Benjamin Grewe, Matthew Cook, Giacomo Indiveri, Daniel Kiper, Wolfger von der Behrens, Valerio Mante
Lecture 12
Asst: Karla Burelo, Vanessa Leite, Nicoletta Risi {kburel, vanessa, nicoletta}@ini.uzh.ch

Solution 12.1: Feed-forward Networks and Backpropagation

1. The error function for a single training sample is

$$E(S) = \frac{1}{2}\left(f(x_1 w_1 + x_2 w_2 + w_0) - d\right)^2 . \tag{1}$$

For general activation functions $f(a)$ there is no way to find an analytical solution (w_1, w_2, w_0) that minimizes the error. However, since we assume that $f(a)$ is differentiable, we can perform a gradient descent to find a local minimum of $E(S)$. For that, we regard the error function to be a function of the weights:

$$E(w_1, w_2, w_0) = \frac{1}{2}(f(x_1 w_1 + x_2 w_2 + w_0) - d)^2. \tag{2}$$

Making repeated use of the chain rule ($\frac{\partial x}{\partial y} = \frac{\partial x}{\partial z}\frac{\partial z}{\partial y}$), we can now compute the partial gradients with respect to any of the weights, *e.g.*,

$$\frac{\partial E(w_1, w_2, w_0)}{\partial w_1} = (f(x_1 w_1 + x_2 w_2 + w_0) - d) \cdot f'(x_1 w_1 + x_2 w_2 + w_0) \cdot x_1, \tag{3}$$

$$\frac{\partial E(w_1, w_2, w_0)}{\partial w_2} = (f(x_1 w_1 + x_2 w_2 + w_0) - d) \cdot f'(x_1 w_1 + x_2 w_2 + w_0) \cdot x_2, \tag{4}$$

and

$$\frac{\partial E(w_1, w_2, w_0)}{\partial w_0} = (f(x_1 w_1 + x_2 w_2 + w_0) - d) \cdot f'(x_1 w_1 + x_2 w_2 + w_0). \tag{5}$$

Note that for the derivation of these gradients we don't need to know $f(a)$ – it is enough if we know that it is differentiable.

If we had more than one training sample we would simply sum up all partial gradients over all training samples (the derivative of a sum is the sum of its derivatives).

2. The output of this simple network is

$$y(x_1, x_2, x_3) = f(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) \tag{6}$$

and the corresponding error function

$$E(w_{11}, w_{12}, w_{10}, w_{21}, w_{22}, w_{20}) = \frac{1}{2}(f(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) - d)^2. \tag{7}$$

The derivative to w_{22} is

$$\begin{aligned} \frac{\partial E}{\partial w_{22}} =& (f(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) - d) \cdot & (8)\\ & f'(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) \cdot & (9)\\ & f(x_2 w_{11} + x_3 w_{12} + w_{10}). & (10) \end{aligned}$$

The derivative to w_{11} is

$$\begin{aligned} \frac{\partial E}{\partial w_{11}} =& (f(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) - d) \cdot & (11)\\ & f'(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) \cdot & (12)\\ & f'(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} \cdot & (13)\\ & x_2 & (14) \end{aligned}$$

and to w_{12}

$$\begin{aligned} \frac{\partial E}{\partial w_{12}} =& (f(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) - d) \cdot & (15)\\ & f'(x_1 w_{21} + f(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} + w_{20}) \cdot & (16)\\ & f'(x_2 w_{11} + x_3 w_{12} + w_{10}) w_{22} \cdot & (17)\\ & x_3. & (18) \end{aligned}$$

The first two lines are identical in each gradient and can be reused once computed. The last line of each gradient ((10), (14), and (18)) is the value of the current input that is multiplied by the respective weight. Between the gradients for w_{11} and w_{12} only the input part is changing.

In general, the non-input part of the gradients is the same for all weights of one unit and is called the *delta term* or *error term*. In our example, the delta term of the right unit consists of lines (8) and (9). The delta term of the left unit consists of lines (11), (12), and (13) or, equivalently, of lines (15), (16), and (17). The delta term of a lower[1] unit can be obtained from the delta terms of the higher units it projects to: the higher delta term is just multiplied by the weight connecting both units and by the derivative of the activation function of the lower unit. In our example, the delta term of the left unit is the delta term of the right unit (lines (8) and (9)) multiplied by $f'(x_2w_{11} + x_3w_{12} + w_{10})w_{22}$, *i.e.*, by the derivative of the activation function of the left unit and the weight connecting the units (see line (13) or (17)).

Thus, it is sufficient to compute the delta terms for each unit to get the derivatives by the weights. The fact that the delta terms represent the error contributed by each unit and that they are computable recursively from higher units gave this learning procedure the name *error back propagation*.

3. We can think of the delta terms as travelling back through the wires. Whenever they leave a neuron through an input, they are multiplied by the weight of the input. Whenever they enter a neuron through its output, they are multiplied by the derivative of the activation function of that input.

   Whenever the output of one neuron is used several times, all delta terms that meet at a junction are simply added.

[1] Here, *lower* means closer to the input. In our example, the left unit is lower than the right unit.