

# Sieci neuronowe - wstęp

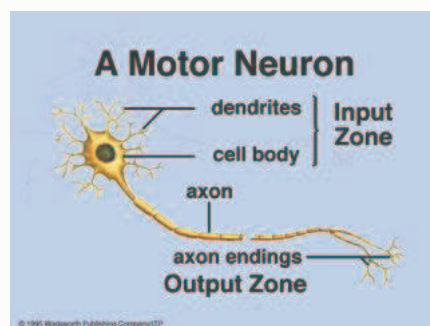
[www.qed.pl/ai/nai2003](http://www.qed.pl/ai/nai2003)

## Literatura:

S. Osowski, *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa 1997.

## Sztuczne sieci neuronowe

Geneza: Naśladowanie działania naturalnych neuronów



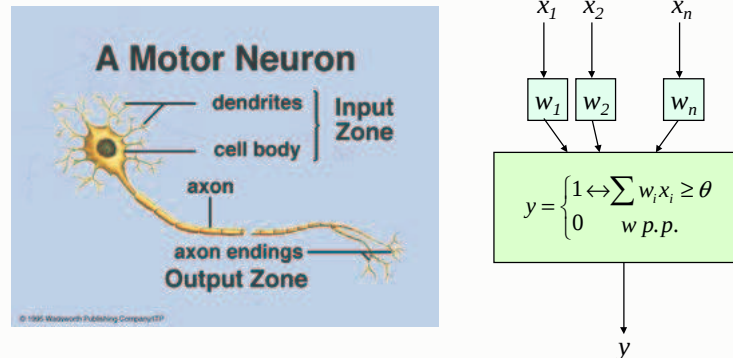
## Perceptron (Rosenblatt 1958)

- Wejście
  - n stanów wejściowych  $x_1, \dots, x_n$
  - stany mogą być cyfrowe lub analogowe
- Wyjście
  - 0 lub 1
- Parametry perceptronu
  - n wag połączeń  $w_1, \dots, w_n \in \mathfrak{R}$
  - wartość progowa  $\theta \in \mathfrak{R}$

## Perceptron

- Zasada działania
  - Do każdego i-tego wejścia przypisana jest waga  $w_i$
  - Dla danych stanów wejściowych  $x_1, \dots, x_n$  liczymy sumę ważoną:
$$s = \sum_{i=1}^n w_i x_i$$
  - Jeżeli  $s \geq \theta$ , to ustawiamy wyjście  $y = 1$ , zaś w przeciwnym przypadku ustawiamy  $y = 0$

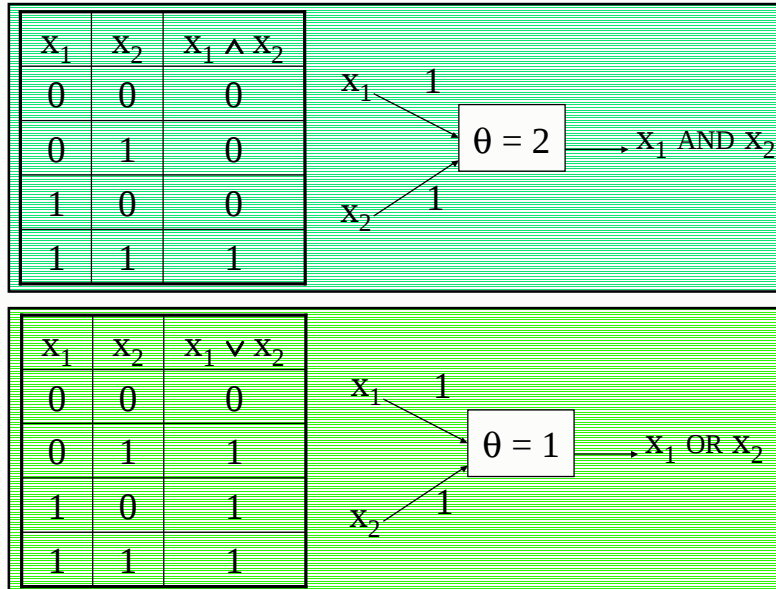
## Analogia z neuronem naturalnym



## Jak opisać perceptron

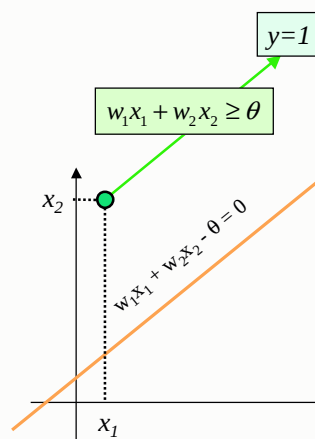
- Perceptron opisuje jednoznacznie zbiór wag  $w_1, \dots, w_n \in \mathfrak{R}$  oraz wartość progowa  $\theta \in \mathfrak{R}$
- Wartości  $x_1, \dots, x_n \in \mathfrak{R}$  to zmienne pojawiające się na wejściu do modelu perceptronu

## Co potrafi perceptron



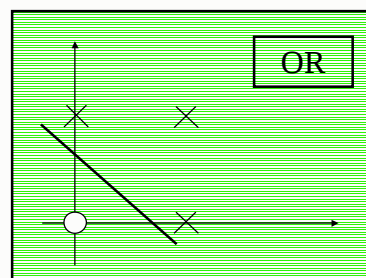
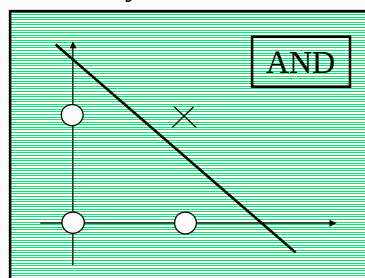
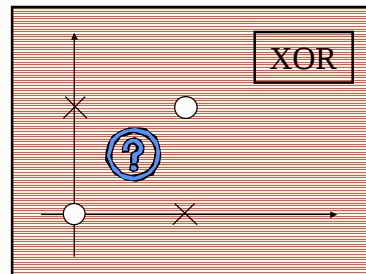
## Co potrafi perceptron

- Równanie perceptronu można potraktować jako równanie prostej (ogólnie: hiperpłaszczyzny w przestrzeni n-wymiarowej).
- Punkty leżące nad ową prostą klasyfikujemy jako 1, zaś pozostałe jako 0.



## Czego perceptron nie potrafi

- Pojedynczy perceptron nie potrafi odróżniać zbiorów nieseparowalnych liniowo, np. funkcji XOR.
- Odkrycie tych ograniczeń (1969) na wiele lat zahamowało rozwój sieci neuronowych.



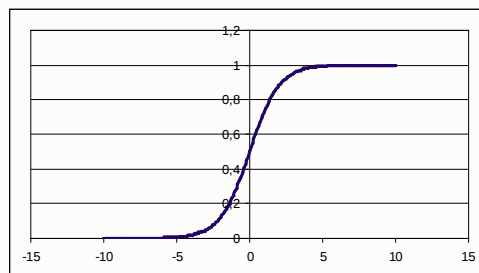
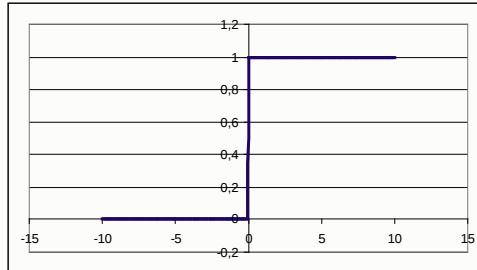
## Czego perceptron nie potrafi

- Zadaniem pojedynczego perceptronu jest jedynie:
  - przetwarzanie jednostkowych informacji
  - podejmowanie prostych decyzji
  - przekazywanie wyników sąsiadom
- Dopiero w połączeniu z innymi węzłami uzyskuje się zdolność podejmowania złożonych decyzji

## Funkcje aktywacji

- Progowe

$$f(z) = \begin{cases} 1 & \Leftrightarrow z \geq 0 \\ 0 & \Leftrightarrow z < 0 \end{cases}$$

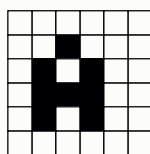


- Sigmoidalne

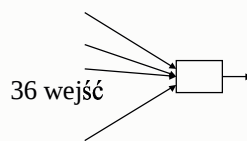
$$f(z) = \frac{1}{1 + e^{-z}}$$

## Uczenie perceptronu

**Przykład:** rozpoznawanie znaków



Siatka  $6 \times 6$

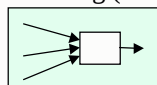


Wyjście: 1, jeśli na wejściu pojawia się litera "A", zaś 0 w p.p.

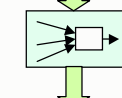
**Zadanie:** dobrać wagi wejść i wartość progową tak, by uzyskać zaplanowany efekt

Dane treningowe  
(znane odpowiedzi)

Dobór wag (uczenie)



Dane testowe



Odpowiedź

## Uczenie perceptronu

- **Wejście:**

- Ciąg przykładów uczących ze znanymi odpowiedziami

- **Proces uczenia:**

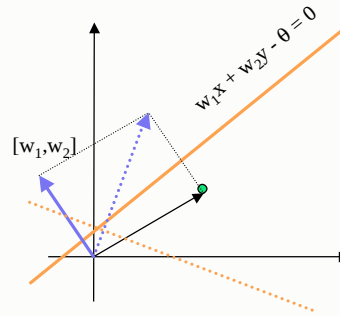
- Inicjujemy wagi losowo
- Dla każdego przykładu, jeśli odpowiedź jest nieprawidłowa, to

$$w_1 += \alpha x_1$$

$$w_2 += \alpha x_2$$

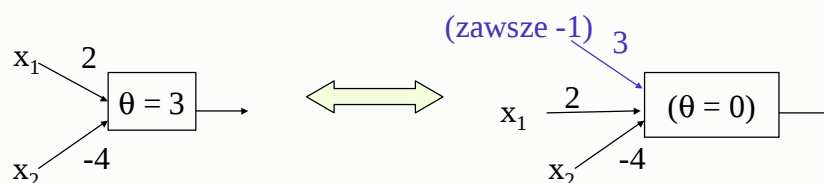
$$\theta -= \alpha$$

gdzie  $\alpha$  jest równe różnicy między odpowiedzią prawidłową a otrzymaną.



## Uczenie perceptronu

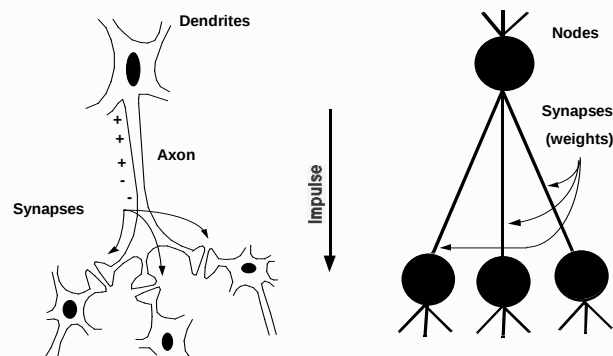
- Często  $\alpha$  mnoży się dodatkowo przez niewielki współczynnik uczenia
- Po wyczerpaniu przykładów, zaczynamy proces uczenia od początku, dopóki następują jakiegolwiek zmiany wag połączeń
- Próg  $\theta$  można traktować jako wagę dodatkowego wejścia o wartości -1:



## Uczenie perceptronu

- Opisany schemat jest w miarę przejrzysty tylko dla pojedynczych perceptronów, lub niewielkich sieci
- Ciężko jest stosować reguły tego typu dla skomplikowanych modeli
  - Tymczasem np. do rozpoznawania wszystkich liter potrzeba by sieci złożonej z 26 takich perceptronów

## Sieci perceptronów



*Ograniczenia pojedynczych perceptronów spowodowały w latach 80-tych wzrost zainteresowania sieciami wielowarstwowymi i opracowanie algorytmu ich uczenia (propagacja wsteczna)*