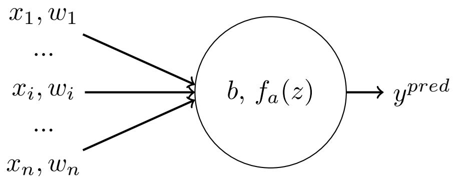
Neuron

Schema of a neuron:



To compute predicted value by the neuron:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

$$y^{pred} = f_a(z) = f_a(\sum_{i=1}^{n} w_i x_i + b) \tag{2}$$

To compute the cost function for m training samples:

$$\begin{aligned} C(w, b) &= \frac{1}{2m} \sum_{j=1}^{m} \left(y_j^{pred} - y_j\right)^2 \\ &= \frac{1}{2m} \sum_{j=1}^{m} \left(f_a(z)_j - y_j\right)^2 \end{aligned} \tag{3}$$

Training the neuron consists on minimizing the cost function. The method used to minimize the cost function is Gradient Descent (`https://en.wikipedia.org/wiki/Gradient_descent`).

$$\mathbf{w}^{n+1} = \mathbf{w}^n - \alpha \nabla C(\mathbf{w}^n, b^n) \tag{4}$$

$$b^{n+1} = b^n - \alpha \nabla C(\mathbf{w}^n, b^n) \tag{5}$$

The equations above compute the gradient descent for the weights and for the bias. α is the so called Learning Rate.
To compute the gradient of the cost function for m training samples, it is necessary to compute the partial derivatives with respect $\mathbf{w}$ and b.
Partial derivative with respect $\mathbf{w}$:

$$\begin{aligned} \frac{\delta C(\mathbf{w}, b)}{\delta w_i} &= \left(\frac{1}{2m} \left(f_a(z) - y\right)^2\right)' \\ &= \frac{1}{m} \left(f_a(z) - y\right) f_a'(z) z' \\ &= \frac{1}{m} \sum_{j=1}^{m} \left(f_a(\mathbf{w} \cdot \mathbf{x} + b) - y_j\right) f_a'(\mathbf{w} \cdot \mathbf{x} + b)(x_i)_j \end{aligned} \tag{6}$$

In a similar way, the partial derivative with respect b:

$$\frac{\delta C(\mathbf{w}, b)}{\delta b} = \frac{1}{m} \sum_{j=1}^{m} \left(f_a(\mathbf{w} \cdot \mathbf{x} + b) - y_j\right) f_a'(\mathbf{w} \cdot \mathbf{x} + b) \tag{7}$$

Activation functions

Sigmoid

Sigmoid function is:

$$\sigma(x) = \frac{1}{1 + \exp^x} \tag{8}$$

Sigmoid derivative is:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x)) \tag{9}$$

ReLU

ReLU function is:

$$ReLU(x) = \max(0, x) \tag{10}$$

The derivative of the ReLU function is:

$$ReLU'(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{11}$$