We want to train an MLP by minimizing a loss over the training set

\[ \mathcal{L}(w, b) = \sum_n \ell(f(x_n; w, b), y_n). \]

To use gradient descent, we need the expression of the gradient of the loss with respect to the parameters:

\[ \frac{\partial \mathcal{L}}{\partial w^{(l)}_{i,j}} \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial b^{(l)}_i}. \]

So, with \( \ell_n = \ell(f(x_n; w, b), y_n) \), what we need is

\[ \frac{\partial \ell_n}{\partial w^{(l)}_{i,j}} \quad \text{and} \quad \frac{\partial \ell_n}{\partial b^{(l)}_i}. \]
For clarity, we consider a single training sample \( x \), and introduce \( s^{(1)}, \ldots, s^{(L)} \) as the summations before activation functions.

\[
x^{(0)} = x \xrightarrow{w^{(1)}, b^{(1)}} s^{(1)} \xrightarrow{\sigma} x^{(1)} \xrightarrow{w^{(2)}, b^{(2)}} s^{(2)} \xrightarrow{\sigma} \ldots \xrightarrow{w^{(L)}, b^{(L)}} s^{(L)} \xrightarrow{\sigma} x^{(L)} = f(x; w, b).
\]

Formally we set \( x^{(0)} = x \),

\[
\forall l = 1, \ldots, L,
\begin{align*}
    s^{(l)} &= w^{(l)}x^{(l-1)} + b^{(l)} \\
    x^{(l)} &= \sigma(s^{(l)}),
\end{align*}
\]

and we set the output of the network as \( f(x; w, b) = x^{(L)} \).

This is the **forward pass**.

The core principle of the back-propagation algorithm is the “chain rule” from differential calculus:

\[
(g \circ f)' = (g' \circ f)f'
\]

which generalizes to longer compositions and higher dimensions

\[
J_{f_N \circ f_{N-1} \circ \cdots \circ f_1}(x) = \prod_{n=1}^{N} J_{f_n}(f_{n-1} \circ \cdots \circ f_1(x)),
\]

where \( J_f(x) \) is the Jacobian of \( f \) at \( x \), that is the matrix of the linear approximation of \( f \) in the neighborhood of \( x \).

The linear approximation of a composition of mappings is the product of their individual linear approximations.

What follows is exactly this principle applied to a MLP.
\[
\ldots \sigma \rightarrow x^{(l-1)} \xrightarrow{w^{(l)}, b^{(l)}} s^{(l)} \sigma \rightarrow x^{(l)} \xrightarrow{w^{(l+1)}, b^{(l+1)}} s^{(l+1)} \sigma \rightarrow \ldots
\]

We have

\[
s_i^{(l)} = \sum_j w_{i,j} x_j^{(l-1)} + b_i^{(l)},
\]

so \(w_{i,j}^{(l)}\) influences \(\ell\) only through \(s_i^{(l)}\), and we get

\[
\frac{\partial \ell}{\partial w_{i,j}^{(l)}} = \frac{\partial \ell}{\partial s_i^{(l)}} \frac{\partial s_i^{(l)}}{\partial w_{i,j}^{(l)}} = \frac{\partial \ell}{\partial s_i^{(l)}} x_j^{(l-1)},
\]

and similarly

\[
\frac{\partial \ell}{\partial b_i^{(l)}} = \frac{\partial \ell}{\partial s_i^{(l)}}.
\]

Since we know \(x_j^{(l-1)}\) from the forward pass, we only need \(\frac{\partial \ell}{\partial s_i^{(l)}}\).

---

\[
\ldots \sigma \rightarrow x^{(l-1)} \xrightarrow{w^{(l)}, b^{(l)}} s^{(l)} \sigma \rightarrow x^{(l)} \xrightarrow{w^{(l+1)}, b^{(l+1)}} s^{(l+1)} \sigma \rightarrow \ldots
\]

We have

\[
x_i^{(l)} = \sigma(s_i^{(l)}),
\]

and since \(s_i^{(l)}\) influences \(\ell\) only through \(x_i^{(l)}\), the chain rule gives

\[
\frac{\partial \ell}{\partial s_i^{(l)}} = \frac{\partial \ell}{\partial x_i^{(l)}} \frac{\partial x_i^{(l)}}{\partial s_i^{(l)}} = \frac{\partial \ell}{\partial x_i^{(l)}} \sigma'(s_i^{(l)}),
\]

Since we know \(s_i^{(l)}\) from the forward pass, we only need \(\frac{\partial \ell}{\partial x_i^{(l)}}\).
Finally, we have

\[ \frac{\partial \ell}{\partial x_i^l} = (\nabla_1 \ell)_i \]

where \( \nabla_1 \ell \) is the gradient of \( \ell \) with respect to its first parameter, that is the predicted value.

Also, \( \forall l = 1, \ldots, L - 1 \), since

\[ s_h^{(l+1)} = \sum_i w_{h,i}^{l+1} x_i^l + b_h^{l+1}, \]

and \( x_i^l \) influences \( \ell \) only through the \( s_h^{(l+1)} \), we have

\[ \frac{\partial \ell}{\partial x_i^l} = \sum_h \frac{\partial \ell}{\partial s_h^{(l+1)}} \frac{\partial s_h^{(l+1)}}{\partial x_i^l} = \sum_h \frac{\partial \ell}{\partial s_h^{(l+1)}} w_{h,i}^{l+1}. \]

To write all this in tensorial form, if \( \psi : \mathbb{R}^N \rightarrow \mathbb{R}^M \), we will use the standard Jacobian notation

\[
\left[ \frac{\partial \psi}{\partial x} \right] = \begin{pmatrix}
\frac{\partial \psi_1}{\partial x_1} & \cdots & \frac{\partial \psi_1}{\partial x_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial \psi_M}{\partial x_1} & \cdots & \frac{\partial \psi_M}{\partial x_N}
\end{pmatrix},
\]

and if \( \psi : \mathbb{R}^{N \times M} \rightarrow \mathbb{R} \), we will use the compact notation, also tensorial

\[
\left[ \frac{\partial \psi}{\partial w} \right] = \begin{pmatrix}
\frac{\partial \psi}{\partial w_{1,1}} & \cdots & \frac{\partial \psi}{\partial w_{1,M}} \\
\vdots & \ddots & \vdots \\
\frac{\partial \psi}{\partial w_{N,1}} & \cdots & \frac{\partial \psi}{\partial w_{N,M}}
\end{pmatrix},
\]

A standard notation (that we do not use here) is

\[
\left[ \frac{\partial \ell}{\partial x^l} \right] = \nabla_x^l \ell \quad \left[ \frac{\partial \ell}{\partial s^l} \right] = \nabla_s^l \ell \quad \left[ \frac{\partial \ell}{\partial b^l} \right] = \nabla_b^l \ell \quad \left[ \frac{\partial \ell}{\partial w^l} \right] = \nabla_w^l \ell.
\]
Forward pass
Compute the activations.
\[ x^{(0)} = x, \quad \forall l = 1, \ldots, L, \quad \begin{cases} 
    s^{(l)} = w^{(l)} x^{(l-1)} + b^{(l)} \\
    x^{(l)} = \sigma (s^{(l)}) 
\end{cases} \]

Backward pass
Compute the derivatives of the loss wrt the activations.
\[
\begin{cases} 
    \frac{\partial \ell}{\partial x^{(l)}} = \nabla_1 \ell (x^{(l)}) \\
    \text{if } l < L, \quad \frac{\partial \ell}{\partial s^{(l)}} = (w^{(l+1)})^T \frac{\partial \ell}{\partial s^{(l+1)}} 
\end{cases}
\]

Compute the derivatives of the loss wrt the parameters.
\[
\frac{\partial \ell}{\partial w^{(l)}} = \left( \frac{\partial \ell}{\partial s^{(l)}} \right) (x^{(l-1)})^T \\
\frac{\partial \ell}{\partial b^{(l)}} = \frac{\partial \ell}{\partial s^{(l)}}
\]

Gradient step
Update the parameters.
\[
w^{(l)} \leftarrow w^{(l)} - \eta \left[ \frac{\partial \ell}{\partial w^{(l)}} \right] \\
b^{(l)} \leftarrow b^{(l)} - \eta \left[ \frac{\partial \ell}{\partial b^{(l)}} \right]
\]
In spite of its hairy formalization, the backward pass is a simple algorithm: apply the chain rule again and again.

As for the forward pass, it can be expressed in tensorial form. Heavy computation is concentrated in linear operations, and all the non-linearities go into component-wise operations.

Regarding computation, since the costly operation for the forward pass is

$$s^{(l)} = w^{(l)} x^{(l-1)} + b^{(l)}$$

and for the backward

$$\left[ \frac{\partial \ell}{\partial x^{(l)}} \right] = \left( w^{(l+1)} \right)^T \left[ \frac{\partial \ell}{\partial s^{(l+1)}} \right]$$

and

$$\left[ \frac{\partial \ell}{\partial w^{(l)}} \right] = \left[ \frac{\partial \ell}{\partial s^{(l)}} \right] \left( x^{(l-1)} \right)^T,$$

the rule of thumb is that the backward pass is twice more expensive than the forward one.