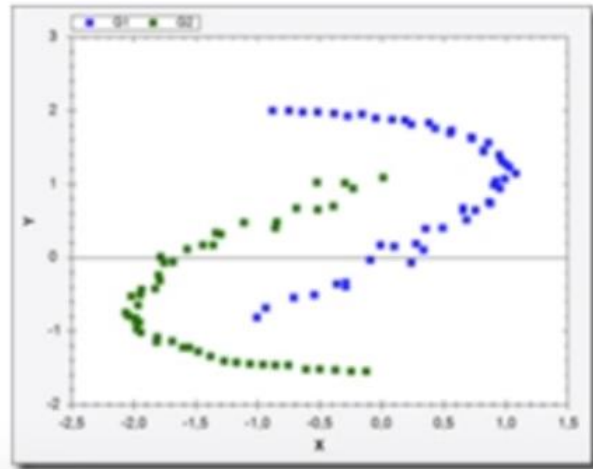
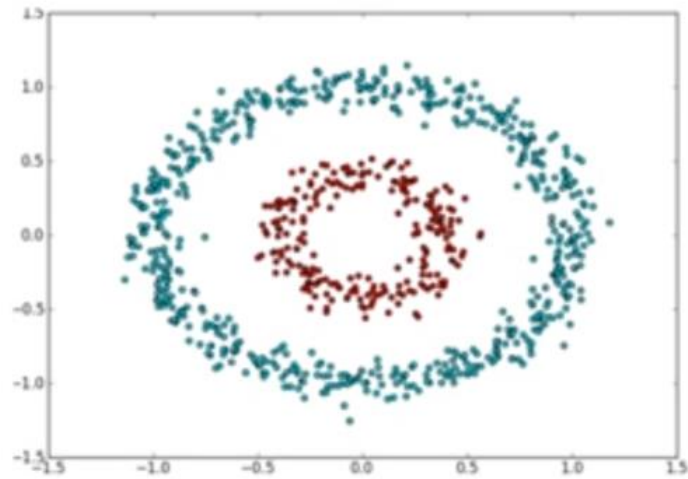


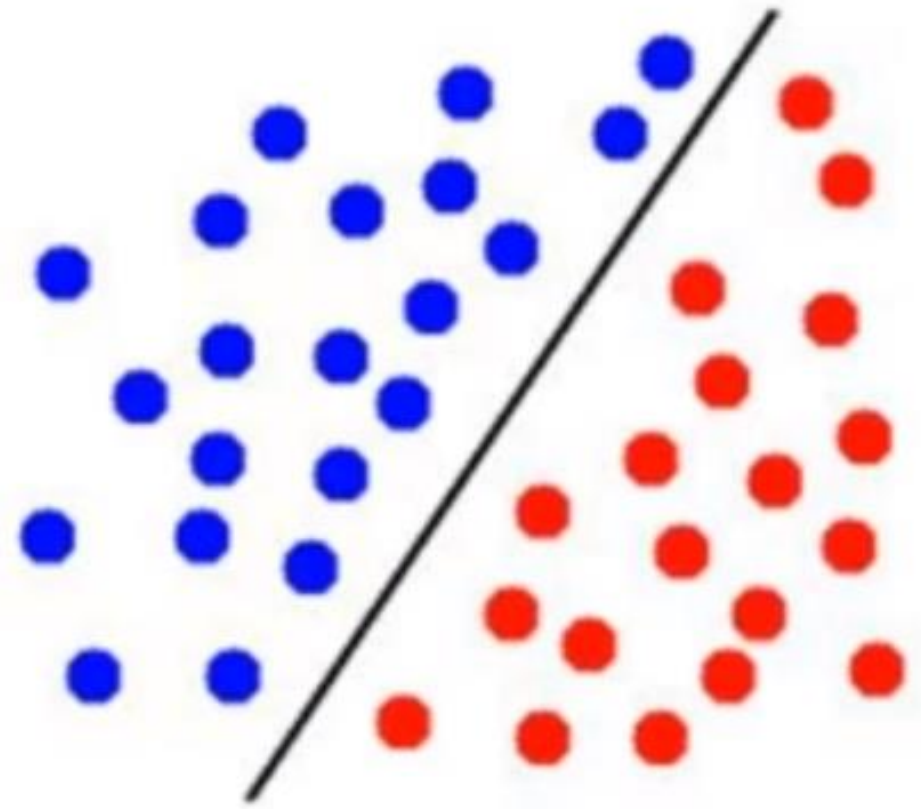
# Czym są sieci neuronowe

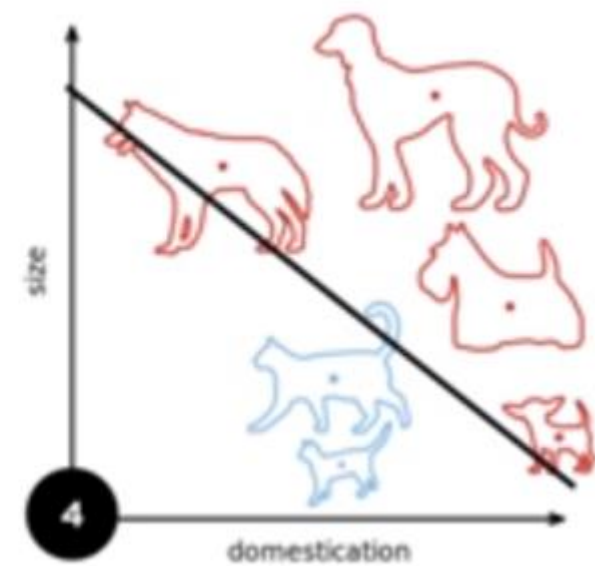
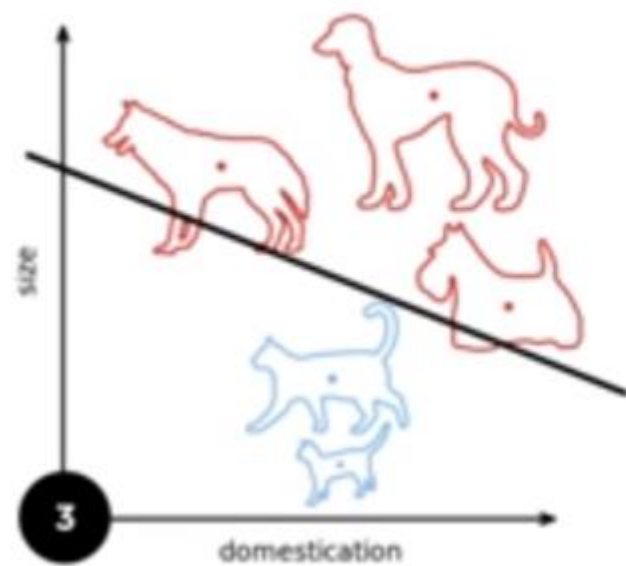
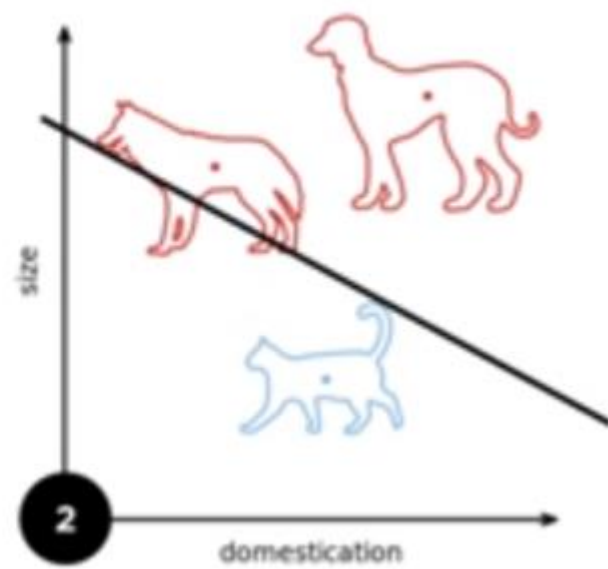
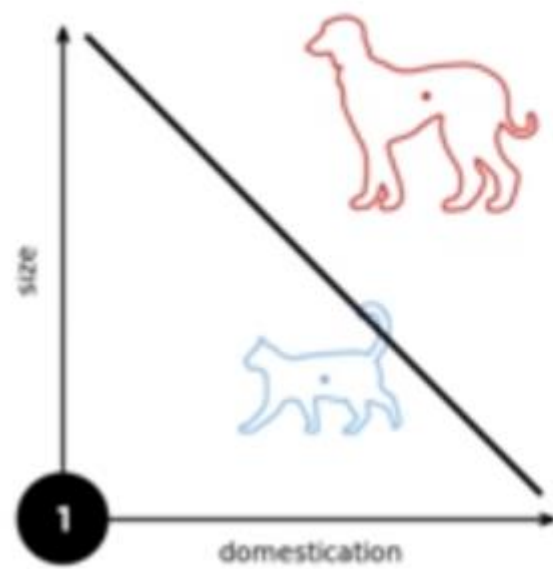
- Matematyczny model działania mózgu
- Przetwarzanie informacji równoległe w przeciwieństwie do szeregowego
- Potrafią się uczyć
- Prosta zasada działania (opowieść o ptakach)
- Skomplikowane zachowanie
- Zastosowanie
  - Rozwiązywanie problemów optymalizacyjnych (komiwojażer)
  - Modele biologiczne

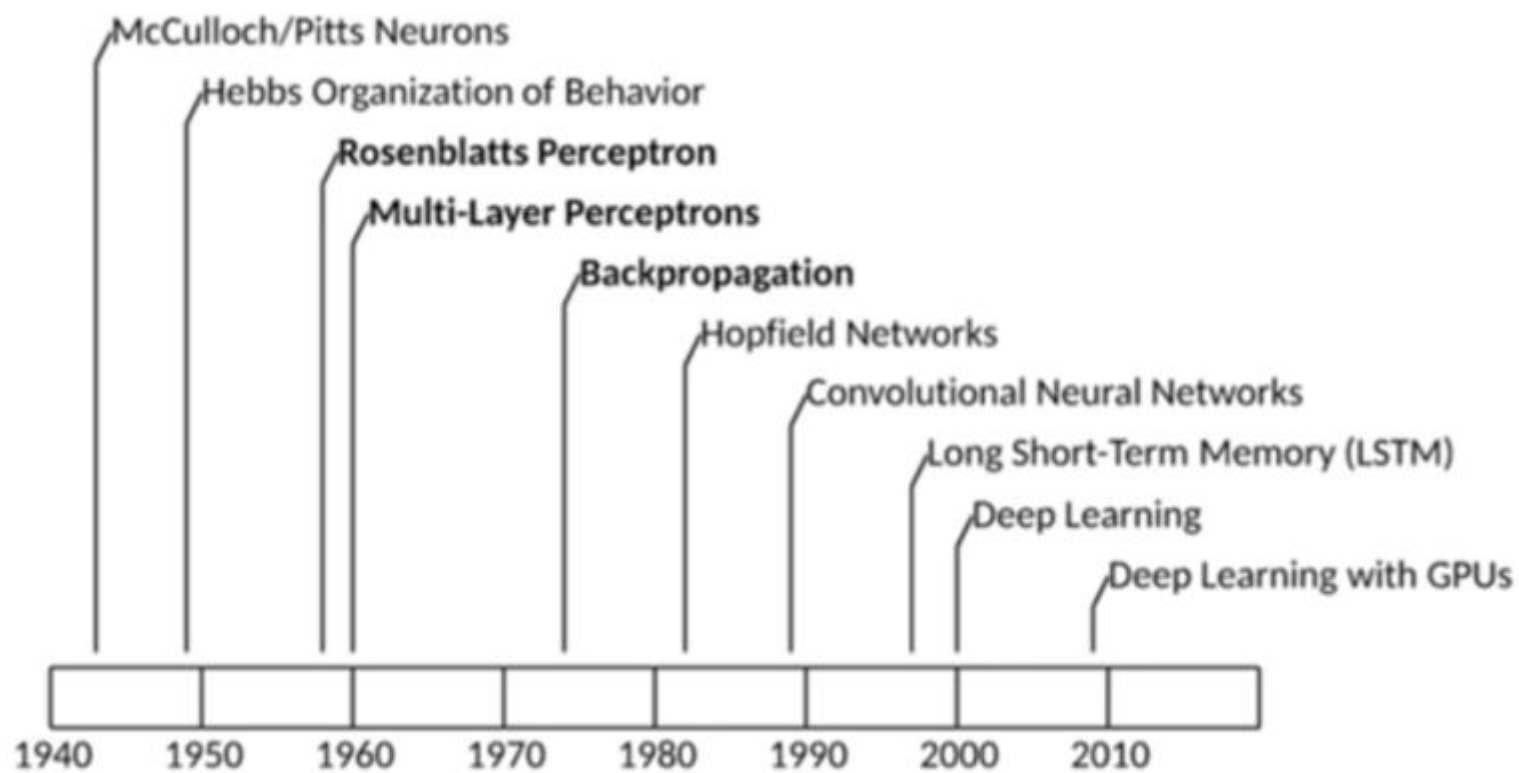
Not Linearly Separable



Linearly Separable

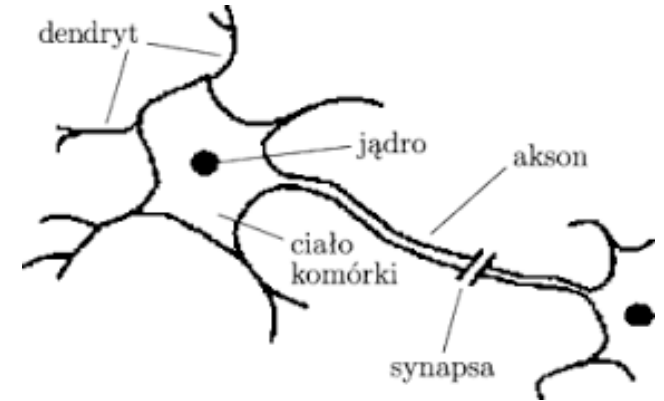






# Neuron biologiczny

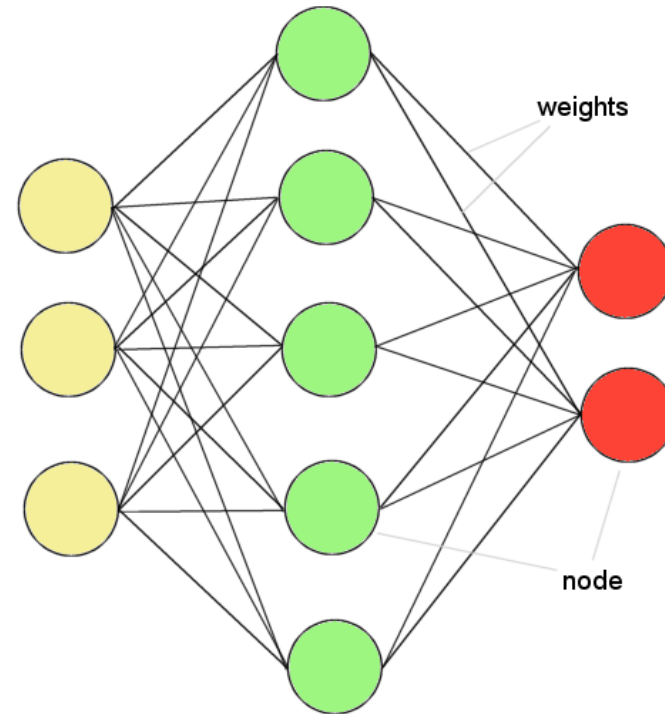
- Rodzimy się ze 100 miliardami neuronów
- Jeden neuron może posiadać do 100 000 połączeń z innymi neuronami
- Sygnały przesyłane za pośrednictwem procesów elektrochemicznych
- Sygnał może aktywować następny neuron



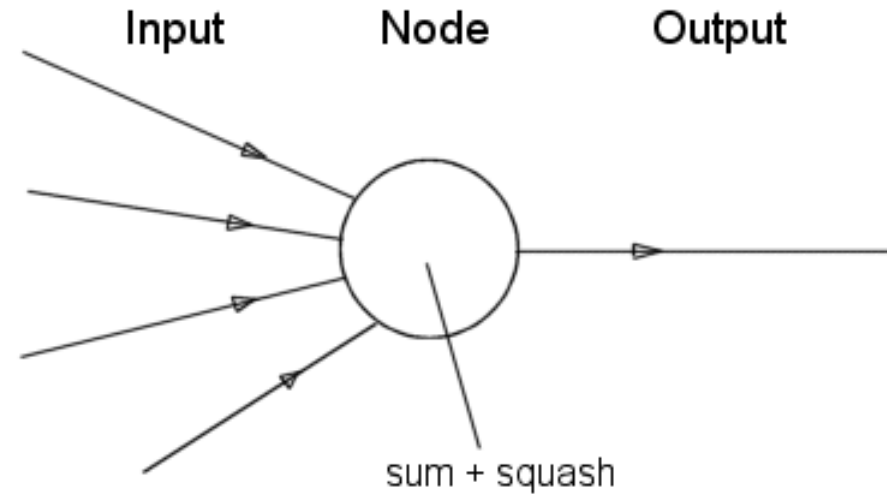
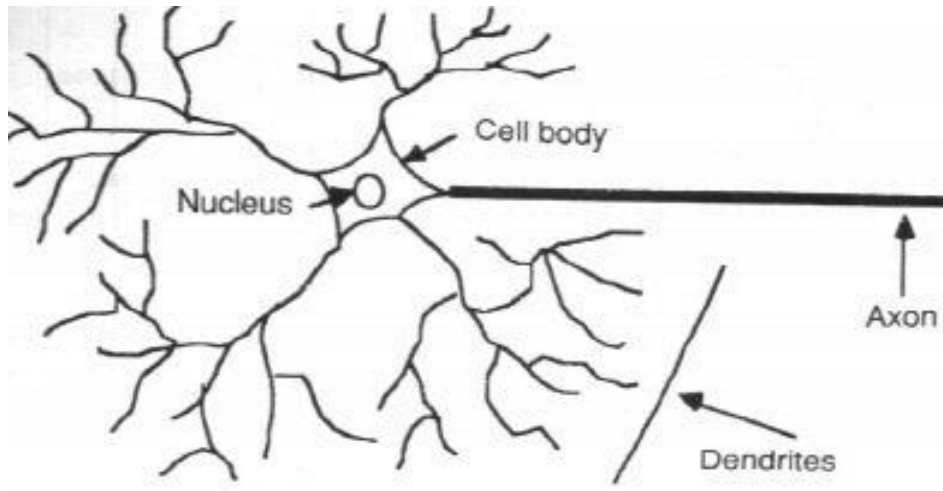
# Sieć neuronowa – zasada działania

- Wzorowana na działaniu mózgu

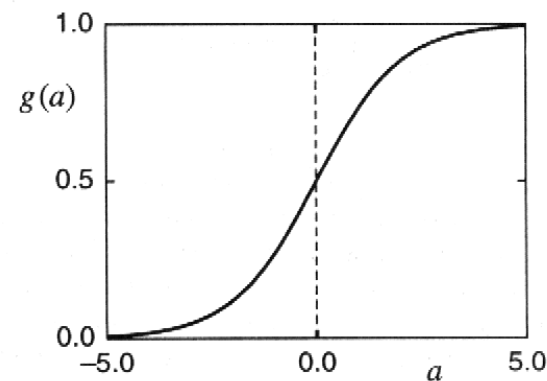
1. Neurony (wierzchołki)
2. Synapsy (wagi)



# Biologiczny, a sztuczny neuron



Funkcja „zgniatająca”



# Sprzężenie w przód

Informacja przepływa w jednym kierunku

Dane wczytywane do *Warstwy wejściowej*

Przekazywane do *Ukrytych warstw*

Przekazywane do *Warstwy wyjściowej*

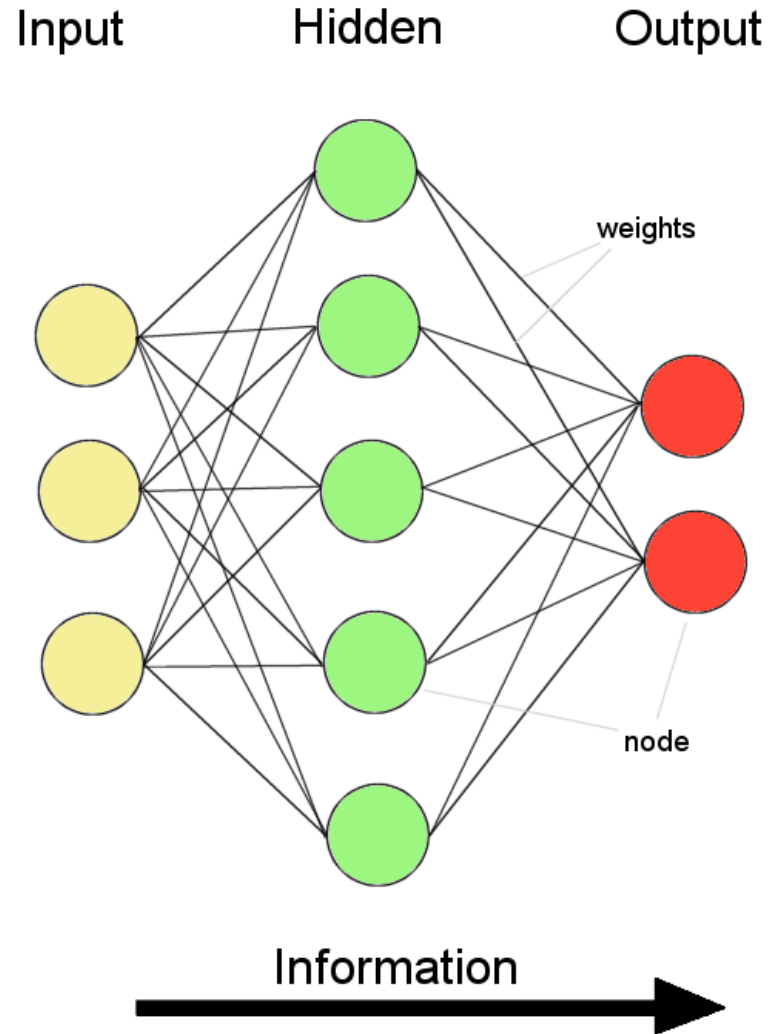
Równoległe przetwarzanie informacji

Przykłady danych wejściowych:

Kolor piksela

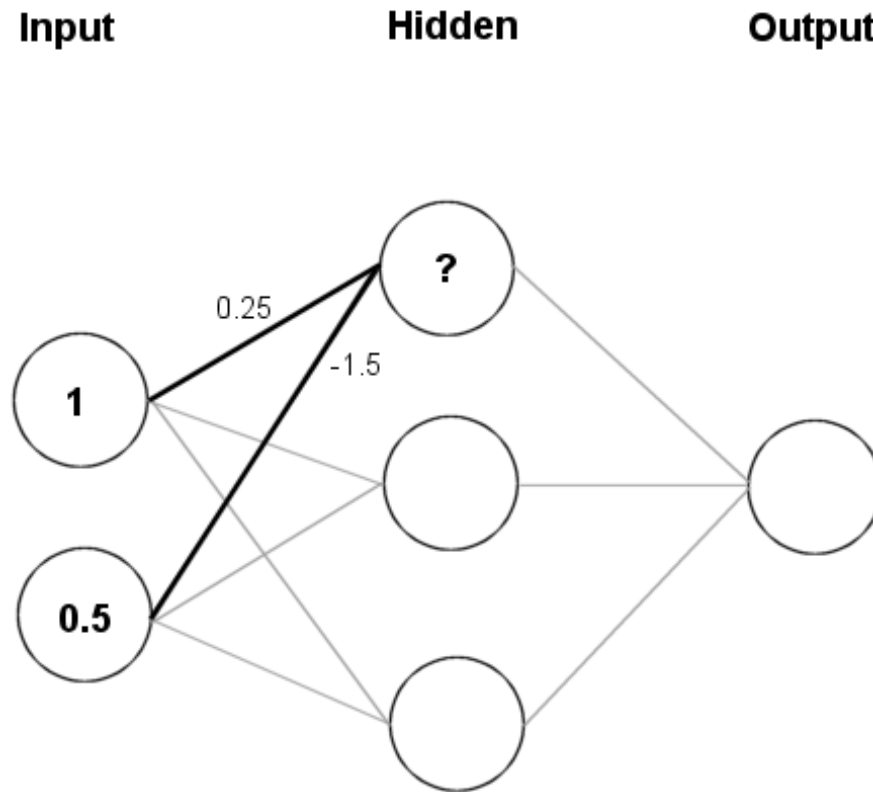
Koncentracja molekuł

Ceny produktów





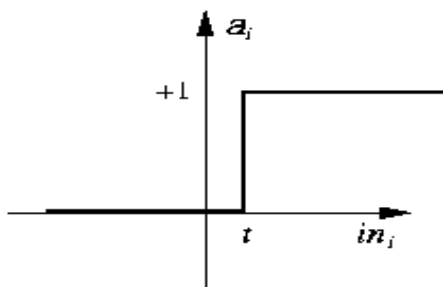
# Sprzężenie w przód



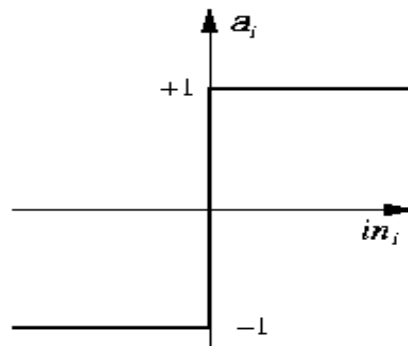
$$(1 \times 0.25) + (0.5 \times (-1.5)) = 0.25 + (-0.75) = -\mathbf{0.5}$$

Squashing:  $\frac{1}{1 + e^{0.5}} = 0.3775$

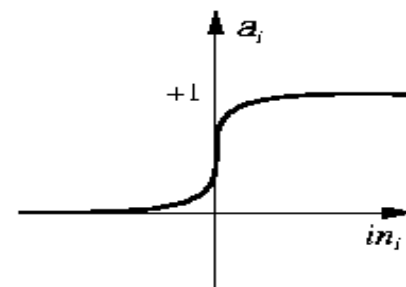
# Funkcje aktywacji



(a) Step function



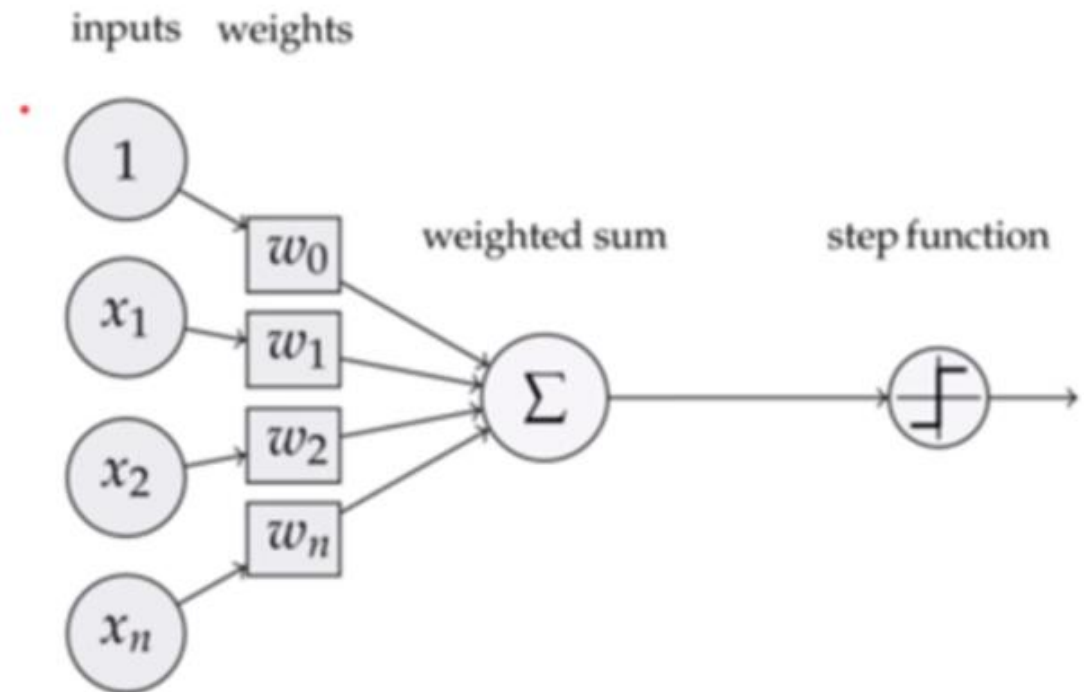
(b) Sign function



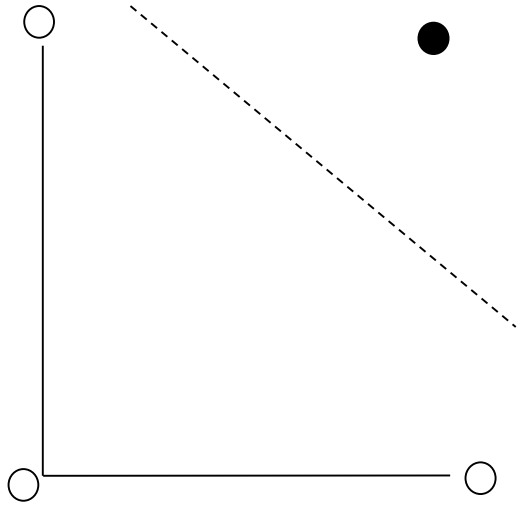
(c) Sigmoid function

## Perceptron (Input $\rightarrow$ Output)

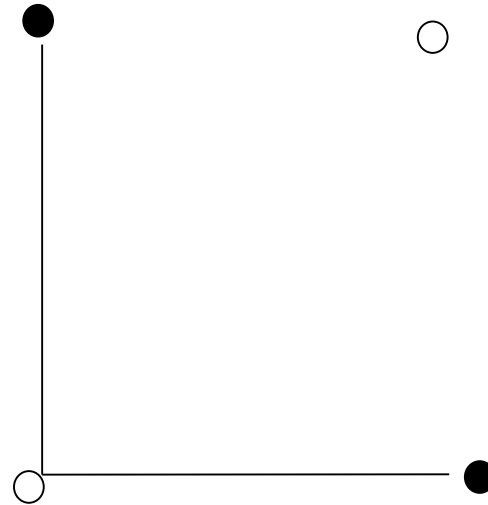
- Input values  
 $(1, x_1, x_2, \dots, x_n)$
- are multiplied by their respective weights  
 $(w_0, w_1, w_2, \dots, w_n)$
- This entire expression is then summed together inside the neuron  
 $(w_0 + x_1 w_1 + x_2 w_2 + \dots + x_n w_n)$
- If the sum is larger than a specified threshold, the step function outputs a 1, otherwise it outputs a 0.



# Co może reprezentować perceptron?



Funkcje liniowo rozłączne.



# Trenowanie/uczenie sieci

## Metoda propagacji wstecznej

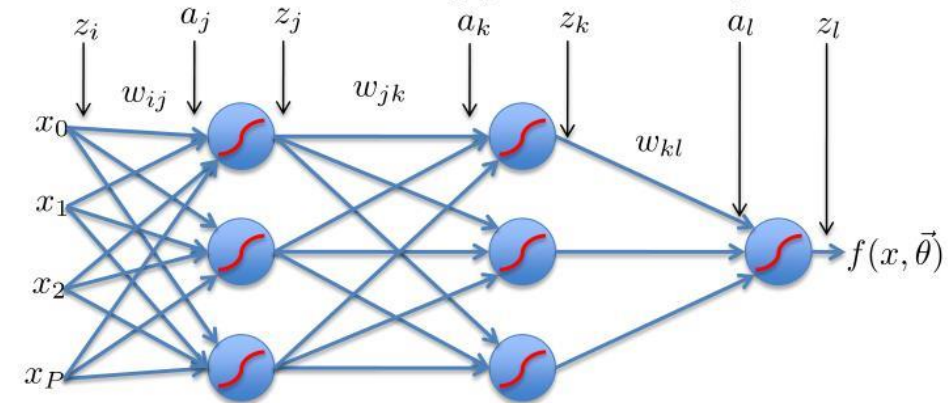
- Wymagany zbiór danych treningowych
- Rozpoczynamy z losowymi wagami
- Funkcja kosztu wykorzystywana do dopasowania wag (uczenie pod nadzorem)

## Możliwe uczenie bez nadzoru

## Error Backpropagation

Repeat for all previous layers

$$\begin{aligned}\frac{\partial R}{\partial w_{kl}} &= \frac{1}{N} \sum_n \left[ \frac{\partial L_n}{\partial a_{l,n}} \right] \left[ \frac{\partial a_{l,n}}{\partial w_{kl}} \right] = \frac{1}{N} \sum_n [-(y_n - z_{l,n}) g'(a_{l,n})] z_{k,n} = \frac{1}{N} \sum_n \delta_{l,n} z_{k,n} \\ \frac{\partial R}{\partial w_{jk}} &= \frac{1}{N} \sum_n \left[ \frac{\partial L_n}{\partial a_{k,n}} \right] \left[ \frac{\partial a_{k,n}}{\partial w_{jk}} \right] = \frac{1}{N} \sum_n \left[ \sum_l \delta_{l,n} w_{kl} g'(a_{k,n}) \right] z_{j,n} = \frac{1}{N} \sum_n \delta_{k,n} z_{j,n} \\ \frac{\partial R}{\partial w_{ij}} &= \frac{1}{N} \sum_n \left[ \frac{\partial L_n}{\partial a_{j,n}} \right] \left[ \frac{\partial a_{j,n}}{\partial w_{ij}} \right] = \frac{1}{N} \sum_n \left[ \sum_k \delta_{k,n} w_{jk} g'(a_{j,n}) \right] z_{i,n} = \frac{1}{N} \sum_n \delta_{j,n} z_{i,n}\end{aligned}$$



# Wady i zalety propagacji wstecznej

## **Zalety**

- Działa

- Działa całkiem szybko

## **Wady**

- Wymagany zbiór treningowy

- Może być powolne

- Odchodzimy od modelu biologicznego

## **Alternatywy**

- Uczenie hebbianowskie

  - Nie działa dla sieci ze sprzężeniem w przód

- Uczenie

  - Only limited success

- Algorytmy ewolucyjne

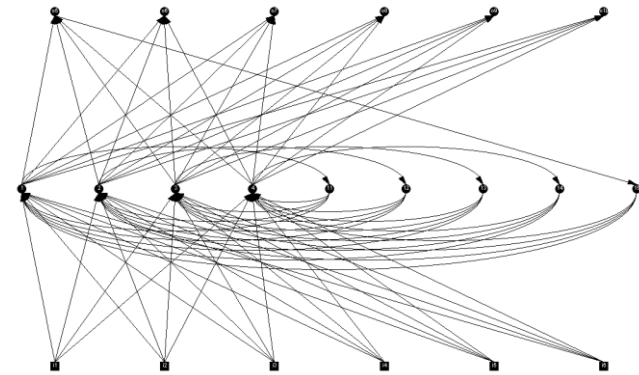
  - Zazwyczaj powolne

# Sieci z powrotami

- Sieci ze sprzężeniem w przód
  - Informacja płynie w jednym kierunku
  - Dany typ wejścia produkuje dany typ wyjścia
  - Brak możliwości zapamiętywania poprzednich stanów
- Powroty
  - Połączenia z poprzednimi węzłami lub pętle
  - Przepływ informacji w wiele stron
  - Możliwość stworzenia pseudopamięci

# Sieci Elmana i Hopppfielda

- sieć częściowo rekurencyjna o strukturze dwuwarstwowej
- sprzężenie zwrotne dotyczy tylko warstwy ukrytej
- neurony warstwy wyjściowej są połączone tylko z neuronami warstwy ukrytej (przepływ sygnałów jednokierunkowy)
- każdy neuron ukryty ma swego odpowiednika w tzw. warstwie kontekstowej



- wejścia to wyjścia innych neuronów
- brak własnego sprzężenia zwrotnego
- symetryczne wagi ( $W_{ij}=W_{ji}$ )
- dwa tryby pracy

