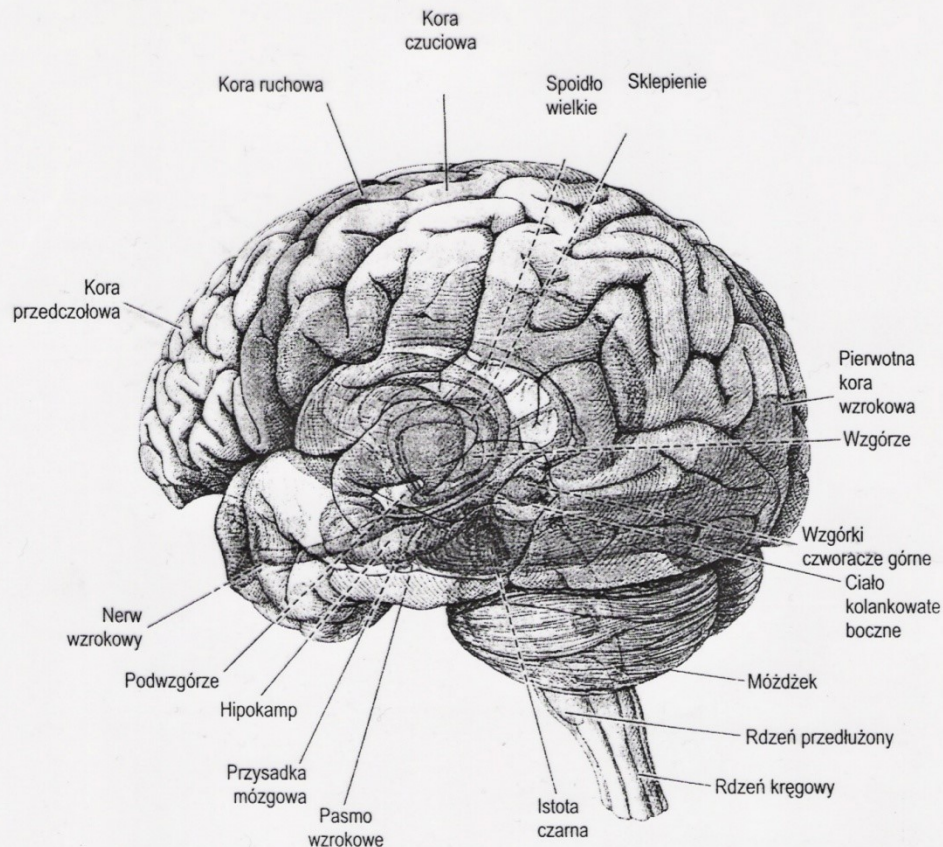


# SZTUCZNE SIECI NEURONOWE

Ang. Artificial neural network

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))

# 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

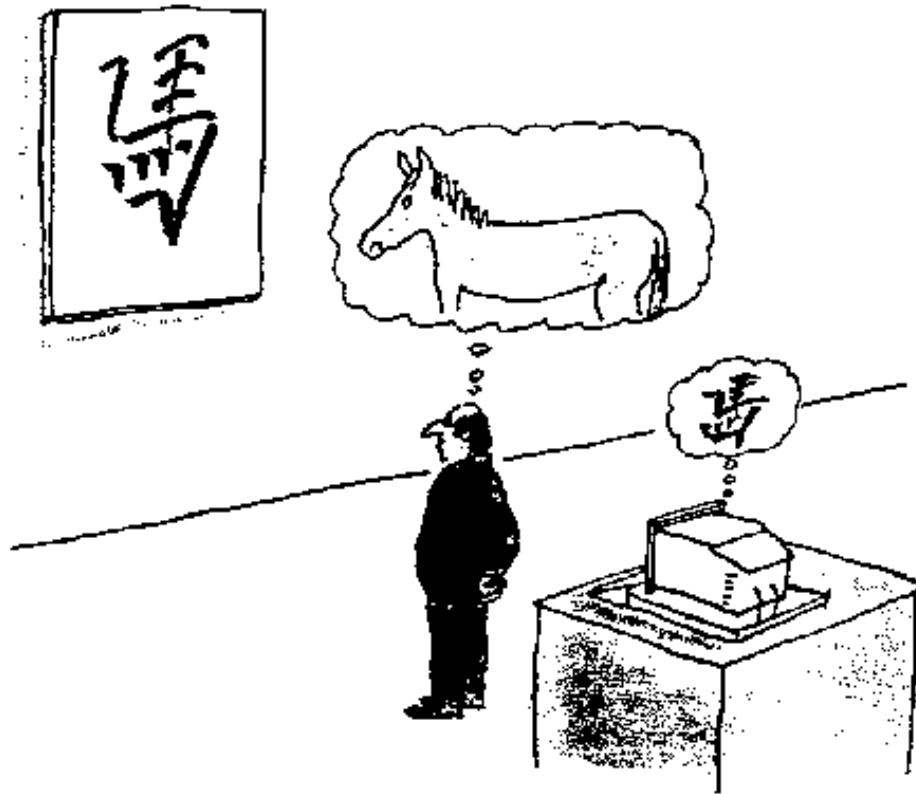
- Małe rozmiary
- $10^{-16}$  Joula na jedną operację na jedną sekundę (krzem  $10^{-6}$ )
- Prędkość "zegara" 102 Hz.

## 3. Duża odporność na uszkodzenia

## 4. Duża odporność na szумы

## 5. Zdolność uczenia i adaptacji, uogólnianie.

# PODSTAWOWA CECHA MÓZGU



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

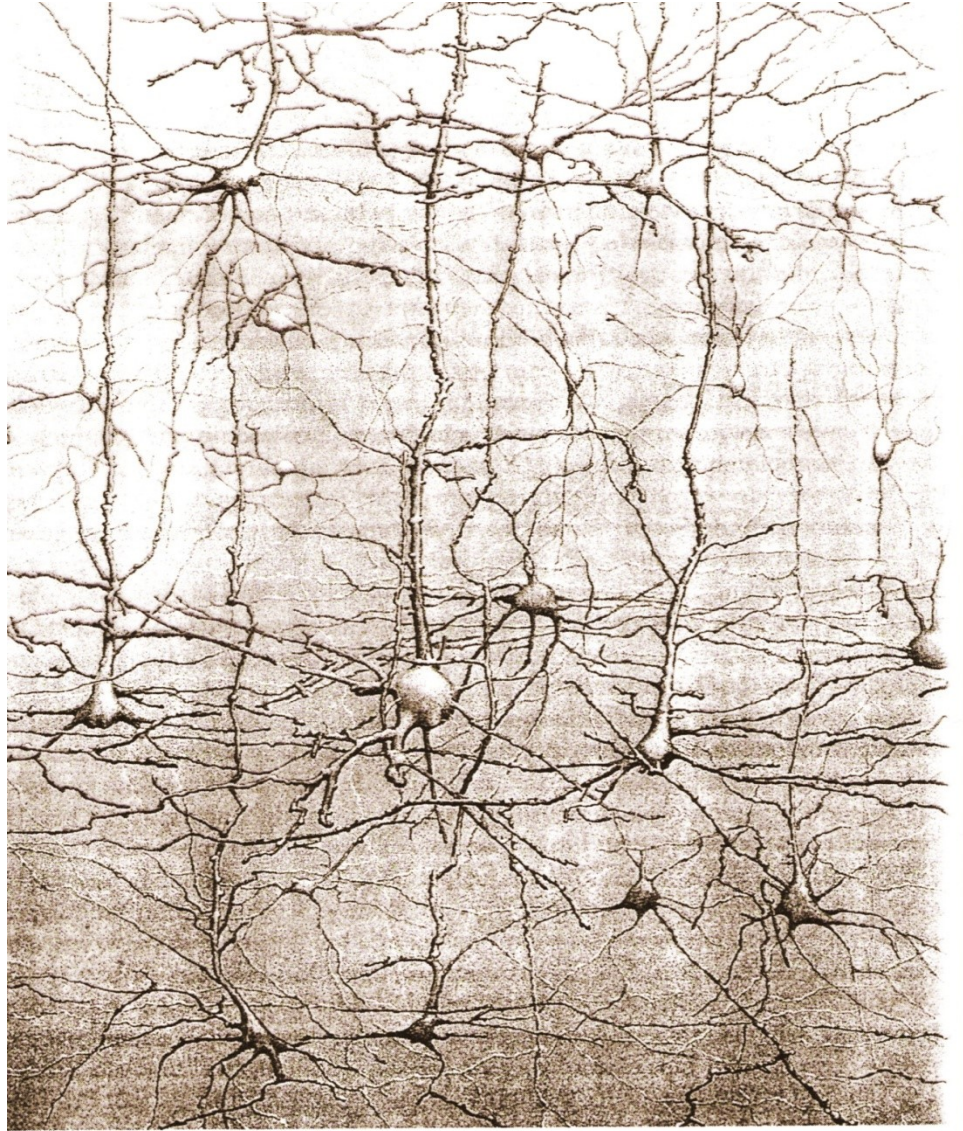
# Mózg vs komputer

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 odtworzenia sygnałów	-
odporność na uszkodzenia	-
zdolność przetwarzania informacji niepełnej i obciążonej błędami	wysoka precyzja obliczeń

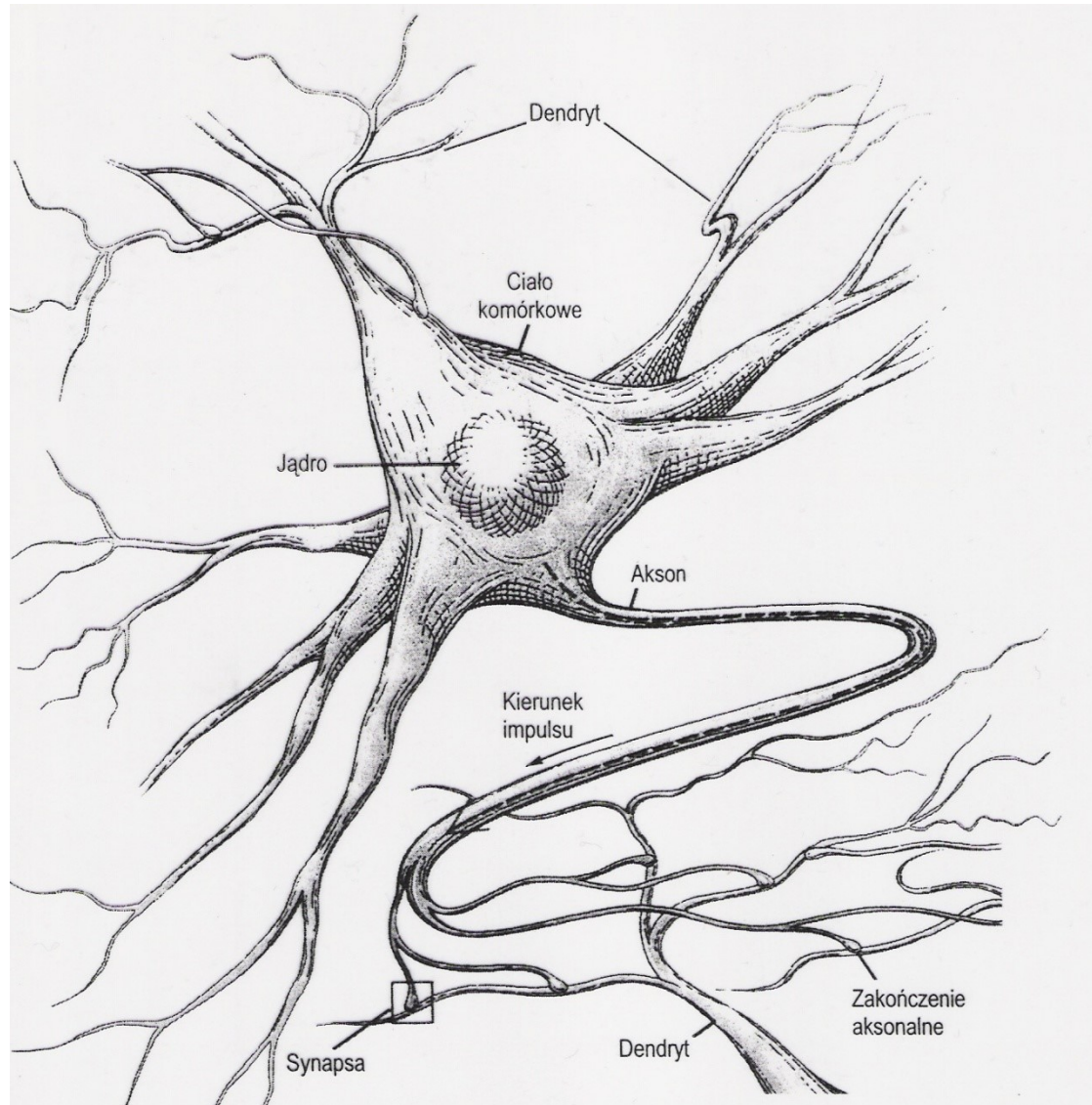
# Budowa mózgu ludzkiego

- Objętość:  $1400 \text{ cm}^3$
- Powierzchnia:  $2000 \text{ cm}^2$
- 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

# Połączenia neuronów – naturalna sieć neuronowa

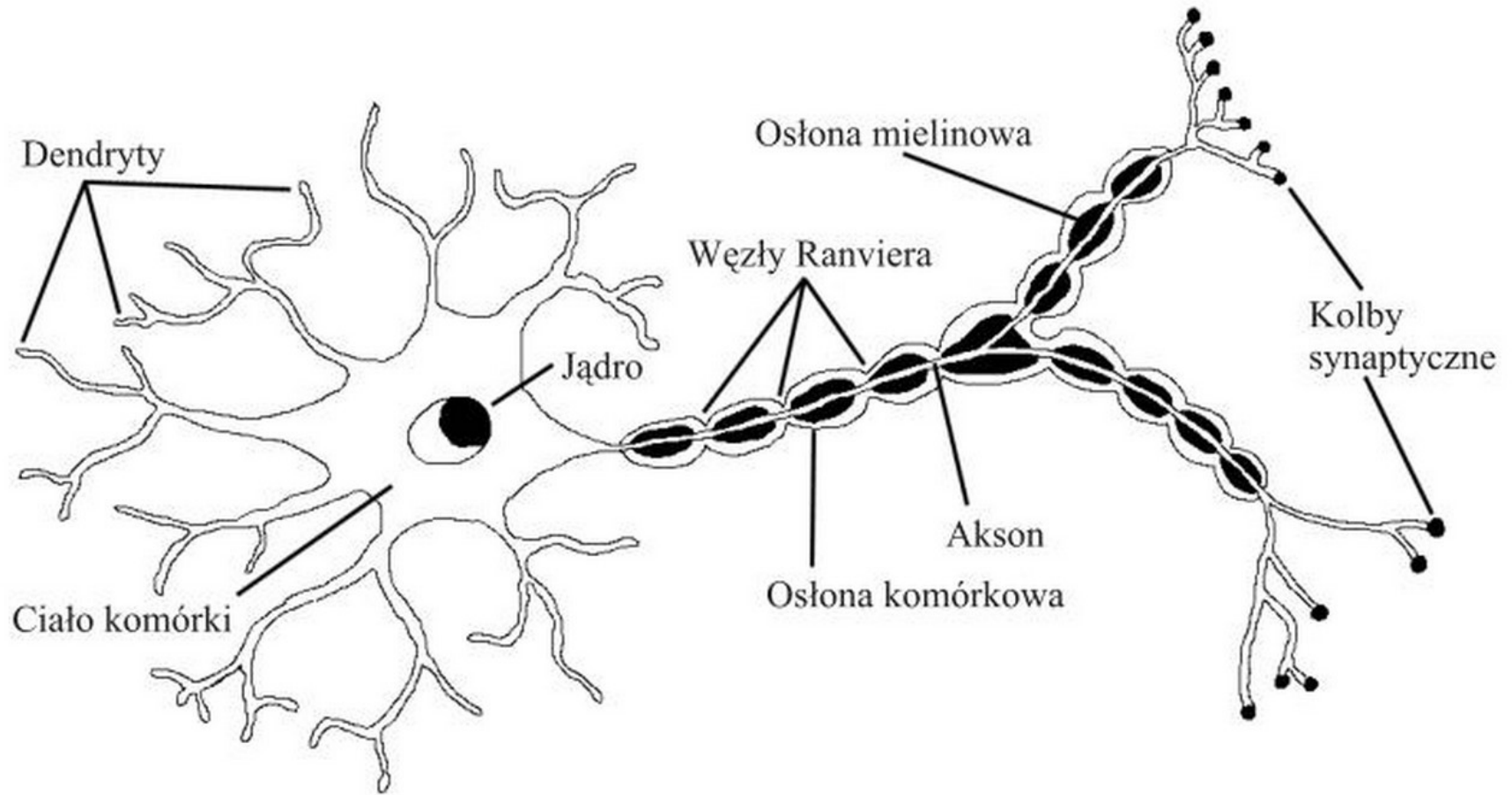


# Budowa neuronu





# Pojedynczy naturalny neuron



# Definicja

## **Sztuczna sieć neuronowa**

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

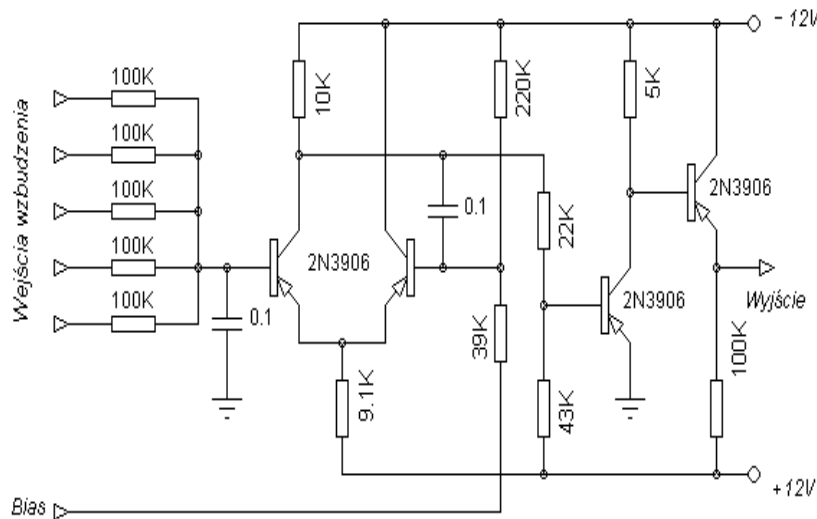
Przypomina mózg w dwóch aspektach:

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

# 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).



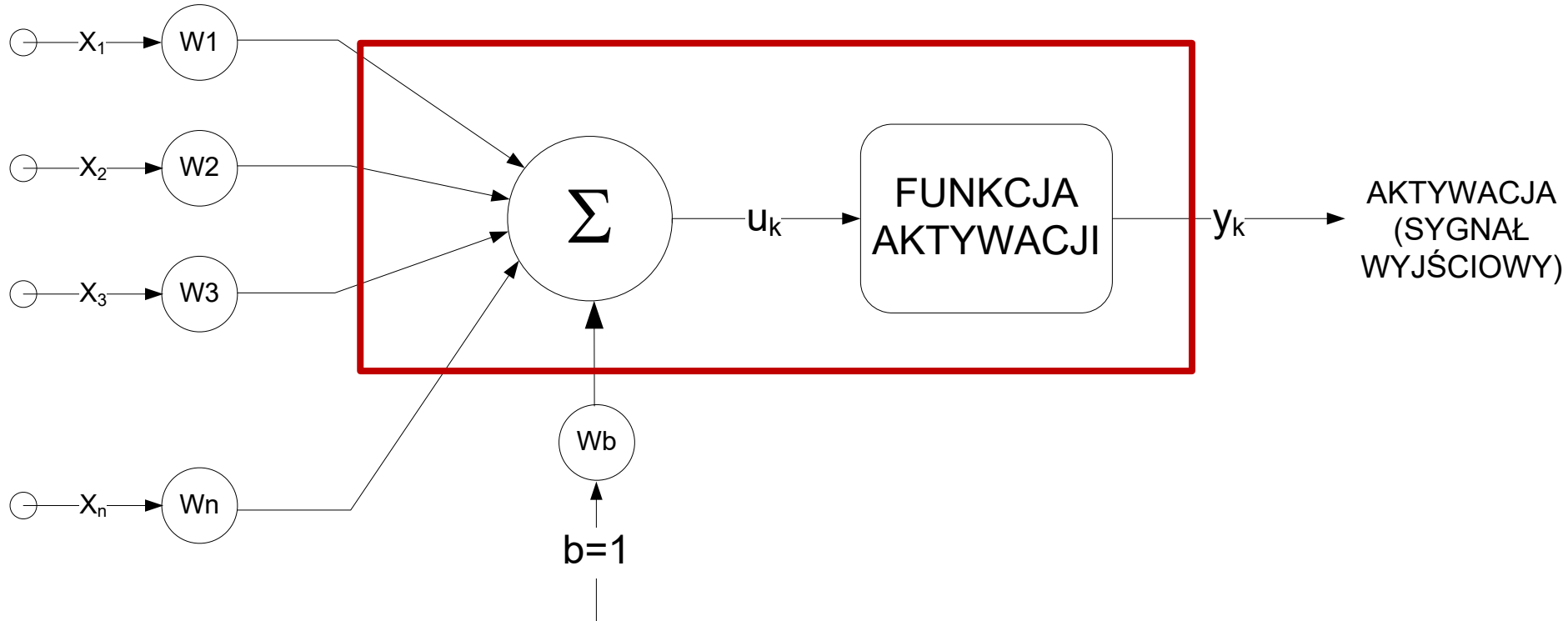
Schemat układu elektronicznego odwzorowującego pojedynczy neuron

```
/** 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)

WAGI SYNAPTYCZNE

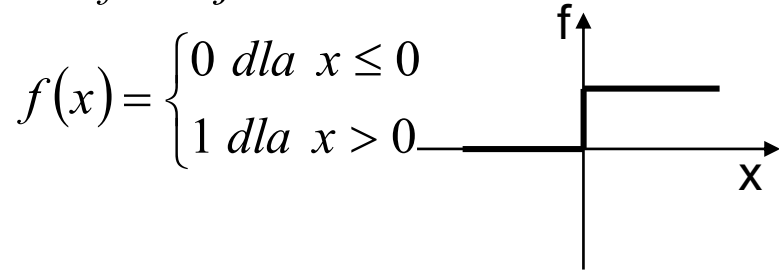


$$u_k = \sum_i w_{ki} x_i + w_b \cdot 1$$

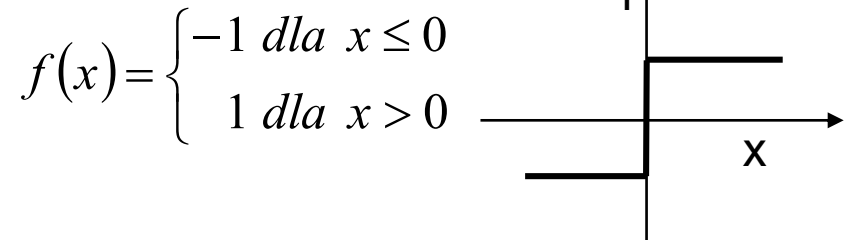
# Funkcje aktywacji

## Perceptron

$\Theta$  – funkcja Heaviside'a dana wzorem :



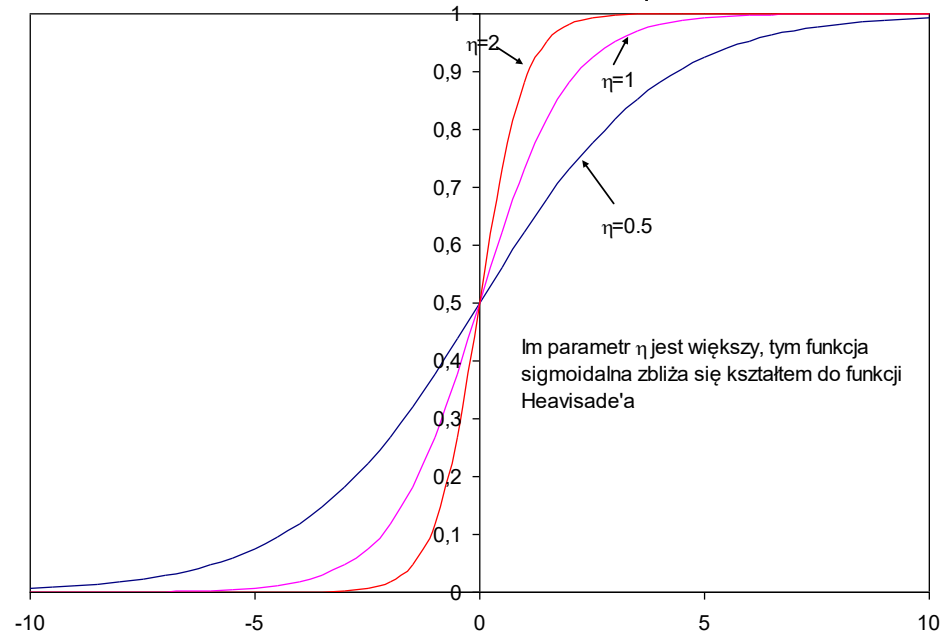
$\Theta$  – funkcja Heaviside'a dana wzorem :



$\Theta$  – funkcja sigmoidalna dana wzorem :

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

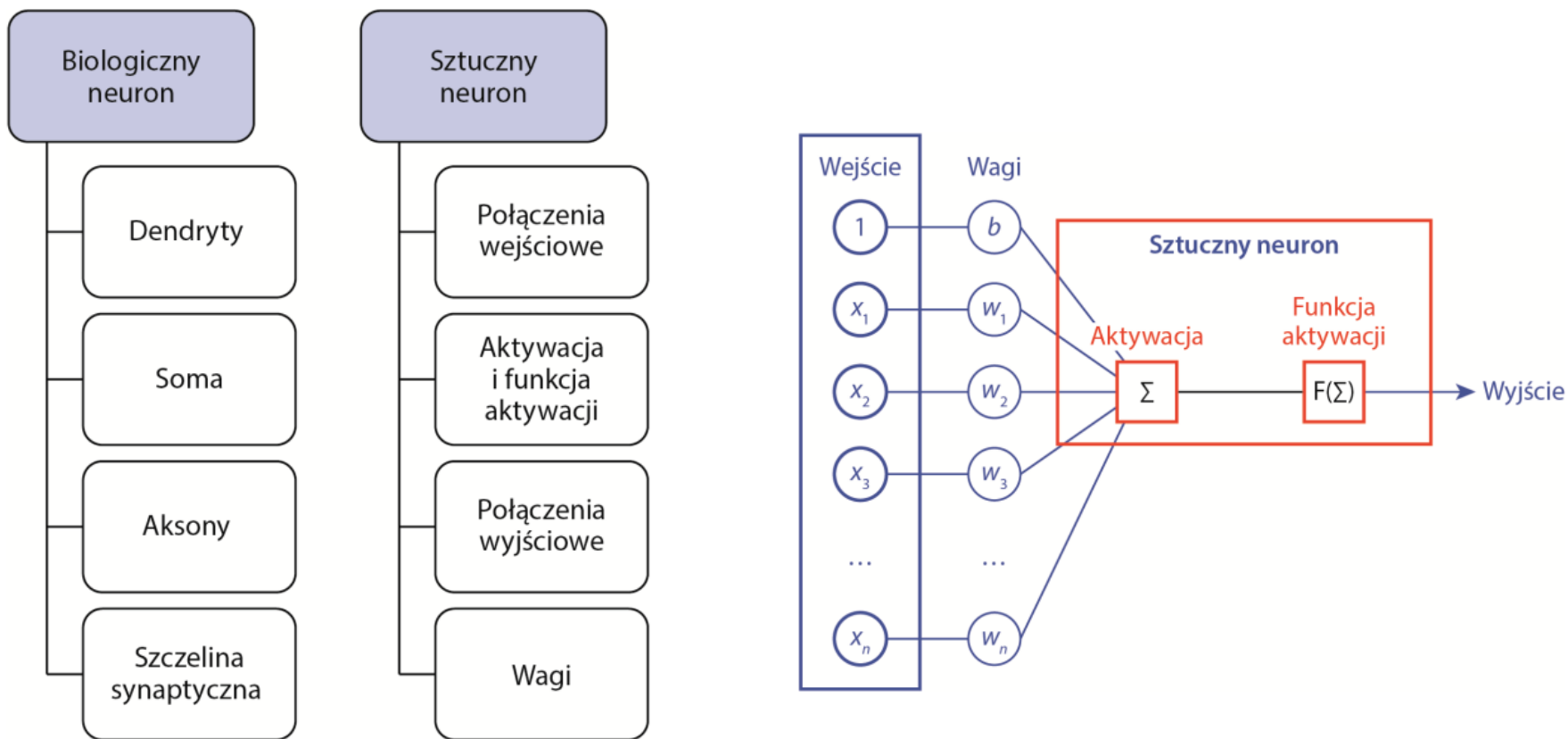
gdzie  $\eta$  – parametr wzmocnienia



# 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	$f(x) = \cos(\beta * x)$	$f'(x) = -\beta \sqrt{1 - f^2(x)}$
$\frac{x}{(1 +  x )}$ (bez nazwy)	$f(x) = \frac{\beta * x}{(1 +  \beta * x )}$	$f'(x) = \frac{\beta}{1 +  \beta * x } - \frac{ \beta * x }{(1 +  \beta * x )^2}$

# Porównanie funkcjonalne neuronu biologicznego z neuronem sztucznym



Źródło: Valentino Zocca, Gianmario Spacagna, Daniel Slater, Peter Roelants.  
Deep Learning. Uczenie głębokie z językiem Python.

# 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 nieznanymi 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

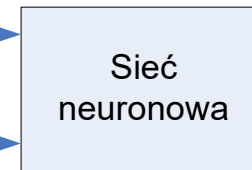
```
t={-3,2,-5,0,1,20,6,7};  
for (i=1,i<=n,i++)  
{  
    if (t(i) < 5) odp=-1;  
    else odp=1;  
}
```

$t = \{-3, 3, -5, 0, 1, 20, 6, 7\}$

$t = \{-3, 2, -5, 0, 1, 20, 6, 7\}$

Wartość odniesienia: 1

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



Zbiór uczący

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

OD  
PO  
WI  
E  
D  
Ź



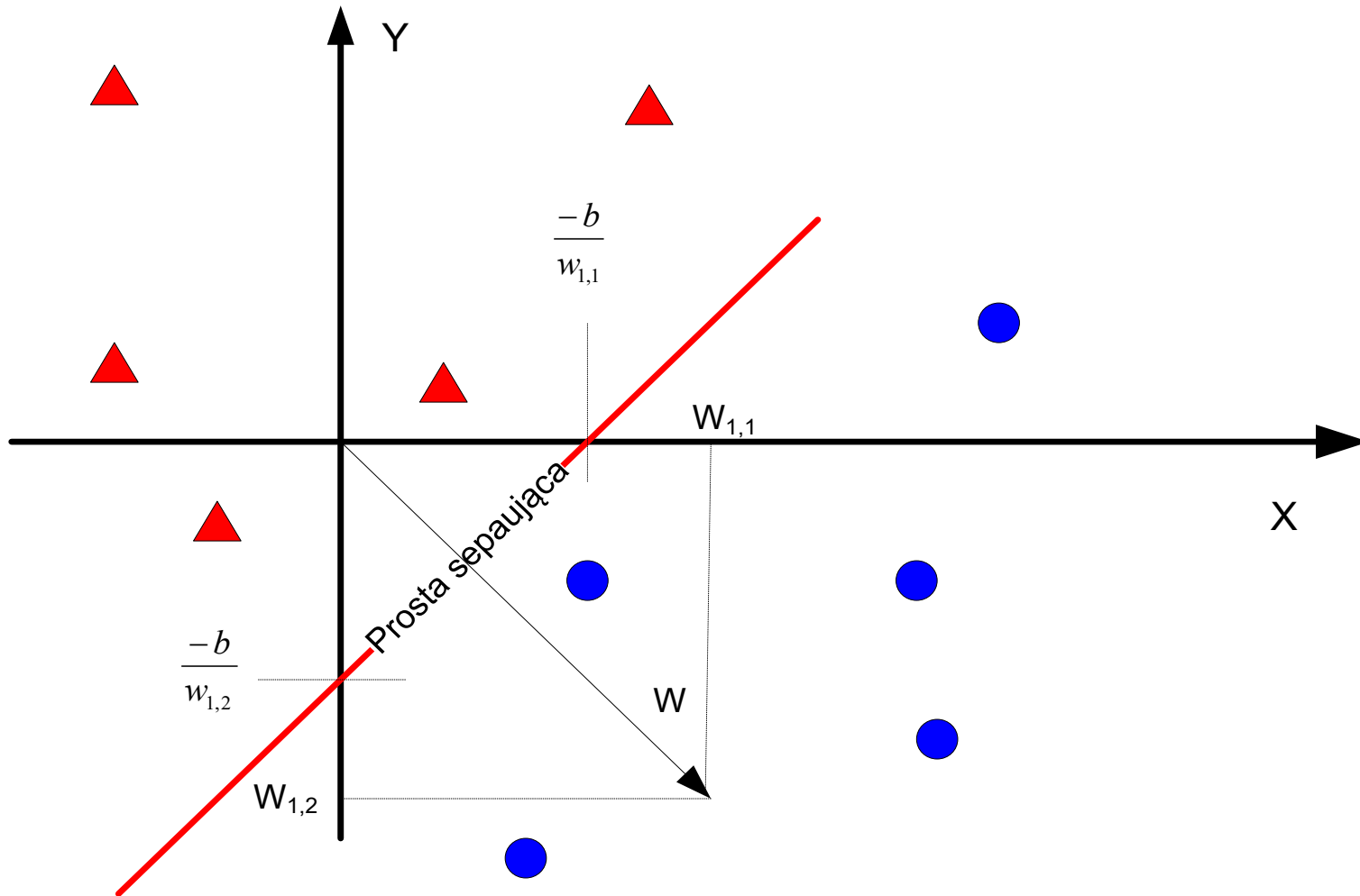
# Uczenie sieci

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.

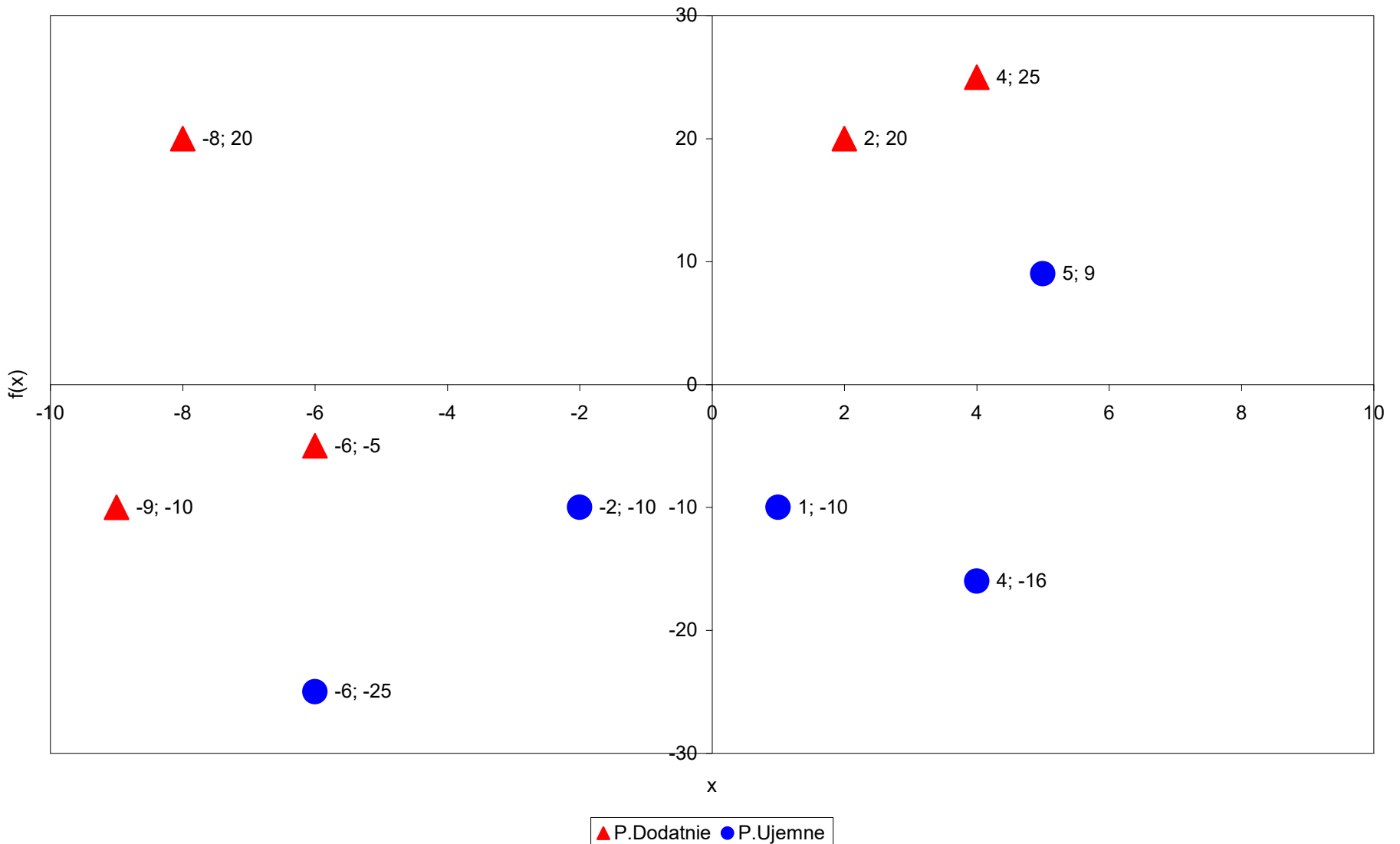


EPOKA

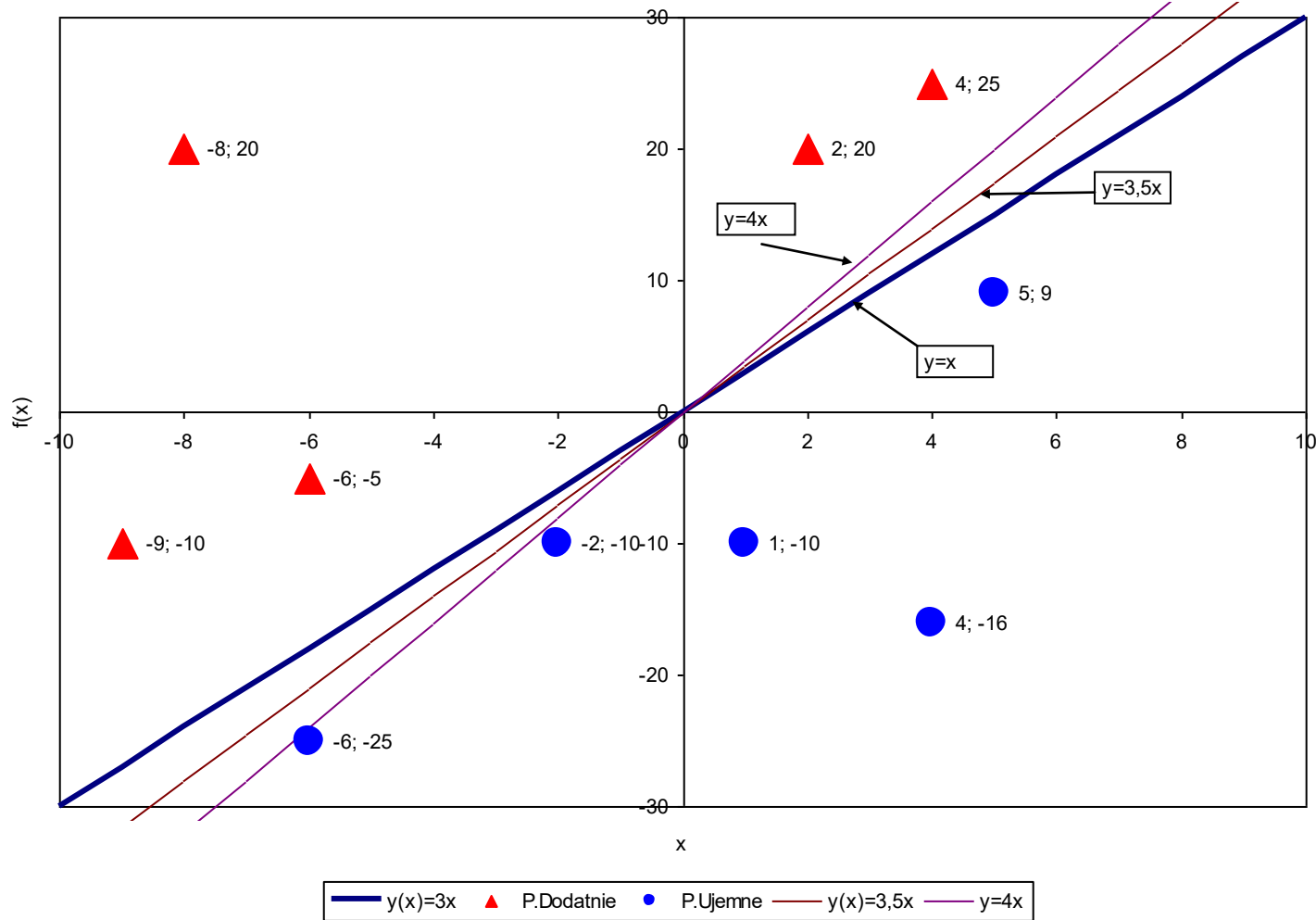
# Przykład klasyfikowania zbiorów



# Klasyfikacja punktów na płaszczyźnie

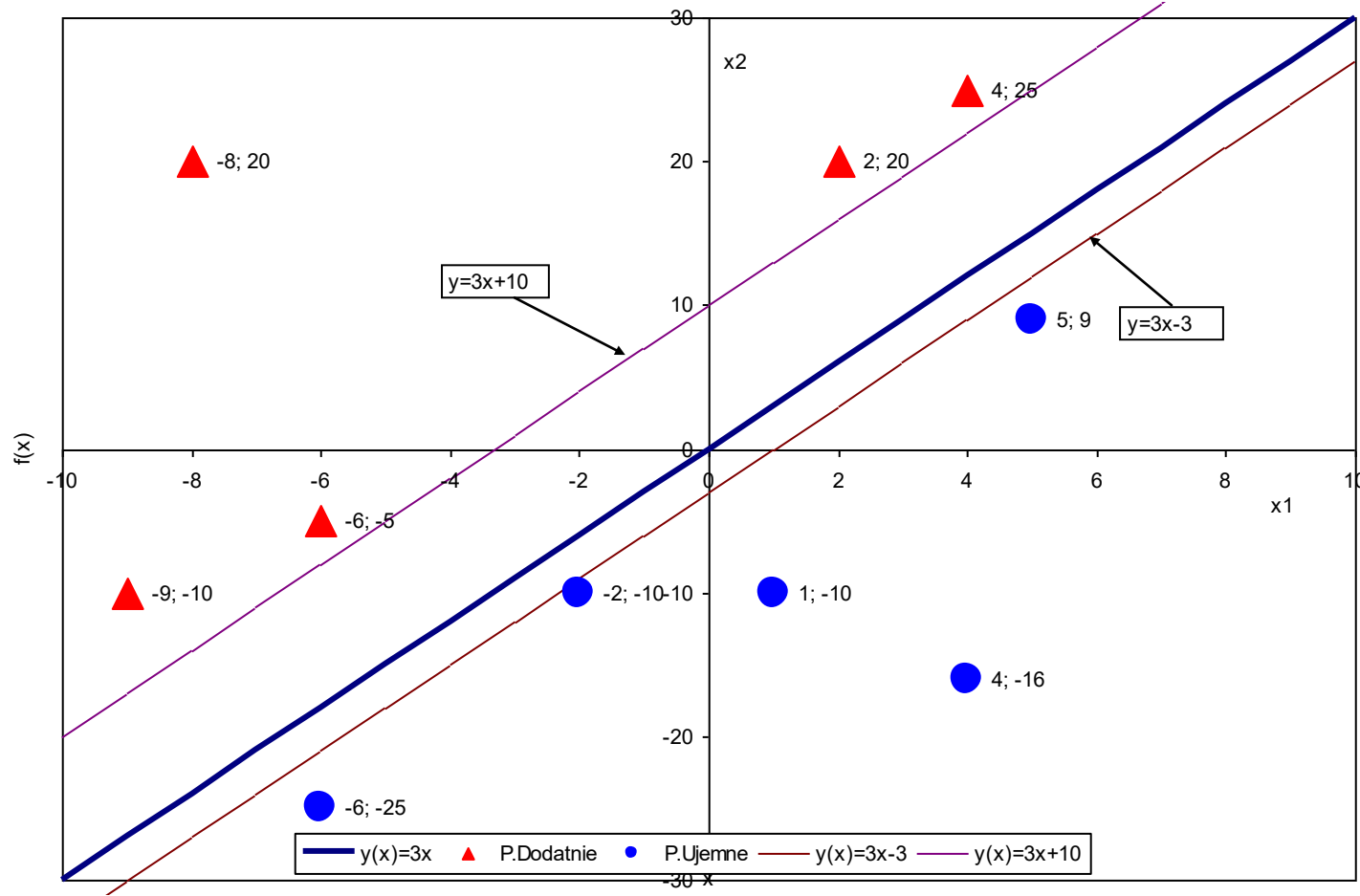


# Klasyfikacja punktów na płaszczyźnie



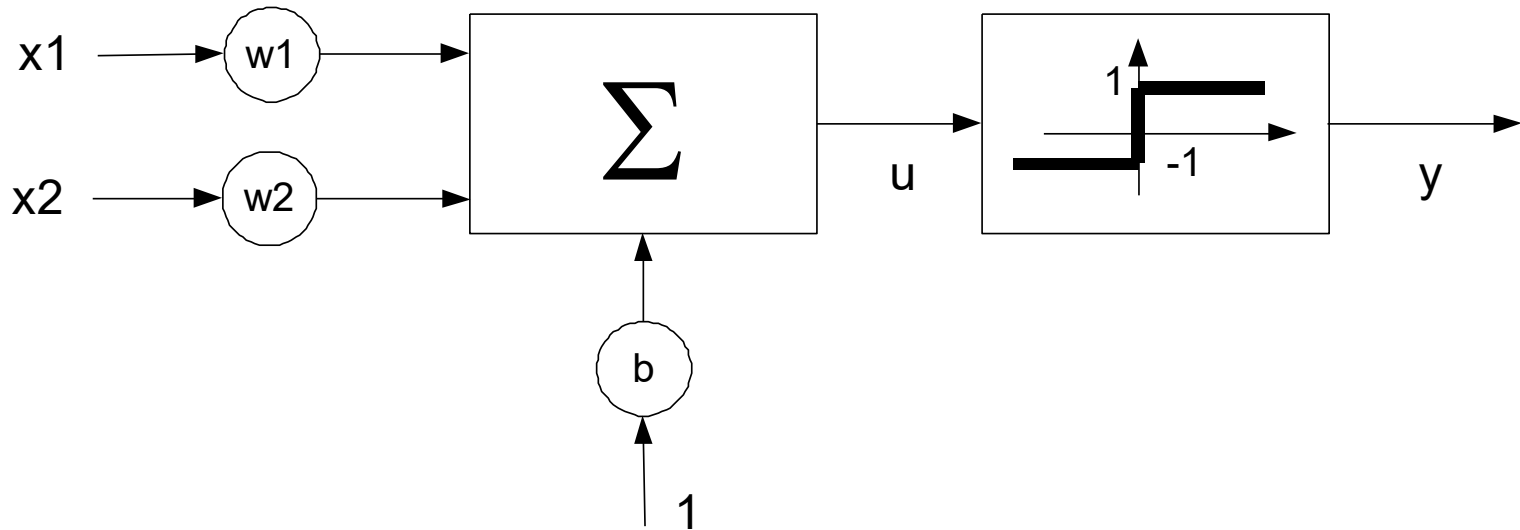
$$y = w \cdot x$$

# Klasyfikacja punktów na płaszczyźnie



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

# Perceptron dwuwejściowy



$$y = f(w_1 \cdot x_1 + w_2 \cdot x_2 + b)$$

$$y = ax + b$$

$$y = f\left([w_1 \quad w_2] * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b\right)$$

$$a \rightarrow [w_1 \quad w_2]$$

$$x \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

# Perceptron dwuwejściowy

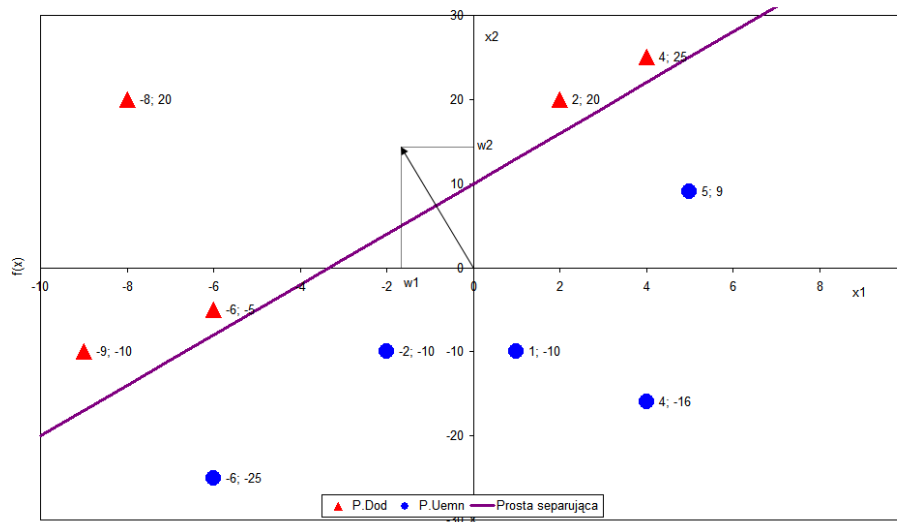
W przypadku dwóch wejść  $[x_1 \ x_2]^T$  perceptron dzieli płaszczyznę na dwie części. Zatem może klasyfikować punkty należące do dwóch umownych klas, np. "dodatnich" i "ujemnych". Podział ten wyznacza prosta o równaniu:

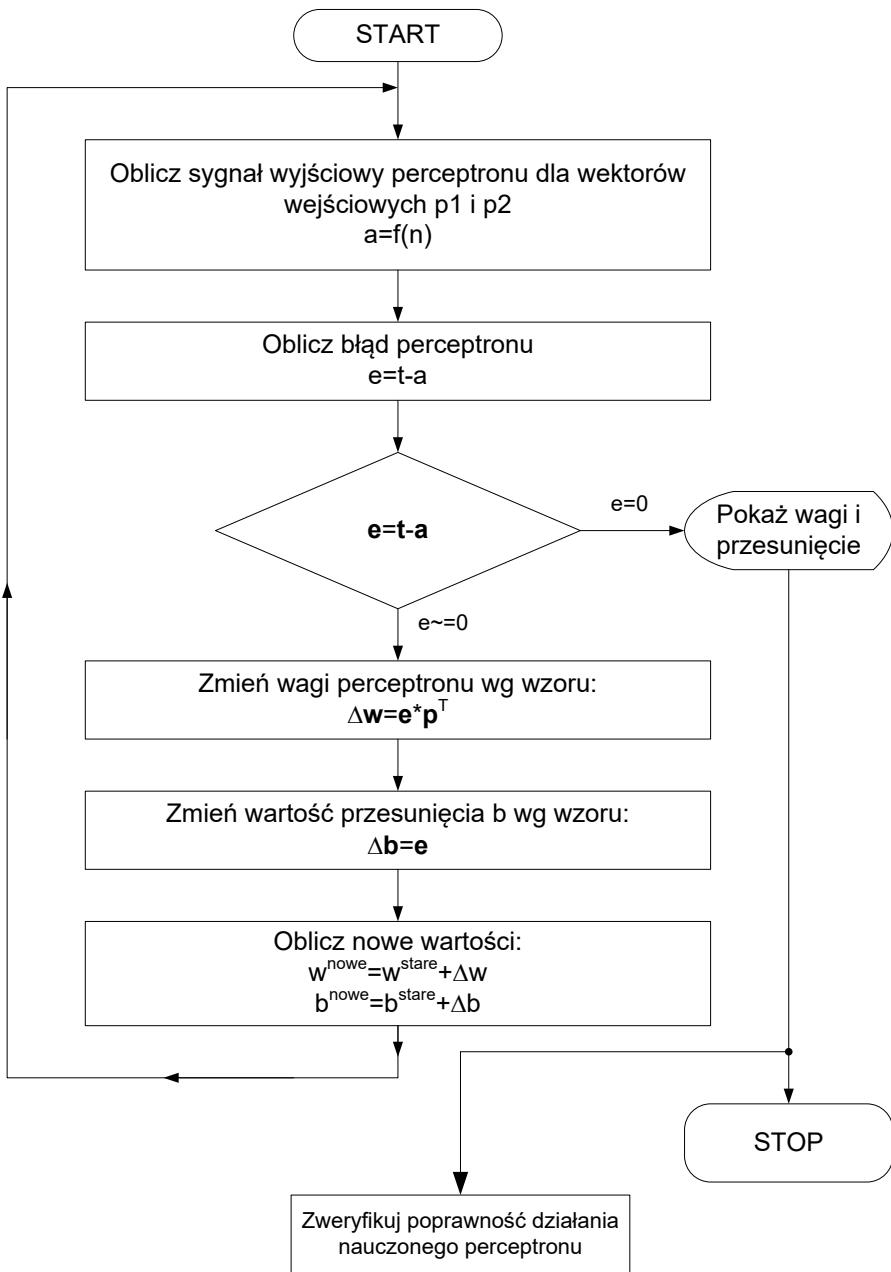
$$w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0$$

Zatem, gdy

$$x_1=0, \text{ mamy } x_2=-b/w_2$$

$$x_2=0, \text{ mamy } x_1=-b/w_1$$





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.



# Weryfikacja działania sieci.

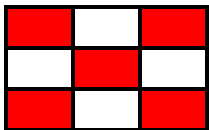
1. Na wejście sieci wprowadza się wektor weryfikujący, który ma te same cechy, co ciąg uczący, czyli musi dokładnie charakteryzować problem i muszą być znane dokładne odpowiedzi. Uwaga! NIE MOŻE TO BYĆ JEDEN Z WEKTORÓW UCZĄCYCH!
2. Przetwarza się wektor weryfikujący rejestrując liczbę odpowiedzi poprawnych.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

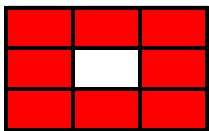
1	0	1
0	1	0
1	0	1

wektory wejściowe

1	0	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---	---

wektory wyjściowe

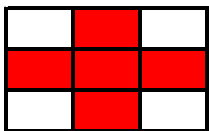
1	0	0	0
---	---	---	---



1	1	1
1	0	1
1	1	1

1	1	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---

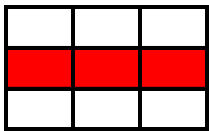
0	1	0	0
---	---	---	---



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

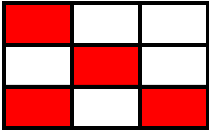
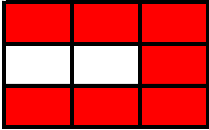
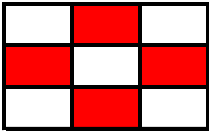
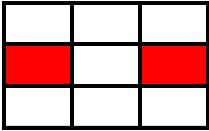
0	0	0	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---

0	0	0	1
---	---	---	---

Wniosek:

Sieć musi posiadać 9 wejść i 4 wyjścia.

# Weryfikacja działania sieci

Znaki	Matryce znaków	wektory wejściowe	wektory wyjściowe																						
	<table border="1"><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table>	1	0	0	0	1	0	1	0	1	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>	1	0	0	0	1	0	1	0	1	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0	0																							
0	1	0																							
1	0	1																							
1	0	0	0	1	0	1	0	1																	
1	0	0	0																						
	<table border="1"><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	0	0	1	1	1	1	<table border="1"><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	0	0	1	1	1	1	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	0	0
1	1	1																							
0	0	1																							
1	1	1																							
1	1	1	0	0	1	1	1	1																	
0	1	0	0																						
	<table border="1"><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	1	0	1	0	1	0	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	1	0	1	0	1	0	<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	1	0
0	1	0																							
1	0	1																							
0	1	0																							
0	1	0	1	0	1	0	1	0																	
0	0	1	0																						
	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	0	1	0	0	0	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	0	1	0	0	0	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	1
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

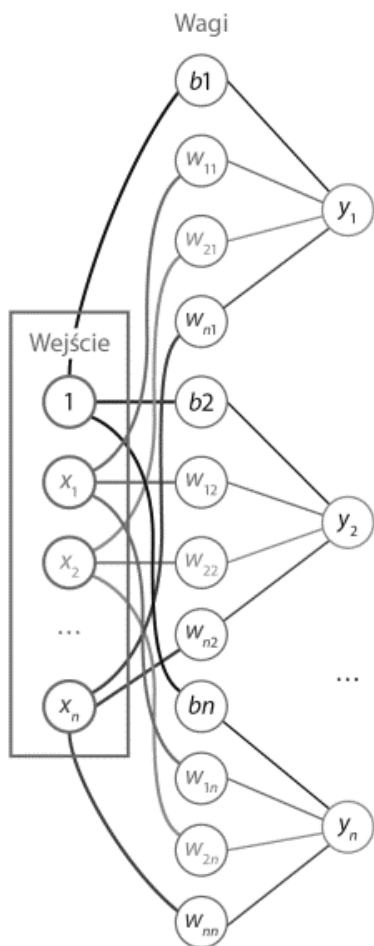
Sieci jednokierunkowe – sygnał rozprzestrzenia się w jednym kierunku

Sieci jednowarstwowe

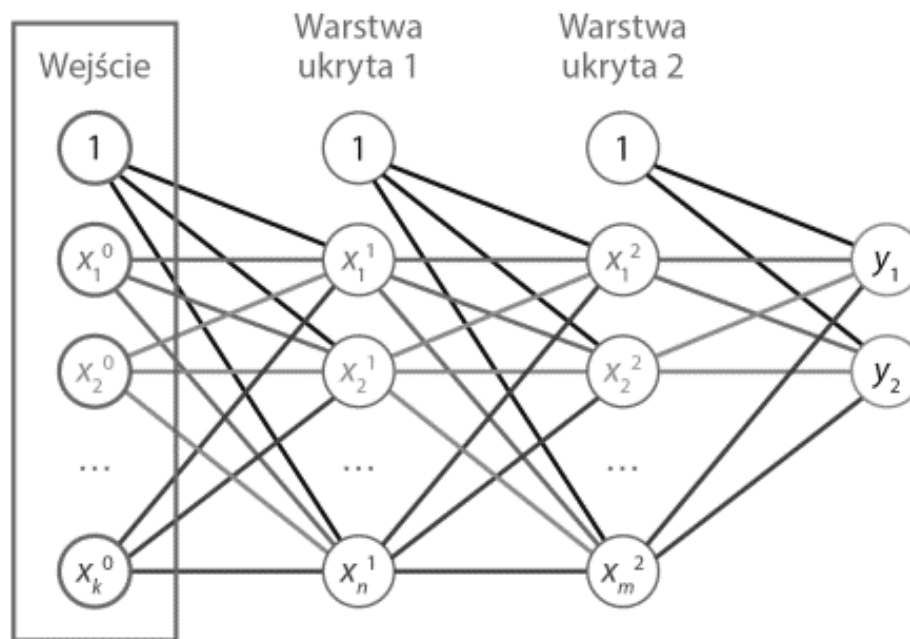
Sieci wielowarstwowe

Sieci rekurencyjne – sieci ze sprzężeniem zwrotnym.

# Rodzaje sieci neuronowych



Jednowarstwowa SSN



Wielowarstwowa SSN

# Cechy sieci neuronowych

- *adaptacyjność*, to znaczy zdolność uczenia się przez dostosowanie parametrów i struktury sieci do zmian otoczenia;
- *uogólnianie*, czyli zdolność wypracowywania rozsądnych decyzji w sytuacjach, z którymi sieć nie miała dotychczas styczności;
- *równoległość obliczeń*, czyli w przeciwieństwie do komputera z architekturą von Neumanna, gdzie instrukcje są wykonywane sekwencyjnie, w sieciach neuronowych obliczenia są wykonywane jednocześnie przez dużą liczbę połączonych wzajemnie jednostek – neuronów;
- *nieliniowość*, rozumiana jako zdolność modelowania nieliniowych zależności wyjściowo-wejściowych przez sieć;
- *odporność na błędy*, czyli zdolność działania mimo pewnych uszkodzeń sieci oraz także zdolność uczenia się na podstawie danych błędnych, niepełnych, rozmytych i nieprecyzyjnych.