Derivation of RNN Backpropagation Equations

AY

March 6, 2019

1 Notations

- Vector dot product: $\langle x, y \rangle = \sum_i x_i y_i$
- Matrix dot product: $A \otimes B = \sum_{i,j} A_{ij} B_{ij}$
- Matrix product: AB
- Entry-wise product: $A * B$

2 Variables

- ith training example at time t: $(x^{(i)\langle t \rangle}, y^{(i)\langle t \rangle})$, where $x^{(i)\langle t \rangle}$ and $y^{(i)\langle t \rangle}$ are column vectors with n_x and n_y components, respectively
- ith output at time t: $\hat{y}^{(i)\langle t \rangle}$
- ith activation at time t: $a^{(i)\langle t \rangle} = \tanh(W_{ax} x^{(i)\langle t \rangle} + W_{aa} a^{(i)\langle t-1 \rangle} + b_a)$ where $a^{(i)\langle t \rangle}$ is a column vector with n_a components
- Inputs at time t: $x^{\langle t \rangle} = (x^{(1)\langle t \rangle}, \ldots, x^{(m)\langle t \rangle})$, an $n_x \times m$ matrix
- Activations at time t: $a^{\langle t \rangle} = (a^{(1)\langle t \rangle}, \ldots, a^{(m)\langle t \rangle})$, an $n_a \times m$ matrix
- Cost at time t: $L^{\langle t \rangle} = \frac{1}{m} \sum_{i=1}^{m} -\langle y^{(i)\langle t \rangle}, \log \hat{y}^{(i)\langle t \rangle} \rangle$
- Total cost: $J = \sum_{t=1}^{T_x} L^{\langle t \rangle}$

3 Dependency

For the purpose of deriving formulas for $\frac{\partial J}{\partial W_{aa}}$, the following functional dependency will suffice:

- $L^{\langle t \rangle} = L^{\langle t \rangle}(a^{\langle t \rangle})$
- $a^{(i)\langle t \rangle} = a^{(i)\langle t \rangle}(W_{aa}, a^{(i)\langle t-1 \rangle})$

4 Computing $\frac{\partial J}{\partial W_{aa}}$ (denoted by $\frac{\partial J}{\partial W}$ for simplicity)

To simplify notations, we write W_{aa} as W, which is an $n_a \times n_a$ matrix with entries $W = (W_{k,l})$, where k is the row index and l is the column index.

The derivative $\frac{\partial J}{\partial W}$ is by definition the matrix

$$\frac{\partial J}{\partial W} = \left(\frac{\partial J}{\partial W_{k,l}} \right).$$

By chain rule,

$$\frac{\partial L^{\langle t \rangle}}{\partial W_{k,l}} = \sum_{s=1}^{t} \left(\frac{\partial L^{\langle t \rangle}}{\partial a^{\langle s \rangle}} \otimes \frac{\partial a^{\langle s \rangle}}{\partial W_{k,l}} \right). \tag{1}$$

The $\frac{\partial L^{\langle t \rangle}}{\partial a^{\langle s \rangle}}$ is a matrix with $\frac{\partial L^{\langle t \rangle}}{\partial a_j^{(i)\langle s \rangle}}$ on its jth row ith column, whereas the $\frac{\partial a^{\langle s \rangle}}{\partial W_{k,l}}$ is a matrix with $\frac{\partial a_j^{(i)\langle s \rangle}}{\partial W_{k,l}}$ on its jth row ith column. The matrix dot product above reads

$$\frac{\partial L^{\langle t \rangle}}{\partial a^{\langle s \rangle}} \otimes \frac{\partial a^{\langle s \rangle}}{\partial W_{k,l}} = \sum_{i=1}^{m} \sum_{j=1}^{n_a} \frac{\partial L^{\langle t \rangle}}{\partial a_j^{(i)\langle s \rangle}} \frac{\partial a_j^{(i)\langle s \rangle}}{\partial W_{k,l}}.$$

Dependency of $a^{\langle s \rangle}$ on $W_{k,l}$ through lower time levels have been taken care of in Eq.(1). Thus, when computing $\frac{\partial a^{\langle s \rangle}}{\partial W_{k,l}}$ through

$$a^{\langle s \rangle} = \tanh(W_{ax} x^{\langle s \rangle} + W_{aa} a^{\langle s-1 \rangle} + b_a),$$

we can treat $a^{\langle s-1 \rangle}$ as a constant.

The derivative of the total cost is

$$\frac{\partial J}{\partial W_{k,l}} = \sum_{t=1}^{T_x} \frac{\partial L^{\langle t \rangle}}{\partial W_{k,l}} = \sum_{t=1}^{T_x} \sum_{s=1}^{t} \left(\frac{\partial L^{\langle t \rangle}}{\partial a^{\langle s \rangle}} \otimes \frac{\partial a^{\langle s \rangle}}{\partial W_{k,l}} \right).$$

Regrouping the terms, it reads

$$\frac{\partial J}{\partial W_{k,l}} = \sum_{t=1}^{T_x} \left(\sum_{s=t}^{T_x} \frac{\partial L^{\langle s \rangle}}{\partial a^{\langle t \rangle}} \right) \otimes \frac{\partial a^{\langle t \rangle}}{\partial W_{k,l}} = \sum_{t=1}^{T_x} Q^{\langle t \rangle} \otimes \frac{\partial a^{\langle t \rangle}}{\partial W_{k,l}},$$

where

$$Q^{\langle t \rangle} = \sum_{s=t}^{T_x} \frac{\partial L^{\langle s \rangle}}{\partial a^{\langle t \rangle}}$$

is a matrix with (j, i)th entry given by

$$Q_j^{(i)\langle t \rangle} = \frac{\partial}{\partial a_j^{(i)\langle t \rangle}} \left(\sum_{s=t}^{T_x} L^{\langle s \rangle} \right).$$

5 The function `rnn_cell_backward`

The variables in the function `rnn_cell_backward` corresponds to the following values

- `da_next` $= Q^{\langle t\rangle}$ (input)
- `da_prev` $= Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial a^{\langle t-1\rangle}}$ (output)
- `dWaa` $= Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W}$ (output)

The term $Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial a^{\langle t-1\rangle}}$ is understood as a matrix with (j,i)th entry equal to

$$Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial a_j^{(i)\langle t-1\rangle}},$$

which is in fact equal to

$$\frac{\partial}{\partial a_j^{(i)\langle t-1\rangle}} \left(\sum_{s=t}^{T_x} L^{\langle s\rangle} \right).$$

Likewise, the term $Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W}$ is understood as a matrix with (k,l)th entry equal to

$$\frac{\partial}{\partial W_{k,l}} \left(\sum_{s=t}^{T_x} L^{\langle s\rangle} \right).$$

6 The function `rnn_backward` and the mysterious `da`

The matrix $Q^{\langle t\rangle}$ can be computed recursively (in backward manner) via

$$\begin{aligned} Q^{\langle t\rangle} &= \sum_{s=t}^{T_x} \frac{\partial L^{\langle s\rangle}}{\partial a^{\langle t\rangle}} = \frac{\partial L^{\langle t\rangle}}{\partial a^{\langle t\rangle}} + \sum_{s=t+1}^{T_x} \frac{\partial L^{\langle s\rangle}}{\partial a^{\langle t\rangle}} \\ &= \frac{\partial L^{\langle t\rangle}}{\partial a^{\langle t\rangle}} + \left(\sum_{s=t+1}^{T_x} \frac{\partial L^{\langle s\rangle}}{\partial a^{\langle t+1\rangle}} \right) \otimes \frac{\partial a^{\langle t+1\rangle}}{\partial a^{\langle t\rangle}} \\ &= \frac{\partial L^{\langle t\rangle}}{\partial a^{\langle t\rangle}} + Q^{\langle t+1\rangle} \otimes \frac{\partial a^{\langle t+1\rangle}}{\partial a^{\langle t\rangle}}. \end{aligned}$$

In the main loop of `rnn_backward`, the following is going on:

- `da`$[:,:,$`t`$] = \frac{\partial L^{\langle t+1\rangle}}{\partial a^{\langle t+1\rangle}}$ (shifted by 1 because Python index starts from 0)
- `da`$[:,:,$`t`$] +$ `da_prevt` $= \frac{\partial L^{\langle t+1\rangle}}{\partial a^{\langle t+1\rangle}} + Q^{\langle t+2\rangle} \otimes \frac{\partial a^{\langle t+2\rangle}}{\partial a^{\langle t+1\rangle}} = Q^{\langle t+1\rangle}$
- `dWaat` $= Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W}$

The values of $\frac{\partial L^{\langle t\rangle}}{\partial a^{\langle t\rangle}}$ are assumed given (computed elsewhere) and stored in `da`$[:,:,$`t`$-$`1`$]$ for $t = 1, 2, \ldots, T_x$. By aggregating `dWaat` over t, we obtain $\frac{\partial J}{\partial W} = \sum_{t=1}^{T_x} Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W}$ when the main loop termintes.

7 Detailed Computations

Recall that

$$a^{(i)\langle t\rangle} = \tanh(W_{ax}x^{(i)\langle t\rangle} + Wa^{(i)\langle t-1\rangle} + b_a).$$

The jth entry reads

$$a_j^{(i)\langle t\rangle} = \tanh\left(\sum_{h=1}^{n_x} W_{ax,j,h} \times x_h^{(i)\langle t\rangle} + \sum_{h=1}^{n_a} W_{j,h} \times a_h^{(i)\langle t\rangle} + b_{a,j}\right).$$

Therefore,

$$\begin{aligned} Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W_{k,l}} &= \sum_{i=1}^{m}\sum_{j=1}^{n_a}\left[Q_j^{(i)\langle t\rangle} \times \frac{\partial a_j^{(i)\langle t\rangle}}{\partial W_{k,l}}\right] \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n_a}\left[Q_j^{(i)\langle t\rangle} \times (1-(a_j^{(i)\langle t\rangle})^2) \times \sum_{h=1}^{n_a}\frac{\partial W_{j,h}}{\partial W_{k,l}} \times a_h^{(i)\langle t\rangle}\right] \end{aligned}$$

As we run over j and h, the term $\frac{\partial W_{j,h}}{\partial W_{k,l}}$ is non-zero (equals 1) only when $j = k$ and $h = l$. Thus,

$$Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W_{k,l}} = \sum_{i=1}^{m}\left[Q_k^{(i)\langle t\rangle} \times (1-(a_k^{(i)\langle t\rangle})^2) \times a_l^{(i)\langle t\rangle}\right]$$

Denote by $Q_k^{\langle t\rangle}$ the kth row of $Q^{\langle t\rangle}$ and $a_l^{\langle t\rangle}$ the lth row of $a^{\langle t\rangle}$. We have

$$Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W_{k,l}} = \left\langle Q_k^{\langle t\rangle} * (1-(a_k^{\langle t\rangle})^2), a_l^{\langle t\rangle}\right\rangle,$$

where $*$ is entry-wise product and $(\cdot)^2$ is the entry-wise square. Note that the (k,l)th entry of the matrix product XY is the vector dot product between the ith row of X and the jth column of Y. Hence, we can recognize the above equation as the vector dot product between the kth row of $Q^{\langle t\rangle} * (1-(a^{\langle t\rangle})^2)$ and the lth column of $(a^{\langle t\rangle})^T$, where T is the matrix transpose. The matrix of derivatives computed in `rnn_cell_backward` is therefore

$$\texttt{dWaa} = Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial W} = \left[Q^{\langle t\rangle} * (1-(a^{\langle t\rangle})^2)\right](a^{\langle t\rangle})^T.$$

Likewise, one can deduce that

$$\texttt{da\_prev} = Q^{\langle t\rangle} \otimes \frac{\partial a^{\langle t\rangle}}{\partial a^{\langle t-1\rangle}} = W^T\left[Q^{\langle t\rangle} * (1-(a^{\langle t\rangle})^2)\right].$$

Similar formulas hold for `dxt` and `dWax`. For `dba`, replace $(a^{\langle t\rangle})^T$ in `dWaa` with a column vector of ones.