

Deep Learning (IST, 2022-23)

Practical 4: Multilayer Perceptron and Backpropagation

André Martins, Andreas Wichert, Luis Sá-Couto, Margarida Campos

Pen-and-Paper Exercises

The following questions should be solved by hand. You can use, of course, tools for auxiliary numerical computations.

Question 1

Consider a network with four layers with the following numbers of units: 4, 4, 3, 3 - including the input and output layers. 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:

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

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

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

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

We start by writting the connection weights and the biases:

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

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

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

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

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

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

We are now ready to do forward propagation:

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

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

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

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

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

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

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

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

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:

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

Gradients of hidden layer parameters:

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

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

Gradients of hidden layer below:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Now the second:

$$\begin{aligned}\mathbf{W}^{[2]} &= \mathbf{W}^{[2]} - \eta \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{W}^{[2]}} \\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} -0.01413 & -0.01413 & -0.01413 & -0.01413 \\ -0.01413 & -0.01413 & -0.01413 & -0.01413 \\ -0.01413 & -0.01413 & -0.01413 & -0.01413 \end{pmatrix} \\ &= \begin{pmatrix} 0.101413 & 0.101413 & 0.101413 & 0.101413 \\ 0.101413 & 0.101413 & 0.101413 & 0.101413 \\ 0.101413 & 0.101413 & 0.101413 & 0.101413 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\mathbf{b}^{[2]} &= \mathbf{b}^{[2]} - \eta \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{b}^{[2]}} \\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} -0.04850 \\ -0.04850 & -0.04850 \end{pmatrix} \\ &= \begin{pmatrix} 0.10485 \\ 0.10485 \\ 0.10485 \end{pmatrix}\end{aligned}$$

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

$$\begin{aligned}\mathbf{W}^{[3]} &= \mathbf{W}^{[3]} - \eta \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{W}^{[3]}} \\ &= \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} 0.03496 & 0.03496 & 0.03496 \\ -0.17825 & -0.17825 & -0.17825 \\ 0.03496 & 0.03496 & 0.03496 \end{pmatrix} \\ &= \begin{pmatrix} 0.0965 & 0.0965 & 0.0965 \\ 0.1178 & 0.1178 & 0.1178 \\ 0.0965 & 0.0965 & 0.0965 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\mathbf{b}^{[3]} &= \mathbf{b}^{[3]} - \eta \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{b}^{[3]}} \\ &= \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} - 0.1 \begin{pmatrix} 0.16396 \\ -0.83604 \\ 0.16396 \end{pmatrix} \\ &= \begin{pmatrix} 0.0836 \\ 0.1836 \\ 0.0836 \end{pmatrix}\end{aligned}$$

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:

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

Solution: Recall the squared loss:

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

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:

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

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 $\mathbf{z} \in \mathbb{R}^d$ (logits) and outputs a vector $\mathbf{p} \in \mathbb{R}^d$ (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 \mathbf{y} (a one-hot vector), the cross-entropy loss is:

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

The gradient of this loss is:

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

We start by writing the connection weights and the biases:

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

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

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

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

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

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

We are now ready to do forward propagation:

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

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

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

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

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

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

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

Recall the cross-entropy loss:

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

Gradient in the output layer:

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

Gradients of hidden layer parameters:

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

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

Gradients of hidden layer below:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Now the second:

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

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

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

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

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

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:

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

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.

Programming Exercises

The following exercises should be solved using Python, you can use the corresponding practical's notebook for guidance.

1. Solve **Question 1** from the Pen-and-Paper exercises with Python.
2. Solve **Question 2** from the Pen-and-Paper exercises with Python.
 - (a) Create functions to compute forward and backpropagation and gradient updates for an MLP with any number of hidden units.
3. Now it's time to try multi-class logistic regression on real data and see what happens. Load the UCI handwritten digits dataset using `scikit-learn`. This is a dataset containing 1797 8x8 input images of digits, each corresponding to one out of 10 output classes.
 - (a) Randomly split this data into training (80%) and test (20%) partitions.
 - (b) Implement a function to do an epoch of MLP training. You can make use of the previously defined functions.
 - (c) Consider a MLP with a single hidden layer of 50 units and a learning rate of 0.001. Run 100 epochs of your algorithm on the training data.
 - (d) Compute accuracies on both train and test sets
 - (e) Run Sklearn's implementation of the MLP and compare the resulting accuracies. never