Backpropagation

1. Why do we need to rescale the input data before training an MLP?

   MLPs use metric information (distances) to determine di error. Therefore, dimensions whose scale is far larger than others will dominate the error, making the smaller-scale features irrelevant.

2. How does the error propagate from the output layer to the hidden layer?

   Through the δ: $\delta_j=(z_j-t_j)z_j(1-z_j)$.

3. Why de we compute the gradient of the error?

   In order to perform gradient descent. The anti-gradient (the direction opposite to the gradient) is the steepest direction for descent.

4. Derive the update rule of the output neurons according to backpropagation.

   We aim to minimise the mean squared error, that is: $E=\frac{1}{2}(y-t)^2$, where y is the output of the MLP, and t is the desired class. The output $y=\sigma(a)$) is a sigmoid function, applied to the dot product between the weights of the output neuron and its inputs (including the bias input): $a=\sum_i w_i z_i$. We derive the gradient of the error with respect to each weight by applying the chain rule: $\frac{\partial E}{\partial w_i}=\frac{\partial E}{\partial a}\frac{\partial a}{\partial w_i}=(y-t)\,y(1-y)z_i$,
   where we used the derivative of the sigmoid: $(\sigma(x))'=\sigma(x)(1-\sigma(x))$. Substituting the gradient into the general update rule for gradient descent: $x_{t+1}=x_t-\eta\nabla f(x_t)$ we obtain the update rule for the output weights:
   $w_{t+1}=w_t-\eta(y-t)\,y(1-y)\,z$

5. Derive the update rule of the hidden neurons according to backpropagation.

   For a hidden neuron j, we need to compute its δ:
   $\delta_j=\frac{\partial E}{\partial a_j}=\sum_k \frac{\partial E}{\partial a_k}\frac{\partial a_k}{\partial a_j}=\sum_k \delta_k w_{jk} z_j(1-z_j)$, where the a_k are the output of the neurons that follow neuron j in the network, $\frac{\partial a_k}{\partial a_j}=\sum_j w_{jk}\sigma(a_j)=w_{jk}\sigma(a_j)(1-\sigma(a_j))$. ,
   and $\sigma(a_j)=z_j$. Then, we can chain this with the derivative of a_j with respect to any of the weights connected to its input, w_{ij} : $\frac{\partial a_j}{\partial w_{ij}}=z_i$. So the gradient of the error with

respect to w_{ij} is $\frac{\partial E}{\partial w_{ij}} = z_i \sum_k \delta_k w_{jk} z_j (1 - z_j)$, which leads to the update rule:

$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta z_i \sum_k \delta_k w_{jk} z_j (1 - z_j)$.