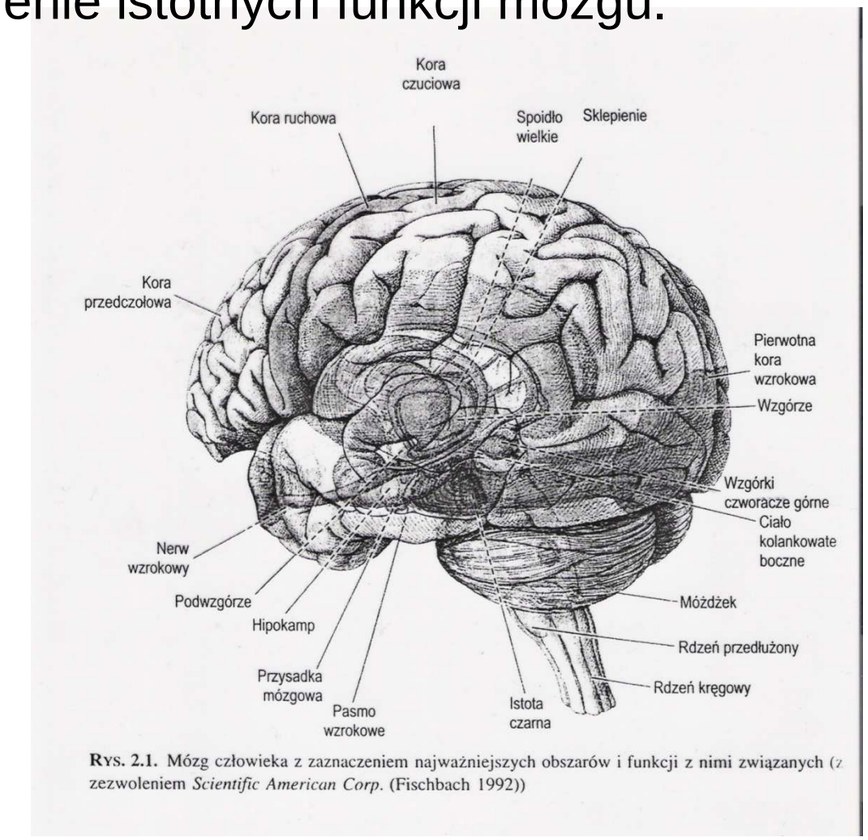
SZTUCZNE SIECI NEURONOWE

Ang. Artifficial neural network

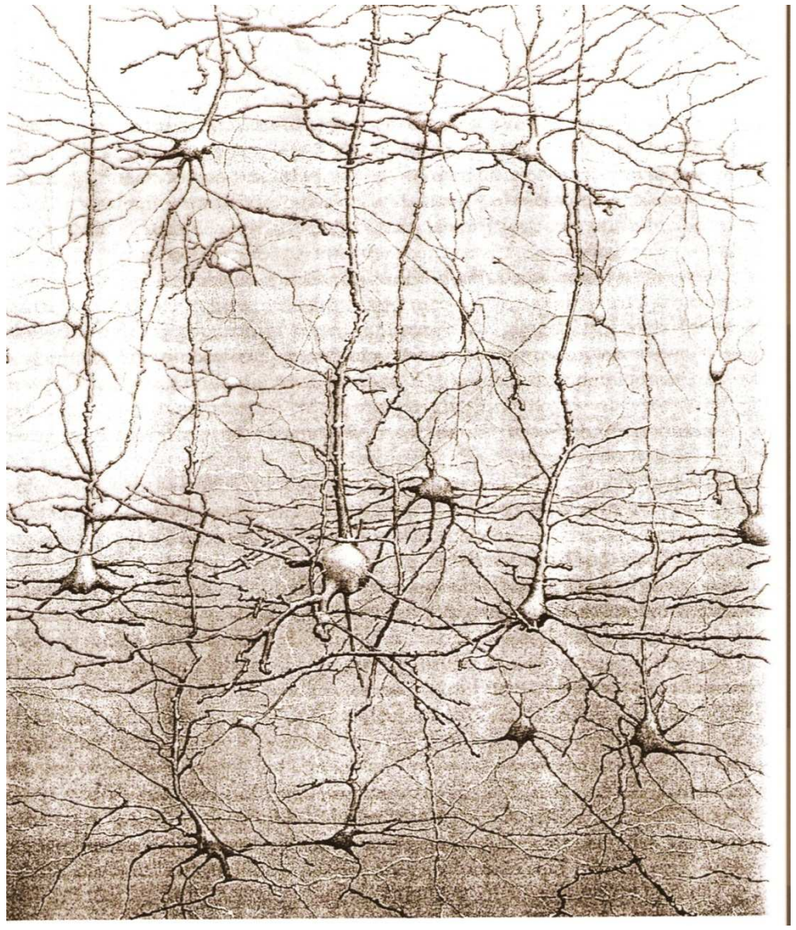
Dr inż. Piotr Urbanek

Mózg człowieka od dawna jest zagadką. Wiek XXI niektórzy postulują nazwać "wiekiem mózgu", gdyż ambicją uczonych jest zrozumienie istotnych funkcji mózgu.



Rys. 2.1. Mózg człowieka z zaznaczeniem najważniejszych obszarów i funkcji z nimi związanych (z zezwoleniem *Scientific American Corp.* (Fischbach 1992))

Połączenia neuronów – naturalna sieć neuronowa



Cechy mózgu (ludzkiego)

1. **Ogromna "moc obliczeniowa"**
- Rozpoznawanie obrazów i mowy (100 − 200 ms).
- Sterowanie : ruch, walka, sport
- Echolokacja (gdzie obiekt się znajduje, w którą stronę należy się udać)

2. **Duża wydajność**
- Małe rozmiary
- 10^{-16} Joula na jedną operację na jedną sekundę (krzem 10^{-6})
- Prędkość "zegara" 102 Hz.

3. **Duża odporność na uszkodzenia**
4. **Duża odporność na szumy**
5. **Zdolność uczenia i adaptacji, uogólnianie.**

Budowa mózgu ludzkiego

- Objętość: 1400 cm^3
- Powierzchnia: 2000 cm^2

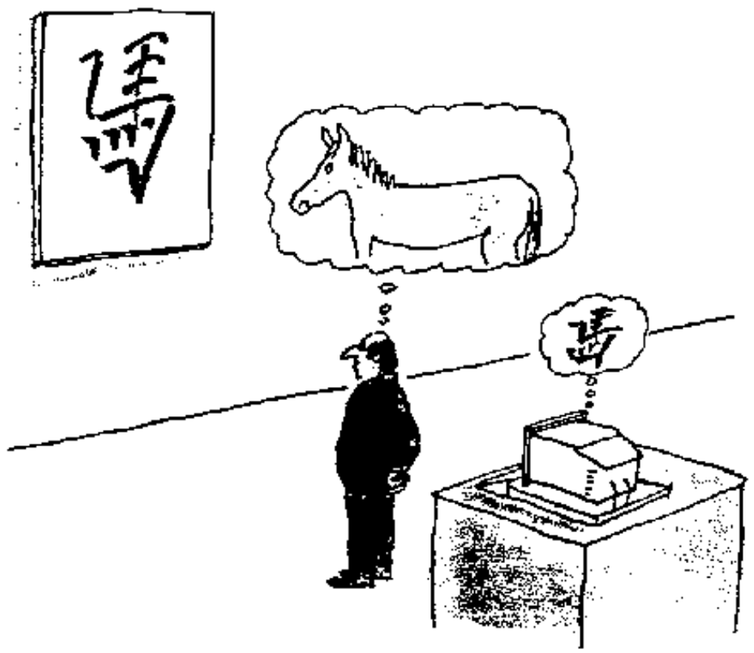
- Liczba neuronów: około 100 miliardów (10^{11})
- Liczba połączeń między komórkami : 10^{15} przy przeciętnym dystansie od 0.01 mm do 1 m.
- Komórki nerwowe wysyłają i przyjmują impulsy o:
  - częstotliwości: 1-100 Hz
  - czasie trwania: 1-2 ms
  - szybkości propagacji: 1-100 m/s.

- Szybkość pracy mózgu: 10^{18} operacji/s

Mózg vs komputer

Mózg	Komputer
Rozpoznawanie. Kojarzenie informacji, Klasyfikacja danych.	obliczenia arytmetyczne
równoległe przetwarzanie danych - wiele neuronów działających w tym samym czasie	bardzo krótki czas przetwarzania jednego polecenia
zdolność do rekonstrukcji i odtwarzania sygnałów	-
odporność na uszkodzenia	-
zdolność przetwarzania informacji niepełnej i obarczonej błędami	wysoka precyzja obliczeń

PODSTAWOWA CECHA MÓZGU



Tu objawia się zdolność uogólniania – porównywania widzianych obrazów z wzorcami zapamiętanymi przez mózg człowieka (i nie tylko).

Definicja

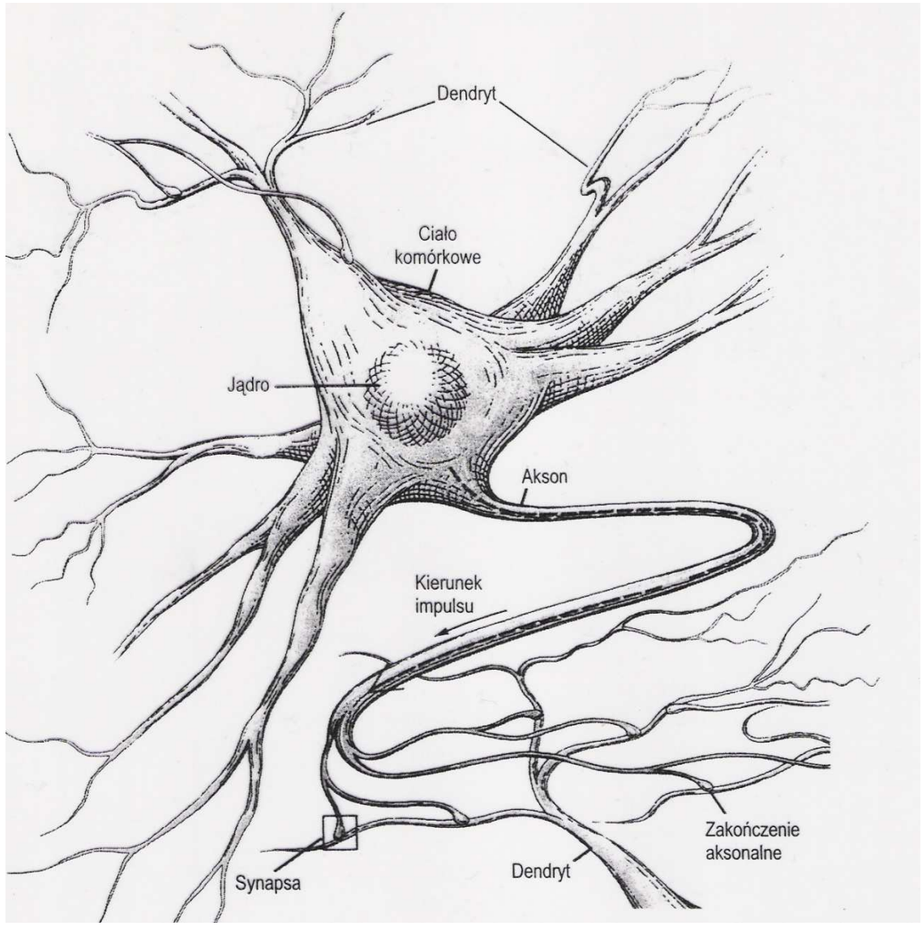
Sztuczna sieć neuronowa

to wysoce równoległy rozproszony p**rocesor** złożony z prostych elementów obliczeniowych mający naturalną skłonność zapamiętywania podawanych informacji.

Przypomina mózg w dwóch aspektach:

- wiedza jest zdobywana przez sieć ze środowiska w toku procesu nauczania,
- wagi połączeń między neuronami używane są do zapamiętywania uzyskanej wiedzy.

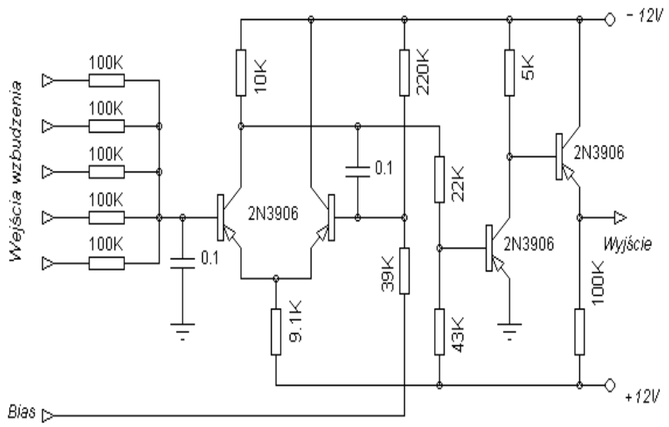
Budowa neuronu



Modelowanie neuronu

Neuron może istnieć jako element w modelu teoretycznym. Nazywa się go wtedy **neuronem formalnym**.

Sztuczny neuron może być realizowany jako układ scalony, bądź w programie numerycznym stanowiącym sztuczną sieć neuronową (element programowy).

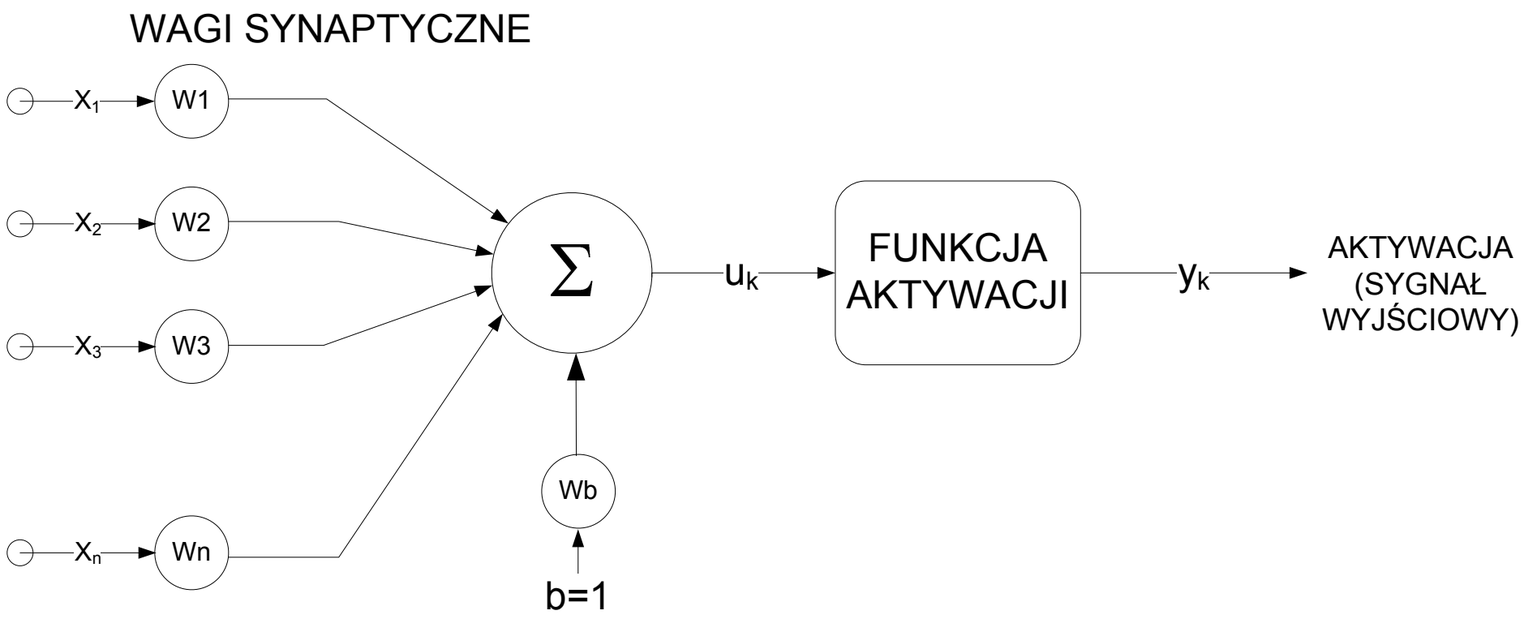


Schemat układu elektronicznego odwzorowującego pojedynczy neuron

```
/*** Generacja wag początkowych ***/
for(i=1;i<LW;i++) for(j=0;j<n[i];j++)  for(k=0;k<=n[i-
1];k++)  {   W[i][j][k] =  (((rand() % 1000000L) /
1700.0) - 9.8)*0.0015;   if(W[i][j][k] == 0.0) W[i][j][k] =
0.01492;  }/*********************************/


/***  Pojedyncze przetworzenie  ***/
for(i=1;i<LW;i++) for(j=0;j<n[i];j++) {  I[i][j] = 0.0;
for(k=0;k<=n[i-1];k++)   I[i][j] += O[i-1][k] * W[i][j][k];
O[i][j] = 1.0 / (1.0 + exp(beta*(-I[i][j])));
}/*********************************/
```

Model pojedynczego neuronu (McCullocha-Pittsa)



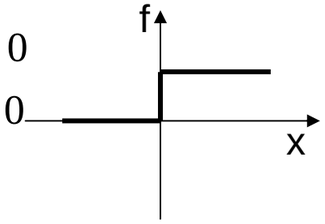
$$u_k = \sum_i w_{ki} x_i + w_b \cdot 1$$

Funkcje aktywacji

Perceptron

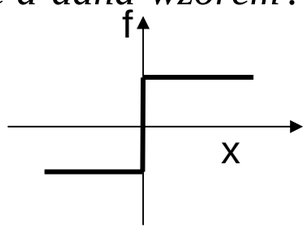
$\Theta -$ *funkcja Heaviside'a dana wzorem*:

$$f(x)=\begin{cases}0 \ dla \ x\le 0\\ 1 \ dla \ x>0\end{cases}$$



$\Theta -$ *funkcja Heaviside'a dana wzorem*:

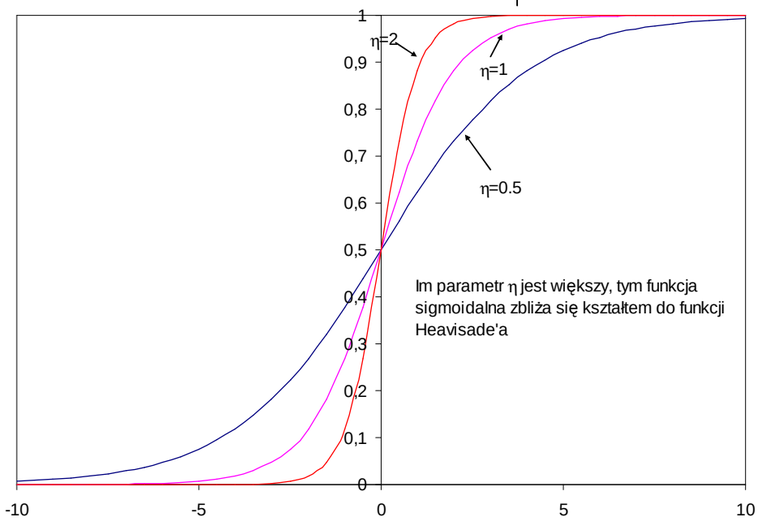
$$f(x)=\begin{cases}-1 \ dla \ x\le 0\\ 1 \ dla \ x>0\end{cases}$$



$\Theta -$ *funkcja* sigmoidalna *dana wzorem*:

$$f(x)=\frac{1}{(1+\exp(-\eta\cdot x))};$$

gdzie η *– parametr wzmocnienia*



Inne funkcje aktywacji

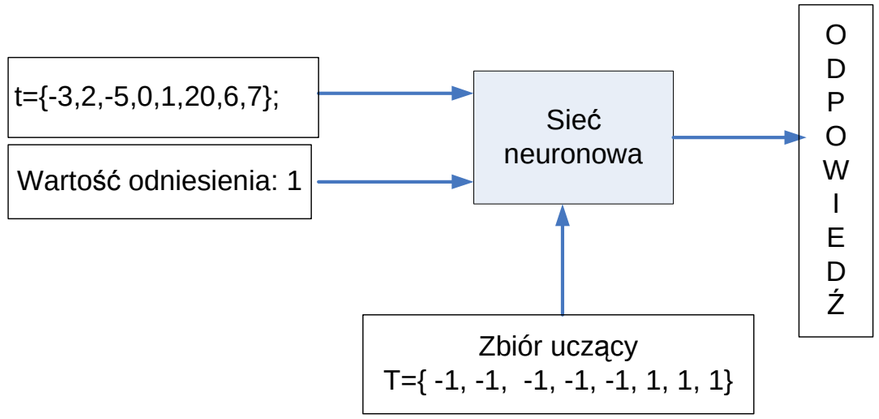
funkcja	Wzór funkcji	Wzór pochodnej
Sigmoida	$f(x) = \frac{1}{1 + e^{-(beta * x)}}$	$f'(x) = beta * (1 - f(x)) * f(x)$
Tangens hiperboliczny	$f(x) = \tanh(beta * x)$	$f'(x) = beta(1 - f^2(x))$
Sinusoida	$f(x) = \sin(beta * x)$	$f'(x) = beta\sqrt{1 - f^2(x)}$
Cosinusoida	$f(x) = \cos(beta * x)$	$f'(x) = -beta\sqrt{1 - f^2(x)}$
$\frac{x}{(1 + \lvert x \rvert)}$ *(bez nazwy)*	$f(x) = \frac{beta * x}{(1 + \lvert beta * x \rvert)}$	$f'(x) = \frac{beta}{1 + \lvert beta * x \rvert} - \frac{\lvert beta * x \rvert}{(1 + \lvert beta * x \rvert)^2}$

Różnica pomiędzy SSN a zwykłym algorytmem

Podstawową cechą różniącą algorytmy SSN od programów realizujących algorytmiczne przetwarzanie informacji jest **zdolność generalizacji** czyli umiejętność uogólniania wiedzy dla nowych wzorców nieznanych wcześniej, czyli nie prezentowanych w trakcie nauki. Określa się to także jako zdolność SSN do aproksymacji wartości funkcji wielu zmiennych w przeciwieństwie do interpolacji możliwej do otrzymania przy przetwarzaniu algorytmicznym.

Zwykły algorytm. ≠ Algorytm Sztucznej Sieci Neuronowej

```
t={-3,2,-5,0,1,20,6,7};
for (i=1,i<=n,i++)
{
        if (t(i) < 5) odp=-1;
        else odp=1;
}
```

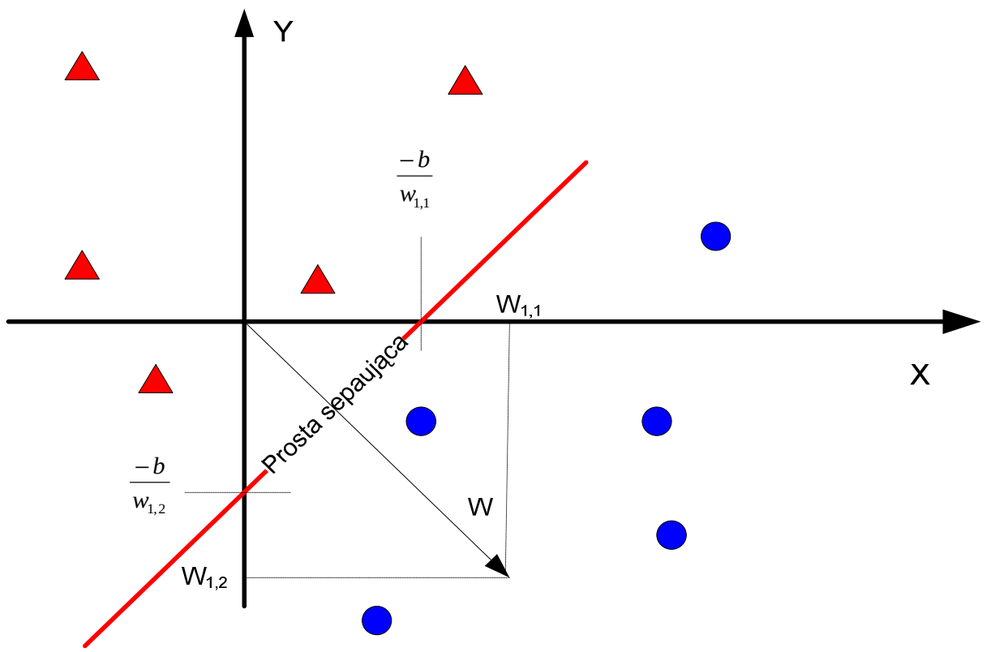


Uczenie sieci

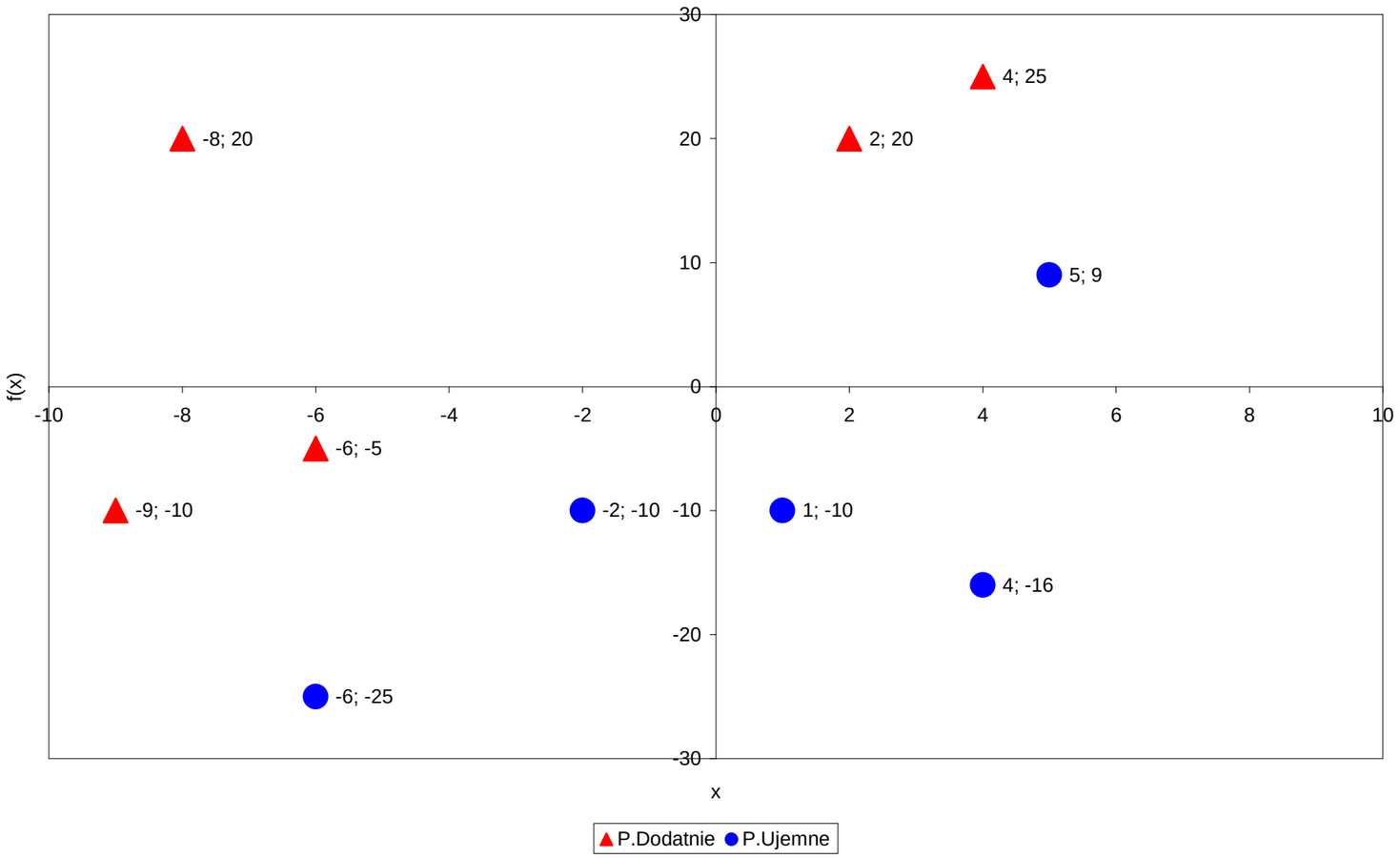
1. Przygotowanie dwóch ciągów: uczącego i weryfikującego. W skład ciągu uczącego wchodzi wektor wejściowy oraz wektor wyjściowy.
2. Ustanowienie początkowych wartości wag (wartości przypadkowe).
3. Po przetworzeniu wektora wejściowego nauczyciel porównuje wartości otrzymane z wartościami oczekiwanymi informując sieć o błędzie odpowiedzi.
4. Jeżeli wartość na wyjściu neuronu nie zgadza się z wartością oczekiwaną, **następuje korekcja wag, tak aby błąd odpowiedzi uzyskany przy powtórnym przetworzeniu wektora wejściowego był mniejszy od poprzedniego.**
5. Czynności 1-4 powtarza się aż do uzyskania błędu mniejszego niż zamierzony.

EPOKA

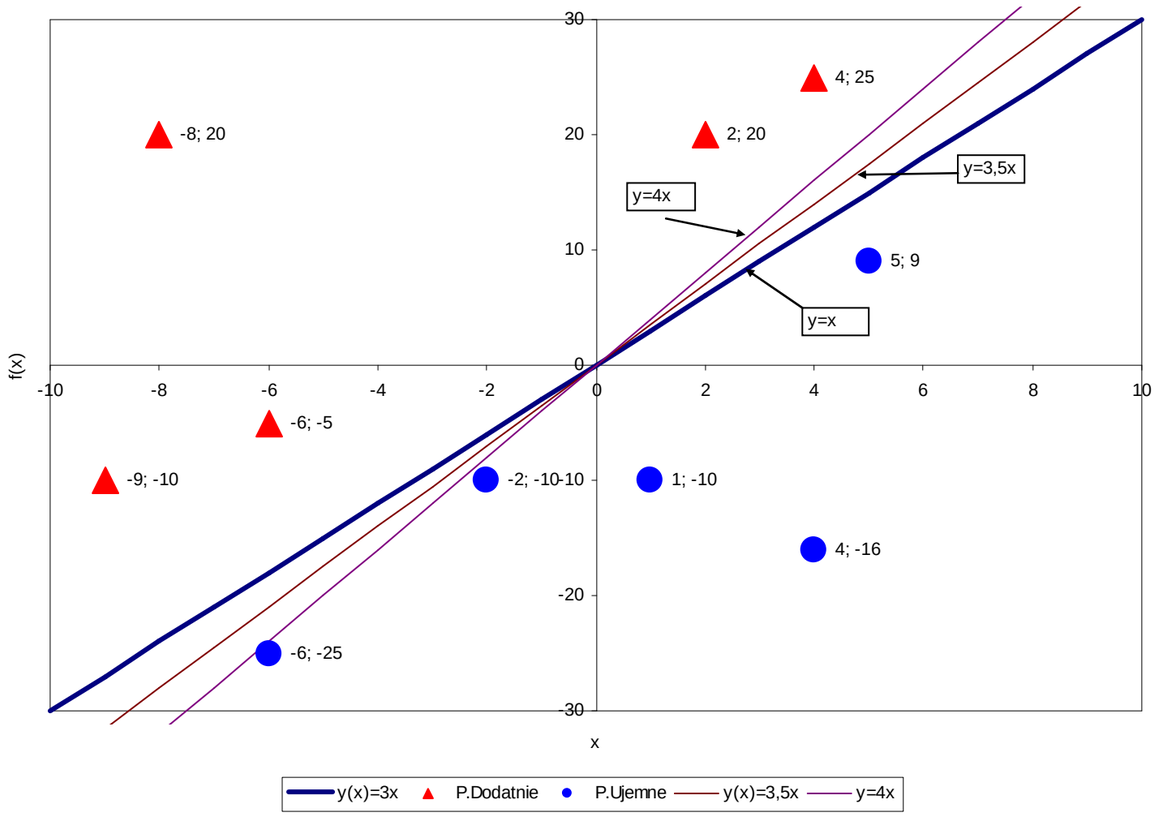
Przykład klasyfikowania zbiorów



Klasyfikacja punktów na płaszczyźnie

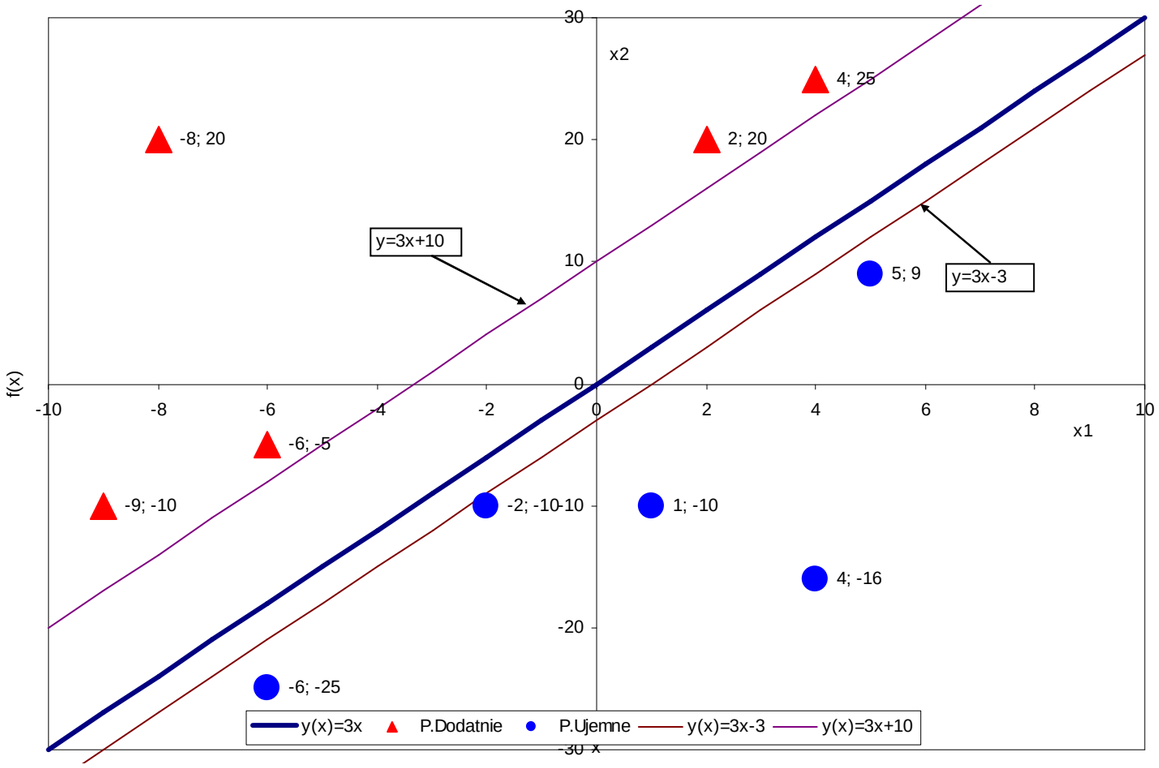


Klasyfikacja punktów na płaszczyźnie



$$y = w \cdot x$$

Klasyfikacja punktów na płaszczyźnie



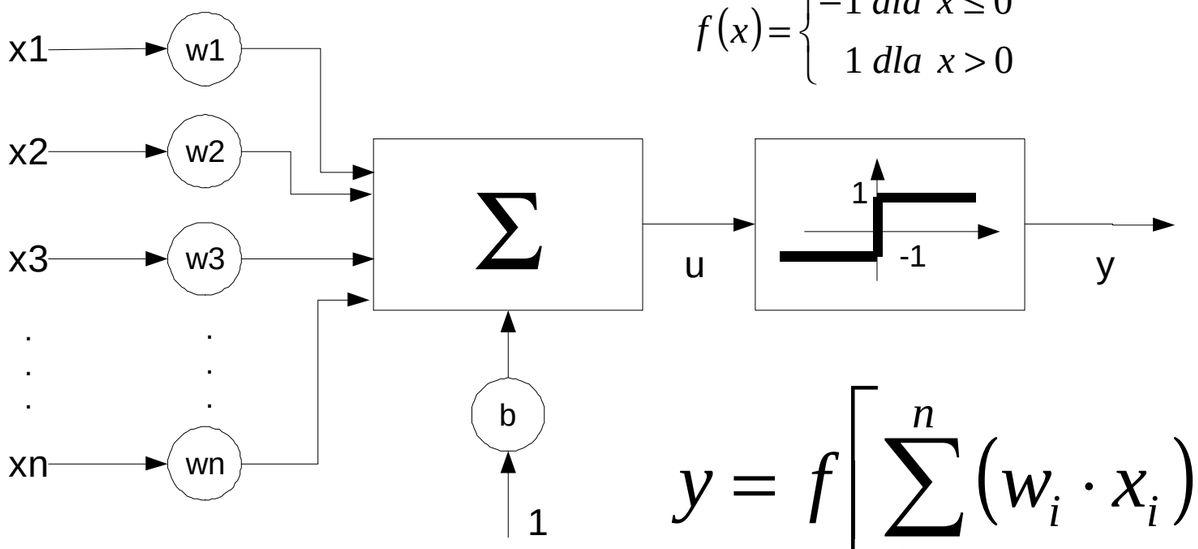
$$y = w \cdot x + b$$

Perceptron

Neuron ze skokową funkcją aktywacji

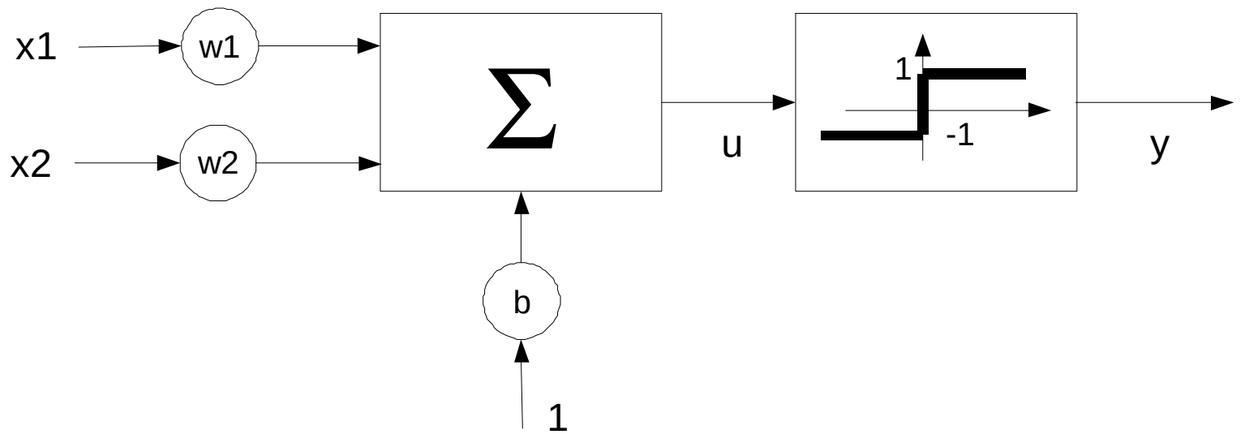
Θ – *funkcja Heaviside'a dana wzorem*:

$$f(x)=\begin{cases}-1\ dla\ x\le 0\\ 1\ dla\ x>0\end{cases}$$



$$y=f\left[\sum_{i=1}^{n}\left(w_i\cdot x_i\right)+b\right]$$

Perceptron dwuwejściowy



$$y = f\left(w1 \cdot x1 + w2 \cdot x2 + b\right)$$

$$y = f\left(\begin{bmatrix} w1 & w2 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \end{bmatrix} + b\right)$$

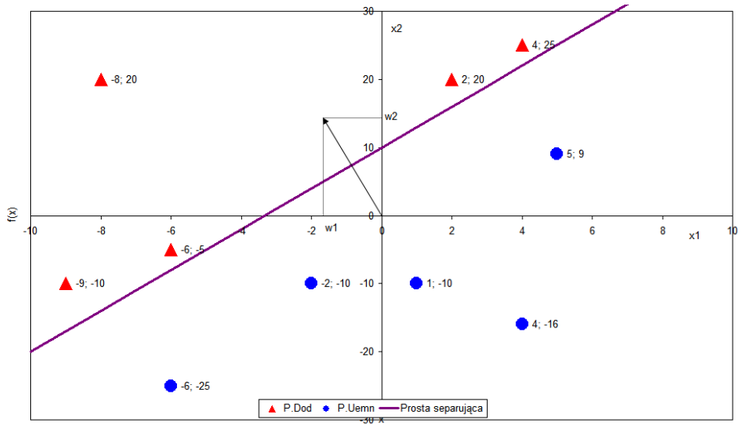
Perceptron dwuwejściowy

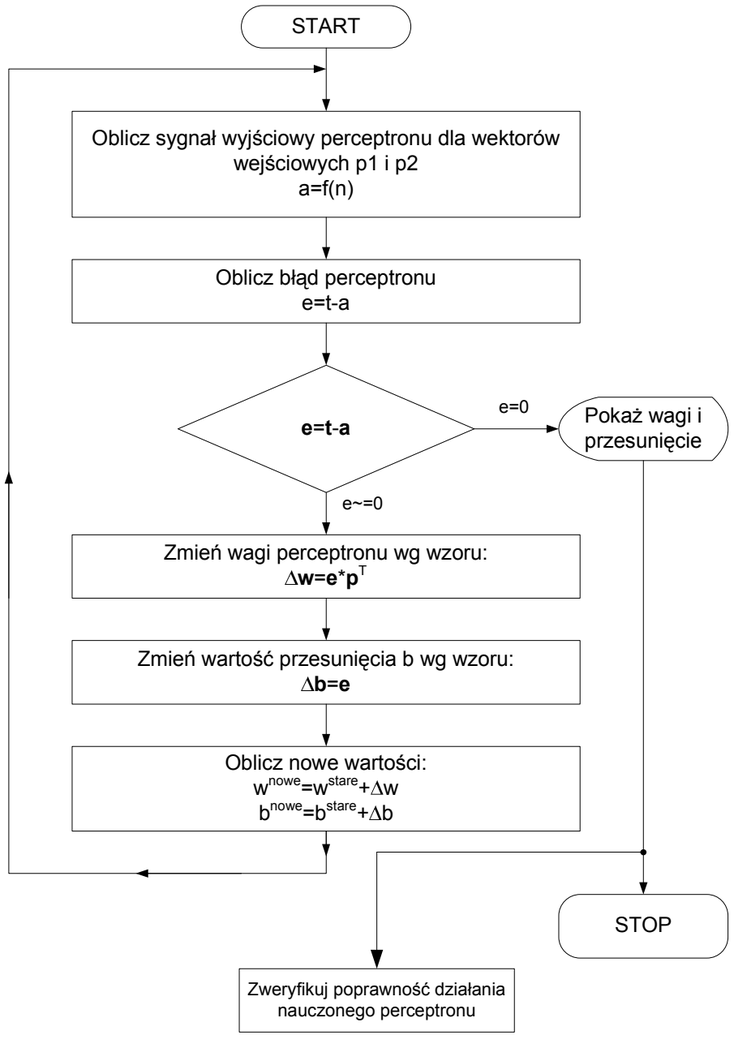
W przypadku dwóch wejść [x1 x2]^T perceptron dzieli płaszczyznę na dwie części. Zatem może klasyfikować punkty należące do dwóch umownych klas, np. "dodatnich" i "ujemnych". Podział ten wyznacza prosta o równaniu:

$$w1 \cdot x1 + w2 \cdot x2 + b = 0$$

Zatem, gdy x1=0, mamy x2=-b/w2

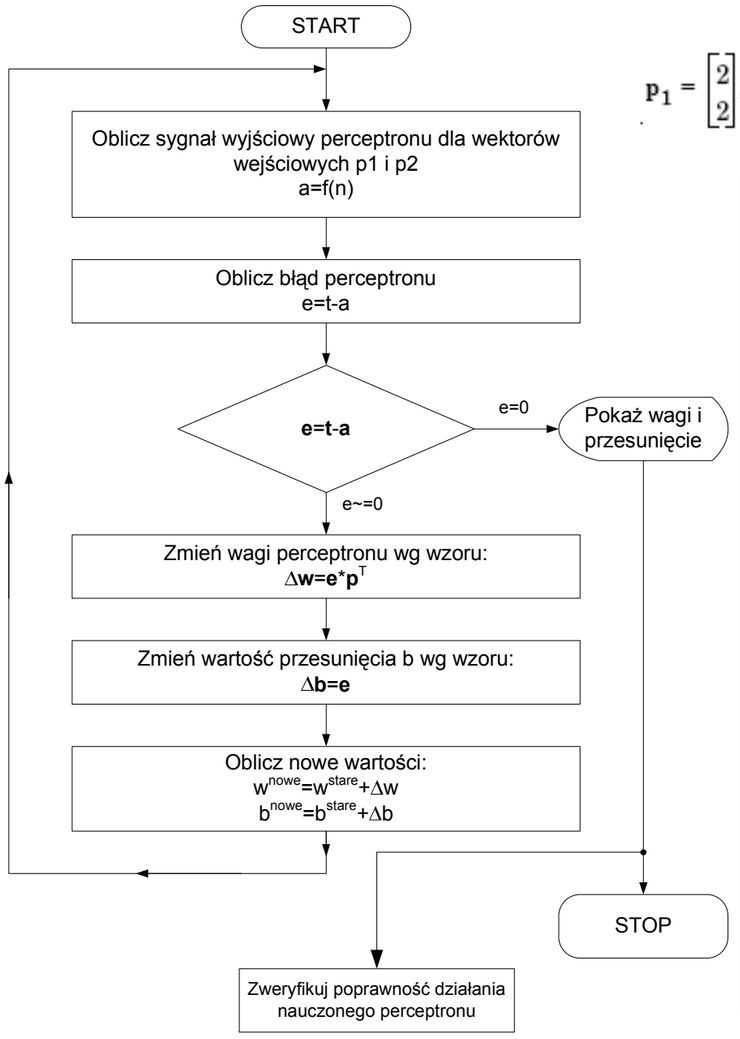
x2=0, mamy x1=-b/w1





1. Przygotowanie dwóch ciągów: uczącego i weryfikującego.
   W skład ciągu uczącego wchodzi wektor wejściowy oraz wektor wyjściowy.
2. Ustanowienie początkowych wartości wag (wartości przypadkowe).
3. Po przetworzeniu wektora wejściowego nauczyciel porównuje wartości otrzymane z wartościami oczekiwanymi informując sieć o błędzie odpowiedzi.
4. Następuje korekcja wag, tak aby błąd odpowiedzi uzyskany przy powtórnym przetworzeniu wektora wejściowego był mniejszy od poprzedniego.
5. Czynności 1-5 powtarza się aż do uzyskania błędu mniejszego niż zamierzony.

Schemat blokowy algorytmu uczenia perceptronu.

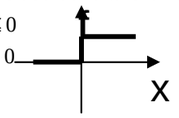


$$\mathbf{p}_1 = \begin{bmatrix}2\\2\end{bmatrix}, t_1 = 0\} \quad \{\mathbf{p}_2 = \begin{bmatrix}1\\-2\end{bmatrix}, t_2 = 1\} \quad \{\mathbf{p}_3 = \begin{bmatrix}-2\\2\end{bmatrix}, t_3 = 0\} \quad \{\mathbf{p}_4 = \begin{bmatrix}-1\\1\end{bmatrix}, t_4 = 1$$

$$\mathbf{W}(0) = \begin{bmatrix}0 & 0\end{bmatrix} \qquad b(0) = 0$$

Θ – *funkcja Heaviside'a dana wzorem*:

$$f(x) = \begin{cases} 0 \text{ dla } x \le 0 \\ 1 \text{ dla } x > 0 \end{cases}$$



$$a = hardlim(\mathbf{W}(0)\mathbf{p}_1 + b(0))$$

$$= hardlim\left(\begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix} + 0\right) = hardlim(0) = 1$$

$$e = t_1 - a = 0 - 1 = -1$$

$$\Delta\mathbf{W} = e\mathbf{p}_1^T = (-1)\begin{bmatrix}2 & 2\end{bmatrix} = \begin{bmatrix}-2 & -2\end{bmatrix}$$

$$\Delta b = e = (-1) = -1$$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{e}\mathbf{p}^T = \begin{bmatrix}0 & 0\end{bmatrix} + \begin{bmatrix}-2 & -2\end{bmatrix} = \begin{bmatrix}-2 & -2\end{bmatrix} = \mathbf{W}(1)$$

$$b^{new} = b^{old} + e = 0 + (-1) = -1 = b(1)$$

$$a = hardlim(\mathbf{W}(1)\mathbf{p}_2 + b(1))$$

$$= hardlim\left(\begin{bmatrix}-2 & -2\end{bmatrix}\begin{bmatrix}-2\\-2\end{bmatrix} - 1\right) = hardlim(1) = 1$$ OK!

$$\mathbf{W}(2) = \mathbf{W}(1) = \begin{bmatrix}-2 & -2\end{bmatrix} \text{ and } p(2) = p(1) = -1 .$$

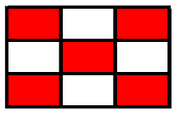
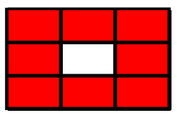
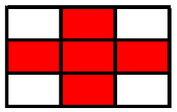
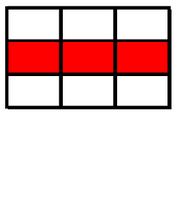
Weryfikacja działania sieci.

1. Na wejście sieci wprowadza się wektor weryfikujący, który ma te same cechy, co ciąg uczący, czyli musi dokładnie charakteryzować problem i muszą być znane dokładne odpowiedzi. Uwaga! NIE MOŻE TO BYĆ JEDEN Z WEKTORÓW UCZĄCYCH!
2. Przetwarza się wektor weryfikujący rejestrując liczbę odpowiedzi poprawnych.
3. Orzekamy, czy sieć spełnia założone wymagania.

Aby zabezpieczyć się przed sytuacjami wyjątkowymi, podczas których SSN nie będzie mogła znaleźć poprawnego rozwiązania wprowadza się mechanizmy kontrolujące szybkość i jakość uczenia.

Są to ***współczynniki uczenia*** oraz ***momentum.*** Wpływają one na stromość funkcji aktywacji i regulują szybkość wpływu zmiany wag na proces uczenia.

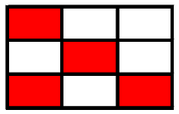
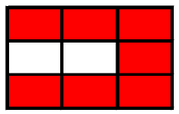
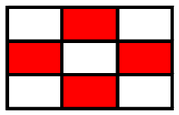
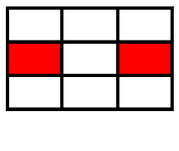
Rozpoznawanie znaków „x, o, +, -”

Znaki	Matryce znaków	wektory wejściowe	wektory wyjściowe
	1 0 1 0 1 0 1 0 1	1 0 1 0 1 0 1 0 1	1 0 0 0
	1 1 1 1 0 1 1 1 1	1 1 1 1 0 1 1 1 1	0 1 0 0
	0 1 0 1 1 1 0 1 0	0 1 0 1 1 1 0 1 0	0 0 1 0
	0 0 0 1 1 1 0 0 0	0 0 0 1 1 1 0 0 0	0 0 0 1

Wniosek:

Sieć musi posiadać 9 wejść i 4 wyjścia.

Weryfikacja działania sieci

Znaki	Matryce znaków	wektory wejściowe	wektory wyjściowe
	1 0 0 0 1 0 1 0 1	1 0 **0** 0 1 0 1 0 1	1 0 0 0
	1 1 1 0 0 1 1 1 1	1 1 1 **0** 0 1 1 1 1	0 1 0 0
	0 1 0 1 0 1 0 1 0	0 1 0 1 **0** 1 0 1 0	0 0 1 0
	0 0 0 1 0 1 0 0 0	0 0 0 1 **0** 1 0 0 0	0 0 0 1

Doświadczenie pokazało, że najlepsze rezultaty uzyskano przy jednej warstwie ukrytej z pięcioma neuronami. Zatem struktura sieci neuronowej składałaby się z 9 neuronów w warstwie wejściowej, 5 neuronów w warstwie ukrytej oraz 4 neuronów w warstwie wyjściowej.

Rodzaje sieci

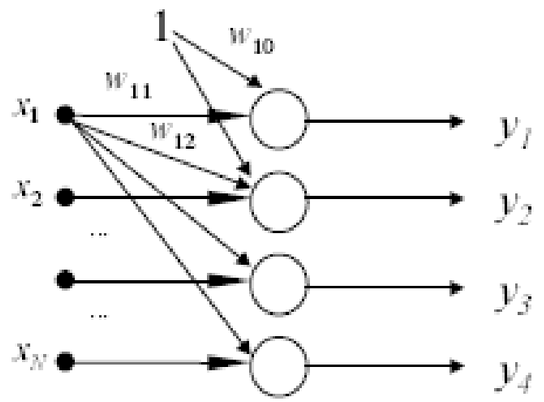
Sieci jednokierunkowe – sygnał rozprzestrzenia się w jednym kierunku

Sieci jednowarstwowe

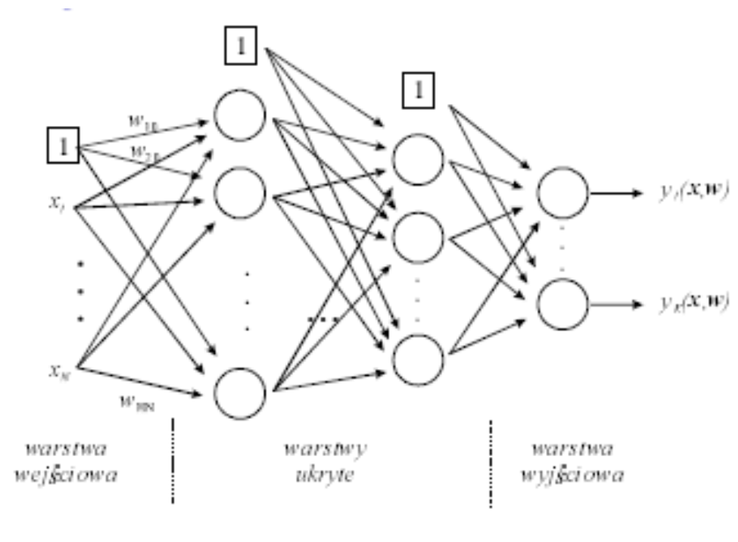
Sieci wielowarstwowe

Sieci rekurencyjne – sieci ze sprzężeniem zwrotnym.

Rodzaje sztucznych sieci neuronowych



Sieć jednokierunkowa, jednowarstwowa



Sieć jednokierunkowa, wielowarstwowa