Learning and Adaptation

3. Delta Learning Rule

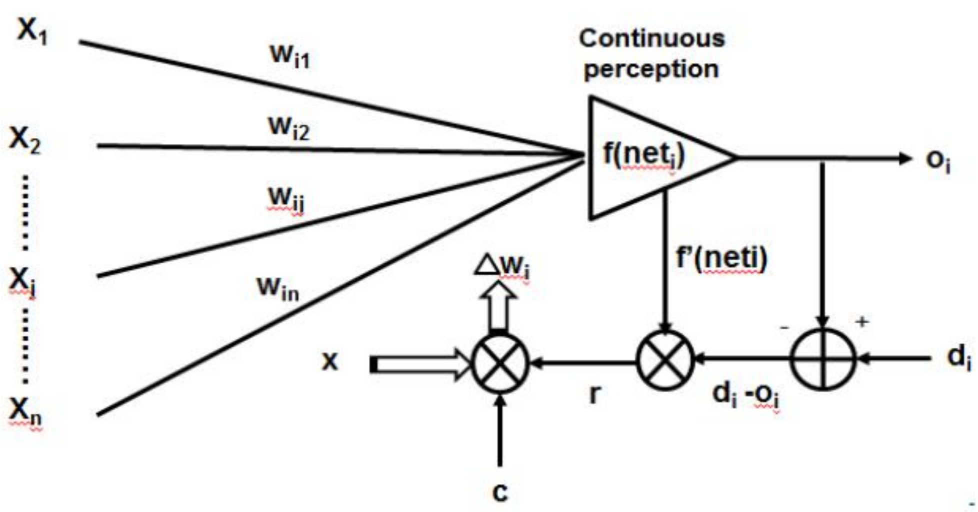
The Algorithm of Delta Learning Rule can be describing as follows:

Delta Learning Rule

- The delta learning rule is only valid for continuous activation functions.
- It is a supervised training mode.
- The single weight w is updated using the following increment:

$$\boxed{\Delta \mathbf{w}_i = c(d_i - o_i)f'(\text{net}_i)\mathbf{x}}$$

- Where f'(neti) is the derivate of the activation function f(neti).



Example 1:

$$\left[x1 = \begin{pmatrix} 1 \\ -2 \\ 0 \\ -1 \end{pmatrix} \quad x2 = \begin{pmatrix} 0 \\ 1.5 \\ -0.5 \\ -1 \end{pmatrix} \quad x3 = \begin{pmatrix} -1 \\ 1 \\ 0.5 \\ -1 \end{pmatrix} \quad w1 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0.5 \end{pmatrix} \right]$$

c = 0.1 and $\lambda = 1$. The desired responses for x_1, x_2, and x_3 are $d_1 = -1$, $d_2 = -1$, $d_3 = 1$. Use continues bipolar activation function f(net)= $\frac{2}{1+\exp(-\lambda net)} - 1$

F'(net)= $\frac{1}{2}(1 - Oi^2)$

$$f'(net) = \frac{(1+\exp(-\lambda net)(0)-(2)(-\lambda \exp(-\lambda net)}{(1+\exp(-\lambda net)^2)} - 0$$

$$= \frac{2\lambda \exp(-\lambda net)}{(1+\exp(-\lambda net)^2)}$$

$$= \frac{\lambda}{2}[1 - \left(\frac{2}{1+\exp(-\lambda net)} - 1\right)^2]$$

$$= \frac{\lambda}{2}(1 - Oi^2)$$

For λ=*1*

F'(net)= $\frac{1}{2}(1 - Oi^2)$

Step 1 :-

Input is xl, desired output is dl :

$$Net^1 = w^{1t} * X^1 = [1\ -1\ 0\ 0.5] \begin{bmatrix} 1 \\ -2 \\ 0 \\ -1 \end{bmatrix} = 2.5$$

$$O^1 = f(net^1) = \frac{2}{1+e^{-2.5}} - 1 = 0.848 \quad di \neq Oi$$

We thus obtain updated weight vector

$$f'(net^1) = \frac{1}{2}[1-(O^1)^2] = = \frac{1}{2}[1-(0.848)^2] = 0.140$$

$$\Delta w^1 = c\,(d^1 - O^1) * f'(net^1) * X^1$$

$$= 0.1 * (-1 - 0.848)(0,140) * \begin{bmatrix} 1 \\ -2 \\ 0 \\ -1 \end{bmatrix} = \begin{pmatrix} -0.0258 \\ 0.0517 \\ 0 \\ 0.0258 \end{pmatrix}$$

$$W^{i+1} = \Delta w^i + W^i$$

$$W^2 = \Delta w^1 + W^1 = \begin{pmatrix} -0.0258 \\ 0.0517 \\ 0 \\ 0.0258 \end{pmatrix} + \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0.5 \end{bmatrix} = \begin{pmatrix} 0.974 \\ -0.948 \\ 0 \\ 0.526 \end{pmatrix}$$

Step 2 :-

Input is x2, desired output is d2 :

$$F\,(Net^2) = w^{2t} * X^2 = \begin{pmatrix} 0.974 \\ -0.948 \\ 0 \\ 0.526 \end{pmatrix}^T * \begin{bmatrix} 0 \\ 1.5 \\ -0.5 \\ -1 \end{bmatrix} = -1.948$$

$$O^2 = f\,(net^2) = \frac{2}{1+e\ \wedge 1.948} - 1 = -\,0.750 \quad di \neq Oi$$

We thus obtain updated weight vector

$$f\,'(net^2) = \frac{1}{2}\,[\,1-(O^2)^2] = = \frac{1}{2}\,[\,1-(-\,0.750)^2] = \ 0.218$$

$$\Delta w^2 = c\,(\,d^2 - O^2\,) * f\,'(net^2) * X^2$$

$$= 0.1 * (-1 + 0.750)\,0.218 * \begin{bmatrix} 0 \\ 1.5 \\ -0.5 \\ -1 \end{bmatrix} = \begin{pmatrix} 0 \\ -8.175 * 10^{-3} \\ 2.725 * 10^{-3} \\ 5.45 * 10^{-3} \end{pmatrix}$$

$$W^{\,i+1} = \Delta w^{\,i} + W^{\,i}$$

$$W^3 = \Delta w^2 + W^2 = \begin{pmatrix} 0 \\ -8.175 * 10^{-3} \\ 2.725 * 10^{-3} \\ 5.45 * 10^{-3} \end{pmatrix} + \begin{pmatrix} 0.974 \\ -0.948 \\ 0 \\ 0.526 \end{pmatrix} = \begin{pmatrix} 0.974 \\ -0.956 \\ 0.002 \\ 0.531 \end{pmatrix}$$

Step 3 :-

Input is x3, desired output is d3 :

$$F(Net^3) = w^{3t} * X^3 = \begin{pmatrix} 0.974 \\ -0.956 \\ 0.002 \\ 0.531 \end{pmatrix}^T * \begin{pmatrix} -1 \\ 1 \\ 0.5 \\ -0.5 \end{pmatrix} = -2.46$$

$$O^3 = f(net^3) = \frac{2}{1+e^{2.46}} - 1 = -0.842 \quad di \neq Oi$$

We thus obtain updated weight vector

$$f'(net^3) = \frac{1}{2}[1-(O^3)^2] = = \frac{1}{2}[1-(-0.842)^2] = 0.145$$

$$\Delta w^3 = c\,(d^3 - O^3) * f'(net^3) * X^3$$

$$= 0.1 * (1 + 0.842) * 0.145 * \begin{pmatrix} -1 \\ 1 \\ 0.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} -0.026 \\ 0.026 \\ 0.0133 \\ -0.0133 \end{pmatrix}$$

$$W^{i+1} = \Delta w^i + W^i$$

$$W^4 = \Delta w^3 + W^3 = \begin{pmatrix} -0.026 \\ 0.026 \\ 0.0133 \\ -0.0133 \end{pmatrix} + \begin{pmatrix} 0.974 \\ -0.956 \\ 0.002 \\ 0.531 \end{pmatrix} = \begin{pmatrix} 0.974 \\ -0.929 \\ 0.016 \\ 0.505 \end{pmatrix}$$

Example 2:

Perform two training steps of neural network, using the delta learning rule for λ = 1 and c = 0.25. Train the network using the following data pairs

$$\left(\mathbf{x}_1 = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix},\quad d_1 = -1\right),\quad \left(\mathbf{x}_2 = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix},\quad d_2 = 1\right)$$

The initial weights are W1 = [1 0 1]t and f(net) is bipolar continuous activation function.

Solution:

activation function $f(net) = \frac{2}{1+\exp(-\lambda net)} - 1$

$F'(net) = \frac{1}{2}(1 - Oi^2)$

$f\,(net) = ½[1-(Oi)^2)]$

$net_1 = W_1 * X_1 = [1\ 0\ 1] * \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = 1$

$O_1 = f(net_1) = 0.47$

Since $Sgn(O_1)$ not equal d1 (-1) the correction is necessary

$F^{-}(net1) = 1/2(1-O_1)^2 = ½(1-(0.47)^2) = 0.39$

$W2 = W1 + c(d_1 - O_1)\, F^{-}\,(net_1) * X_1$

$W_2 = [1\ 0\ 1] + 0.25(-1-0.47)*0.39 * \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = [1\ 0\ 1] + (-\,0.14) * \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}$

$= \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.28 \\ 0 \\ 0.14 \end{bmatrix} = \begin{bmatrix} 0.72 \\ 0 \\ 1.14 \end{bmatrix}$

net2 = W_2* X_2 = $[0.72\ 0\ \ 1.14] * \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} = -0.42$

O_2= f(net_2) = -0.2

Since Sgn(O_2) not equal d2 (1) the correction is necessary

$F^-(net_2)=1/2(1-O^2) = ½ (1- (-0.2)^2) = 0.48$

$W_3=W_2 + c(d_2-O_2)\ F'(net_2)*X_2$

$$W_3 = \begin{bmatrix} 0.72 \\ 0 \\ 1.14 \end{bmatrix} + 0.25\ (1+0.2) * 0.48 * \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 0.72 \\ 0 \\ 1.14 \end{bmatrix} + \begin{bmatrix} 0.144 \\ -0.288 \\ -0.144 \end{bmatrix} = \begin{bmatrix} 0.864 \\ -0.288 \\ 0.996 \end{bmatrix}$$