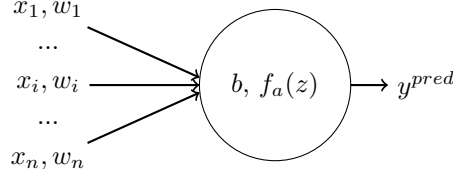


Neuron

Schema of a neuron:



To compute predicted value by the neuron:

$$z = \sum_{i=1}^n w_i x_i + b \quad (1)$$

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

To compute the cost function for m training samples:

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

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).

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

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

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} :

$$\begin{aligned} \frac{\delta C(\mathbf{w}, b)}{\delta w_i} &= \left(\frac{1}{2m} (f_a(z) - y)^2 \right)' \\ &= \frac{1}{m} (f_a(z) - y) f'_a(z) z' \\ &= \frac{1}{m} \sum_{j=1}^m (f_a(\mathbf{w}^T \cdot \mathbf{x} + b) - y_j) f'_a(\mathbf{w}^T \cdot \mathbf{x} + b) (x_i)_j \end{aligned} \quad (6)$$

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

$$\frac{\delta C(\mathbf{w}, b)}{\delta b} = \frac{1}{m} \sum_{j=1}^m (f_a(\mathbf{w}^T \cdot \mathbf{x} + b) - y_j) f'_a(\mathbf{w}^T \cdot \mathbf{x} + b) \quad (7)$$

Activation functions

Sigmoid

Sigmoid function is:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

Sigmoid derivative is:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x)) \quad (9)$$

ReLU

ReLU function is:

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

The derivative of the ReLU function is:

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

Network

