Recursive Deep Models for Semantic Compositionality Over A Sentiment Treebank
Derivations for Gradient Equations

Alain Soltani

In the following sections, for a vector b, $b_{[i]}$ denotes the i-th item ; for a matrix A, $A_{[i,j]}, A_{[i,:]}, A_{[:,j]}$ respectively denote the (i,j) coefficient, i-th row and the j-th column.

1 Slices of Tensor V

The RNTN error function for the network's top node p_2 is :

$$E^{p_2} = \sum_{j=1} t^{p_2}_{[j]} \log y^{p_2}_{[j]} = \log y^{p_2}_{[r]}$$

where r is the index of the label (using a 0-1 encoding for the target distribution), so that $t^{p_2}_{[j]} = \mathbb{1}_{\{j=r\}}$.
Thus we can derive the following d-dimensional gradient :

$$\begin{aligned}
\frac{\partial E^{p_2}}{\partial p_2} &= \frac{1}{y^{p_2}_{[r]}} \frac{\partial y^{p_2}_{[r]}}{\partial p_2} \\
&= \frac{1}{y^{p_2}_{[r]}} \frac{\partial}{\partial p_2} \left[\frac{\exp(W^s_{[r,:]} p_2 + b^s_{[r]})}{\sum_{l=1}^{d} \exp(W^s_{[l,:]} p_2 + b^s_{[l]})} \right] \\
&= \frac{1}{y^{p_2}_{[r]}} [W^s_{[r,:]} y^{p_2}_{[r]} - y^{p_2}_{[r]} \sum_{l=1}^{d} W^s_{[l,:]} y^{p_2}_{[l]}], \text{ via a classical product derivation} \\
&= W^s_{[r,:]} - \sum_{l=1}^{d} W^s_{[l,:]} y^{p_2}_{[l]} \\
&= \sum_{l=1}^{d} W^s_{[l,:]} t^{p_2}_{[l]} - W^s_{[l,:]} y^{p_2}_{[l]}, \text{ as } t^{p_2}_{[l]} = \mathbb{1}_{\{l=r\}} \\
&= \begin{bmatrix} \sum_{l=1}^{d} W^s_{[l,1]} (t^{p_2}_{[l]} - y^{p_2}_{[l]}) \\ \cdots \\ \sum_{l=1}^{d} W^s_{[l,d]} (t^{p_2}_{[l]} - y^{p_2}_{[l]}) \end{bmatrix} = (W^s)^T (t^{p_2} - y^{p_2}).
\end{aligned} \tag{1}$$

In the meantime, one can form the following gradient for $V^{[k]}, k \in \langle 1, d \rangle$, k-th slice of tensor V :

$$\frac{\partial p_{2[j]}}{\partial V^{[k]}} = \frac{\partial}{\partial V^{[k]}} f(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[j]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W_{[j,:]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + b_{[j]}), \tag{2}$$

where W, b are respectively the weight and bias matrices.
One can immediately note that $\frac{\partial p_{2[j]}}{\partial V^{[k]}} = 0_{(2d,2d)}$ if $j \neq k, j \in \langle 1, d \rangle$.
For $j = k$:

$$\begin{aligned} \frac{\partial p_{2[k]}}{\partial V^{[k]}} &= f'(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[k]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W_{[k,:]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + b_{[k]}) \begin{bmatrix} a \\ p_1 \end{bmatrix} \begin{bmatrix} a \\ p_1 \end{bmatrix}^T \\ &= f'(x^{p_2}_{[k]}) \begin{bmatrix} a \\ p_1 \end{bmatrix} \begin{bmatrix} a \\ p_1 \end{bmatrix}^T . \end{aligned} \tag{3}$$

by chain derivation on the function's first term.

Finally, we obtain the following gradient :

$$\begin{aligned} \frac{\partial E^{p_2}}{\partial V^{[k]}} &= \frac{\partial E^{p_2}}{\partial p_2} \frac{\partial p_2}{\partial V^{[k]}} = \frac{\partial E^{p_2}}{\partial p_{2[k]}} \frac{\partial p_{2[k]}}{\partial V^{[k]}} \text{ as there is only one non-zero slice in tensor } \frac{\partial p_2}{\partial V^{[k]}} \\ &= [(W^s)^T (t^{p_2} - y^{p_2}) \bigotimes f'(x^{p_2})]_{[k]} \begin{bmatrix} a \\ p_1 \end{bmatrix} \begin{bmatrix} a \\ p_1 \end{bmatrix}^T \\ &= \delta^{p_2,com}_{[k]} \begin{bmatrix} a \\ p_1 \end{bmatrix} \begin{bmatrix} a \\ p_1 \end{bmatrix}^T = \delta^{p_2,com}_{[k]} \begin{bmatrix} a \\ p_1 \end{bmatrix} \bigodot \begin{bmatrix} a \\ p_1 \end{bmatrix} , \end{aligned} \tag{4}$$

$\bigodot$ denoting the outer product, $\bigotimes$ the Hadamard product.

2 Weight W

Very similarly as in (2), we can calculate the following gradient, related to W :

$$\begin{aligned} \frac{\partial p_2}{\partial W} &= \frac{\partial}{\partial W} f(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix} + b) \\ &= f'(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix} + b) \begin{bmatrix} a \\ p_1 \end{bmatrix}^T \\ &= f'(x^{p_2}) \begin{bmatrix} a \\ p_1 \end{bmatrix}^T . \end{aligned} \tag{5}$$

Hence

$$\begin{aligned} \frac{\partial E^{p_2}}{\partial W} &= \frac{\partial E^{p_2}}{\partial p_2} \frac{\partial p_2}{\partial W} = (W^s)^T (t^{p_2} - y^{p_2}) \bigotimes f'(x^{p_2}) \begin{bmatrix} a \\ p_1 \end{bmatrix}^T \\ &= \delta^{p_2,com} \bigodot \begin{bmatrix} a \\ p_1 \end{bmatrix} . \end{aligned} \tag{6}$$

3 Bias b

This one is rather straightforward :

$$\begin{aligned}
\frac{\partial p_2}{\partial b} &= \frac{\partial}{\partial b} f(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix} + b) \\
&= f'(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix} + b) = f'(x^{p_2}).
\end{aligned} \tag{7}$$

Thus, we obtain by chain derivation

$$\frac{\partial E^{p_2}}{\partial b} = \frac{\partial E^{p_2}}{\partial p_2}\frac{\partial p_2}{\partial b} = (W^s)^T(t^{p_2} - y^{p_2}) \bigotimes f'(x^{p_2}) = \delta^{p_2,com}. \tag{8}$$

4 Classification weight W^s

Let us consider Eq. (1) from Socher's paper for computing the posterior probability over labels :

$$y^a = \mathrm{softmax}(W^s a + b^s). \tag{9}$$

when including the bias b^s. At the network's top node, this becomes :

$$y^{p_2} = \mathrm{softmax}(W^s p_2 + b^s).$$

From this follows the gradient's calculus for coefficient (i, j) from W^s :

$$\frac{\partial y^{p_2}_{[r]}}{\partial W^s_{[i,j]}} = \frac{\partial}{\partial W^s_{[i,j]}} \left[\frac{\exp(W^s_{[r,:]} p_2 + b^s_{[r]})}{\sum_{l=1}^{d} \exp(W^s_{[l,:]} p_2 + b^s_{[l]})} \right]$$

- If $r = i$:

$$\begin{aligned}
\frac{\partial y^{p_2}_{[r]}}{\partial W^s_{[i,j]}} &= -\frac{y^{p_2}[r]}{\sum_{l=1}^{d} \exp(W^s_{[l,:]} p_2 + b^s_{[l]})} \frac{\partial}{\partial W^s_{[i,j]}} \left[\exp(W^s_{[i,:]} p_2 + b^s_{[i]})\right] \\
&= -y^{p_2}_{[r]} y^{p_2}_{[i]} p^2_{[j]}.
\end{aligned}$$

- If $r \neq i$:

$$\frac{\partial y^{p_2}_{[r]}}{\partial W^s_{[i,j]}} = y^{p_2}_{[r]} p^2_{[j]} - y^{p_2}_{[r]} y^{p_2}_{[i]} p^2_{[j]}.$$

Finally :

$$\frac{\partial E^{p_2}}{\partial W^s_{[i,j]}} = \frac{1}{y^{p_2}_{[r]}} \frac{\partial y^{p_2}_{[r]}}{\partial W^s_{[i,j]}} = p^2_{[j]}(\mathbb{1}_{\{r=i\}} - y^{p_2}_{[i]}) = p^2_{[j]}(t^{p_2}_{[i]} - y^{p_2}_{[i]})$$

i.e.

$$\begin{aligned}
\frac{\partial E^{p_2}}{\partial W^s_{[i,j]}} &= [(t^{p_2} - y^{p_2}) \bigodot p^2]_{[i,j]} \\
\text{and } \frac{\partial E^{p_2}}{\partial W^s} &= (t^{p_2} - y^{p_2}) \bigodot p^2.
\end{aligned} \tag{10}$$

5 Classification bias b^s

Similarly, one can obtain the classification bias' gradient :

$$\begin{aligned}\frac{\partial E^{p_2}}{\partial b^s_{[i]}} &= \frac{1}{y^{p_2}_{[r]}}\frac{\partial y^{p_2}_{[r]}}{\partial b^s_{[i]}} = \frac{1}{y^{p_2}_{[r]}}\frac{\partial}{\partial b^s_{[i]}}\left[\frac{\exp(W^s_{[r,:]}p_2 + b^s_{[r]})}{\sum_{l=1}^{d}\exp(W^s_{[l,:]}p_2 + b^s_{[l]})}\right] \\ &= \mathbb{1}_{\{r=i\}} - y^{p_2}[i] = t^{p_2}[i] - y^{p_2}[i]\end{aligned} \tag{11}$$

and

$$\frac{\partial E^{p_2}}{\partial b^s} = t^{p_2} - y^{p_2}. \tag{12}$$

6 Moving through the nodes

One can then make the calculus for the children's node by first back-propagating the error :

$$\delta^{p_2,down} = (W^T\delta^{p_2,com} + S) \bigotimes f'(\begin{bmatrix} a \\ p_1 \end{bmatrix}) \tag{13}$$

This equation is a classical RNN back-propagation error, except for the additional term S accounting for all the tensor slices' contributions :

$$S = \sum_{k=1}^{d} \delta^{p_2,com}(V^{[k]} + (V^{[k]})^T)\begin{bmatrix} a \\ p_1 \end{bmatrix}. \tag{14}$$

We decompose the stacked error $\delta^{p_2,down}$ in two sub-errors, $\delta^{p_2,down}_{[1:d]}$ and $\delta^{p_2,down}_{[d+1:2d]}$.
These are used to compute the children errors :

$$\begin{aligned}&\bullet\ \delta^{p_1,com} = \delta^{p_1,s} + \delta^{p_2,down}_{[d+1:2d]} \\ &\bullet\ \delta^{a,com} = \delta^{p_2,down}_{[1:d]}\end{aligned} \tag{15}$$

Thus the full derivative for each parameter (tensor slice, weight, bias, etc.) sums up all node gradients :

$$\begin{aligned}\frac{\partial E}{\partial V^{[k]}} &= \frac{\partial E^{p_2}}{\partial V^{[k]}} + \delta^{p_1,com}_{[k]}\begin{bmatrix} b \\ c \end{bmatrix} \bigodot \begin{bmatrix} b \\ c \end{bmatrix} \\ \frac{\partial E}{\partial W} &= \frac{\partial E^{p_2}}{\partial W} + \delta^{p_1,com} \bigodot \begin{bmatrix} b \\ \\ c \end{bmatrix} \\ \frac{\partial E}{\partial b} &= \frac{\partial E^{p_2}}{\partial b} + \delta^{p_1,com} \\ &\text{etc.}\end{aligned} \tag{16}$$