Backpropagation

Gregor Hartl Watters

February 2024

1 Single-Hidden-Layer Neural Network

The input to the network is $\mathbf{x}$, which goes through a matrix multiplication, followed by an activation and then another matrix multiplication to produce the output, $\mathbf{f}$,

$$\mathbf{x} \to \mathbf{g} = V\mathbf{x} \to \mathbf{h} = \sigma(\mathbf{g}) \to \mathbf{f} = W\mathbf{h}, \tag{1}$$

where

$$\mathbf{x} \in \mathbb{R}^D, \tag{2}$$

$$\mathbf{g}, \mathbf{h} \in \mathbb{R}^H, \tag{3}$$

$$\mathbf{y}, \mathbf{f} \in \mathbb{R}^F, \tag{4}$$

$$V \in \mathbb{R}^{H \times D}, \tag{5}$$

and

$$W \in \mathbb{R}^{F \times H}. \tag{6}$$

Note that the input vector $\mathbf{x}$ has a 1 appended as an extra dimension and that all matrices of weights have an extra column appended with bias terms $b_1, b_2, ..., b_N$ (these are already assumed to be included within the above dimensions).

1.1 Finding the Rate of Change of the Loss w.r.t. the Second Set of Weights, $\frac{\partial L}{\partial W}$

The loss function is

$$L = |\mathbf{y} - \mathbf{f}|^2, \tag{7}$$

which can be rewritten as a summation,

$$L = \sum_{i=1}^{F} (y_i - f_i)^2. \tag{8}$$

Thus, the rate of change of loss function with respect to an output f_i is given as

$$\frac{\partial L}{\partial f_i} = 2(f_i - y_i), \tag{9}$$

which can be written in vector form (as a column vector),

$$\frac{\partial L}{\partial \mathbf{f}} = 2(\mathbf{f} - \mathbf{y}). \tag{10}$$

Now, the rate of change of the loss with respect to the output of the hidden layer will be given by

$$\frac{\partial L}{\partial \mathbf{h}} = \left(\left(\frac{\partial L}{\partial \mathbf{f}} \right)^T \frac{\partial \mathbf{f}}{\partial \mathbf{h}} \right)^T, \tag{11}$$

where the transposes arise from wishing to express $\frac{\partial L}{\partial \mathbf{h}}$ as a column vector (and since $\frac{\partial L}{\partial \mathbf{f}}$ is also a column vector). For a single component h_j,

$$\frac{\partial L}{\partial h_j} = \frac{\partial L}{\partial \mathbf{f}} \cdot \frac{\partial \mathbf{f}}{\partial h_j} = \sum_{i=1}^{F} \frac{\partial L}{\partial f_i} \frac{\partial f_i}{\partial h_j}. \tag{12}$$

The rate of change of a single component in the output vector f_i with respect to a single component in the output of the hidden layer h_j is simply the weight in the matrix which is multiplied by the component in the hidden layer, since

$$f_i = \sum_{j=1}^{H} W_{ij} h_j, \tag{13}$$

so

$$\frac{\partial f_i}{\partial h_j} = W_{ij}, \tag{14}$$

which means

$$\frac{\partial \mathbf{f}}{\partial \mathbf{h}} = W. \tag{15}$$

Therefore,

$$\frac{\partial L}{\partial h_j} = \sum_{i=1}^{F} 2(f_i - y_i) W_{ij}, \tag{16}$$

which can be used to write the entire equation in vector form,

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{h}} &= \left(\left(\frac{\partial L}{\partial \mathbf{f}} \right)^T W \right)^T \\ &= 2W^T(\mathbf{f} - \mathbf{y}) \end{aligned} \tag{17}$$

Now I will derive $\frac{\partial L}{\partial W}$ itself using all of the above. I start with noting the rate of change of the loss with respect to a single component of W,

$$\frac{\partial L}{\partial W_{kl}} = \sum_{i=1}^{F} \frac{\partial L}{\partial f_i} \frac{\partial f_i}{\partial W_{kl}}. \tag{18}$$

Equation (18) is formally the case because

$$\begin{aligned} L &= L(\mathbf{f}) = L(f_1, f_2, \dots, f_F) = L(f_1(W), f_2(W), \dots, f_F(W)) \\ &= L(f_1(W_{11}, W_{12}, \dots, W_{FH}), f_2(W_{11}, W_{12}, \dots, W_{FH}), \dots, f_F(W_{11}, W_{12}, \dots, W_{FH})). \end{aligned} \tag{19}$$

However, the sum in eq. (18) can be simplified by noting that

$$f_i = \sum_{j=1}^{H} W_{ij} h_j, \tag{20}$$

i.e., that f_i is actually only a function of the weights in W that are in row i, $f_i = f_i(W_{i1}, W_{i2}, \dots, W_{iH})$, so that the derivative of a single component with respect to a given weight W_{kl} becomes

$$\frac{\partial f_i}{\partial W_{kl}} = \delta_{ik} \sum_{j=1}^{H} \delta_{lj} h_j = \delta_{ik} h_l. \tag{21}$$

Thus, this can be plugged into eq. (18) to produce

$$\begin{aligned} \frac{\partial L}{\partial W_{kl}} &= \sum_{i=1}^{F} 2(f_i - y_i)\delta_{ik} h_l = 2h_l \sum_{i=1}^{F} \delta_{ik}(f_i - y_i) \\ &= 2(f_k - y_k)h_l. \end{aligned} \tag{22}$$

Finally, it is simple to note that the above can be rewritten as an outer product,

$$\frac{\partial L}{\partial W} = 2(\mathbf{f} - \mathbf{y})\mathbf{h}^T. \tag{23}$$

At this point, using eq. (23), it would already be possible to optimise the second set of weights W to minimise the prediction error of the NN.

1.2 Finding the Rate of Change of the Loss w.r.t. the First Set of Weights, $\frac{\partial L}{\partial V}$

In eq. (16), I already derived $\frac{\partial L}{\partial h_j}$, which can be used to find the rate of change of the loss w.r.t. a component of the input to the hidden layer, g_k,

$$\frac{\partial L}{\partial g_k} = \sum_{j=1}^{H} \frac{\partial L}{\partial h_j} \frac{\partial h_j}{\partial g_k}. \tag{24}$$

Again, the above summation can be simplified enormously by nothing that $h_j = h_j(g_j)$ only (i.e., the j^{th} component of $\mathbf{h}$ is a function only of the j^{th} component of $\mathbf{g}$), namely

$$h_j(g_k) = \delta_{jk}\sigma(g_k), \tag{25}$$

where σ is the activation function in question. Therefore,

$$\frac{\partial h_j}{\partial g_k} = \frac{\partial(\delta_{jk}\sigma(g_k))}{\partial g_k} = \delta_{jk}\sigma'(g_k), \tag{26}$$

which can be inserted into eq. (24) to yield

$$\begin{aligned} \frac{\partial L}{\partial g_k} &= \sum_{j=1}^{H} \frac{\partial L}{\partial h_j} \delta_{jk}\sigma'(g_k) \\ &= \frac{\partial L}{\partial h_k}\sigma'(g_k) \qquad (27) \\ &= \left(\sum_{i=1}^{F} 2(f_i - y_i)W_{ik} \right) \sigma'(g_k). \qquad (28) \end{aligned}$$

Though eq. (27) could be written in vector form as

$$\frac{\partial L}{\partial \mathbf{g}} = \left(\left(\frac{\partial L}{\partial \mathbf{h}} \right)^T \frac{\partial \mathbf{h}}{\partial \mathbf{g}} \right)^T, \tag{29}$$

given that $\frac{\partial \mathbf{h}}{\partial \mathbf{g}}$ is a diagonal matrix, where $\left(\frac{\partial \mathbf{h}}{\partial \mathbf{g}}\right)_{ii} = \sigma'(g_i)$, it is more instructive to write eq. (29) as a simple Hadamard product between the derivative of the loss w.r.t. the hidden layer output and the vector of derivatives of each input to the hidden layer g_i w.r.t. its output h_i,

$$\frac{\partial L}{\partial \mathbf{g}} = \frac{\partial L}{\partial \mathbf{h}} \odot \sigma'(\mathbf{g}). \tag{30}$$

Finally, it is now possible to derive the rate of change of the loss w.r.t. a weight in the first layer,

$$\frac{\partial L}{\partial V_{kl}} = \sum_{i=1}^{H} \frac{\partial L}{\partial g_i} \frac{\partial g_i}{\partial V_{kl}}. \tag{31}$$

From eq. (27), we know $\frac{\partial L}{\partial g_i}$. To find $\frac{\partial g_i}{\partial V_{kl}}$, it can be noted that

$$g_i = \sum_{j=1}^{D} V_{ij} x_j, \tag{32}$$

thus,

$$\begin{aligned} \frac{\partial g_i}{\partial V_{kl}} &= \delta_{ik} \sum_{j=1}^{D} \delta_{lj} x_j \qquad (33) \\ &= \delta_{ik} x_l. \qquad (34) \end{aligned}$$

This can be substituted into eq. (31) to produce

$$\begin{aligned} \frac{\partial L}{\partial V_{kl}} &= \sum_{i=1}^{H} \frac{\partial L}{\partial g_i} \delta_{ik} x_l \qquad (35) \\ &= \frac{\partial L}{\partial g_k} x_l \qquad (36) \end{aligned}$$

This reveals that the full matrix of derivatives of L with respect to each component of V is

$$\frac{\partial L}{\partial V} = \begin{bmatrix} \frac{\partial L}{\partial g_1} x_1 & \frac{\partial L}{\partial g_1} x_2 & \cdots & \frac{\partial L}{\partial g_1} x_D \\ \frac{\partial L}{\partial g_2} x_1 & \frac{\partial L}{\partial g_2} x_2 & \cdots & \frac{\partial L}{\partial g_2} x_D \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial g_H} x_1 & \frac{\partial L}{\partial g_H} x_2 & \cdots & \frac{\partial L}{\partial g_H} x_D \end{bmatrix}, \tag{37}$$

which can easily be rewritten as an outer product,

$$\frac{\partial L}{\partial V} = \frac{\partial L}{\partial \mathbf{g}} \mathbf{x}^T. \tag{38}$$

Finally, the full form of eq. (38) can be written by making use of eq. (30) and eq. (17),

$$\begin{aligned} \frac{\partial L}{\partial V} &= \left(\frac{\partial L}{\partial \mathbf{h}} \odot \sigma'(\mathbf{g}) \right) \mathbf{x}^T \qquad (39) \\ &= 2 \left(W^T (\mathbf{f} - \mathbf{y}) \odot \sigma'(\mathbf{g}) \right) \mathbf{x}^T \qquad (40) \end{aligned}$$

2 Multi-layer Neural Network

2.1 Network Architecture

I will start by defining the architecture of the full network. The network will consist of N hidden layers, followed by a single linear (output) layer. Passing an input through hidden layer n consists of a matrix multiplication by weights (with incorporated biases) W^n, followed by the application of activation function σ_n.

The input to the n^{th} hidden layer is taken as $\mathbf{h}^{n-1}$. The intermediate vector in the n^{th} layer produced by the matrix multiplication of W^n by $\mathbf{h}^{n-1}$ is taken as $\mathbf{g}^n$, and the output of the layer (after the activation function σ_n is applied) is taken as $\mathbf{h}^n$. Hence, as was probably already clear, the output of layer n is taken as the input to layer $n+1$ (so $\mathbf{h}^n$ is taken as the input in layer $n+1$). This architecture is more clearly defined below.

$$\begin{array}{lrll}
\mathbf{h}^0 \rightarrow & \mathbf{g}^1 = W^1\mathbf{h}^0 \rightarrow \mathbf{h}^1 = \sigma_1(\mathbf{g}^1) & \mathbf{g}^1, \mathbf{h}^1 \in \mathbb{R}^{D_1} & W^1 \in \mathbb{R}^{D_1 \times D_0} \\
\mathbf{h}^1 \rightarrow & \mathbf{g}^2 = W^2\mathbf{h}^1 \rightarrow \mathbf{h}^2 = \sigma_2(\mathbf{g}^2) & \mathbf{g}^2, \mathbf{h}^2 \in \mathbb{R}^{D_2} & W^2 \in \mathbb{R}^{D_2 \times D_1} \\
\mathbf{h}^2 \rightarrow & \mathbf{g}^3 = W^3\mathbf{h}^2 \rightarrow \mathbf{h}^3 = \sigma_3(\mathbf{g}^3) & \mathbf{g}^3, \mathbf{h}^3 \in \mathbb{R}^{D_3} & W^3 \in \mathbb{R}^{D_3 \times D_2} \\
\vdots & & \vdots & \vdots \\
\mathbf{h}^{N-1} \rightarrow & \mathbf{g}^N = W^N\mathbf{h}^{N-1} \rightarrow \mathbf{h}^N = \sigma_N(\mathbf{g}^N) & \mathbf{g}^N, \mathbf{h}^N \in \mathbb{R}^{D_N} & W^N \in \mathbb{R}^{D_N \times D_{N-1}} \\
\mathbf{h}^N \rightarrow & \mathbf{f} = W^F\mathbf{h}^N & \mathbf{f} \in \mathbb{R}^{D_F} & W^F \in \mathbb{R}^{D_F \times D_N}
\end{array} \tag{41}$$

As is probably apparent, $\mathbf{h}^0$ is the initial input to the entire network (equivalent to $\mathbf{x}$ used in section 1). $\mathbf{f}$ is the output of the entire network, and $\mathbf{y}$ (not shown yet), again, is taken as the vector of "true" values the network is attempting to predict. Importantly, in this section I will be treating derivatives of the loss function with respect to a vector as a row vector - I made this choice since this format is the "natural" output one obtains from applying the chain rule with vectors and matrices.

2.2 Finding $\frac{\partial L}{\partial \mathbf{f}}$

This process will be almost identical to that in the single-hidden-layer network. Again, the loss is given by

$$\begin{aligned} L &= |\mathbf{y} - \mathbf{f}|^2 \\ &= \sum_{i=1}^{D_F} (y_i - f_i)^2 . \end{aligned} \tag{42}$$

Thus, the derivative of the loss function w.r.t. a component of the output vector $\mathbf{f}$ is still

$$\frac{\partial L}{\partial f_i} = 2(f_i - y_i), \tag{43}$$

which makes the derivative w.r.t. the entire vector

$$\frac{\partial L}{\partial \mathbf{f}} = 2(\mathbf{f} - \mathbf{y})^T, \tag{44}$$

though the transpose is taken in order to ensure the derivative is a row vector (for reasons explained above).

2.3 Finding $\frac{\partial L}{\partial W^F}$

I start by taking the rate of change of the loss w.r.t. a single component of W^F,

$$\frac{\partial L}{\partial W^F_{kl}} = \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} \frac{\partial f_i}{\partial W^F_{kl}}. \tag{45}$$

Noting the fact that

$$f_i = \sum_{l=1}^{D_N} W^F_{il} h^N_l , \tag{46}$$

the derivative of a single component of $\mathbf{f}$ w.r.t. a single component of W^F can be written as

$$\begin{aligned} \frac{\partial f_i}{\partial W^F_{kl}} &= \delta_{ik} \sum_{j=1}^{D_N} \delta_{lj} h^N_j \\ &= \delta_{ik} h^N_l , \end{aligned} \tag{47}$$

where the outer Kronecker delta arises from the fact that f_i is only a function of the weights in row i of W^F, hence the gradient of f_i w.r.t. a weight in W^F which does not contribute to f_i is zero. Thus, eq. (45) can be rewritten as

$$\begin{aligned}\frac{\partial L}{\partial W^F_{kl}} &= \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} \delta_{ik} h^N_l \\ &= \frac{\partial L}{\partial f_k} h^N_l. \end{aligned} \tag{48}$$

The above equation can be used for all elements of W^F, which produces

$$\begin{aligned}\frac{\partial L}{\partial W^F} &= \begin{bmatrix} \frac{\partial L}{\partial f_1} h^N_1 & \frac{\partial L}{\partial f_1} h^N_2 & \cdots & \frac{\partial L}{\partial f_1} h^N_{D_N} \\ \frac{\partial L}{\partial f_2} h^N_1 & \frac{\partial L}{\partial f_2} h^N_2 & \cdots & \frac{\partial L}{\partial f_2} h^N_{D_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial f_{D_F}} h^N_1 & \frac{\partial L}{\partial f_{D_F}} h^N_2 & \cdots & \frac{\partial L}{\partial f_{D_F}} h^N_{D_N} \end{bmatrix} \\ &= \left(\frac{\partial L}{\partial \mathbf{f}}\right)^T \left(\mathbf{h}^N\right)^T, \end{aligned} \tag{49}$$

i.e., the rate of change of the loss w.r.t. every single element of the final matrix W^F of the multi-layer network. Finally, by substituting in eq. (44), this can be rewritten as

$$\frac{\partial L}{\partial W^F} = 2(\mathbf{f} - \mathbf{y})^T \left(\mathbf{h}^N\right)^T. \tag{50}$$

2.4 Finding $\frac{\partial L}{\partial \mathbf{h}^N}$

Now, in order to derive $\frac{\partial L}{\partial \mathbf{h}^N}$, I start by noting the rate of a change of L w.r.t. a single component of $\mathbf{h}^N$,

$$\frac{\partial L}{\partial h^N_j} = \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} \frac{\partial f_i}{\partial h^N_j}, \tag{51}$$

where the sum is required because $L = L(\mathbf{f}) = L(f_1, f_2, \ldots, f_{D_F})$, i.e., the loss is a function of all components of $\mathbf{f}$, and $f_i = f_i(h^N_1, h^N_2, \ldots, h^N_{D_N})$, i.e., each component of $\mathbf{f}$ is a function of all components of $\mathbf{h}^N$. Given eq. (46), the derivative of this component with respect to a given component h^N_j is simply

$$\begin{aligned}\frac{\partial f_i}{\partial h^N_j} &= \sum_{l=1}^{D_N} \delta_{jl} W^F_{il} \\ &= W^F_{ij}. \end{aligned} \tag{52}$$

Thus, it is easy to see that the derivative of the full output $\mathbf{f}$ w.r.t. the full output of the last hidden layer $\mathbf{h}^N$ (i.e. the Jacobian) is

$$\begin{aligned}
\frac{\partial \mathbf{f}}{\partial \mathbf{h}^N} &= \begin{bmatrix} \frac{\partial f_1}{\partial h_1^N} & \frac{\partial f_1}{\partial h_2^N} & \cdots & \frac{\partial f_1}{\partial h_{D_N}^N} \\ \frac{\partial f_2}{\partial h_1^N} & \frac{\partial f_2}{\partial h_2^N} & \cdots & \frac{\partial f_2}{\partial h_{D_N}^N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{D_F}}{\partial h_1^N} & \frac{\partial f_{D_F}}{\partial h_2^N} & \cdots & \frac{\partial f_{D_F}}{\partial h_{D_N}^N} \end{bmatrix} \\
&= \begin{bmatrix} W_{11}^F & W_{12}^F & \cdots & W_{1D_N}^F \\ W_{21}^F & W_{22}^F & \cdots & W_{2D_N}^F \\ \vdots & \vdots & \ddots & \vdots \\ W_{D_F1}^F & W_{D_F2}^F & \cdots & W_{D_FD_N}^F \end{bmatrix} \\
&= W^F
\end{aligned} \tag{53}$$

Now, eq. (51) can be rewritten using eq. (52),

$$\frac{\partial L}{\partial h_j^N} = \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} W_{ij}^F. \tag{54}$$

Since the above is just one component of the row vector of the rate of change of the loss w.r.t. the output from the last hidden layer, the full vector can be written as

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{h}^N} &= \begin{bmatrix} \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} W_{i1}^F & \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} W_{i2}^F & \cdots & \sum_{i=1}^{D_F} \frac{\partial L}{\partial f_i} W_{iD_N}^F \end{bmatrix} \\
&= \frac{\partial L}{\partial \mathbf{f}} W^F \\
&= \frac{\partial L}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \mathbf{h}^N}.
\end{aligned} \tag{55}$$

Finally, by substituting in eq. (44), the above can be rewritten as

$$\frac{\partial L}{\partial \mathbf{h}^N} = 2(\mathbf{f} - \mathbf{y})^T W^F. \tag{56}$$

2.5 Finding $\frac{\partial L}{\partial \mathbf{g}^N}$

The rate of change of the loss w.r.t. a single element of $\mathbf{g}^N$ will be given by

$$\frac{\partial L}{\partial g_j^N} = \sum_{i=1}^{D_N} \frac{\partial L}{\partial h_i^N}\frac{\partial h_i^N}{\partial g_j^N}. \tag{57}$$

However, since $h_i^N = \sigma^N(g_i^N)$, $h_i^N = h_i^N(g_i^N)$, i.e., h_i^N is a function of g_i^N only, so

$$\frac{\partial h_i^N}{\partial g_j^N} = \delta_{ij}\frac{dh_i^N}{dg_i^N} = \delta_{ij}\frac{d\sigma^N\left(g_i^N\right)}{dg_i^N}, \tag{58}$$

which means eq. (57) can be rewritten as

$$\begin{aligned}\frac{\partial L}{\partial g_j^N} &= \sum_{i=1}^{D_N} \frac{\partial L}{\partial h_i^N}\delta_{ij}\frac{d\sigma^N\left(g_i^N\right)}{dg_i^N} \\ &= \frac{\partial L}{\partial h_j^N}\frac{d\sigma^N\left(g_j^N\right)}{dg_j^N} \\ &= \frac{\partial L}{\partial h_j^N}\sigma^{N\prime}\left(g_j^N\right)\end{aligned} \tag{59}$$

Thus, the full derivative of the loss function w.r.t. each component of $\mathbf{g}^N$ can be written as

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{g}^N} &= \begin{bmatrix}\frac{\partial L}{\partial g_1^N} & \frac{\partial L}{\partial g_2^N} & \cdots & \frac{\partial L}{\partial g_{D_N}^N}\end{bmatrix} \\ &= \begin{bmatrix}\frac{\partial L}{\partial h_1^N}\sigma^{N\prime}\left(g_1^N\right) & \frac{\partial L}{\partial h_2^N}\sigma^{N\prime}\left(g_2^N\right) & \cdots & \frac{\partial L}{\partial h_{D_N}^N}\sigma^{N\prime}\left(g_{D_N}^N\right)\end{bmatrix},\end{aligned} \tag{60}$$

which, clearly, must be the Hadamard product

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{g}^N} &= \begin{bmatrix}\frac{\partial L}{\partial h_1^N} \\ \frac{\partial L}{\partial h_2^N} \\ \vdots \\ \frac{\partial L}{\partial h_{D_N}^N}\end{bmatrix}^T \odot \begin{bmatrix}\sigma^{N\prime}\left(g_1^N\right) \\ \sigma^{N\prime}\left(g_2^N\right) \\ \vdots \\ \sigma^{N\prime}\left(g_{D_N}^N\right)\end{bmatrix}^T \\ &= \frac{\partial L}{\partial \mathbf{h}^N} \odot \sigma^{N\prime}\left(\mathbf{g}^N\right).\end{aligned} \tag{61}$$

2.6 Finding $\frac{\partial L}{\partial W^N}$

I start by noting the rate of change of the loss w.r.t. a single component of the vector $\mathbf{g}^N$,

$$\frac{\partial L}{\partial W^N_{kl}} = \sum_{i=1}^{D_N} \frac{\partial L}{\partial g^N_i} \frac{\partial g^N_i}{\partial W^N_{kl}}. \tag{62}$$

However, given that

$$g^N_i = \sum_{j=1}^{D_{N-1}} W^N_{ij} h^{N-1}_j \tag{63}$$

and, thus,

$$\begin{aligned} \frac{\partial g^N_i}{\partial W^N_{kl}} &= \delta_{ik} \sum_{j=1}^{D_{N-1}} \delta_{lj} h^{N-1}_j \\ &= \delta_{ik} h^{N-1}_l, \end{aligned} \tag{64}$$

eq. (62) can be rewritten as

$$\begin{aligned} \frac{\partial L}{\partial W^N_{kl}} &= \sum_{i=1}^{D_N} \frac{\partial L}{\partial g^N_i} \delta_{ik} h^{N-1}_l \\ &= \frac{\partial L}{\partial g^N_k} h^{N-1}_l. \end{aligned} \tag{65}$$

Finally, the full rate of change of the loss w.r.t. the entire entire matrix of weights of the N^{th} layer can be expressed as

$$\begin{aligned} \frac{\partial L}{\partial W^N} &= \begin{bmatrix} \frac{\partial L}{\partial g^N_1} h^{N-1}_1 & \frac{\partial L}{\partial g^N_1} h^{N-1}_2 & \cdots & \frac{\partial L}{\partial g^N_1} h^{N-1}_{D_{N-1}} \\ \frac{\partial L}{\partial g^N_2} h^{N-1}_1 & \frac{\partial L}{\partial g^N_2} h^{N-1}_2 & \cdots & \frac{\partial L}{\partial g^N_2} h^{N-1}_{D_{N-1}} \\ \cdots & \cdots & \ddots & \vdots \\ \frac{\partial L}{\partial g^N_{D_N}} h^{N-1}_1 & \frac{\partial L}{\partial g^N_{D_N}} h^{N-1}_2 & \cdots & \frac{\partial L}{\partial g^N_{D_N}} h^{N-1}_{D_{N-1}} \end{bmatrix} \\ &= \left(\frac{\partial L}{\partial \mathbf{g}^N}\right)^T \left(\mathbf{h}^{N-1}\right)^T \end{aligned} \tag{66}$$

2.7 Finding $\frac{\partial L}{\partial \mathbf{h}^{N-1}}$

Again, I will start by writing the rate of change of the loss w.r.t. the rate of change of a single component of $\mathbf{h}^{N-1}$,

$$\frac{\partial L}{\partial h_j^{N-1}} = \sum_{i=1}^{D_N} \frac{\partial L}{\partial g_i^N} \frac{\partial g_i^N}{\partial h_j^{N-1}}, \tag{67}$$

where the summation is present because the loss is a function of every single component of $\mathbf{g}^N$, and every single component of $\mathbf{g}^N$ is, in turn, a function of every single component of $\mathbf{h}^{N-1}$. Next, it can be noted that

$$\mathbf{g}^N = W^N \mathbf{h}^{N-1}, \tag{68}$$

which means that

$$g_i^N = \sum_{l=1}^{D_{N-1}} W_{il}^N h_l^{N-1}, \tag{69}$$

making the derivative of a single element of $\mathbf{g}^N$ w.r.t. a single element of $\mathbf{h}^{N-1}$

$$\begin{aligned} \frac{\partial g_i^N}{\partial h_j^{N-1}} &= \sum_{l=1}^{D_{N-1}} \delta_{jl} W_{il}^N \\ &= W_{ij}^N. \end{aligned} \tag{70}$$

This makes it clear that

$$\begin{aligned} \frac{\partial \mathbf{g}^N}{\partial \mathbf{h}^{N-1}} &= \begin{bmatrix} W_{11}^N & W_{12}^N & \cdots & W_{1D_{N-1}}^N \\ W_{21}^N & W_{22}^N & \cdots & W_{2D_{N-1}}^N \\ \vdots & \vdots & \ddots & \vdots \\ W_{D_N1}^N & W_{D_N2}^N & \cdots & W_{D_N D_{N-1}}^N \end{bmatrix} \\ &= W^N. \end{aligned} \tag{71}$$

Now, eq. (70) can be substituted into eq. (67) to produce

$$\frac{\partial L}{\partial h_j^{N-1}} = \sum_{i=1}^{D_N} \frac{\partial L}{\partial g_i^N} W_{ij}^N. \tag{72}$$

Finally, this can be used to write the full rate of change of the loss w.r.t. the entire vector output of the $(N-1)^{\text{th}}$ layer,

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{h}^{N-1}} &= \left[\sum_{i=1}^{D_N} \frac{\partial L}{\partial g_i^N} W_{i1}^N \quad \sum_{i=1}^{D_N} \frac{\partial L}{\partial g_i^N} W_{i2}^N \quad \cdots \quad \sum_{i=1}^{D_N} \frac{\partial L}{\partial g_i^N} W_{iD_{N-1}}^N\right] \\ &= \frac{\partial L}{\partial \mathbf{g}^N} W^N \end{aligned} \tag{73}$$

2.8 Subsequent Layers

For the subsequent layers of the network (i.e. going futher back, towards the input layer), we wish to be able to find $\frac{\partial L}{\partial W^n}$ for every layer n, so that the weights W^n may be updated. Instead of re-deriving the above quantities for every layer, it can be noted that the equation for $\frac{\partial L}{\partial W^N}$ (i.e., the rate of change of the loss w.r.t. the rate of change of the weights in the penultimate layer of the network), eq. (66), is expressed in quantities that have been previously worked out, i.e.,

$$\frac{\partial L}{\partial W^N} = \left(\frac{\partial L}{\partial \mathbf{g}^N}\right)^T \left(\mathbf{h}^{N-1}\right)^T. \tag{74}$$

Since there is nothing special about the N^{th} layer of the network, the derivations for another layer n would have resulted in an identical equation, simply switching N for n, given that all layers simply receive an input, perform a matrix multiplication and then an activation. Thus, eq. (74) can be rewritten as

$$\frac{\partial L}{\partial W^n} = \left(\frac{\partial L}{\partial \mathbf{g}^n}\right)^T \left(\mathbf{h}^{n-1}\right)^T. \tag{75}$$

Similarly, eq. (61) can be rewritten as

$$\frac{\partial L}{\partial \mathbf{g}^n} = \frac{\partial L}{\partial \mathbf{h}^n} \odot \sigma^{n\prime}\left(\mathbf{g}^n\right) \tag{76}$$

and eq. (73) can be rewritten as

$$\frac{\partial L}{\partial \mathbf{h}^{n-1}} = \frac{\partial L}{\partial \mathbf{g}^n} W^n. \tag{77}$$

With eqs. (75) to (77), the rate of change of the loss w.r.t. the rate of change of every single one of the parameters of the neural network can be computed, thus allowing each matrix W^n to be updated via

$$W_{t+1}^n = W_t^n - \eta \frac{\partial L}{\partial W_t^n} \tag{78}$$

where η is the learning rate and t is a given training iteration. The negative sign arises because $\frac{\partial L}{\partial W_t^n}$ gives the "direction of steepest ascent" in weight-space, i.e., the direction within $(D_n \times D_{n-1})$-dimensional hyperspace in which the loss increases most rapidly for an infinitesimally small step in this space. Therefore, the weights must be adjusted in order to move in the opposite direction to the steepest ascent, i.e., to take a step in the direction of steepest descent, which is the direction in which an infinitesimally small step in weight-space produces the largest decrease in the loss.

All of the derivations of backpropagation in this document started with the definition of the loss being $L = |\mathbf{y} - \mathbf{f}|^2$, i.e., squared loss. However, as you'll have noticed, eqs. (75) to (77) are expressed completely generally, and can, thus, be used for a neural network with any loss function. Therefore, the only equations that would have to be redefined are those pertaining to the last layer of the network, particularly $\frac{\partial L}{\partial \mathbf{f}}$. If any other transformations are performed between the final matrix multiplication and the calculation of the loss, then these would also have to be treated. Nonetheless, the equations derived in this document provide a comprehensive means of updating all the parameters of a neural network via backpropagation.