POJĘCIE SZTUCZNEJ SIECI NEURONOWEJ

- **Sztuczna sieć neuronowa** jest to struktura rozproszona mająca zdolność do nauki.
- Strukturę tę stanowi zespół wzajemnie połączonych, w określony luźny sposób, prostych elementów — zwanych neuronami — których działanie jest analogiczne do zasad funkcjonowania biologicznych komórek nerwowych.

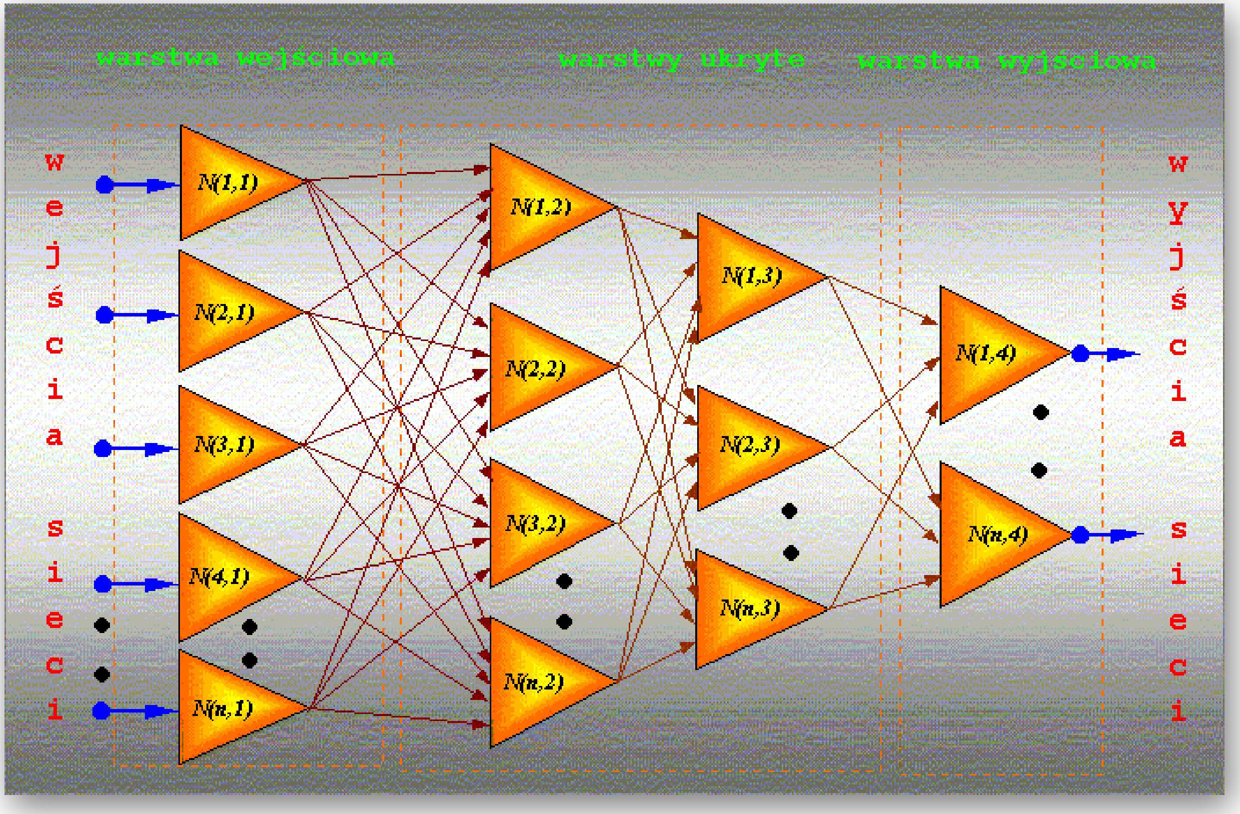
Podstawowy schemat jednokierunkowej wielowarstwowej sieci neuronowej:

Jednokierunkowa wielowarstwowa sieć neuronowa to odpowiednio połączone ze sobą neurony rozmieszczone w wielu warstwach.

W każdej sieci wielowarstwowej muszą istnieć co najmniej dwie warstwy:

wejściowa i **wyjściowa**.

Między nimi mogą się znajdować **warstwy ukryte**.



http://www.ai.c-labtech.net/sn/pod_prakt.html

- Każda sieć musi mieć co najmniej dwie warstwy: wejściową i wyjściową

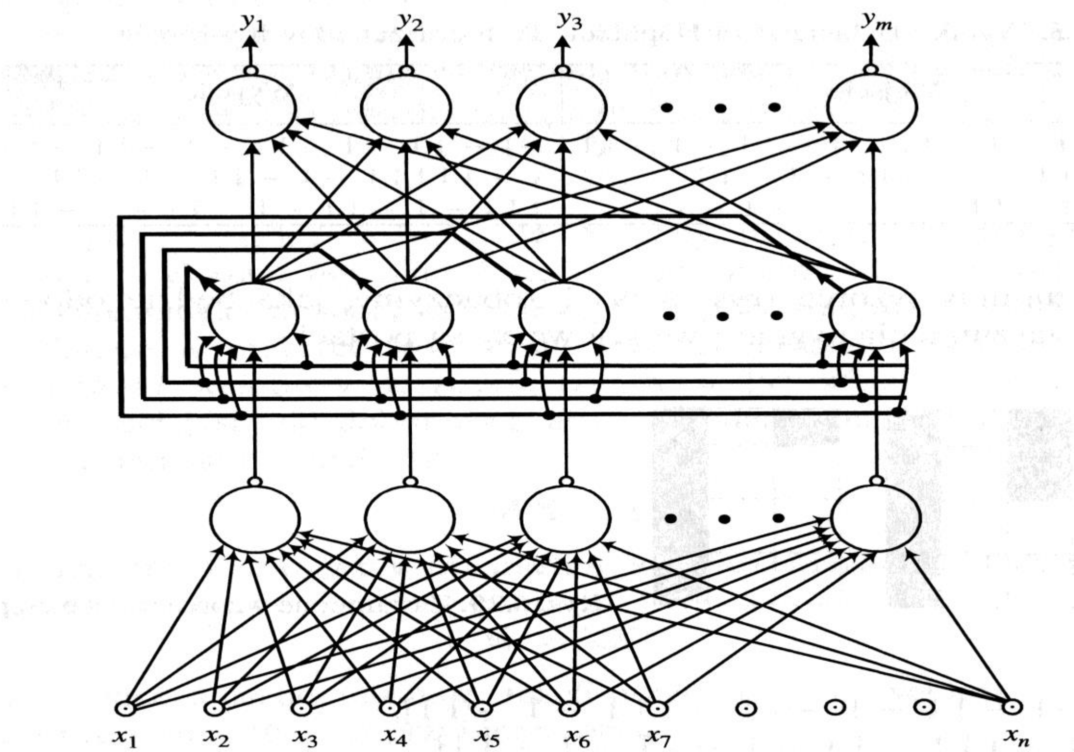
- **Warstwa wejściowa** wprowadza informacje do sieci (i nic więcej nie robi).

- **Warstwa wyjściowa** przetwarza informacje i wyprowadza je na zewnątrz.

- **Warstwy ukryte** przetwarzają informacje (Użytkownik sieci nie ma do nich dostępu).

Schemat rekurencyjnej wielowarstwowej sieci neuronowej:

Trójwarstwowa sieć Hamminga stosowana do klasyfikacji obrazów:

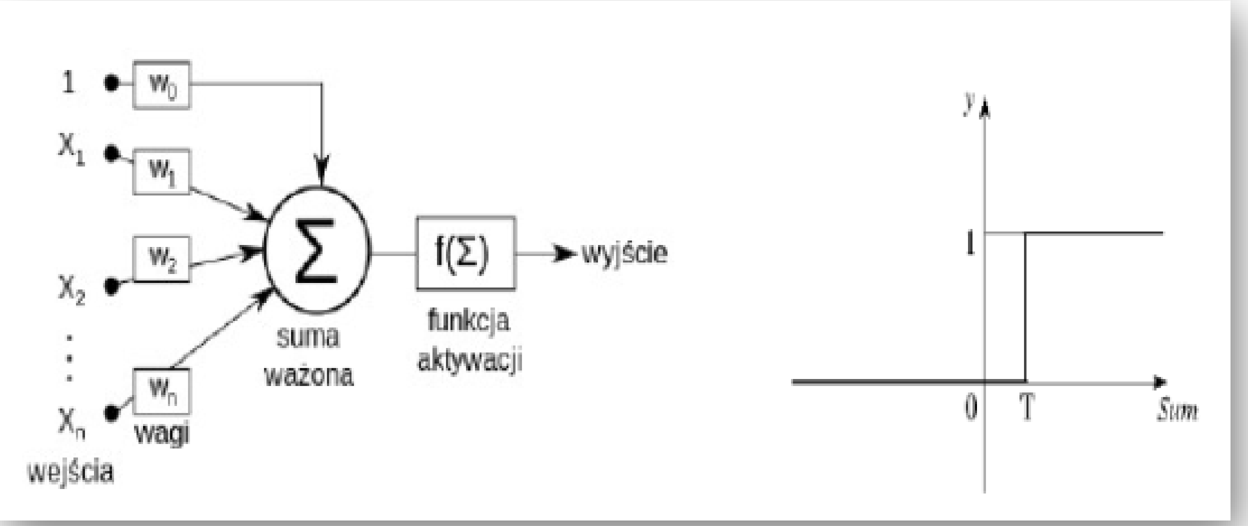


- W rekurencyjnej sieci neuronowej występują tzw. sprzężenia zwrotne, za pomocą których sygnały przetworzone przez daną warstwę mogą wracać na jej wejście (lub do warstw wcześniejszych), wpływając na wartości wag danej warstwy.

Warto dodać, że istnieje jeszcze wiele innych typów sieci neuronowych, tu nie wspominanych.

Najprostsza sieć jednokierunkowa - Neuron McCullocha-Pittsa

- Cechuje go elegancja i precyzja matematycznej definicji
- Zawiera szereg uproszczeń:
  - Dopuszcza tylko dwa stany binarne: 0 i 1.
  - Zakłada dyskretność czasu i synchronizacje neuronów w sieci.
  - Wagi i progi są niezmienne.



Prosty perceptron jest zwykłym neuronem McCullocha-Pittsa o odpowiednio przyjętej strategii uczenia i rozszerzonej skokowej funkcji aktywacji

- Najpopularniejszą metodą uczenia (przeprogramowania) jest tzw. *metoda perceptronu,*
- Dobór wag odbywa się w następujących krokach:
  1. Przy zadanych wstępnie najczęściej losowo wartościach wag, podaje się wektor uczący x_k , referencyjny sygnał wyjściowy d_k
  2. Oblicza się sygnał wyjściowy y_k.
  3. W wyniku porównania dokonuje się uaktualnienia wektora wag wg. reguły:

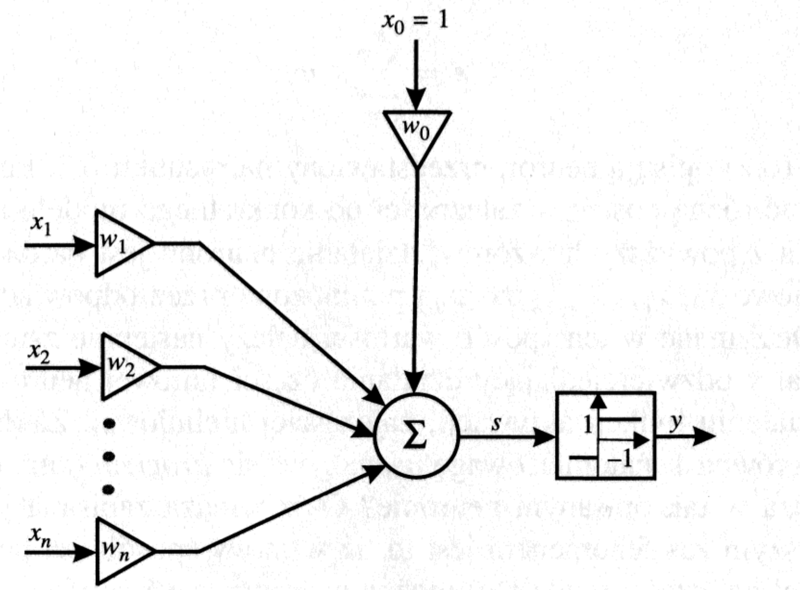
     jeżeli $y_k = d_k$, wagi pozostają niezmienione

     jeżeli $y_k = 0$ i $d_k = 1$, $w_{ki}\,(t+1) = w_{ki}\,(t) + x_i$

     jeżeli $y_k = 1$ i $d_k = 0$, $w_{ki}\,(t+1) = w_{ki}\,(t) - x_i$

Perceptron

Funkcja aktywacji perceptronu
jest **funkcją bipolarną**:



$$y = f(s) = \operatorname{sgn}(s) = \begin{cases} +1 & dla \;\; s \geq 0 \\ -1 & dla \;\; s < 0 \end{cases} \qquad y \in \{-1, +1\} \;,$$

gdzie sterowanie *s* jest opisane zależnością:

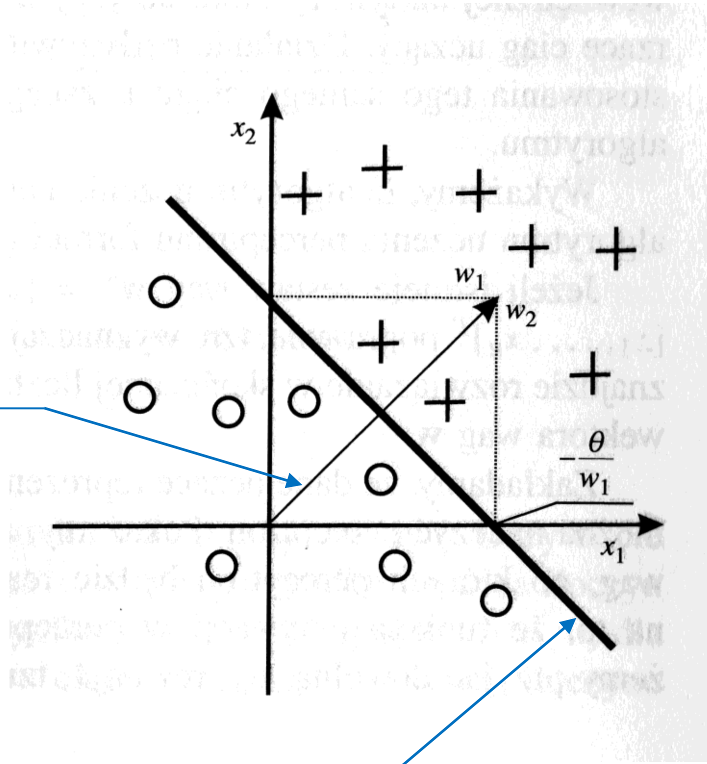
$$s = \sum_{i=1}^{n} w_i x_i + w_0$$

Perceptron, z uwagi na postać funkcji aktywacji, podane na wejściu sygnały **x** może klasyfikować do jednej z **dwóch klas**.

- Perceptron **z jednym wejściem** ocenia czy sygnał wejściowy jest dodatni czy ujemny.
- Perceptron **z dwoma wejściami** x_1 i x_2 potrafi podzielić płaszczyznę na dwie części. Podział ten wyznacza prosta o równaniu:

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$

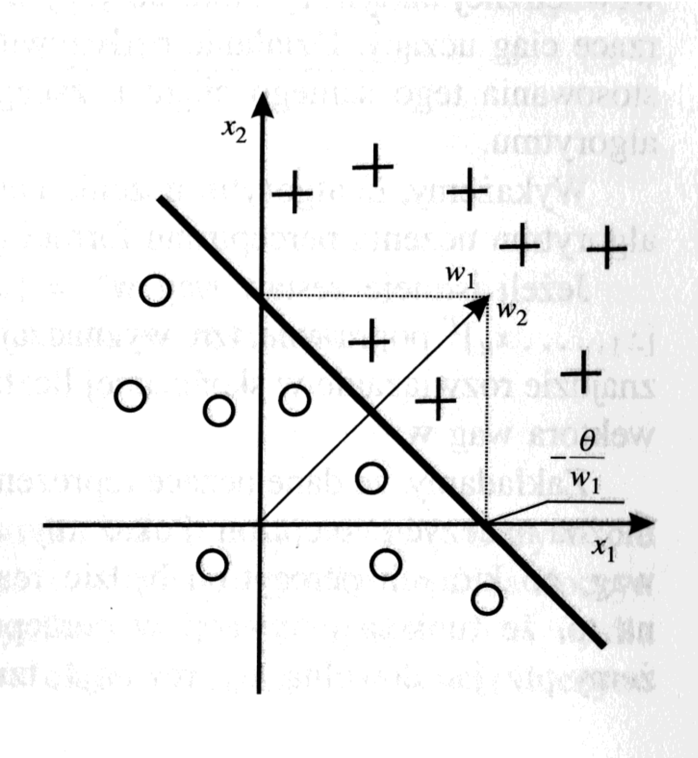
lub inaczej $$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$



Perceptron można uczyć. W czasie tego procesu jego wagi są modyfikowane.

- Perceptron **z** *n* **wejściami** dzieli *n*-wymiarową przestrzeń wektorów wejściowych **x** na dwie półprzestrzenie (*n*-1)-wymiarową hiperpłaszczyzną, daną zależnością:

$$\sum_{i=1}^{n} w_i x_i + w_0 = 0$$



Sposób działania perceptronu zależy od wartości jego wag w_i.

Perceptron można uczyć. W czasie tego procesu jego wagi są modyfikowane.

Algorytm uczenia perceptronu:

Dane:

Zbiory wektorów uczących $\mathbf{x}_k$ i odpowiadających im wzorcowych wyjść $d = d(\mathbf{x}_k)$
$k = 1, 2, 3, \ldots, K$

epoka

1. Wybór losowy początkowych wag $\mathbf{w}_0$; $k = 1$.

2. Podanie na wejściu wektora uczącego $\mathbf{x}_k$.

3. Obliczenie wartości sterowania s_k i funkcji wyjściowej $y(s_k)$.

4. Jeżeli:

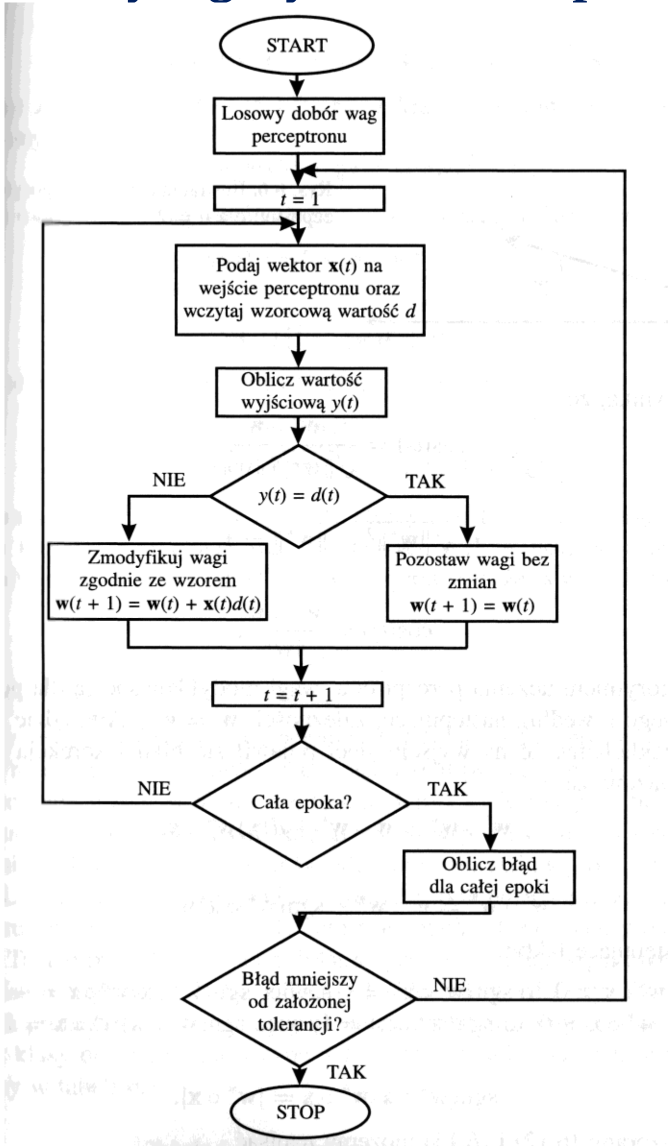
$y(s_k) \neq d(\mathbf{x}_k)$ to: $\mathbf{w}_{k+1} = \mathbf{w}_k + d(\mathbf{x}_k)\mathbf{x}_k$ (wagi są modyfikowane)

inaczej: $\mathbf{w}_{k+1} = \mathbf{w}_k$ (wagi pozostają bez zmian)

5. $k = k + 1$; Jeżeli $k <= K$ to: powrót do kroku 2, inaczej: krok 6.

6. Jeżeli dla wszystkich $k = 1, 2, 3, \ldots, K$ zachodzi $\left| y(s_k) \neq d(\mathbf{x}_k) \right| \leq \varepsilon$ (tolerancja)
to: **koniec uczenia** inaczej: $k = 1$, powrót do kroku 2.

Schemat blokowy algorytm uczenia perceptronu:

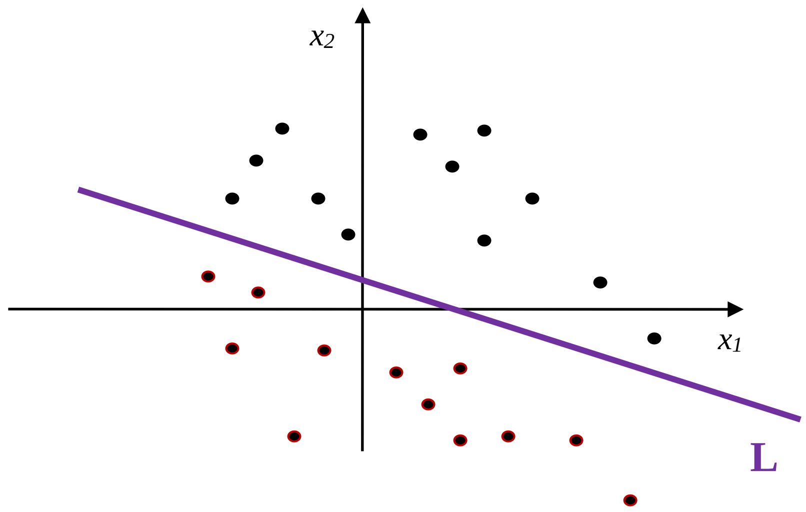


Przykład uczenia perceptronu:

Dane są dwa zbiory punktów leżące na płaszczyźnie Ox_1x_2 rozdzielonej prostą L.

Zbiór I leży ponad prostą L

Zbiór II leży poniżej prostej L



Perceptron, po nauczeniu, dla punktów ze zbioru I ma podawać wartość 1, zaś dla punktów ze zbioru II wartość -1.

Ciąg uczący perceptronu:

x_1 :	2	2	0	-2	-2	0	4
x_2:	1	2	6	8	0	0	-20
$d(\mathbf{x})$:	1	1	1	-1	-1	-1	-1

Początkowe wartości wag:

$w_1 = 2$, $w_2 = 2$, $w_0 = 4$ (prosta K)

Wartości wag po 10 epokach uczenia:

$w_1 = 4$, $w_2 = 1$, $w_0 = -1$ (prosta M)

Wszystkie próbki są prawidłowo klasyfikowane chociaż prosta M nie pokrywa się z wzorcową prostą L.