General feed forward neural networks

Kristian Wichmann

July 19, 2017

1 Formalism and nomenclature

Consider a feed forward neural net with $L+1$ layers, in the sense that layer zero is the input layer and layer L the output layer. Each non-input layer has its own activation function σ_l.

The size of layer l we will denote S_l. So layer l has a $S_l \times S_{l-1}$ matrix of weights $W^{(l)}$ and a bias vector $b^{(l)}$ with dimension S_l. So, given an input vector x (with dimension S_{l-1}), the pre-activation and the activation of layer l can be expressed as:

$$z^{(l)} = W^{(l)}x + b^{(l)}, \quad a^{(l)} = \sigma_l(z^{(l)}) = \sigma_l(W^{(l)}x + b^{(l)}) \tag{1.1}$$

We will consider N data points, each with a feature vector of dimension S_0. We group these into an $S_0 \times N$ matrix X:

$$X = \begin{pmatrix} | & \cdots & | \\ x_1 & \cdots & x_N \\ | & \cdots & | \end{pmatrix} \tag{1.2}$$

Note that this is the transpose of the usual "data frame" structure.

We can now write the pre-activations and activations of the first layer as:

$$Z^{(1)} = W^{(1)}X + b^{(1)}, \quad A^{(1)} = \sigma_1(Z^{(1)}) = \sigma_1(W^{(1)}X + b^{(1)}) \tag{1.3}$$

Here, we've used "$+b^{(1)}$" as a shorthand for adding the vector $b^{(1)}$ to every column. We could write this as "$+b^{(1)}J_N^t$" if we wanted to be accurate. (J_N is a column vector of N ones).

Similarly, we may generally write the pre-activations and activations of layer l as:

$$Z^{(l)} = W^{(l)}A^{(l-1)} + b^{(l)}, \quad A^{(l)} = \sigma_l(Z^{(l)}) = \sigma_l(W^{(l)}A^{(l-1)} + b^{(l)}) \tag{1.4}$$

We will also identify X with the activations of "layer zero": $A^{(0)} = X$.

Finally, we have an error function J which measures the distance to some target data $T \in \mathbb{R}^{S_L \times N}$:

$$T = \begin{pmatrix} | & \cdots & | \\ t_1 & \cdots & t_N \\ | & \cdots & | \end{pmatrix} \tag{1.5}$$

We will assume the error function is of the form $J = J(T, A^{(L)})$, taking on real values. I.e. it only depends on the targets and the activations of the output layer.

2 Backpropagation

Forward propagation through the network is described by equation 1.1. The procedure is assumed to be done before we look at how to determine partial derivatives of J through backpropagation.

2.1 Output layer

The derivatives with respect to the output layer weights and biases will take a rather abstract form in this general formalism:

$$\frac{\partial J}{\partial W_{ij}^{(L)}} = \sum_{k=1}^{S_L} \sum_{n=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}} \frac{\partial A_{kn}^{(L)}}{\partial W_{ij}^{(L)}}, \quad \frac{\partial J}{\partial b_i^{(L)}} = \sum_{k=1}^{S_L} \sum_{n=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}} \frac{\partial A_{kn}^{(L)}}{\partial b_i^{(L)}} \tag{2.1}$$

We may find the derivatives of $A^{(L)}$ with respect to the weights and biases:

$$\frac{\partial A_{kn}^{(L)}}{\partial W_{ij}^{(L)}} = \frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}} \frac{\partial Z_{kn}^{(L)}}{\partial W_{ij}^{(L)}} \tag{2.2}$$

Finally, we may calculate the derivatives of $Z^{(L)}$ with respect to the weights:

$$\frac{\partial Z_{kn}^{(L)}}{\partial W_{ij}^{(L)}} = \frac{\partial}{\partial W_{ij}^{(L)}} (W^{(L)} A^{(L-1)} + b^{(L)})_{kn} = \frac{\partial}{\partial W_{ij}^{(L)}} \left(\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)} A_{ln}^{(L-1)} + b_k^{(L)} \right) \tag{2.3}$$

Differentiating $W^{(L)}$ with respect to $W^{(L)}$ yields two Kronecker deltas:

$$\frac{\partial Z_{kn}^{(L)}}{\partial W_{ij}^{(L)}} = \sum_{l=1}^{S_{L-1}} \delta_{ik} \delta_{jl} A_{ln}^{(L-1)} = \delta_{ik} A_{jn}^{(L-1)} \tag{2.4}$$

Now, we may insert equations 2.2 and 2.4 into 2.1:

$$\frac{\partial J}{\partial W_{ij}^{(L)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}} \frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}} \delta_{ik} A_{jn}^{(L-1)} = \sum_{n=1}^{N} \frac{\partial J}{\partial A_{in}^{(L)}} \frac{\partial \sigma_L}{\partial Z_{in}^{(L)}} A_{jn}^{(L-1)} \tag{2.5}$$

We can rewrite this using the Hadamard product between the two derivatives and swapping the indices of $A^{(L-1)}$, turning into a transpose:

$$\sum_{n=1}^{N} \left[\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right]_{in} \left(A^{(L-1)}\right)^t_{nj} \tag{2.6}$$

Or in matrix form:

$$\frac{\partial J}{\partial W^{(L)}} = \left[\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] \left(A^{(L-1)}\right)^t = H^{(L)} \left(A^{(L-1)}\right)^t \tag{2.7}$$

Here we've introduced $H^{(L)}$, the matrix of the Hadamard product, for notational ease:

$$H^{(L)} = \frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \tag{2.8}$$

The procedure for the biases is the same until we get to the $Z^{(L)}$ derivative:

$$\frac{\partial Z_{kn}^{(L)}}{\partial b_i^{(L)}} = \frac{\partial}{\partial b_i^{(L)}} \left(\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)} A_{ln}^{(L-1)} + b_k^{(L)}\right) = \delta_{ik} \tag{2.9}$$

Reinserting all the way back to equation 2.1 we get:

$$\frac{\partial J}{\partial b_i^{(L)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}} \frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}} \delta_{ik} = \sum_{n=1}^{N} \frac{\partial J}{\partial A_{in}^{(L)}} \frac{\partial \sigma_L}{\partial Z_{in}^{(L)}} \tag{2.10}$$

Again, we can write this is matrix notation using the Hadamard product:

$$\frac{\partial J}{\partial b^{(L)}} = \left[\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] J_N^t = H^{(L)} J_N^t \tag{2.11}$$

2.2 Output layer "error"

We will call the quantity $\frac{\partial J}{\partial A^{(L)}}$ the output layer "error":

$$\Delta^{(L)} = \frac{\partial J}{\partial A^{(L)}} \tag{2.12}$$

The quotes are used, because there's no need for this to be equal/proportional to what we usually call errors, i.e. distance between the output $A^{(L)}$ and T.

However, often this is the case (or rather, J is specifically chosen to make this so - see below). At any rate, we may now write:

$$
\begin{aligned}
H^{(L)} =& \Delta^{(L)} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} && (2.13) \\
\frac{\partial J}{\partial W^{(L)}} =& H^{(L)} \left(A^{(L-1)}\right)^t = \left[\Delta^{(L)} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] \left(A^{(L-1)}\right)^t && (2.14) \\
\frac{\partial J}{\partial b^{(L)}} =& H^{(L)} J_N^t = \left[\Delta^{(L)} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] J_N^t && (2.15)
\end{aligned}
$$

2.2.1 Example error function: Euclidean distance

A common form of error function is:

$$J(T, A^{(L)}) = \frac{1}{2N} \sum_{m=1}^{N} ||a_m^{(L)} - t_m||^2 \tag{2.16}$$

Here, $a_m^{(L)}$ is the m'th column of $A^{(L)}$ and the double dashes is the usual Euclidean norm in S_L dimensions. So we may rewrite:

$$J(T, A^{(L)}) = \frac{1}{2N} \sum_{m=1}^{N} \sum_{j=1}^{S_L} (A_{jm}^{(L)} - T_{jm})^2 \tag{2.17}$$

Now we can find the derivative:

$$\Delta_{in}^{(L)} = \frac{\partial J}{\partial A_{in}^{(L)}} = \frac{1}{2N} \sum_{m=1}^{N} \sum_{j=1}^{S_L} \frac{\partial}{\partial A_{in}^{(L)}} (A_{jm}^{(L)} - T_{jm})^2 \tag{2.18}$$

Again, we get two Kronecker deltas:

$$\frac{1}{2N} \sum_{m=1}^{N} \sum_{j=1}^{S_L} 2(A_{jm}^{(L)} - T_{jm}) \delta_{ij} \delta_{nm} = \frac{1}{N} (A_{in}^{(L)} - T_{in}) \tag{2.19}$$

The quantity in parenthesis is the ij element of the error term matrix for the output layer. So indeed we get:

$$\Delta^{(L)} = \frac{1}{N} \left(A_{in}^{(L)} - T_{in}\right) \tag{2.20}$$

Or in matrix form:

$$\Delta^{(L)} = \frac{1}{N} \left(A^{(L)} - T\right) \tag{2.21}$$

2.3 Last hidden layer

2.3.1 Weights

Now, let's consider derivatives with respect to weights in layer $L-1$. Simply applying the chain rule, we get:

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{k=1}^{S_L} \sum_{n=1}^{N} \sum_{k'=1}^{S_{L-1}} \sum_{n'=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}} \frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}} \frac{\partial Z_{kn}^{(L)}}{\partial A_{k'n'}^{(L-1)}} \frac{\partial \sigma_{L-1}}{\partial Z_{k'n'}^{(L-1)}} \frac{\partial Z_{k'n'}^{(L-1)}}{\partial W_{ij}^{(L-1)}} \tag{2.22}$$

Now, this will obviously be very similar to the calculations above, but to be certain, let's proceed carefully. The only term in the above we have not calculated yet is:

$$\frac{\partial Z_{kn}^{(L)}}{\partial A_{k'n'}^{(L-1)}} = \frac{\partial}{\partial A_{k'n'}^{(L-1)}} \left(W^{(L)} A^{(L-1)} + b^{(L)}\right)_{kn} = \tag{2.23}$$

$$\frac{\partial}{\partial A_{k'n'}^{(L-1)}} \left(\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)} A_{ln}^{(L-1)} + b_k^{(L)} \right) = \tag{2.24}$$

$$\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)} \delta_{lk'} \delta_{nn'} = W_{kk'}^{(L)} \delta_{nn'} \tag{2.25}$$

Now we're ready to insert into equation 2.22:

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{k=1}^{S_L} \sum_{n=1}^{N} \sum_{k'=1}^{S_{L-1}} \sum_{n'=1}^{N} \Delta_{kn}^{(L)} \frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}} W_{kk'}^{(L)} \delta_{nn'} \frac{\partial \sigma_{L-1}}{\partial Z_{k'n'}^{(L-1)}} \delta_{ik'} A_{jn'}^{(L-2)} = \tag{2.26}$$

$$\sum_{k=1}^{S_L} \sum_{n=1}^{N} H_{kn}^{(L)} W_{ki}^{(L)} \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}} A_{jn}^{(L-2)} \tag{2.27}$$

Use the trick of rearranging terms and swapping indices:

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{k=1}^{S_L} \sum_{n=1}^{N} \left(W^{(L)}\right)_{ik}^{t} H_{kn}^{(L)} \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}} \left(A^{(L-2)}\right)_{nj}^{t} = \tag{2.28}$$

$$\sum_{n=1}^{N} \left[\left(W^{(L)}\right)^t H^{(L)}\right]_{in} \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}} \left(A^{(L-2)}\right)_{nj}^{t} \tag{2.29}$$

Once again, we can collect the first two terms into a Hadamard product:

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{n=1}^{N} \left[\left(W^{(L)}\right)^t H^{(L)} \odot \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}}\right]_{in} \left(A^{(L-2)}\right)_{nj}^{t} = \tag{2.30}$$

$$\left[\left(W^{(L)}\right)^t H^{(L)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}} \left(A^{(L-2)}\right)^t\right]_{ij} \tag{2.31}$$

Or in matrix form:

$$\frac{\partial J}{\partial W^{(L-1)}} = \left[\underbrace{\left(W^{(L)}\right)^t H^{(L)}}_{\Delta^{(L-1)}} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}} \right] \left(A^{(L-2)}\right)^t \tag{2.32}$$

Here, we've defined the underbraced part to be the "error" for layer $L - 1$. The formula now takes a form very similar to equation 2.13:

$$\frac{\partial J}{\partial W^{(L-1)}} = \left[\Delta^{(L-1)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}} \right] \left(A^{(L-2)}\right)^t = H^{(L-1)} \left(A^{(L-2)}\right)^t \tag{2.33}$$

Here, we've defined the H for layer $L - 1$ as:

$$H^{(L-1)} = \Delta^{(L-1)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}} \tag{2.34}$$

2.3.2 Biases

This will be very similar to the weights case. The only thing that changes, is that the last term in the chain rule decomposition is:

$$\frac{\partial Z_{k'n'}^{(L-1)}}{\partial b_i^{(L-1)}} = \delta_{ik'} \tag{2.35}$$

So all of the calculations play out the same way as above, except there's no multiplication by $A^{(L-2)}$. Instead we get:

$$\frac{\partial J}{\partial b_i^{(L-1)}} = \sum_{n=1}^{N} \left[\left(W^{(L)}\right)^t H^{(L)} \odot \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}} \right]_{in} = \sum_{n=1}^{N} H_{in}^{(L-1)} \tag{2.36}$$

Or in matrix form:

$$\frac{\partial J}{\partial b^{(L-1)}} = H^{(L-1)} J_N^t \tag{2.37}$$