Back Propagation

Marc Emanuel

March 9, 2015

Abstract

We consider a neural network with k layers. The cost function we write with q running over the samples as and summing over repeated indices (Einstein summing convention):

$$J = \frac{1}{m}\sum_q y_i \log[\sigma(z_i^{(k)})] + (1-y_i)\log[1-\sigma(z_i^{(k)})] \tag{1}$$

$$z_i^{(l)} = \theta_{i,j}^{(l-1)}\sigma(z_j^{(l-1)}) \tag{2}$$

$$z_i^{(2)} = \theta_{i,j}^{(1)} X_j \tag{3}$$

The second equation is a recursion and the third the initial condition. σ is the sigmoid function with derivative :

$$\sigma'(z) = \sigma(z)(1-\sigma(z)) = \frac{1}{2}\frac{1}{1+\cosh(z)} \tag{4}$$

The first form is as given in the programming exercise, but it is in fact not a good expression to calculate it computationally since it is pretty sensitive to rounding errors especially when z is larger than zero when you subtract two numbers that are almost equal. The second form behaves much better. We will use the first form in the analytical calculations.

The question is how to calculate the gradient wrt to $\theta_{i,j}^{(\ell)}$ for an $\ell \in (k, 1]$

Lets do it using induction: The first step is (I will leave out the q summation for clarity) just applying the chain-rule

$$\partial_{\theta_{r,s}^{(k-1)}} J = \left\{ y_i \frac{\sigma(z_i^{(k)})[1-\sigma(z_i^{(k)})]}{\sigma(z_i^{(k)})} - (1-y_i)\frac{\sigma(z_i^{(k)})[1-\sigma(z_i^{(k)})]}{1-\sigma(z_i^{(k)})} \right\} \partial_{\theta_{r,s}^{(k-1)}} z_i^{(k)} \tag{5}$$

and lo and behold lots of terms cancel and using (2) this leaves us with

$$\partial_{\theta_{r,s}^{(k-1)}} J = [y_r - \sigma(z_r^{(k)})]\sigma(z_s^{(k-1)}) = [y_r - a_r^{(k)}].a_s^{(k-1)} =: \delta_r^{(k)} a_s^{(k-1)} \tag{6}$$

To streamline the iteration for the rest of the chain of maps we consider at each level the $a_{i_\ell}^{(\ell)}$ as the coordinates of an n_ℓ -dimensional space, $\mathtt{C}_\ell$ which is at least locally a normal Euclidean space for which a_i's are local coordinates

To keep the overview we index the index with the level, thus we know in local coordinates on what level we are. We next define maps:

$$F^{(k-1)}_{i_\ell, j_{\ell-1}} : \mathtt{C}_{\ell-1} \to \mathtt{C}_\ell \qquad \text{through:} \qquad F^{(k-1)}_{i_\ell, j_{\ell-1}}(a^{(\ell-1)}_{j_{\ell-1}}) := \sigma(\theta^{\ell-1}_{i_\ell, j_{\ell-1}} a^{(\ell-1)}_{j_{\ell-1}})$$

This tells us the output, of the neural network is given by the composition :

$$a^{(k)} = F^{(k-1)} \circ F^{(k-2)} \circ \cdots F^{(1)}(a^{(1)})$$

Now the parameter matrix $\theta^{(\ell)}$ only appears in the map(s) F^ℓ . And so to obtain the partial derivatives of the J we chain rule downwards along the map chain till we bump into the corresponding map:

$$\frac{\partial J}{\partial \theta^{(\ell-1)}_{i_\ell, j_{\ell-1}}} = \frac{\partial J}{\partial a^{(k-1)}_{r_{k-1}}} \frac{\partial a^{(k-1)}_{r_{k-1}}}{\theta^{(\ell-1)}_{i_\ell, j_{\ell-1}}} \tag{7}$$

$$\tag{8}$$

The first derivative we can do along the lines of the derivation of (), exchanging the role played by the linear map, $\theta^{(k-1)}_{i_k, j_{k-1}}$ and the coordinates ,$a^{(k-1)}_{i_{k-1}}$. The second one is the derivative of the chain of maps

$$\begin{aligned}
\frac{\partial J}{\partial \theta^{(\ell)}_{i_{\ell+1}, r_\ell}} &= \delta^{(k)}_{r_k} \theta^{(k-1)}_{r_k, s_{k-1}} \frac{\partial}{\theta^{(\ell)} i_{\ell+1}, j_\ell} (F^{(k-2)}_{s_{k-1}, \cdots} \circ \cdots F^{(1)(a^{(1)})}) \\
&= \delta^{(k)}_{r_k} \theta^{(k-1)}_{r_k, r_{k-1}} d_a F^{k-2}_{r_{k-1}, r_{k-2}} d_a F^{k-3}_{r_{k-2}, r_{k-3}} \cdots \frac{\partial}{\theta^{(\ell)}_{i_{\ell+1}, r_\ell}} F^{(\ell)}_{r_{\ell+1}, r_\ell} \\
&= \delta^{(k)}_{r_k} \theta^{(k-1)}_{r_k, r_{k-1}} d_a F^{k-2}_{r_{k-1}, r_{k-2}} d_a F^{k-3}_{r_{k-2}, r_{k-3}} \cdots \sigma'(\theta^{(\ell)}_{i_{\ell+1}, r_\ell} a^{(\ell)}_{r_\ell}) a^{(\ell)}_{j_\ell} \\
&=: \delta^{\ell+1}_{i_{\ell+1}} a^{(\ell)}_{j_\ell}
\end{aligned} \tag{9}$$

With daF denoting the derivative of F wrt its argument. Now it straightforward to validate the back propagation. This is of course pretty messy all these indices , so I have the feeling that the neater way to show this is to consider the gradient as a one form and in that case is the back propagation nothing but a pull back of the gradient. Perhaps TA's know if this has been done in this context ?