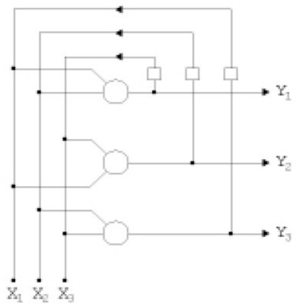
Sieć Hopfielda



Program „odtwarzający literki”.
Zbiór uczący jest postaci:
P = {p1, p2, p3, p4},
p_1 =11111 1-1-1-11 1-1-1-11 11111 1-1-1-11 1-1-1-11 1-1-1-11 (odpowiada literce A)
p_2 =11111 1-1-1-1-1 1-1-1-1-1 1-1-1-1-1 1-1-1-1-1 1-1-1-1-1 11111 (odpowiada literce C)
Analogicznie przedstawiają się literki X oraz I.

Algorytm:
1) Podanie na wejście sieci wektora x. Wektor x podawany jest na wejście sieci tylko raz, aby zainicjalizować jej działanie (np. wyliczenie wag). W kolejnych krokach rolę sygnału wejściowego pełnić będą sygnały sprzężenia zwrotnego.

Zapisujemy do pamięci wektory wzorcowe s_m, $m = 1, ..., p$ o składowych −1 lub +1. Wagi wyznaczamy według wzoru

$$w_{ij} = (1 - \delta_{ij}) \sum_{m=1}^{p} s_i^m s_j^m$$

gdzie delta jest delta Kroneckera:

$$\delta_{ij} = \begin{cases} 1 & gdy \quad i = j \\ 0 & gdy \quad i \neq j. \end{cases}$$

2) Obliczenie wartości sygnałów wyjściowych. Asynchroniczna aktualizacja stanów neuronów. Sygnały wyjściowe stają się nowym sygnałem wejściowym.
Funkcja wyjścia

$$y(t+1) = \begin{cases} +1 & gdy \quad net > 0 \\ y(t) & gdy \quad net = 0 \\ -1 & gdy \quad net < 0 \end{cases}$$

3) Porównanie poprzedniego i obecnego sygnału wejściowego (sprawdzamy stabilność sieci). Jeśli są one identyczne to kończymy działanie sieci (przechodzimy do „odtwarzania”). W przeciwnym razie powracamy do punktu 2.