - \tanh^2(\boldsymbol{c}_t)) + \frac{\partial L}{\partial \boldsymbol{c}_{t+1}}\frac{\partial \boldsymbol{c}_{t+1}}{\partial \boldsymbol{c}_t} \\
&= \boldsymbol{g}_t V \boldsymbol{e}_t (1 - \tanh^2(\boldsymbol{c}_t)) + \frac{\partial L}{\partial \boldsymbol{c}_{t+1}} \boldsymbol{f}_{t+1}
\end{aligned}$$

Then, in order to compute the gradients all the gradients for the hidden units and the memory cell, we need to

(1) calculate gradients for $\boldsymbol{i}_{t+1}$, $\boldsymbol{f}_{t+1}$, $\boldsymbol{e}_{t+1}$, and $\tilde{\boldsymbol{c}}_{t+1}$,

(2) calculate the gradient for $\boldsymbol{h}_t$ using $\boldsymbol{h}_{t+1}$ and (1), and

(3) calculate the gradient for $\boldsymbol{c}_t$ using (2) and $\boldsymbol{c}_{t+1}$.

(4) calculate the gradients for $\boldsymbol{i}_t$, $\boldsymbol{f}_t$, $\boldsymbol{e}_t$, and $\tilde{\boldsymbol{c}}_t$ using (2) and (3).

III - gradients for gates and memory cell

Further, for notational convenience, we define

$$\tilde{\boldsymbol{g}}_t := \frac{\partial L}{\partial \boldsymbol{h}_t}, \quad \text{and} \quad \hat{\boldsymbol{g}}_t := \frac{\partial L}{\partial \boldsymbol{c}_t}$$

such that we can expand the terms in the above equation per the follwing:

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{i}_t}\frac{\partial \boldsymbol{i}_t}{\partial \boldsymbol{h}_{t-1}} &= \hat{\boldsymbol{g}}_t\, \tilde{\boldsymbol{c}}_t \sigma(\boldsymbol{a}_{i,t})(1 - \sigma(\boldsymbol{a}_{i,t}))W_i = \hat{\boldsymbol{g}}_t\, \tilde{\boldsymbol{c}}_t \boldsymbol{i}_t (1 - \boldsymbol{i}_t) W_i \\
\frac{\partial L}{\partial \boldsymbol{f}_t}\frac{\partial \boldsymbol{f}_t}{\partial \boldsymbol{h}_{t-1}} &= \hat{\boldsymbol{g}}_t\, \boldsymbol{c}_{t-1} \sigma(\boldsymbol{a}_{f,t})(1 - \sigma(\boldsymbol{a}_{f,t}))W_f = \hat{\boldsymbol{g}}_t\, \boldsymbol{c}_{t-1} \boldsymbol{f}_t (1 - \boldsymbol{f}_t) W_f \\
\frac{\partial L}{\partial \boldsymbol{e}_t}\frac{\partial \boldsymbol{e}_t}{\partial \boldsymbol{h}_{t-1}} &= \tilde{\boldsymbol{g}}_t\, \tanh(\boldsymbol{c}_t) \sigma(\boldsymbol{a}_{e,t})(1 - \sigma(\boldsymbol{a}_{e,t}))W_e = \tilde{\boldsymbol{g}}_t\, \tanh(\boldsymbol{c}_t) \boldsymbol{e}_t (1 - \boldsymbol{e}_t) W_e \\
\frac{\partial L}{\partial \tilde{\boldsymbol{c}}_t}\frac{\partial \tilde{\boldsymbol{c}}_t}{\partial \boldsymbol{h}_{t-1}} &= \hat{\boldsymbol{g}}_t\, \boldsymbol{i}_t (1 - \tanh^2(\boldsymbol{a}_{c,t}))W_c = \hat{\boldsymbol{g}}_t\, \boldsymbol{i}_t (1 - \tilde{\boldsymbol{c}}_t^2) W_c
\end{aligned}$$

V - putting it all together

Finally, given that we have iteratively computed all the gradients for the hidden units, the gates, the memory cell and their respective activations, we have that

$$\frac{\partial L}{\partial W_i} = \sum_{t=1}^{T} \hat{\boldsymbol{g}}_t \, \tilde{\boldsymbol{c}}_t \boldsymbol{i}_t (1 - \boldsymbol{i}_t) \boldsymbol{h}_{t-1} \tag{1}$$

$$\frac{\partial L}{\partial U_i} = \sum_{t=1}^{T} \hat{\boldsymbol{g}}_t \, \tilde{\boldsymbol{c}}_t \boldsymbol{i}_t (1 - \boldsymbol{i}_t) \boldsymbol{x}_t \tag{2}$$

$$\frac{\partial L}{\partial W_f} = \sum_{t=1}^{T} \hat{\boldsymbol{g}}_t \, \boldsymbol{c}_{t-1} \boldsymbol{f}_t (1 - \boldsymbol{f}_t) \boldsymbol{h}_{t-1} \tag{3}$$

$$\frac{\partial L}{\partial U_f} = \sum_{t=1}^{T} \hat{\boldsymbol{g}}_t \, \boldsymbol{c}_{t-1} \boldsymbol{f}_t (1 - \boldsymbol{f}_t) \boldsymbol{x}_t \tag{4}$$

$$\frac{\partial L}{\partial W_e} = \sum_{t=1}^{T} \tilde{\boldsymbol{g}}_t \, \tanh(\boldsymbol{c}_t) \boldsymbol{e}_t (1 - \boldsymbol{e}_t) \boldsymbol{h}_{t-1} \tag{5}$$

$$\frac{\partial L}{\partial U_e} = \sum_{t=1}^{T} \tilde{\boldsymbol{g}}_t \, \tanh(\boldsymbol{c}_t) \boldsymbol{e}_t (1 - \boldsymbol{e}_t) \boldsymbol{x}_t \tag{6}$$

$$\frac{\partial L}{\partial W_c} = \sum_{t=1}^{T} \hat{\boldsymbol{g}}_t \, \boldsymbol{i}_t (1 - \tilde{\boldsymbol{c}}_t^2) \boldsymbol{h}_{t-1} \tag{7}$$

$$\frac{\partial L}{\partial U_c} = \sum_{t=1}^{T} \hat{\boldsymbol{g}}_t \, \boldsymbol{i}_t (1 - \tilde{\boldsymbol{c}}_t^2) \boldsymbol{x}_t \tag{8}$$

$$\frac{\partial L}{\partial V} = \sum_{t=1}^{T} \boldsymbol{g}_t \boldsymbol{h}_t \tag{9}$$

Multi-layer LSTM Backpropagation

Forward pass

For a multi-layer LSTM, the forward pass follows that of the single-layer LSTM for each layer, with the exception that for all layers but the first, we replace the original input, $\boldsymbol{x}_t$, with the output of the previous layer. That is, for any layer k, for $k > 1$, we have that

$$\boldsymbol{x}_t^{(k)} = \boldsymbol{o}_t^{(k-1)}$$

such that we can describe a general forward pass as

$$\begin{aligned}
\boldsymbol{i}_t^{(k)} &= \sigma(W_i \boldsymbol{h}_{t-1}^{(k)} + U_i \boldsymbol{o}_t^{(k-1)}) \\
\boldsymbol{f}_t^{(k)} &= \sigma(W_f \boldsymbol{h}_{t-1}^{(k)} + U_f \boldsymbol{o}_t^{(k-1)}) \\
\boldsymbol{e}_t^{(k)} &= \sigma(W_e \boldsymbol{h}_{t-1}^{(k)} + U_e \boldsymbol{o}_t^{(k-1)}) \\
\tilde{\boldsymbol{c}}_t^{(k)} &= \tanh(W_c \boldsymbol{h}_{t-1}^{(k)} + U_c \boldsymbol{o}_t^{(k-1)}) \\
\boldsymbol{c}_t^{(k)} &= \boldsymbol{f}_t^{(k)} \bullet \boldsymbol{c}_{t-1}^{(k)} + \boldsymbol{i}_t^{(k)} \bullet \tilde{\boldsymbol{c}}_t^{(k)} \\
\boldsymbol{h}_t^{(k)} &= \boldsymbol{e}_t^{(k)} \bullet \tanh(\boldsymbol{c}_t^{(k)}) \\
\boldsymbol{o}_t^{(k)} &= V \boldsymbol{h}_t^{(k)}
\end{aligned}$$

where for $k = 1$ we have that $\boldsymbol{o}_t^{(k-1)} := \boldsymbol{x}_t$ and for $k = K$ we apply a softmax operation to the output, such that the final output of the network is given by

$$\boldsymbol{p}_t = \text{softmax}(\boldsymbol{o}_t^{(K)})$$

Backward pass

When calculating the gradients for the final layer, i.e. $k = K$, we can proceed as in the case of a single-layer LSTM, with the exception of using $\boldsymbol{o}_t^{(K-1)}$ instead of $\boldsymbol{x}_t$ from the forward pass. However, when considering the gradients for the preceding layer, i.e. $k = K - 1$, we start by considering the gradient of the loss with respect to the output:

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{o}_t^{(K-1)}} &= \frac{\partial L}{\partial \boldsymbol{i}_t^{(K)}} \frac{\partial \boldsymbol{i}_t^{(K)}}{\partial \boldsymbol{o}_t^{(K-1)}} + \frac{\partial L}{\partial \boldsymbol{f}_t^{(K)}} \frac{\partial \boldsymbol{f}_t^{(K)}}{\partial \boldsymbol{o}_t^{(K-1)}} + \frac{\partial L}{\partial \boldsymbol{e}_t^{(K)}} \frac{\partial \boldsymbol{e}_t^{(K)}}{\partial \boldsymbol{o}_t^{(K-1)}} + \frac{\partial L}{\partial \tilde{\boldsymbol{c}}_t^{(K)}} \frac{\partial \tilde{\boldsymbol{c}}_t^{(K)}}{\partial \boldsymbol{o}_t^{(K-1)}} \\
&= \hat{\boldsymbol{g}}_t^{(K)}\, \tilde{\boldsymbol{c}}_t^{(K)} \boldsymbol{i}_t^{(K)} (1 - \boldsymbol{i}_t^{(K)}) U_i^{(K)} \\
&\quad + \hat{\boldsymbol{g}}_t^{(K)}\, \boldsymbol{c}_{t-1}^{(K)} \boldsymbol{f}_t^{(K)} (1 - \boldsymbol{f}_t^{(K)}) U_f^{(K)} \\
&\quad + \tilde{\boldsymbol{g}}_t^{(K)}\, \tanh(\boldsymbol{c}_t^{(K)}) \boldsymbol{e}_t^{(K)} (1 - \boldsymbol{e}_t^{(K)}) U_e^{(K)} \\
&\quad + \hat{\boldsymbol{g}}_t^{(K)}\, \boldsymbol{i}_t^{(K)} (1 - (\tilde{\boldsymbol{c}}_t^{(K)})^2) U_c^{(K)}
\end{aligned}$$

We recognize that this iterative pattern is indeed true for all layers for which $k < K$, and define a general expression for the gradient w.r.t. $\boldsymbol{o}_t^{(k)}$, and set $\boldsymbol{g}_t^{(k)} := \frac{\partial L}{\partial \boldsymbol{o}_t^{(k)}}$ such that

$$\boldsymbol{g}_t^{(K)} = -(\boldsymbol{y}_t - \boldsymbol{p}_t)^T, \quad \forall\, t = 1, 2, \ldots, T$$

and, for $k = 1, 2, \ldots, K-1$:

$$\begin{aligned}
\boldsymbol{g}_t^{(k)} &= \hat{\boldsymbol{g}}_t^{(k+1)}\, \tilde{\boldsymbol{c}}_t^{(k+1)} \boldsymbol{i}_t^{(k+1)}(1 - \boldsymbol{i}_t^{(k+1)})U_i^{(k+1)} \\
&+ \hat{\boldsymbol{g}}_t^{(k+1)}\, \boldsymbol{c}_{t-1}^{(k+1)} \boldsymbol{f}_t^{(k+1)}(1 - \boldsymbol{f}_t^{(k+1)})U_f^{(k+1)} \\
&+ \tilde{\boldsymbol{g}}_t^{(k+1)}\, \tanh(\boldsymbol{c}_t^{(k+1)})\boldsymbol{e}_t^{(k+1)}(1 - \boldsymbol{e}_t^{(k+1)})U_e^{(k+1)} \\
&+ \hat{\boldsymbol{g}}_t^{(k+1)}\, \boldsymbol{i}_t^{(k+1)}(1 - (\tilde{\boldsymbol{c}}_t^{(k+1)})^2)U_c^{(k+1)}
\end{aligned}$$

for $t = 1, 2, \ldots, T$. Now, we may proceed similarly to the one-layer case and define the gradient with respect to the hidden unit and the memory cell:

$$\tilde{\boldsymbol{g}}_t^{(k)} = \frac{\partial L}{\partial \boldsymbol{h}_t^{(k)}} \quad \text{and} \quad \hat{\boldsymbol{g}}_t^{(k)} = \frac{\partial L}{\partial \boldsymbol{c}_t^{(k)}}$$

I - the case of $t = T$

Given that we have computed the gradients for the next layer, i.e. $k+1$, we can compute $\boldsymbol{g}_t^{(k)}$ and then compute the gradient with respect to the hidden unit in layer k as

$$\tilde{\boldsymbol{g}}_t^{(k)} = \frac{\partial L}{\partial \boldsymbol{h}_t^{(k)}} = \boldsymbol{g}_t^{(k)} V^{(k)}$$

and similarly for the memory cell:

$$\hat{\boldsymbol{g}}_t^{(k)} = \frac{\partial L}{\partial \boldsymbol{c}_t^{(k)}} = \boldsymbol{g}_t^{(k)} V^{(k)} \boldsymbol{e}_t^{(k)}(1 - \tanh^2(\boldsymbol{c}_t^{(k)}))$$

for $t = T$.

II - the case of $t < T$

Then, for $t = 1, 2, \ldots, T-1$, we have for the hidden unit and the memory cell that

$$\tilde{\boldsymbol{g}}_t^{(k)} = \boldsymbol{g}_t^{(k)} V^{(k)} + \frac{\partial L}{\partial \boldsymbol{i}_{t+1}^{(k)}} \frac{\partial \boldsymbol{i}_{t+1}^{(k)}}{\partial \boldsymbol{a}_{i,t+1}^{(k)}} W_i^{(k)} + \frac{\partial L}{\partial \boldsymbol{f}_{t+1}^{(k)}} \frac{\partial \boldsymbol{f}_{t+1}^{(k)}}{\partial \boldsymbol{a}_{f,t+1}^{(k)}} W_f^{(k)} + \frac{\partial L}{\partial \boldsymbol{e}_{t+1}^{(k)}} \frac{\partial \boldsymbol{e}_{t+1}^{(k)}}{\partial \boldsymbol{a}_{et+1}^{(k)}} W_e^{(k)} + \frac{\partial L}{\partial \tilde{\boldsymbol{c}}_{t+1}^{(k)}} \frac{\partial \tilde{\boldsymbol{c}}_{t+1}^{(k)}}{\partial \boldsymbol{a}_{c,t+1}^{(k)}} W_c^{(k)}$$

$$\hat{\boldsymbol{g}}_t^{(k)} = \boldsymbol{g}_t^{(k)} V^{(k)} \boldsymbol{e}_t^{(k)}(1 - \tanh^2(\boldsymbol{c}_t^{(k)})) + \frac{\partial L}{\partial \boldsymbol{c}_{t+1}^{(k)}} \boldsymbol{f}_{t+1}^{(k)}$$