Some Neural Network Derivative Calculations

David S. Rosenberg

1 Affine Transformation

$$y = Wx + b$$

where y and b are $m \times 1$, x is $d \times 1$, and W is $m \times d$.

Now there is also some function $f : \mathbf{R}^m \to \mathbf{R}$, and let's write $J = f(Wx + b)$. Our goal is to find the partial derivative of J with respect to each element of W, namely $\partial J / \partial W_{ij}$. Suppose we have already computed the partial derivatives of J with respect to the intermediate variable y, namely $\frac{\partial J}{\partial y_i}$ for $i = 1, \ldots, m$. Then by the chain rule, we have

$$\frac{\partial J}{\partial W_{ij}} = \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \frac{\partial y_r}{\partial W_{ij}}.$$

Now $y_r = W_{r \cdot} x + b_r = b_r + \sum_{k=1}^{d} W_{rk} x_k$. So

$$\frac{\partial y_r}{\partial W_{ij}} = x_k \delta_{ir} \delta_{jk} = x_j \delta_{ir},$$

where $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$.

Putting it together we get

$$\begin{aligned} \frac{\partial J}{\partial W_{ij}} &= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} x_j \delta_{ir} \\ &= \frac{\partial J}{\partial y_i} x_j \end{aligned}$$

We can represent these partial derivatives as a matrix and compute it where the ij'th entry of $\frac{\partial J}{\partial W}$ is $\frac{\partial J}{\partial W_{ij}}$, i.e. the partial derivative of J w.r.t. the parameter W_{ij}. It's gonna be

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial y} x^T,$$

where $\frac{\partial J}{\partial y}$ is $m \times 1$ and x is $d \times 1$. So this is an outer product of two vectors, yielding an $m \times d$ matrix.

We'll also need the derivative w.r.t x – if it's actually data, we don't need the derivative w.r.t. x, but when we chain things together, x will be the output of another unit:

$$\frac{\partial y_r}{x_i} = W_{ri}$$

$$\begin{aligned}\frac{\partial J}{\partial x_i} &= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \frac{\partial y_r}{\partial x_i} \\ &= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} W_{ri} \\ &= \left(\frac{\partial J}{\partial y}\right)^T W_{\cdot i}\end{aligned}$$

and

$$\frac{\partial J}{\partial x} = W^T \left(\frac{\partial J}{\partial y}\right)$$

will give us a column vector.

Similarly,

$$\begin{aligned}\frac{\partial J}{\partial b_i} &= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \frac{\partial y_r}{\partial b_i} \\ &= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \delta_{ir} \\ &= \frac{\partial J}{\partial y_i}\end{aligned}$$

Let's repeat the same calculations for a minibatch. Let's suppose we have n inputs $x_1, \ldots, x_n \in \mathbf{R}^d$, and we stack them in the usual way as rows in a $n \times d$ design matrix X. For each x_i there's an intermediate output $y_i = Wx_i + b$. Let's consider stacking these as rows as well, so each row is $y_i^T = x_i^T W^T + b^T$. Let's write Y for the $n \times m$ matrix, which stacks the n row vectors y_i^T on top of each other. Then we have

$$Y = XW^T + b^T,$$

and the rs'th entry is given by

$$\begin{aligned} Y_{rs} &= X_{r\cdot}\left(W^T\right)_{\cdot s} + 1b^T, \\ &= \sum_{k=1}^{d} X_{rk}\left(W^T\right)_{ks} + b_s \\ &= \sum_{k=1}^{d} X_{rk}W_{sk} + b_s \end{aligned}$$

whee 1 is an $n \times 1$ column vector. where the notation $X_{r\cdot}$ refers the the rth row of X, as a row matrix, and similarly $X_{\cdot s}$ refers to the sth column of X, as a column matrix. Now

$$\begin{aligned} \frac{\partial Y_{rs}}{\partial W_{ij}} &= X_{rk}\delta_{is}\delta_{jk} = X_{rj}\delta_{is} \\ \frac{\partial Y_{rs}}{\partial b_i} &= \delta_{is} \\ \frac{\partial Y_{rs}}{\partial X_{ij}} &= \sum_{k=1}^{d} W_{sk}\delta_{ir}\delta_{jk} = W_{sj}\delta_{ir} \end{aligned}$$

(Note – the necessity for the δ_{ir} should be obvious if we understand what rows of Y and X are.)

Now we have a function $f : \mathbf{R}^{n\times m} \to \mathbf{R}$ that operates on a full minibatch and produces a single scalar. This would typically be the average of the

$f(Wx_i + b)$ over $i = 1, \ldots, n$. So

$$\begin{aligned}
\frac{\partial J}{\partial W_{ij}} &= \sum_{r=1}^{n}\sum_{s=1}^{m} \frac{\partial J}{\partial Y_{rs}} \frac{\partial Y_{rs}}{\partial W_{ij}} \\
&= \sum_{r=1}^{n}\sum_{s=1}^{m} \frac{\partial J}{\partial Y_{rs}} X_{rj}\delta_{is} \\
&= \sum_{r=1}^{n} \frac{\partial J}{\partial Y_{ri}} X_{rj} \\
&= \left[\left(\frac{\partial J}{\partial Y}\right)_{\cdot i}\right]^T X_{\cdot j}
\end{aligned}$$

where $\frac{\partial J}{\partial Y}$ is the $n \times m$ matrix with $\frac{\partial J}{\partial Y_{ij}}$ in the ij'th entry. So

$$\frac{\partial J}{\partial W} = \left(\frac{\partial J}{\partial Y}\right)^T X$$

and

$$\begin{aligned}
\frac{\partial J}{\partial b_i} &= \sum_{r=1}^{n}\sum_{s=1}^{m} \frac{\partial J}{\partial Y_{rs}} \frac{\partial Y_{rs}}{\partial b_i} \\
&= \sum_{r=1}^{n}\sum_{s=1}^{m} \frac{\partial J}{\partial Y_{rs}} \delta_{is} \\
&= \sum_{r=1}^{n} \frac{\partial J}{\partial Y_{ri}} \\
&= 1^T \left(\frac{\partial J}{\partial Y}\right)_{\cdot i}
\end{aligned}$$

and if we let $\frac{\partial J}{\partial b}$ be the $b \times 1$ vector of derivatives $\frac{\partial J}{\partial b_i}$, then we can write

$$\frac{\partial J}{\partial b} = \left(\frac{\partial J}{\partial Y}\right)^T 1.$$

Finally,

$$
\begin{aligned}
\frac{\partial J}{\partial X_{ij}} &= \sum_{r=1}^{n}\sum_{s=1}^{m}\frac{\partial J}{\partial Y_{rs}}\frac{\partial Y_{rs}}{\partial X_{ij}} \\
&= \sum_{r=1}^{n}\sum_{s=1}^{m}\frac{\partial J}{\partial Y_{rs}}W_{sj}\delta_{ir} \\
&= \sum_{s=1}^{m}\frac{\partial J}{\partial Y_{is}}W_{sj}
\end{aligned}
$$

So

$$
\frac{\partial J}{\partial X} = \frac{\partial J}{\partial Y}W
$$

2 Softmax

Consider an input vector of scores s is $d \times 1$ and output vector y also $d \times 1$, where y encodes a probability distribution over d classes. Then the ith entry of the output is given by

$$
y_i = \frac{\exp(s_i)}{\sum_{c=1}^{k}\exp(s_c)}.
$$

Then

$$
\begin{aligned}
\frac{\partial y_i}{\partial s_j} &= \frac{\frac{\partial}{\partial s_j}(\exp(s_i))}{\sum_{c=1}^{k}\exp(s_c)} - \frac{\exp(s_i)\frac{\partial}{\partial s_j}\left(\sum_{c=1}^{k}\exp(s_c)\right)}{\left[\sum_{c=1}^{k}\exp(s_c)\right]^2} \\
&= \frac{\exp(s_i)\,\delta_{ij}}{\sum_{c=1}^{k}\exp(s_c)} - \frac{\exp(s_i)\exp(s_j)}{\left[\sum_{c=1}^{k}\exp(s_c)\right]^2} \\
&= \sigma(s_i)\delta_{ij} - \sigma(s_i)\sigma(s_j) \\
&= \sigma(s_i)\left(\delta_{ij} - \sigma(s_j)\right)
\end{aligned}
$$

Now there is also some function $f : \mathbf{R}^d \to \mathbf{R}$, and let's write $J = f(\sigma(s))$. Our goal is to find the partial derivative of J with respect to each element of s, namely $\partial J/\partial s_j$. Suppose we have already computed all partial derivatives of J with respect to the intermediate vector $y = \sigma(s)$, namely $\frac{\partial J}{\partial y_i}$ for $i = 1, \ldots, d$. Then by the chain rule, we have

$$\begin{aligned}
\frac{\partial J}{\partial s_j} &= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \frac{\partial y_r}{\partial s_j} \\
&= \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \sigma(s_r) \left(\delta_{rj} - \sigma(s_j)\right) \\
&= \frac{\partial J}{\partial y_j} \sigma(s_j) - \sum_{r=1}^{m} \frac{\partial J}{\partial y_r} \sigma(s_r) \sigma(s_j)
\end{aligned}$$

so

$$\frac{\partial J}{\partial s} = \left(\frac{\partial J}{\partial y} - \left[\left(\frac{\partial J}{\partial y} \right)^T \sigma(s) \right] 1 \right) * \sigma(s)$$

Now suppose we are using a minibatch, in which case we have