# Deep Learning (IST, 2021-22)

# Practical 4: Multilayer Perceptron and Backpropagation

André Martins, Andreas Wichert, Luis Sá-Couto

## Question 1

Consider a network with four layers with the following numbers of units: $4, 4, 3, 3$. Assume all units, except the ones in the output layer, use the hyperbolic tangent activation function.

1. Initialize all connection weights and biases to 0.1. Using the squared error loss do a **stochastic gradient descent** update (with learning rate $\eta = 0.1$) for the training example:

$$\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^\top, \quad \boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

**Solution:** Before moving forward, we recall and derive the tanh activation function.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{2}{1 + \exp(-2x)} - 1$$

$$
\begin{aligned}
\frac{\partial}{\partial x}\tanh(x) &= \frac{\partial}{\partial x}\left(\frac{2}{1+\exp(-2x)}-1\right)\\
&= \frac{\partial}{\partial x}\left(\frac{2}{1+\exp(-2x)}\right)\\
&= 2\frac{\partial}{\partial x}\left(\frac{1}{1+\exp(-2x)}\right)\\
&= 2\left(\frac{\partial\left(\frac{1}{1+\exp(-2x)}\right)}{\partial(1+\exp(-2x))}\frac{\partial(1+\exp(-2x))}{\partial(\exp(-2x))}\frac{\partial(\exp(-2x))}{\partial(-2x)}\frac{\partial(-2x)}{\partial x}\right)\\
&= 2\left(-\frac{1}{(1+\exp(-2x))^2}(1)\exp(-2x)(-2)\right)\\
&= 4\left(\frac{\exp(-2x)}{(1+\exp(-2x))^2}\right)\\
&= 4\left(\frac{1+\exp(-2x)-1}{(1+\exp(-2x))^2}\right)\\
&= 4\left(\frac{1+\exp(-2x)}{(1+\exp(-2x))^2}-\frac{1}{(1+\exp(-2x))^2}\right)\\
&= 4\left(\frac{1}{1+\exp(-2x)}-\frac{1}{(1+\exp(-2x))^2}\right)\\
&= \frac{4}{1+\exp(-2x)}\left(1-\frac{1}{1+\exp(-2x)}\right)\\
&= 2\frac{2}{1+\exp(-2x)}\left(1-\frac{1}{1+\exp(-2x)}\right)\\
&= \frac{2}{1+\exp(-2x)}\left(2-\frac{2}{1+\exp(-2x)}\right)\\
&= (\tanh(x)+1)(2-(\tanh(x)+1))\\
&= (\tanh(x)+1)(2-\tanh(x)-1)\\
&= (\tanh(x)+1)(1-\tanh(x))\\
&= \tanh(x)-\tanh(x)^2+1-\tanh(x)\\
& 1-\tanh(x)^2
\end{aligned}
$$

We start by writting the connection weights and the biases:

$$
\mathbf{W}^{[1]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}
$$

$$
\mathbf{b}^{[1]} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}
$$

$$
\mathbf{W}^{[2]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}
$$

$$\mathbf{b}^{[2]} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

$$\mathbf{W}^{[3]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}$$

$$\boldsymbol{b}^{[3]} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

We are now ready to do forward propagation:

$$\boldsymbol{x} = \boldsymbol{h}^{[0]} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\boldsymbol{z}^{[1]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{pmatrix}$$

$$\boldsymbol{h}^{[1]} = \tanh\left(\begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{pmatrix}\right) = \begin{pmatrix} 0.2913 \\ 0.2913 \\ 0.2913 \\ 0.2913 \end{pmatrix}$$

$$\boldsymbol{z}^{[2]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \begin{pmatrix} 0.2913 \\ 0.2913 \\ 0.2913 \\ 0.2913 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.2165 \\ 0.2165 \\ 0.2165 \end{pmatrix}$$

$$\boldsymbol{h}^{[2]} = \tanh\left(\begin{pmatrix} 0.2165 \\ 0.2165 \\ 0.2165 \end{pmatrix}\right) = \begin{pmatrix} 0.2132 \\ 0.2132 \\ 0.2132 \end{pmatrix}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{z}^{[3]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} \begin{pmatrix} 0.2132 \\ 0.2132 \\ 0.2132 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.16396 \\ 0.16396 \\ 0.16396 \end{pmatrix}$$

Now we want to do the backward phase. Recall the squared error measure:

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{2}\sum_{i=1}^{3}(\hat{y}_i - y_i)^2 = 0.376.$$

In general, we will need to know how to derive all functions in our network. Let us compute them beforehand:

Gradient in the output layer:

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[L]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{y}}} = \frac{\partial \frac{1}{2}\|\hat{\boldsymbol{y}}-\boldsymbol{y}\|^2}{\partial \hat{\boldsymbol{y}}} = \hat{\boldsymbol{y}}-\boldsymbol{y}.$$

Gradients of hidden layer parameters:

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[l]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[l]}} \boldsymbol{h}^{[l-1]^\top}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[l]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[l]}}$$

Gradients of hidden layer below:

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[l]}} = \mathbf{W}^{[l+1]^\top} \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[l+1]}}$$

Gradients of hidden layer below (before activation $\boldsymbol{h}^{[l]} = \boldsymbol{g}(\boldsymbol{z}^{[l]})$):

$$\begin{aligned}\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[l]}} &= \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[l]}} \odot \boldsymbol{g}'(\boldsymbol{z}^{[l]}) \\ &= \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[l]}} \odot (1-(\tanh(\boldsymbol{z}^{[l]}))^2) \\ &= \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[l]}} \odot (1-(\boldsymbol{h}^{[l]})^2).\end{aligned}$$

We compute these quantities recursively from the last layer to the first one:

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} = \hat{\boldsymbol{y}}-\boldsymbol{y} = \left(\begin{pmatrix}0.16396\\0.16396\\0.16396\end{pmatrix} - \begin{pmatrix}0\\1\\0\end{pmatrix}\right) = \begin{pmatrix}0.16396\\-0.83604\\0.16396\end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[3]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} \boldsymbol{h}^{[2]^\top} = \begin{pmatrix}0.16396\\-0.83604\\0.16396\end{pmatrix}\begin{pmatrix}0.2132\\0.2132\\0.2132\end{pmatrix}^\top = \begin{pmatrix}0.03496 & 0.03496 & 0.03496\\-0.17825 & -0.17825 & -0.17825\\0.03496 & 0.03496 & 0.03496\end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[3]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} = \begin{pmatrix}0.16396\\-0.83604\\0.16396\end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[2]}} = \mathbf{W}^{[3]^\top} \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} = \begin{pmatrix}0.1 & 0.1 & 0.1\\0.1 & 0.1 & 0.1\\0.1 & 0.1 & 0.1\end{pmatrix}^\top \begin{pmatrix}0.16396\\-0.83604\\0.16396\end{pmatrix} = \begin{pmatrix}-0.050812\\-0.050812\\-0.050812\end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[2]}} \odot (1-(\boldsymbol{h}^{[2]})^2) = \begin{pmatrix}-0.050812\\-0.050812\\-0.050812\end{pmatrix} \odot \left(1 - \begin{pmatrix}0.2132\\0.2132\\0.2132\end{pmatrix}^2\right) = \begin{pmatrix}-0.04850\\-0.04850\\-0.04850\end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[2]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}}\boldsymbol{h}^{[1]\top} = \begin{pmatrix} -0.04850 \\ -0.04850 \\ -0.04850 \end{pmatrix}\begin{pmatrix} 0.2913 \\ 0.2913 \\ 0.2913 \\ 0.2913 \end{pmatrix}^{\top} = \begin{pmatrix} -0.01413 & -0.01413 & -0.01413 & -0.01413 \\ -0.01413 & -0.01413 & -0.01413 & -0.01413 \\ -0.01413 & -0.01413 & -0.01413 & -0.01413 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[2]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}} = \begin{pmatrix} -0.04850 \\ -0.04850 \\ -0.04850 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[1]}} = \mathbf{W}^{[2]\top}\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}^{\top}\begin{pmatrix} -0.04850 \\ -0.04850 \\ -0.04850 \end{pmatrix} = \begin{pmatrix} -0.01455 \\ -0.01455 \\ -0.01455 \\ -0.01455 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[1]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[1]}} \odot (1-(\boldsymbol{h}^{[1]})^2) = \begin{pmatrix} -0.01455 \\ -0.01455 \\ -0.01455 \\ -0.01455 \end{pmatrix} \odot \left(1 - \begin{pmatrix} 0.2913 \\ 0.2913 \\ 0.2913 \\ 0.2913 \end{pmatrix}^{2}\right) = \begin{pmatrix} -0.01332 \\ -0.01332 \\ -0.01332 \\ -0.01332 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[1]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[1]}}\boldsymbol{x}^{\top} = \begin{pmatrix} -0.01332 \\ -0.01332 \\ -0.01332 \\ -0.01332 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}^{\top} = \begin{pmatrix} -0.01332 & 0 & -0.01332 & 0 \\ -0.01332 & 0 & -0.01332 & 0 \\ -0.01332 & 0 & -0.01332 & 0 \\ -0.01332 & 0 & -0.01332 & 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[1]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[1]}} = \begin{pmatrix} -0.01332 \\ -0.01332 \\ -0.01332 \\ -0.01332 \end{pmatrix}.$$

Finally, we can go to the last phase and perform the updates. We start with the first layer:

$$\begin{aligned}
\mathbf{W}^{[1]} &= \mathbf{W}^{[1]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[1]}} \\
&= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1\begin{pmatrix} -0.01332 & 0 & -0.01332 & 0 \\ -0.01332 & 0 & -0.01332 & 0 \\ -0.01332 & 0 & -0.01332 & 0 \\ -0.01332 & 0 & -0.01332 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0.101332 & 0.1 & 0.101332 & 0.1 \\ 0.101332 & 0.1 & 0.101332 & 0.1 \\ 0.101332 & 0.1 & 0.101332 & 0.1 \\ 0.101332 & 0.1 & 0.101332 & 0.1 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{b}^{[1]} &= \boldsymbol{b}^{[1]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[1]}} \\
&= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1\begin{pmatrix} -0.01332 \\ -0.01332 \\ -0.01332 \\ -0.01332 \end{pmatrix} \\
&= \begin{pmatrix} 0.101332 \\ 0.101332 \\ 0.101332 \\ 0.101332 \end{pmatrix}
\end{aligned}$$

Now the second:

$$\mathbf{W}^{[2]} = \mathbf{W}^{[2]} - \eta \frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[2]}}$$

$$= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} -0.01413 & -0.01413 & -0.01413 & -0.01413 \\ -0.01413 & -0.01413 & -0.01413 & -0.01413 \\ -0.01413 & -0.01413 & -0.01413 & -0.01413 \end{pmatrix}$$

$$= \begin{pmatrix} 0.101413 & 0.101413 & 0.101413 & 0.101413 \\ 0.101413 & 0.101413 & 0.101413 & 0.101413 \\ 0.101413 & 0.101413 & 0.101413 & 0.101413 \end{pmatrix}$$

$$\boldsymbol{b}^{[2]} = \boldsymbol{b}^{[2]} - \eta \frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[2]}}$$

$$= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} -0.04850 \\ -0.04850 \ - 0.04850 \end{pmatrix}$$

$$= \begin{pmatrix} 0.10485 \\ 0.10485 \\ 0.10485 \end{pmatrix}$$

All that is left is to update the parameters for the output layer.

$$\mathbf{W}^{[3]} = \mathbf{W}^{[3]} - \eta \frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[3]}}$$

$$= \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} 0.03496 & 0.03496 & 0.03496 \\ -0.17825 & -0.17825 & -0.17825 \\ 0.03496 & 0.03496 & 0.03496 \end{pmatrix}$$

$$= \begin{pmatrix} 0.0965 & 0.0965 & 0.0965 \\ 0.1178 & 0.1178 & 0.1178 \\ 0.0965 & 0.0965 & 0.0965 \end{pmatrix}$$

$$\boldsymbol{b}^{[3]} = \boldsymbol{b}^{[3]} - \eta \frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[3]}}$$

$$= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} 0.16396 \\ -0.83604 \\ 0.16396 \end{pmatrix}$$

$$= \begin{pmatrix} 0.0836 \\ 0.1836 \\ 0.0836 \end{pmatrix}$$

2. Reusing the computations from the previous exercise do a **gradient descent** update (with learning rate $\eta = 0.1$) for the batch with the training example from the a) and the following:

$$\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 10 & 0 \end{bmatrix}^\top, \quad \boldsymbol{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

**Solution:** Recall the squared loss:

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{2} \sum_{j=1}^{2} \| \hat{\boldsymbol{y}}^{(j)} - \boldsymbol{y}^{(j)} \|^2$$

Notice that the derivative of the sum is equal to the sum of the derivatives. For this reason, all we have to do is to compute the derivative for the new example and the final gradient will be the sum of both individual derivatives: the one for the new example and the one from the previous exercise.

## Question 2

Let us repeat the exact same exercise as in 1) but this time we will change:

- The output units have a softmax activation function
- The error function is cross-entropy

1. Initialize all connection weights and biases to 0.1. Using the cross-entropy loss do a **stochastic gradient descent** update (with learning rate $\eta = 0.1$) for the training example:

$$\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^\top, \quad \boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

**Solution:** Before moving forward, we recall and derive the *softmax* activation function. This activation function will work differently from what we have seen so far. This happens because each unit's output depends not only on its input but on the inputs of the other units. Let us make this more concrete. This function receives a vector $\boldsymbol{z} \in \mathbb{R}^d$ (logits) and outputs a vector $\boldsymbol{p} \in \mathbb{R}^d$ (probability values).

$$\text{softmax}\left(\begin{bmatrix} z_1 & z_2 & \cdots & z_d \end{bmatrix}^\top\right) = \begin{bmatrix} p_1 & p_2 & \cdots & p_d \end{bmatrix}^\top$$

with $p_i = \frac{\exp(z_i)}{\sum_{k=1}^d \exp(z_k)}$. Notice how each $p_i$ depends not only on $z_i$ like before.

Given the true output $\boldsymbol{y}$ (a one-hot vector) , the cross-entropy loss is:

$$\begin{aligned} L(\boldsymbol{y}, \boldsymbol{p}) &= -\sum_{k=1}^{d} y_k \log p_k \\ &= -\sum_{k=1}^{d} y_k \left( z_k - \log \sum_j \exp(z_j) \right) \\ &= -\underbrace{\sum_{k=1}^{d} y_k z_k}_{\boldsymbol{y}^\top \boldsymbol{z}} + \underbrace{\left( \sum_{k=1}^{d} y_k \right)}_{=1} \log \sum_j \exp(z_j) \\ &= -\boldsymbol{y}^\top \boldsymbol{z} + \log \sum_j \exp(z_j). \end{aligned}$$

The gradient of this loss is:

$$\begin{aligned} \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{z}} &= -\boldsymbol{y} + \text{softmax}(\boldsymbol{z}) \\ \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{z}} &= -\boldsymbol{y} + \boldsymbol{p}. \end{aligned}$$

We start by writing the connection weights and the biases:

$$\mathbf{W}^{[1]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}$$

$$\boldsymbol{b}^{[1]} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

$$\mathbf{W}^{[2]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}$$

$$\boldsymbol{b}^{[2]} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

$$\mathbf{W}^{[3]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}$$

$$\boldsymbol{b}^{[3]} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

We are now ready to do forward propagation:

$$\boldsymbol{x} = \boldsymbol{h}^{[0]} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\boldsymbol{z}^{[1]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{pmatrix}$$

$$\boldsymbol{h}^{[1]} = \tanh\left(\begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{pmatrix}\right) = \begin{pmatrix} 0.2913 \\ 0.2913 \\ 0.2913 \\ 0.2913 \end{pmatrix}$$

$$\boldsymbol{z}^{[2]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \begin{pmatrix} 0.2913 \\ 0.2913 \\ 0.2913 \\ 0.2913 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.2165 \\ 0.2165 \\ 0.2165 \end{pmatrix}$$

$$\boldsymbol{h}^{[2]} = \tanh\left(\begin{pmatrix} 0.2165 \\ 0.2165 \\ 0.2165 \end{pmatrix}\right) = \begin{pmatrix} 0.2132 \\ 0.2132 \\ 0.2132 \end{pmatrix}$$

$$\boldsymbol{z}^{[3]} = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} \begin{pmatrix} 0.2132 \\ 0.2132 \\ 0.2132 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.16396 \\ 0.16396 \\ 0.16396 \end{pmatrix}$$

$$\boldsymbol{p} = \mathrm{softmax}\left( \begin{pmatrix} 0.16396 \\ 0.16396 \\ 0.16396 \end{pmatrix} \right) = \begin{pmatrix} 0.3333 \\ 0.3333 \\ 0.3333 \end{pmatrix}$$

Recall the cross-entropy loss:

$$L\left(\boldsymbol{y}, \boldsymbol{p}\right) = -\sum_{k=1}^{d} y_k \log p_k = 1.0986.$$

Gradient in the output layer:

$$\frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{z}^{[L]}} = -\frac{\partial \sum_{k=1}^{d} y_k \log p_k}{\partial \boldsymbol{z}^{[L]}} = \boldsymbol{p} - \boldsymbol{y}.$$

Gradients of hidden layer parameters:

$$\frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \mathbf{W}^{[l]}} = \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{z}^{[l]}} \boldsymbol{h}^{[l-1]^\top}$$

$$\frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{b}^{[l]}} = \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{z}^{[l]}}$$

Gradients of hidden layer below:

$$\frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{h}^{[l]}} = \mathbf{W}^{[l+1]^\top} \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{p}}$$

Gradients of hidden layer below (before activation $\boldsymbol{h}^{[l]} = \boldsymbol{g}(\boldsymbol{z}^{[l]})$):

$$\begin{aligned} \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{z}^{[l]}} &= \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{h}^{[l]}} \odot \boldsymbol{g}'(\boldsymbol{z}^{[l]}) \\ &= \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{h}^{[l]}} \odot (1 - (\tanh(\boldsymbol{z}^{[l]}))^2) \\ &= \frac{\partial L(\boldsymbol{y}, \boldsymbol{p})}{\partial \boldsymbol{h}^{[l]}} \odot (1 - (\boldsymbol{h}^{[l]})^2). \end{aligned}$$

We compute these quantities recursively from the last layer to the first one:

$$\frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} = \boldsymbol{p} - \boldsymbol{y} = \left( \begin{pmatrix} 0.3333 \\ 0.3333 \\ 0.3333 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 0.3333 \\ -0.6667 \\ 0.3333 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[3]}} = \frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} \boldsymbol{h}^{[2]^\top} = \begin{pmatrix} 0.3333 \\ -0.6667 \\ 0.3333 \end{pmatrix} \begin{pmatrix} 0.2132 \\ 0.2132 \\ 0.2132 \end{pmatrix}^\top = \begin{pmatrix} 0.07107 & 0.07107 & 0.07107 \\ -0.14214 & -0.14214 & -0.14214 \\ 0.07107 & 0.07107 & 0.07107 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[3]}} = \frac{\partial L(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} = \begin{pmatrix} 0.3333 \\ -0.6667 \\ 0.3333 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[2]}} = \mathbf{W}^{[3]\top}\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[3]}} = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}^{\top} \begin{pmatrix} 0.3333 \\ -0.6667 \\ 0.3333 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[2]}} \odot (1-(\boldsymbol{h}^{[2]})^2) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[2]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}}\boldsymbol{h}^{[1]\top} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[2]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[1]}} = \mathbf{W}^{[2]\top}\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[2]}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[1]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{h}^{[1]}} \odot (1-(\boldsymbol{h}^{[1]})^2) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[1]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[1]}}\boldsymbol{x}^{\top} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}^{\top} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[1]}} = \frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{z}^{[1]}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Finally, we can go to the last phase and perform the updates. We start with the first layer:

$$\begin{aligned} \mathbf{W}^{[1]} &= \mathbf{W}^{[1]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[1]}} \\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \end{aligned}$$

$$\begin{aligned}\boldsymbol{b}^{[1]} &= \boldsymbol{b}^{[1]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[1]}}\\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}\end{aligned}$$

Now the second:

$$\begin{aligned}\mathbf{W}^{[2]} &= \mathbf{W}^{[2]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[2]}}\\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\boldsymbol{b}^{[2]} &= \boldsymbol{b}^{[2]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[2]}}\\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1\begin{pmatrix} 0 \\ 0\ 0 \end{pmatrix}\\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}\end{aligned}$$

So far, no parameters changed. All that is left is to update the parameters for the output layer.

$$\begin{aligned}\mathbf{W}^{[3]} &= \mathbf{W}^{[3]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \mathbf{W}^{[3]}}\\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1\begin{pmatrix} 0.07107 & 0.07107 & 0.07107 \\ -0.14214 & -0.14214 & -0.14214 \\ 0.07107 & 0.07107 & 0.07107 \end{pmatrix}\\ &= \begin{pmatrix} 0.09289 & 0.09289 & 0.09289 \\ 0.11421 & 0.11421 & 0.11421 \\ 0.09289 & 0.09289 & 0.09289 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\boldsymbol{b}^{[3]} &= \boldsymbol{b}^{[3]} - \eta\frac{\partial L(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \boldsymbol{b}^{[3]}}\\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1\begin{pmatrix} 0.3333 \\ -0.6667 \\ 0.3333 \end{pmatrix}.\\ &= \begin{pmatrix} 0.0667 \\ 0.1667 \\ 0.0667 \end{pmatrix}\end{aligned}$$

2. Reusing the computations from the previous exercise do a **gradient descent** update (with learning rate $\eta = 0.1$) for the batch with the training example from the a) and the following:

$$\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 10 & 0 \end{bmatrix}^\top, \quad \boldsymbol{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

**Solution:** Notice that the derivative of the sum is equal to the sum of the derivatives. For this reason, all we have to do is to compute the derivative for the new example and the final gradient will be the sum of both individual derivatives: the one for the new example and the one from the previous exercise.

## Question 3

Now it's time to train a multi-layer perceptron on real data and see what happens.

1. Load the UCI handwritten digits dataset using `scikit-learn`:

```
from sklearn.datasets import load_digits
data = load_digits()
```

This is a dataset containing 1797 8x8 input images of digits, each corresponding to one out of 10 output classes. You can print the dataset description and visualize some input examples with:

```
print(data.DESCR)

import matplotlib.pyplot as plt
plt.gray()
for i in range(10):
    plt.matshow(data.images[i])
plt.show()
```

Randomly split this data into training (80%) and test (20%) partitions. This can be done with:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
```

2. Run your implementation of the multi-layer perceptron on this dataset for 100 epochs, using stochastic gradient descent with $\eta = 0.001$. Use a single hidden layer with 50 hidden units. Measure the training and test accuracy.

3. Use `scikit-learn`'s implementation of a single-layer multi-layer perceptron. This can be done with

```
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(hidden_layer_sizes=(50),
                    activation='tanh',
                    solver='sgd',
                    learning_rate='constant',
                        learning_rate_init=0.001,
                        nesterovs_momentum=False,
                        random_state=1,
                        max_iter=1000)
    clf.fit(X_train, y_train)
    print(clf.score(X_train, y_train))
    print(clf.score(X_test, y_test))
```

Compare the resulting accuracies.