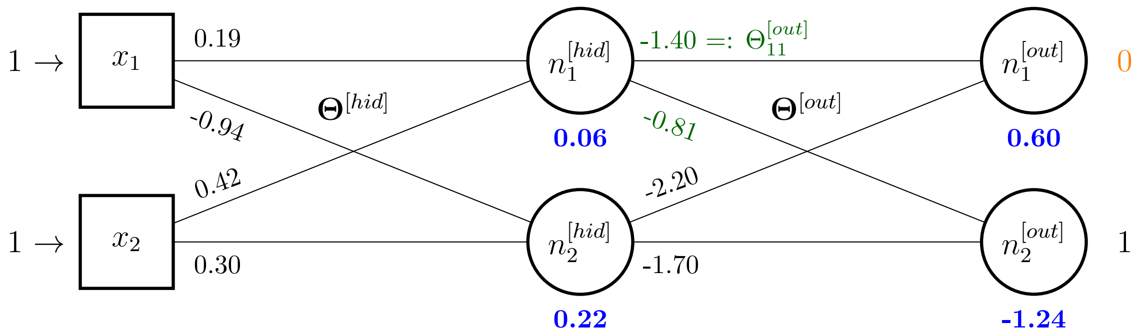
W3WI DS304.1 Applied Machine Learning Fundamentals

Derivation of the Backpropagation Formulas by Example

Input to the network: $\boldsymbol{x} := (1,1)^\intercal$. Desired output: $\boldsymbol{y} := (0,1)^\intercal$. The network is depicted in the following figure:



The following table lists the preactivations and activations of all neurons (result of the forward pass through the network):

Neuron	**Preactivation** p	**Activation** z	**Activation function** $g(\cdot)$
$n_1^{[hid]}$	0.61	0.61	ReLU
$n_2^{[hid]}$	-0.64	0.00	ReLU
$n_1^{[out]}$	-0.85	0.30	Sigmoid σ
$n_2^{[out]}$	-0.49	0.38	Sigmoid σ

Step 1) Computation of the error gradient in the output layer: In the following we are going to use the least squares error given by $\mathcal{J}(\boldsymbol{\Theta}) := \sum_{k=1}^{K} \left(z_{n_k^{[out]}} - y_k\right)^2$.

$$\frac{\partial \mathcal{J}(\boldsymbol{\Theta})}{\partial z_{n_k^{[out]}}} := 2 \cdot \left(z_{n_k^{[out]}} - y_k\right) \tag{1}$$

Example for neuron $n_1^{[out]}$: (see bold face blue number below neuron in figure above)

$$\frac{\partial \mathcal{J}(\boldsymbol{\Theta})}{\partial z_{n_1^{[out]}}} = 2 \cdot (0.30 - 0) = 0.60$$

Step 2) Computation of the error gradient in the hidden layer:

$$\frac{\partial \mathcal{J}(\boldsymbol{\Theta})}{\partial z_{n_t^{[hid]}}} := \sum_{k=1}^{K} \overbrace{\frac{\partial \mathcal{J}(\boldsymbol{\Theta})}{\partial z_{n_k^{[out]}}}}^{\text{see step } \mathbf{1)}} \cdot g'\left(p_{n_k^{[out]}}\right) \cdot \Theta_{kt}^{[out]} \tag{2}$$

Example for neuron $n_1^{[hid]}$: (see bold face blue number below neuron in figure above)

$$\begin{aligned}\frac{\partial \mathcal{J}(\boldsymbol{\Theta})}{\partial z_{n_1^{[hid]}}} &= \Big[0.60 \cdot \overbrace{\sigma(-0.85) \cdot (1 - \sigma(-0.85))}^{\text{Derivative Sigmoid}} \cdot (-1.4)\Big] \\ &\quad + \Big[-1.24 \cdot \sigma(-0.49) \cdot (1 - \sigma(-0.49)) \cdot (-0.81)\Big] \\ &= 0.06\end{aligned}$$

Step 3) Computation of the weight gradient in the output layer:

$$\frac{\partial \mathcal{J}(\Theta)}{\partial \Theta_{kt}^{[out]}} := \overbrace{\frac{\partial \mathcal{J}(\boldsymbol{\Theta})}{\partial z_{n_k^{[out]}}}}^{\text{see } \mathbf{1)}} \cdot g'(p_{n_k^{[out]}}) \cdot z_{n_t^{[hid]}} \tag{3}$$

Example for weight $\Theta_{11}^{[out]}$:

$$\begin{aligned}\frac{\partial \mathcal{J}(\Theta)}{\partial \Theta_{11}^{[out]}} &= 0.60 \cdot \overbrace{\sigma(-0.85) \cdot (1 - \sigma(-0.85))}^{\text{Derivative Sigmoid}} \cdot 0.61 \\ &= 0.077\end{aligned}$$

Step 4) Computation of the weight gradient in the hidden layer:

The weight gradient in the hidden layer is computed analogously to step 3. However, we use the input to the network instead of $g(p_{n_t^{[hid]}})$.

Step 5) Update the network parameters:

The parameters are updated according to the gradient descent update rule.

Example for weight $\Theta_{11}^{[out]}$ with learning rate $\alpha := 0.1$:

$$\begin{aligned}\Theta_{11}^{[out]} &\longleftarrow \Theta_{11}^{[out]} - \alpha \cdot \overbrace{\frac{\partial \mathcal{J}(\Theta)}{\partial \Theta_{11}^{[out]}}}^{\text{see } \mathbf{3)}} \\ &\longleftarrow -1.40 - 0.1 \cdot 0.077 \\ &\longleftarrow -1.4077\end{aligned}$$

Derivation of the formulas:

It is essential to remember the **chain rule** for derivatives: Let $f, g, h : \mathbb{R} \to \mathbb{R}$ be real-valued functions. The derivative of $f(g(h(x)))$ is given by:

$$\frac{\mathrm{d}}{\mathrm{d}x} f(g(h(x))) = \frac{\mathrm{d}f}{\mathrm{d}g} \cdot \frac{\mathrm{d}g}{\mathrm{d}h} \cdot \frac{\mathrm{d}h}{\mathrm{d}x} = f'(g(h(x))) \cdot g'(h(x)) \cdot h'(x) \tag{4}$$

The cost function $\mathcal{J}$ of the neural network depicted above is given by:

$$\mathcal{J}(\mathbf{\Theta}) := \sum_{k=1}^{2} \left(z_{n_k^{[out]}} - y_k\right)^2 = \left(z_{n_1^{[out]}} - y_1\right)^2 + \left(z_{n_2^{[out]}} - y_2\right)^2$$

where:

$$z_{n_1^{[out]}} := \sigma\left(p_{n_1^{[out]}}\right)$$

$$z_{n_2^{[out]}} := \sigma\left(p_{n_2^{[out]}}\right)$$

$$p_{n_1^{[out]}} := z_{n_1^{[hid]}} \cdot \Theta_{11}^{[out]} + z_{n_2^{[hid]}} \cdot \Theta_{12}^{[out]}$$

$$p_{n_2^{[out]}} := z_{n_1^{[hid]}} \cdot \Theta_{21}^{[out]} + z_{n_2^{[hid]}} \cdot \Theta_{22}^{[out]}$$

$$z_{n_1^{[hid]}} := \mathrm{ReLU}\left(p_{n_1^{[hid]}}\right)$$

$$z_{n_2^{[hid]}} := \mathrm{ReLU}\left(p_{n_2^{[hid]}}\right)$$

$$p_{n_1^{[hid]}} := x_1 \cdot \Theta_{11}^{[hid]} + x_2 \cdot \Theta_{12}^{[hid]}$$

$$p_{n_2^{[hid]}} := x_1 \cdot \Theta_{21}^{[hid]} + x_2 \cdot \Theta_{22}^{[hid]}$$

Let us now compute the partial derivative of the cost function $\mathcal{J}$ with respect to the model parameter $\Theta_{11}^{[out]}$ (in the output layer). We notice that this parameter is only relevant for the computation of $z_{n_1^{[out]}}$. Therefore, the second addend of the cost function is constant with respect to this parameter and its derivative is therefore equal to zero.

$$\frac{\partial \mathcal{J}(\mathbf{\Theta})}{\partial \Theta_{11}^{[out]}} = \underbrace{\overbrace{\frac{\partial \mathcal{J}(\mathbf{\Theta})}{\partial z_{n_1^{[out]}}}}^{\substack{\text{Error gradient} \\ \text{cf. (1)}}} \cdot \frac{\partial z_{n_1^{[out]}}}{\partial p_{n_1^{[out]}}} \cdot \frac{\partial p_{n_1^{[out]}}}{\partial \Theta_{11}^{[out]}}}_{\substack{\text{Weight gradient} \\ \text{cf. (3)}}} \tag{5}$$

We compute the partial derivatives appearing in the equation:

$$\frac{\partial \mathcal{J}(\mathbf{\Theta})}{\partial z_{n_1^{[out]}}} = 2 \cdot \left(z_{n_1^{[out]}} - y_1\right)$$

$$\frac{\partial z_{n_1^{[out]}}}{\partial p_{n_1^{[out]}}} = \sigma\left(p_{n_1^{[out]}}\right) \cdot \left(1 - \sigma\left(p_{n_1^{[out]}}\right)\right)$$

$$\frac{\partial p_{n_1^{[out]}}}{\partial \Theta_{11}^{[out]}} = z_{n_1^{[hid]}}$$

Please compare the result with equation (3).

Let us now compute the weight gradient for $\Theta_{11}^{[hid]}$. We notice that the parameter is relevant to both addends in the cost function.

$$\frac{\partial \mathcal{J}(\mathbf{\Theta})}{\partial \Theta_{11}^{[hid]}} = \underbrace{\overbrace{\left(\sum_{k=1}^{2} \frac{\partial \mathcal{J}(\mathbf{\Theta})}{\partial z_{n_k^{[out]}}} \cdot \frac{\partial z_{n_k^{[out]}}}{\partial p_{n_k^{[out]}}} \cdot \frac{\partial p_{n_k^{[out]}}}{\partial z_{n_1^{[hid]}}} \right)}^{\substack{\text{Error gradient} \\ \text{cf. (2)}}} \cdot \frac{\partial z_{n_1^{[hid]}}}{\partial p_{n_1^{[hid]}}} \cdot \frac{\partial p_{n_1^{[hid]}}}{\partial \Theta_{11}^{[hid]}}}_{\text{Weight gradient}} \tag{6}$$

Again we compute the partial derivatives occurring in the formula:

$$\frac{\partial \mathcal{J}(\mathbf{\Theta})}{\partial z_{n_k^{[out]}}} = 2 \cdot \left(z_{n_k^{[out]}} - y_k \right)$$

$$\frac{\partial z_{n_k^{[out]}}}{\partial p_{n_k^{[out]}}} = \sigma\left(p_{n_k^{[out]}}\right) \cdot \left(1 - \sigma\left(p_{n_k^{[out]}}\right)\right)$$

$$\frac{\partial p_{n_k^{[out]}}}{\partial z_{n_1^{[hid]}}} = \Theta_{k1}^{[out]}$$

$$\frac{\partial z_{n_1^{[hid]}}}{\partial p_{n_1^{[hid]}}} = \mathrm{ReLU}'\left(p_{n_1^{[hid]}}\right) = \begin{cases} 0 & \text{if } p_{n_1^{[hid]}} \leq 0 \\ 1 & \text{if } p_{n_1^{[hid]}} > 0 \end{cases}$$

$$\frac{\partial p_{n_1^{[hid]}}}{\partial \Theta_{11}^{[hid]}} = x_1$$