

# Algorytm

Definiujemy w następujący sposób kolejne elementy zbioru uczącego:

$$p_1 = \{-1, -1, 1\}$$

$$p_2 = \{-1, 1, 1\}$$

$$p_3 = \{1, -1, 1\}$$

$$p_4 = \{1, 1, 1\}$$

$$t_1 = \{0\}$$

$$t_2 = \{1\}$$

$$t_3 = \{1\}$$

$$t_4 = \{0\}$$

1. Wybór  $\eta > 0$  (współczynnik uczenia),  $E_{max} > 0$  (maksymalny błąd jaki chcemy osiągnąć),  $C_{max} > 0$  (ilość kroków uczenia).
2. Losowy wybór początkowych wartości wag jako niewielkich liczb (na przykład z przedziału  $[-1, 1]$ );  $c := 0$ .
3.  $l := 0$ ,  $E := 0$ .
4. Podanie jednego z obrazów ze zbioru P na wejścia sieci.
5. Obliczenie sygnału wyjściowego sieci, czyli  $y_1^2$ .
6. Obliczenie sygnałów błęd

$$\delta_1^2 = (t - f(net_1^2))f'(net_1^2)$$

$$\delta_1^1 = \delta_1^2 w_{11}^2 f'(net_1^1)$$

$$\delta_2^1 = \delta_1^2 w_{12}^2 f'(net_2^1)$$

7. Uaktualnienie wartości wag według wzoru

$$w_{bc}^a = w_{bc}^a + \eta \delta_b^a x_c^a$$

8. Obliczenie błędu

$$E = E + \frac{1}{2}(t - y)^2$$

9. Jeśli  $l < ilość\_obrazów$  to  $l := l + 1$  i przejście do kroku 4.
10. Jeśli  $E < E_{max}$ , to kończymy algorytm. Jeśli  $c < C_{max}$ , to  $c := c + 1$  i przechodzimy do kroku 3. W przeciwnym razie kończymy algorytm.

