# General feed forward neural networks

Kristian Wichmann

August 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 $\sigma_l$.

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 a cost 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 cost 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. We will make further assumptions about $J$ later.

# 2 Backpropagation - Output layer

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 Weights

The derivatives with respect to the output layer weights and biases can be found through the chain rule:

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

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

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

But since $\frac{\partial A_{kn}^{(L)}}{\partial \sigma_L}$ is simply one, this reduces to:

$$\frac{\partial A_{kn}^{(L)}}{\partial W_{ij}^{(L)}} = \frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}} \frac{\partial Z_{kn}^{(L)}}{\partial W_{ij}^{(L)}} \tag{2.3}$$

This also means that when writing the $J$-derivative in matrix form, there will be a Hadamard product between the first two terms instead of ordinary matrix multiplication:

$$\frac{\partial J}{\partial W^{(L)}} = \frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \frac{\partial Z^{(L)}}{\partial W^{(L)}} \tag{2.4}$$

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

$$\frac{\partial Z_{kn}^{(L)}}{\partial W_{ij}^{(L)}} = \frac{\partial}{\partial W_{ij}^{(L)}}(W^{(L)}A^{(L-1)} + b^{(L)})_{kn} = \frac{\partial}{\partial W_{ij}^{(L)}}\left(\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)}A_{ln}^{(L-1)} + b_k^{(L)}\right) \tag{2.5}$$

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

$$\frac{\partial Z_{kn}^{(L)}}{\partial W_{ij}^{(L)}} = \sum_{l=1}^{S_{L-1}} \delta_{ik}\delta_{jl}A_{ln}^{(L-1)} = \delta_{ik}A_{jn}^{(L-1)} \tag{2.6}$$

Now, we may insert equations 2.3 and 2.6 into 2.1:

$$\frac{\partial J}{\partial W_{ij}^{(L)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}}\frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}}\delta_{ik}A_{jn}^{(L-1)} = \sum_{n=1}^{N} \frac{\partial J}{\partial A_{in}^{(L)}}\frac{\partial \sigma_L}{\partial Z_{in}^{(L)}}A_{jn}^{(L-1)} \tag{2.7}$$

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

$$\sum_{n=1}^{N}\left[\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right]_{in} \left(A^{(L-1)}\right)_{nj}^{t} \tag{2.8}$$

Or in matrix form:

$$\frac{\partial J}{\partial W^{(L)}} = \left[\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] \left(A^{(L-1)}\right)^{t} = H^{(L)}\left(A^{(L-1)}\right)^{t} \tag{2.9}$$

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

$$H^{(L)} = \frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \tag{2.10}$$

## 2.2 Biases

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

$$\frac{\partial Z_{kn}^{(L)}}{\partial b_i^{(L)}} = \frac{\partial}{\partial b_i^{(L)}}\left(\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)}A_{ln}^{(L-1)} + b_k^{(L)}\right) = \delta_{ik} \tag{2.11}$$

Reinserting all the way back to equation 2.1 we get:

$$\frac{\partial J}{\partial b_i^{(L)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N} \frac{\partial J}{\partial A_{kn}^{(L)}}\frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}}\delta_{ik} = \sum_{n=1}^{N} \frac{\partial J}{\partial A_{in}^{(L)}}\frac{\partial \sigma_L}{\partial Z_{in}^{(L)}} \tag{2.12}$$

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

$$\frac{\partial J}{\partial b^{(L)}} = \left[\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] J_N^t = H^{(L)}J_N^t \tag{2.13}$$

## 2.3 Output layer "error"

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

$$\Delta^{(L)} = \frac{\partial J}{\partial A^{(L)}} \tag{2.14}$$

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 $\Delta$, $H$, or $\delta$ defined below equal/proportional to it - see below). At any rate, we may now write:

$$H^{(L)} = \Delta^{(L)} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \tag{2.15}$$

$$\frac{\partial J}{\partial W^{(L)}} = H^{(L)} \left(A^{(L-1)}\right)^t = \left[\Delta^{(L)} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] \left(A^{(L-1)}\right)^t \tag{2.16}$$

$$\frac{\partial J}{\partial b^{(L)}} = H^{(L)} J_N^t = \left[\Delta^{(L)} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}}\right] J_N^t \tag{2.17}$$

# 3 The cost function

Let's take a closer look at the cost function. Usually, it can be written as an average of a *loss function* $\mathcal{L}(t, a)$, where $t$ and $a$ are the desired and actual activations for the output layer. Then the cost function can be written:

$$J(T, A^{(L)}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(t_i, a_i^{(L)}) \tag{3.1}$$

The "error" term is now:

$$\Delta^{(L)} = \frac{\partial J}{\partial A^{(L)}} = \frac{1}{m} \sum_{i=1}^{N} \frac{\partial}{\partial A^{(L)}} \mathcal{L}(t_i, a_i^{(L)}) \tag{3.2}$$

The derivative is only non-zero when differentiating with respect to $a_i^{(L)}$, so this is the same as:

$$\Delta^{(L)} = \frac{1}{m} \sum_{i=1}^{N} \underbrace{\frac{\partial}{\partial a_i^{(L)}} \mathcal{L}(t_i, a_i^{(L)})}_{\delta_i^{(L)}} \tag{3.3}$$

Here, we've defined the error function for specific a data point $\delta_i^{(L)}$.

## 3.1 Example: Euclidean distance cost function

A common form of cost function is:

$$J(T, A^{(L)}) = \frac{1}{2N} \sum_{m=1}^{N} ||a_m^{(L)} - t_m||^2 \tag{3.4}$$

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 with the notation above:

$$\mathcal{L}(t, a^{(L)}) = \frac{1}{2}||a^{(L)} - t||^2 \tag{3.5}$$

The error for a data point is:

$$\delta_i^{(L)} = \frac{1}{2} \frac{\partial}{\partial a_i^{(L)}} ||a_i^{(L)} - t_i||^2 = a_i^{(L)} - t_i \tag{3.6}$$

This conforms with the usual notion of error for a data point. The total "error" term becomes:

$$\Delta^{(L)} = \frac{1}{N} \left( A^{(L)} - T \right) \tag{3.7}$$

## 3.2 Example: Cross-entropy cost function with logistic sigmoid activation function

Here the cost function is:

$$J(T, A^{(L)}) = -\frac{1}{N} \sum_{m=1}^{N} \left[ t_i \log a_i^{(L)} + (1 - t_i) \log(1 - a_i^{(L)}) \right] \tag{3.8}$$

Or in other words:

$$\mathcal{L}(t, a^{(L)}) = - \left[ t \log a^{(L)} + (1 - t) \log(1 - a^{(L)}) \right] \tag{3.9}$$

The data point error is:

$$\delta_i^{(L)} = - \left[ \frac{t_i}{a_i^{(L)}} - \frac{1 - t_i}{1 - a_i^{(L)}} \right] = \frac{-t_i(1 - a_i^{(L)}) + (1 - t_i) a_i^{(L)}}{a_i^{(L)}(1 - a_i^{(L)})} = \frac{a_i^{(L)} - t_i}{a_i^{(L)}(1 - a_i^{(L)})} \tag{3.10}$$

This looks like what we usually think of as the error, except for the denominator. However, we also see that the same denominator is equal to the derivative of a logistic sigmoid function. Hence we will re-find the familiar from when we get to the Hadamard product. Let's start with $\Delta$:

$$\Delta^{(L)} = \frac{1}{N}\sum_{i=1}^{N}\frac{a_i^{(L)} - t_i}{a_i^{(L)}(1 - a_i^{(L)})} \tag{3.11}$$

As noted above, for $H$ the denominator cancels and we get:

$$H^{(L)} = \frac{1}{N}(A^{(L)} - T) \tag{3.12}$$

# 4 Backpropagation - Last hidden layer

## 4.1 Weights

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

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N}\sum_{k'=1}^{S_{L-1}}\sum_{n'=1}^{N}\frac{\partial J}{\partial A_{kn}^{(L)}}\frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}}\frac{\partial Z_{kn}^{(L)}}{\partial A_{k'n'}^{(L-1)}}\frac{\partial \sigma_{L-1}}{\partial Z_{k'n'}^{(L-1)}}\frac{\partial Z_{k'n'}^{(L-1)}}{\partial W_{ij}^{(L-1)}} \tag{4.1}$$

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:

$$\frac{\partial Z_{kn}^{(L)}}{\partial A_{k'n'}^{(L-1)}} = \frac{\partial}{\partial A_{k'n'}^{(L-1)}}\left(W^{(L)}A^{(L-1)} + b^{(L)}\right)_{kn} = \tag{4.2}$$

$$\frac{\partial}{\partial A_{k'n'}^{(L-1)}}\left(\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)}A_{ln}^{(L-1)} + b_k^{(L)}\right) = \tag{4.3}$$

$$\sum_{l=1}^{S_{L-1}} W_{kl}^{(L)}\delta_{lk'}\delta_{nn'} = W_{kk'}^{(L)}\delta_{nn'} \tag{4.4}$$

Now we're ready to insert into equation 4.1:

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N}\sum_{k'=1}^{S_{L-1}}\sum_{n'=1}^{N}\Delta_{kn}^{(L)}\frac{\partial \sigma_L}{\partial Z_{kn}^{(L)}}W_{kk'}^{(L)}\delta_{nn'}\frac{\partial \sigma_{L-1}}{\partial Z_{k'n'}^{(L-1)}}\delta_{ik'}A_{jn'}^{(L-2)} = \tag{4.5}$$

$$\sum_{k=1}^{S_L}\sum_{n=1}^{N} H_{kn}^{(L)}W_{ki}^{(L)}\frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}}A_{jn}^{(L-2)} \tag{4.6}$$

Use the trick of rearranging terms and swapping indices:

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{k=1}^{S_L}\sum_{n=1}^{N} \left(W^{(L)}\right)_{ik}^{t} H_{kn}^{(L)} \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}} \left(A^{(L-2)}\right)_{nj}^{t} = \tag{4.7}$$

$$\sum_{n=1}^{N} \left[\left(W^{(L)}\right)^{t} H^{(L)}\right]_{in} \frac{\partial \sigma_{L-1}}{\partial Z_{in}^{(L-1)}} \left(A^{(L-2)}\right)_{nj}^{t} \tag{4.8}$$

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

$$\frac{\partial J}{\partial W_{ij}^{(L-1)}} = \sum_{n=1}^{N} \left[\left(W^{(L)}\right)^{t} H^{(L)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}}\right]_{in} \left(A^{(L-2)}\right)_{nj}^{t} = \tag{4.9}$$

$$\left[\left(W^{(L)}\right)^{t} H^{(L)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}} \left(A^{(L-2)}\right)^{t}\right]_{ij} \tag{4.10}$$

Or in matrix form:

$$\frac{\partial J}{\partial W^{(L-1)}} = \left[\underbrace{\left(W^{(L)}\right)^{t} H^{(L)}}_{\Delta^{(L-1)}} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}}\right] \left(A^{(L-2)}\right)^{t} \tag{4.11}$$

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 2.15:

$$\frac{\partial J}{\partial W^{(L-1)}} = \left[\Delta^{(L-1)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}}\right] \left(A^{(L-2)}\right)^{t} = H^{(L-1)} \left(A^{(L-2)}\right)^{t} \tag{4.12}$$

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

$$H^{(L-1)} = \Delta^{(L-1)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}} \tag{4.13}$$

## 4.2 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:

$$\frac{\partial Z_{k'n'}^{(L-1)}}{\partial b_i^{(L-1)}} = \delta_{ik'} \tag{4.14}$$

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

$$\frac{\partial J}{\partial b_i^{(L-1)}} = \sum_{n=1}^{N} \left[\left(W^{(L)}\right)^{t} H^{(L)} \odot \frac{\partial \sigma_{L-1}}{\partial Z^{(L-1)}}\right]_{in} = \sum_{n=1}^{N} H_{in}^{(L-1)} \tag{4.15}$$

Or in matrix form:

$$\frac{\partial J}{\partial b^{(L-1)}} = H^{(L-1)} J_N^{t} \tag{4.16}$$

# 5 Backpropagation - General layer

Here, we wish to prove that in general, the formula for derivatives of $J$ with respect to weights and biases from any layer $l$ can be written:

$$\frac{\partial J}{\partial W^{(l)}} = H^{(l)} \left(A^{(l-1)}\right)^t, \quad \frac{\partial J}{\partial b^{(l)}} = H^{(l)} J_N^t \tag{5.1}$$

Here, $H^{(l)}$ is defined recursively:

$$H^{(l)} = \Delta^{(l)} \odot \frac{\partial \sigma_l}{\partial Z^{(l)}}, \quad \Delta^{(l)} = \left(W^{(l+1)}\right)^t H^{(l+1)}, \quad \Delta^{(L)} = \frac{\partial J}{\partial A^{(L)}} \tag{5.2}$$

## 5.1 A useful lemma

To show this it is turns out to be useful to start by proving the following:

$$\frac{\partial J}{\partial A^{(l)}} = \Delta^{(l)} \tag{5.3}$$

This is done through induction, although backwards from $l = L$ down to $l = 1$.

### 5.1.1 Induction start

This is corresponds to $l = L$. Here, this is true by definition:

$$\frac{\partial J}{\partial A^{(L)}} = \Delta^{(L)} \tag{5.4}$$

### 5.1.2 Induction step

So we need to prove $(l) \Rightarrow (l-1)$. Notice the following:

$$\frac{\partial J}{\partial A^{(l)}} = \frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \odot \cdots \odot \frac{\partial \sigma_{l+1}}{\partial Z^{(l+1)}} \frac{\partial Z^{(l+1)}}{\partial A^{(l)}} \tag{5.5}$$

$$\frac{\partial J}{\partial A^{(l-1)}} = \underbrace{\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \odot \cdots \odot \frac{\partial \sigma_{l+1}}{\partial Z^{(l+1)}} \frac{\partial Z^{(l+1)}}{\partial A^{(l)}}}_{\frac{\partial J}{\partial A^{(l)}}} \odot \frac{\partial \sigma_l}{\partial Z^{(l)}} \frac{\partial Z^{(l)}}{\partial A^{(l-1)}} \tag{5.6}$$

The Hadamard products follow from the same logic that led to equation 2.4. By the induction assumption, $\frac{\partial J}{\partial A^{(l)}} = \Delta^{(l)}$, so we need to calculate:

$$\frac{\partial J}{\partial A^{(l-1)}} = \Delta^{(l)} \odot \frac{\partial \sigma_l}{\partial Z^{(l)}} \frac{\partial Z^{(l)}}{\partial A^{(l-1)}} \tag{5.7}$$

In coordinate form:

$$\left[\frac{\partial J}{\partial A^{(l-1)}}\right]_{in} = \sum_{j=1}^{S_l}\sum_{n'=1}^{N}\left(\Delta_{jn'}^{(l)}\frac{\partial \sigma_l}{\partial Z_{jn'}^{(l)}}\right)\frac{\partial Z_{jn'}^{(l)}}{\partial A_{in}^{(l-1)}} \tag{5.8}$$

But we've already calculated the last derivative:

$$\sum_{j=1}^{S_l}\sum_{n'=1}^{N}\left(\Delta_{jn'}^{(l)}\frac{\partial \sigma_l}{\partial Z_{jn'}^{(l)}}\right)W_{ji}^{(l)}\delta_{nn'} = \sum_{j=1}^{S_l}\left(\Delta_{jn}^{(l)}\frac{\partial \sigma_l}{\partial Z_{jn}^{(l)}}\right)W_{ji}^{(l)} \tag{5.9}$$

Now use the usual trick of swapping indices:

$$\left[\frac{\partial J}{\partial A^{(l-1)}}\right]_{in} = \sum_{j=1}^{S_l}\left(W^{(l)}\right)_{ij}^{t}\left(\Delta_{jn}^{(l)}\frac{\partial \sigma_l}{\partial Z_{jn}^{(l)}}\right) = \left[\left(W^{(l)}\right)^{t}H^{(l)}\right]_{ij} \tag{5.10}$$

This is exactly the $ij$'th element of $\Delta^{(l-1)}$, as desired.

## 5.2 Weights

The lemma makes it easy to derive the formula for weights in the $l$'th layer:

$$\frac{\partial J}{\partial W^{(l)}} = \underbrace{\frac{\partial J}{\partial A^{(L)}}\odot\frac{\partial \sigma_L}{\partial Z^{(L)}}\frac{\partial Z^{(L)}}{\partial A^{(L-1)}}\odot\cdots\odot\frac{\partial \sigma_{l+1}}{\partial Z^{(l+1)}}\frac{\partial Z^{(l+1)}}{\partial A^{(l)}}}_{\frac{\partial J}{\partial A^{(l)}}}\odot\frac{\partial \sigma_l}{\partial Z^{(l)}}\frac{\partial Z^{(l)}}{\partial W^{(l)}} = \tag{5.11}$$

$$\Delta^{(l)}\odot\frac{\partial \sigma_l}{\partial Z^{(l)}}\frac{\partial Z^{(l)}}{\partial W^{(l)}} \tag{5.12}$$

Element-wise, using all the (by now) usual tricks:

$$\frac{\partial J}{\partial W_{ij}^{(l)}} = \sum_{k=1}^{S_l}\sum_{n=1}^{N}\Delta_{kn}^{(l)}\frac{\partial \sigma_l}{\partial Z_{kn}^{(l)}}\frac{\partial Z_{kn}^{(l)}}{\partial W_{ij}^{(l)}} = \sum_{k=1}^{S_l}\sum_{n=1}^{N}\Delta_{kn}^{(l)}\frac{\partial \sigma_l}{\partial Z_{kn}^{(l)}}A_{jn}^{(l)}\delta_{ki} = \tag{5.13}$$

$$\sum_{n=1}^{N}\Delta_{in}^{(l)}\frac{\partial \sigma_l}{\partial Z_{in}^{(l)}}A_{jn}^{(l)} = \sum_{n=1}^{N}\Delta_{in}^{(l)}\frac{\partial \sigma_l}{\partial Z_{in}^{(l)}}\left(A^{(l)}\right)_{nj}^{t} = \tag{5.14}$$

$$\sum_{n=1}^{N}\left[\Delta^{(l)}\odot\frac{\partial \sigma_l}{\partial Z^{(l)}}\right]_{in}\left(A^{(l)}\right)_{nj}^{t} \tag{5.15}$$

Back in matrix form:

$$\frac{\partial J}{\partial W^{(l)}} = H^{(l)}\left(A^{(l)}\right)^{t} \tag{5.16}$$

### 5.3 Biases

This is almost the same as for the weights:

$$\frac{\partial J}{\partial b^{(l)}} = \underbrace{\frac{\partial J}{\partial A^{(L)}} \odot \frac{\partial \sigma_L}{\partial Z^{(L)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \odot \cdots \odot \frac{\partial \sigma_{l+1}}{\partial Z^{(l+1)}} \frac{\partial Z^{(l+1)}}{\partial A^{(l)}}}_{\frac{\partial J}{\partial A^{(l)}}} \odot \frac{\partial \sigma_l}{\partial Z^{(l)}} \frac{\partial Z^{(l)}}{\partial b^{(l)}} = \tag{5.17}$$

$$\Delta^{(l)} \odot \frac{\partial \sigma_l}{\partial Z^{(l)}} \frac{\partial Z^{(l)}}{\partial b^{(l)}} \tag{5.18}$$

Elementwise:

$$\left[\frac{\partial J}{\partial b^{(l)}}\right]_i = \sum_{j=1}^{S_l} \sum_{n=1}^{N} \Delta_{jn}^{(l)} \frac{\partial \sigma_l}{\partial Z_{jn}^{(l)}} \delta_{ji} = \sum_{n=1}^{N} \Delta_{in}^{(l)} \frac{\partial \sigma_l}{\partial Z_{in}^{(l)}} = \sum_{n=1}^{N} H_{in}^{(l)} \tag{5.19}$$

In matrix form:

$$\frac{\partial J}{\partial b^{(l)}} = H^{(l)} J_N^t \tag{5.20}$$

# 6 Example: Two-layer tanh-network for binary classification

This example is from week 3 of the Coursera course "Neural Networks and Deep Learning" by Andrew Ng. The input layer has 2 dimensions, and the output 1. Since it is a binary classification model, the output activation function is sigmoid, while the hidden layer is tanh:

$$\sigma_1(z) = \tanh(z), \quad \sigma_2(z) = \sigma(z) \tag{6.1}$$

The training set consists of 400 data points, so $X =\in \mathbb{R}^{2\times 400}$. The labels $T$ are called $Y$, and are one-hot encodings of the true condition, so $T = Y \in \mathbb{R}^{1\times 400}$. The hidden layer has $n_h$ layers, so the dimensions of the weights and biases are:

$$W^{(1)} \in \mathbb{R}^{n_h\times 2}, \quad b^{(1)} \in \mathbb{R}^{n_h\times 1}, \quad W^{(2)} \in \mathbb{R}^{1\times n_h}, \quad b^{(2)} \in \mathbb{R}^{1\times 1} \tag{6.2}$$

The loss function is the average cross-entropy for the training set. This means that we're in the same case as in section 3.2. Therefore the Hadamard term for the output layer is:

$$H^{(2)} = \frac{1}{m}(A^{(2)} - Y) \tag{6.3}$$

So, the $W^{(2)}$-derivative is:

$$\frac{\partial J}{\partial W^{(2)}} = H^{(2)} A^{(1)} = \frac{1}{m} \sum_{i=1}^{m} \left( A_i^{(2)} - Y_i \right) (A_i^{(1)})^t \tag{6.4}$$

And the $b^{(2)}$-derivative:

$$\frac{\partial J}{\partial b^{(2)}} = H^{(2)} A^{(1)} = \frac{1}{m} \sum_{i=1}^{m} \left( A_i^{(2)} - Y_i \right) \tag{6.5}$$

Now, we backpropagate the error term:

$$\Delta^{(1)} = (W^{(2)})^t H^{(2)} = \frac{1}{m} (W^{(2)})^t (A^{(2)} - Y) \tag{6.6}$$

And using the derivative of tanh, the Hadamard term is:

$$H_i^{(1)} = \Delta_i^{(1)} (1 - (a_i^{(1)})^2) \tag{6.7}$$

Now the derivatives of $W^{(1)}$ and $b^{(1)}$ follow the usual formulas:

$$\frac{\partial J}{\partial W^{(1)}} = H^{(1)} (A^{(1)})^t, \quad \frac{\partial J}{\partial b^{(1)}} = H^{(1)} (J_2)^t \tag{6.8}$$