One layer neural-nets (perceptrons)

The input to a neural net is denoted by the variables $x_1, \ldots, x_n$. The output of a single unit perceptron is denoted by O. It is obtained by applying a function g to a weighted sum of the input:

$$\begin{array}{rcl} O & = & g(h) \\ h & = & \sum_{i=1}^{n} w_i x_i \end{array} \tag{1}$$

The input to the learning process is a set of training examples. An example is a set of values of $x_1, \ldots, x_n$ with a desired response y. Thus, an example e_t is given by a set of $n+1$ numbers:

$$e_t = (x_1^t, \ldots, x_n^t, y^t)$$

The objective of a learning algorithm is to obtain a set of weights $w_1, \ldots, w_n$ such that the output O (as given above by Equation 1. is a good approximation of the desired response y for all the examples. Specifically, we try to minimize the following error:

$$E = \sum_t (O^t - y^t)^2$$

where the summation is over all the examples e_t, O^t is the value of O that is obtained by putting $x_i = x_i^t$ in Equation 1, and y^t is the desired response of the example e_t.

The error E can be minimized by the ϵ-step gradient descent algorithm. This algorithm is implemented by repeatedly changing the values of the weights $w_1, \ldots, w_n$ according to:

$$w_i \leftarrow w_i + \epsilon \sum_t (y^t - O^t) g'(h^t) x_i^t \qquad \text{for} \quad i = 1, \ldots, n$$

A simplified implementation of this rule, motivated by the generic on-line learning protocol, changes the values of $w_1, \ldots, w_n$ for a single example $e = (x_1, \ldots, x_n, y)$ according to:

$$w_i \leftarrow w_i + \epsilon (y - O) g'(h) x_i \qquad \text{for} \quad i = 1, \ldots, n$$

where O is given by Equation 1. Since the term $(y - O)g'(h)$ is independent of i we usually write it as:

$$\begin{array}{l} w_i \leftarrow w_i + \epsilon \delta x_i \qquad \text{for} \quad i = 1, \ldots, n, \\ \text{where} \quad \delta = (y - O) g'(h) \end{array}$$

In this form it is called *the delta rule*.

The delta rule for a linear unit

The simplest delta rule is obtained by choosing:

$$g(h) = h.$$

In this case we have:

$$O = \sum_{i=1}^{n} w_i x_i$$

and the delta rule is:

$$\begin{array}{l} w_i \leftarrow w_i + \epsilon \delta x_i \qquad \text{for} \quad i = 1, \ldots, n, \\ \text{where} \quad \delta = (y - O) \end{array}$$

This update rule is sometimes called *The Adaline Rule*, or *The Widrow-Hoff Rule*.

The delta rule for a sigmoidal unit

The delta rule cannot be applied directly when the function g is a threshold. However, it can be applied when the threshold is approximated by a sigmoid. The sigmoid function is given by:

$$g(h) = \frac{1}{1 + \exp\{-2\beta h\}} \qquad \beta > 0$$

and we have:

$$g'(h) = 2\beta \frac{\exp\{-2\beta h\}}{(1 + \exp\{-2\beta h\})^2} = 2\beta g(1 - g)$$

Therefore, in this case we have:

$$h = \sum_{i=1}^{n} w_i x_i$$

and the delta rule is:

$$\begin{array}{l} \text{for} \quad i = 1, \ldots, n, \quad w_i \leftarrow w_i + \epsilon \delta x_i \\ \text{where} \quad \delta = O(1 - O)(y - O) \end{array}$$

The delta rule for hyperbolic tangent

The hyperbolic tangent function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Take:

$$g(h) = \tanh(\beta h)$$

then:

$$g' = \beta(1 - g^2)$$

This gives:

$$\delta = (1 - O^2)(y - O)$$