Wykład 2: Wprowadzenie do Sztucznych Sieci Neuronowych

S. Hoa Nguyen

Katedra: Systemy Inteligentne

hoa@pjwstk.edu.pl

http://www.users.pjwstk.edu.pl/~hoa/www/

Narzędzie Sztucznej Inteligencji

Plan wykładu

- Wprowadzenie w tematykę sieci neuronowych
- Neuron sztuczny odpowiednik neuronu biologicznego
- Architektury sieci
- Uczenie perceptronu

Historia

- Początki
  - 1943 – W.McCulloch, W.Pitts – pierwszy formalny model neuronu;
  - 1949 – Donald Hebb – „The organization of behaviour” – reguła uaktualniania wag połączeń neuronów.
- Pierwsze sukcesy
  - 1957-58 – F.Rosenblatt, Ch. Wightman – PERCEPTRON;
  - 1960 – B.Widrow, M.Hoff – ADALINE;

Historia

- Okres zastoju
  - 1969 – M.Minsky, S.Papert – publikacja „Perceptrons”
  - 1972-82 - T.Kohonen – pamięć skojarzeniowa
  - 1974,82 – S.Grossberg, G.Carpenter – teoria sieci rezonansowych.
- Ponowny rozkwit
  - 1983-86 – prace Johna Hopfielda;
  - 1986 - James McCleeland, David Rumelhard „Parallel Distributed Processing” – „odkrycie” metody uczenia perceptronów wielowarstwowych.

Funkcje sieci neuronowych i Zastosowania

- Rozpoznawanie obrazów
- Rozpoznawanie i synteza mowy

Rozpoznawanie wzorców

- Typowania w wyścigach konnych
- Dobór pracowników
- Selekcja celów śledztwa w kryminalistyce
- Diagnostyka chorób

Klasyfikacja

- Prognozowanie sprzedaży
- Prognozowanie giełdy

Predykcja

- Analiza sygnałów radarowych;
- Kompresja obrazów;

Asocjacja

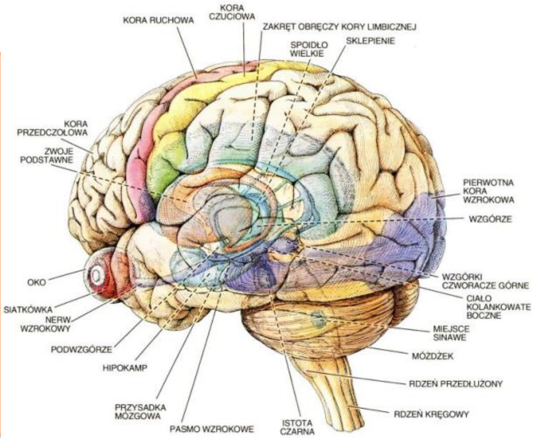
- Sterowanie robotami przemysłowymi
- Modelowanie funkcji nieliniowych

Aproksymacja oraz Sterowanie

Mózg – insperacja neurofizjologiczna

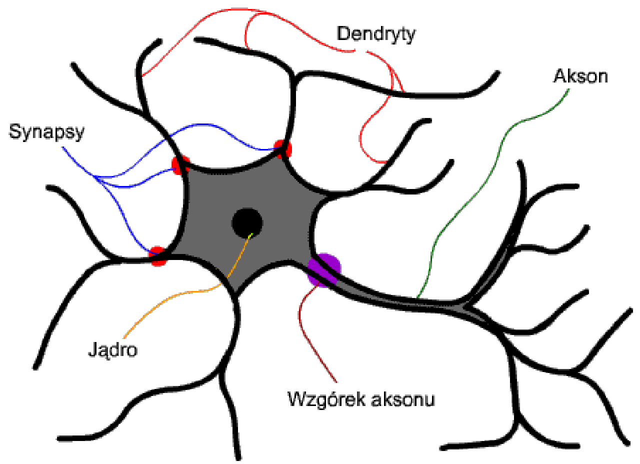
- Liczba połączeń synaptycznych w mózgu: $10^{10} – 10^{11}$;
- Gęstość połączeń synaptycznych: $\sim 10^4$/neuron
- Częstość generacji sygnałów przez neuron: $\sim 1 – 100$ Hz;
- Szybkość pracy: $\sim 10^{18}$ operacji/s

(najszybsze komputery: $\sim 10^{12}$ operacji/s)



- Odporny na uszkodzenia;
- Elastyczny – łatwo dostosowuje się do zmiennego otoczenia;
- Uczy się – nie musi być programowany
- Potrafi radzić sobie z informacją rozmytą, zaszumioną lub niespójną;
- W wysokim stopniu równoległy;

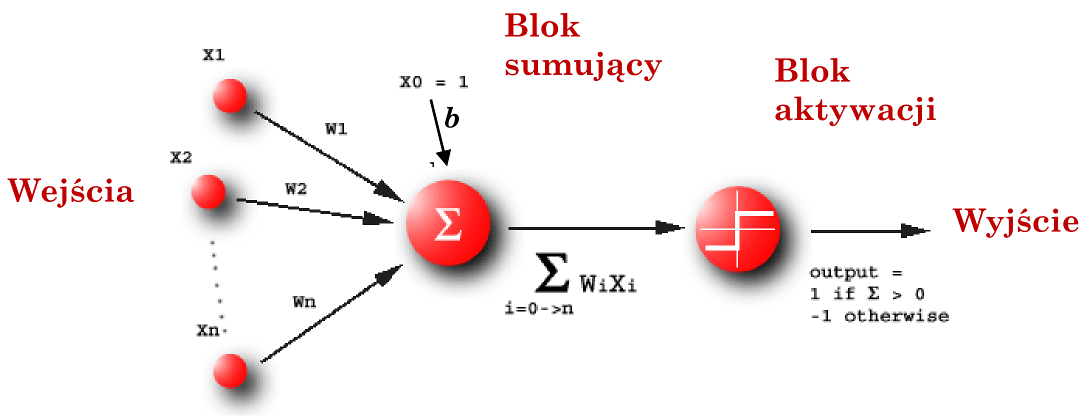
Neuron biologiczny i jego funkcjonowanie

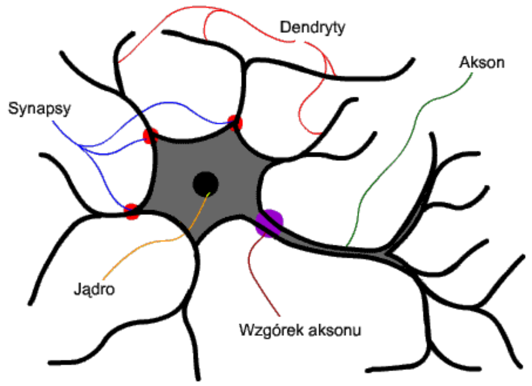


Jądro - "centrum obliczeniowe" neuronu.
Akson - "wyjście" neuronu.
Wzgórek aksonu - stąd wysyłany jest sygnał wyjściowy.
Dendryt – „wejście" neuronu.
Synapsa - . Może ona zmienić moc sygnału napływającego poprzez dendryt.

- **Potencjały** odebrane z innych komórek za pomocą dendrytów są zbierane na błonie ciała komórki.
- Gdy zebrane potencjały przekroczą **wartość progową** neuron staje się aktywny i wysyła sygnały elektryczne (elektrochemiczne) przez akson.
- Inne neurony odbierają sygnał zależnie od **przepustowości synaps.**

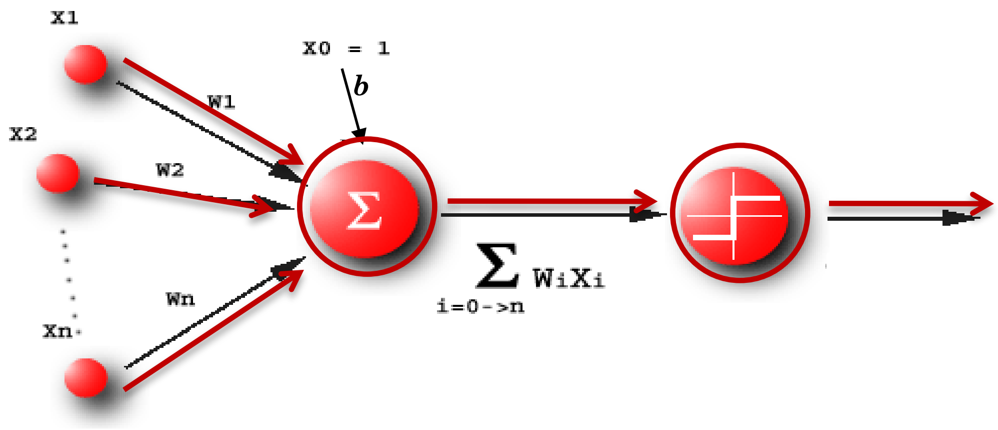
Model sztucznego neuronu (McCulloch'a i Pitts'a 1943)





- **Wejście**
  - n stanów wejściowych $x_1,\ldots,x_n$
  - stany mogą być cyfrowe lub analogowe
- **Wyjście**
  - 0 lub 1
- **Parametry perceptronu**
  - n wag połączeń $w_1,\ldots,w_n \in \Re$ (wagi synaptyczne)
  - wartość progową (bias) $b \in \Re$

Sztuczny neuron – Zasada działania

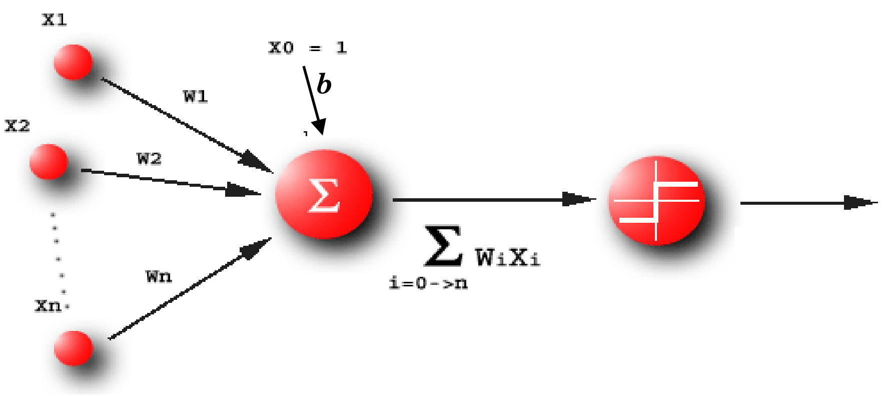


- Do każdego i-tego wejścia przypisana jest **waga w_i**
- Dla danych stanów wejściowych $x_1,\ldots,x_n$ liczymy **sumę ważoną**:

$$net = \sum_{i=1}^{n} w_i x_i + b$$

- Jeżeli $net \geq 0$, to ustawiamy wyjście $\boldsymbol{y = 1}$, zaś w przeciwnym przypadku ustawiamy $\boldsymbol{y = 0}$

Jak opisać perceptron?

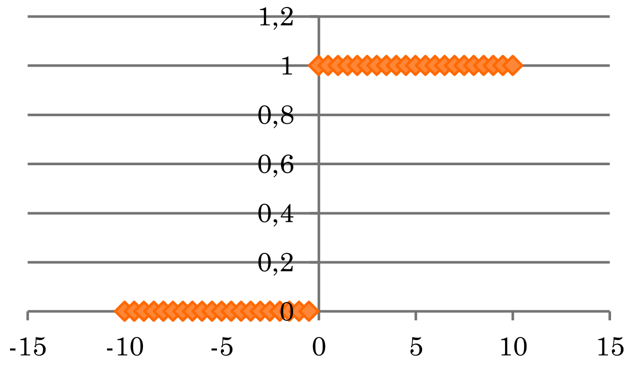


- **Wejścia**
- **Zbiór wag** $w_1,\ldots,w_n \in \Re$
- **Wartość progowa (bias)** $b \in \Re$
- **Funkcja aktywacji:** funkcji, według której obliczana jest wartość wyjścia neuronów sieci neuronowych

Funkcje aktywacji - dyskretne

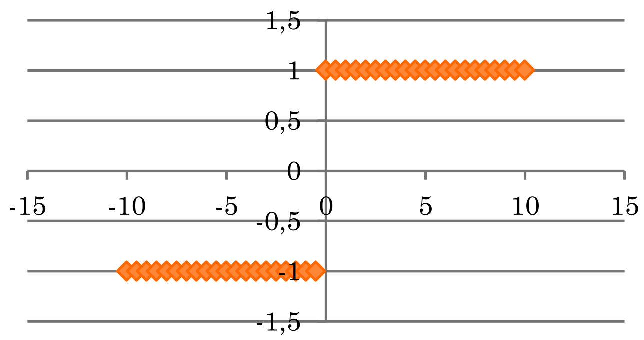
Progowa unipolarna

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



Progowa bipolarna

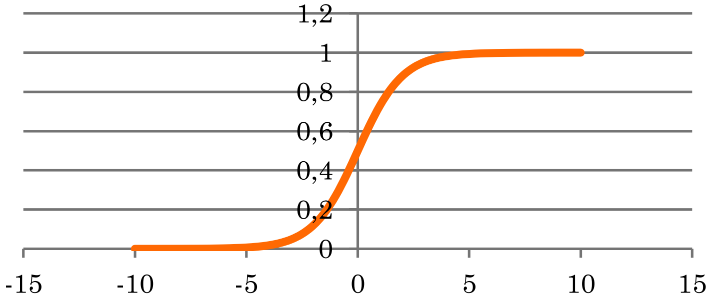
$$f(net) = \begin{cases} 1 \Leftrightarrow net \geq 0 \\ -1 \Leftrightarrow net < 0 \end{cases}$$



Funkcje aktywacji - ciągłe

Sigmoidalna unipolarna

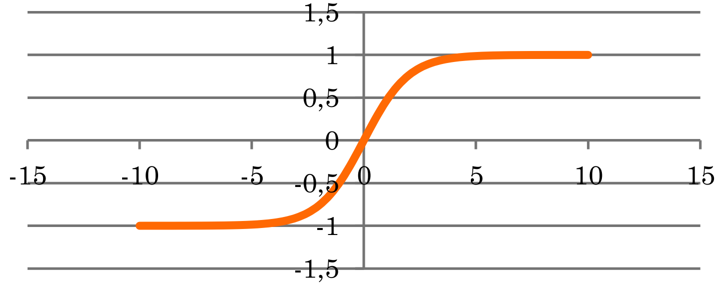
$$f(net) = \frac{1}{1+e^{-\lambda net}}$$



Sigmoidalna bipolarna

(tangens hiperpoliczny)

$$f(net) = \frac{2}{1+e^{-\lambda net}} - 1$$



Funkcja liniowa : $f(net) = a.net + b$

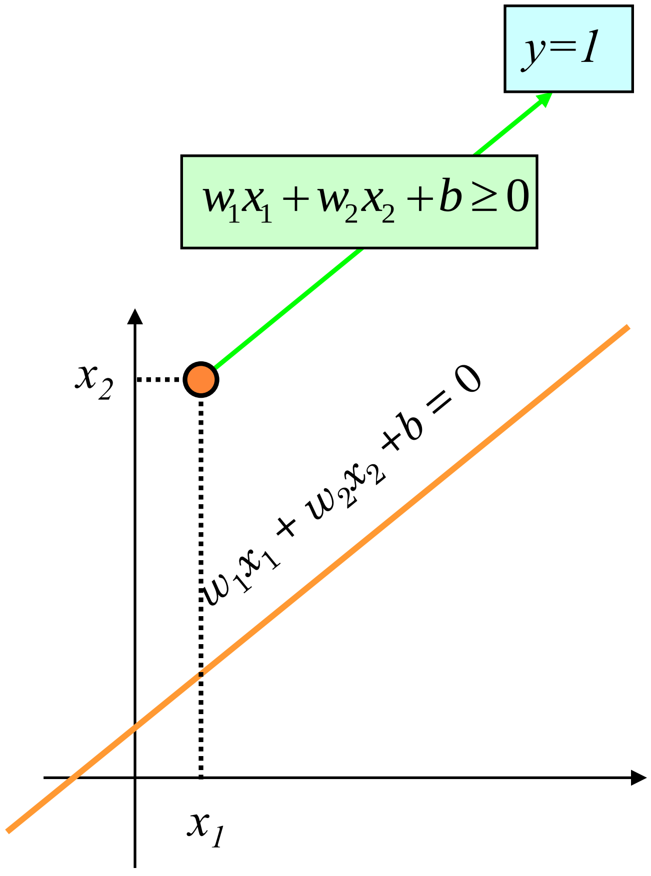
„Obcięta" funkcja liniowa: $f(net) = \begin{cases} -1 & net \le -1 \\ net & -1 < net < 1 \\ 1 & 1 \le net \end{cases}$

Intepretacja geometryczna (neuron dyskretny)

- **Równanie perceptronowe**:

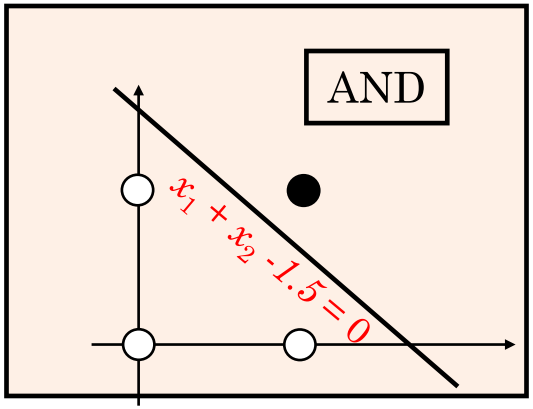
$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b = 0$$

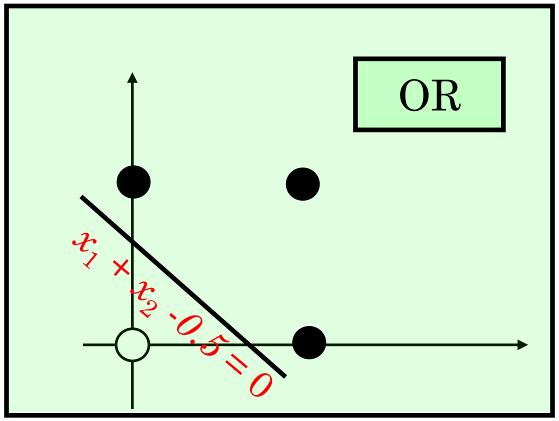
- Równanie perceptronu można potraktować jako równanie prostej (ogólnie: hiperpłaszczyzny w przestrzeni n-wymiarowej).
- Punkty leżące nad ową prostą klasyfikujemy jako **1**, zaś pozostałe jako **0**.
- Quize: Jak interpretować neuron ciągły (z sigmoidalną funkcją aktywacji)?

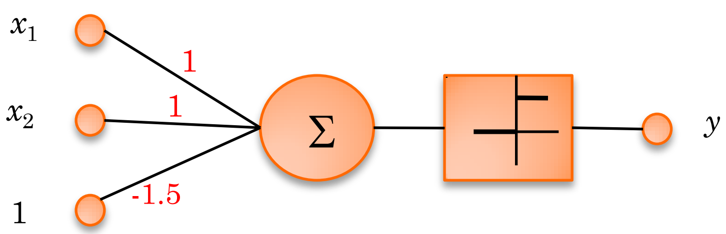


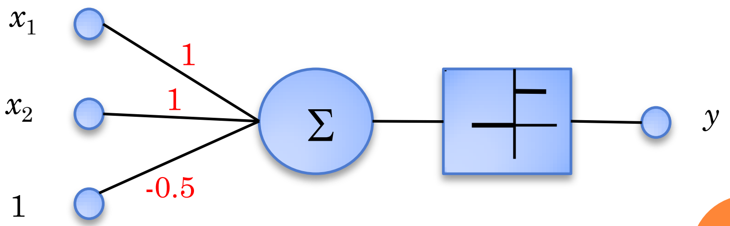
Co potrafi perceptron?

Pojedynczy perceptron potrafi odróżniać zbiorów **liniowo separowalnych**.





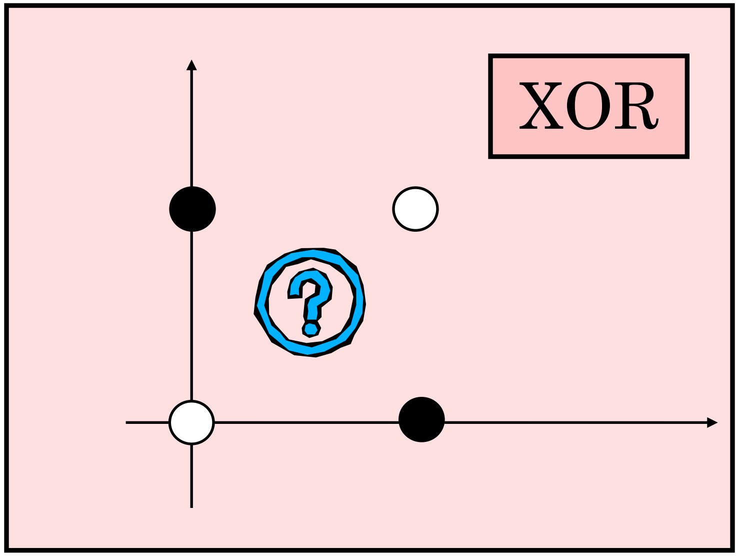




Czego nie potrafi perceptron?

- Pojedyńczy perceptron nie potrafi odróżniać zbiorów **nieseparowalnych liniowo**, np. funkcji XOR.

- Odkrycie tych ograniczeń (1969) na wiele lat zahamowało rozwój sieci neuronowych.

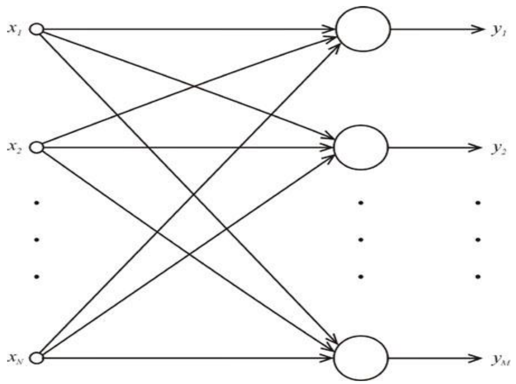


Zadanie perceptronu

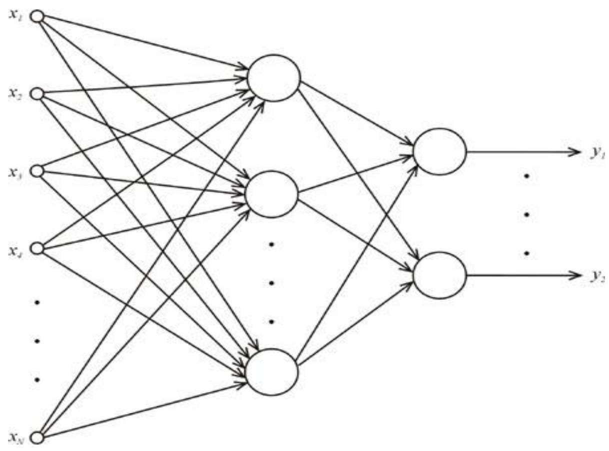
- Zadaniem pojedyńczego perceptronu jest jedynie:
  - przetwarzanie jednostkowych informacji
  - podejmowanie prostych decyzji
  - przekazywanie wyników sąsiadom
- Dopiero w połączeniu z innymi węzłami uzyskuje się zdolność podejmowania złożonych decyzji

Zadanie perceptronu Architektury sieci neuronowych

- Sieć jednokierunkowa **jednowartswowa**

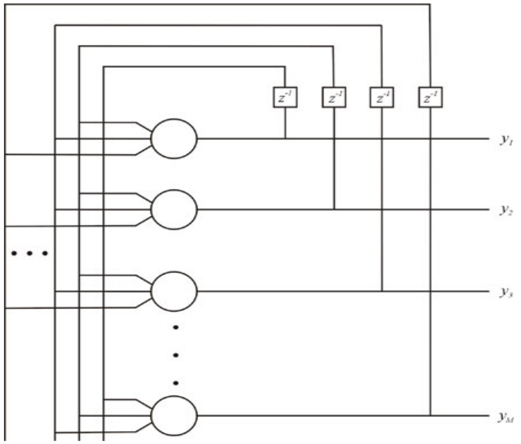


- Sieć jednokierunkowa **wielowarstwowa**
  - Uogólnienie perceptronu Rosenblatta
  - Funkcja aktywacji nieliniowa (sigmoidalna)



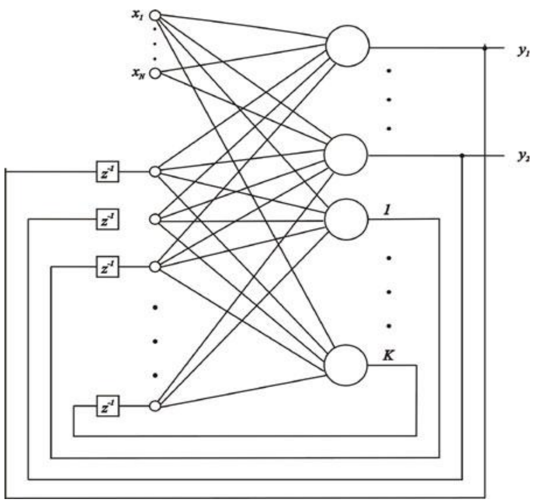
ARCHITEKTURY SIECI NEURONOWYCH

- Sieć **rekurencyjna jednowarstwowa**



- Sieć rekurencyjna **z jedną ukrytą wartswą**

  Sieć Hopfielda



Uczenie perceptronu

- Sposób działania perceptronu (wartości wag) w praktycznych problemach nie jest ustawiany ręcznie, tylko wyuczany na podstawie przykładów.
- Potrzebujemy zarówno metody uczenia jednego neuronu, jak i procedury obejmującej całą sieć.

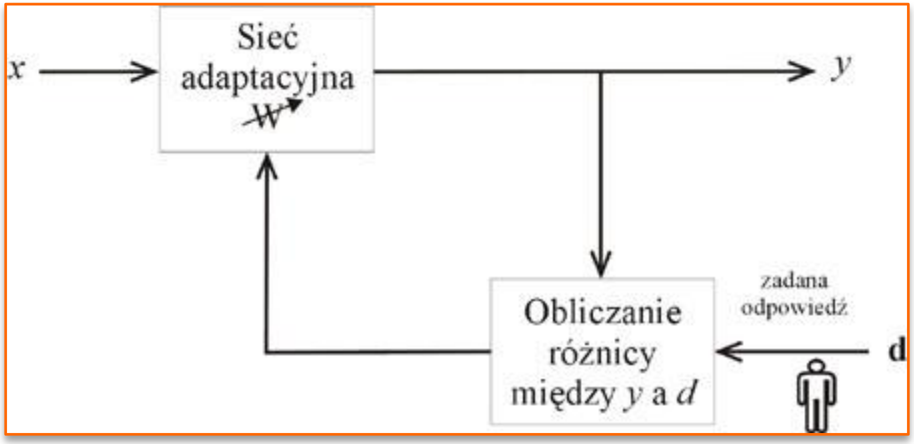
Grupy algorytmów uczących

- Nadzorowane (z nauczycielem)
- Bez nadzoru (bez nauczyciela)

Uczenie perceptronu z nauczycielem

- **Wejście:**
  - Ciąg przykładów uczących ze znanymi odpowiedziami: $\{(\boldsymbol{x}_1,d_1), (\boldsymbol{x}_2,d_2),\ldots(\boldsymbol{x}_n,d_n)\}$
- **Wyjście**: Wagi $\{w_1,w_2,\ldots w_n\}$
- **Celem**: Zminimalizować funkcję błędu:

$$\text{Error} = \Sigma_k (d_k-y_k)^2$$



Reguły adaptacji wag

- Przykład uczący:
  $\boldsymbol{x} = [x_1, x_2, \ldots, x_n, 1]$
- Żądana odpowiedź:
  $\boldsymbol{d} = [d_1, d_2, \ldots, d_m]$
- **Reguła perceptronowa**

(dla neuronów dyskretnych):

$$\Delta w_{ij} = \eta.(d_i - y_i).x_j$$
$$b_j = \eta.(d_i - y_i)$$

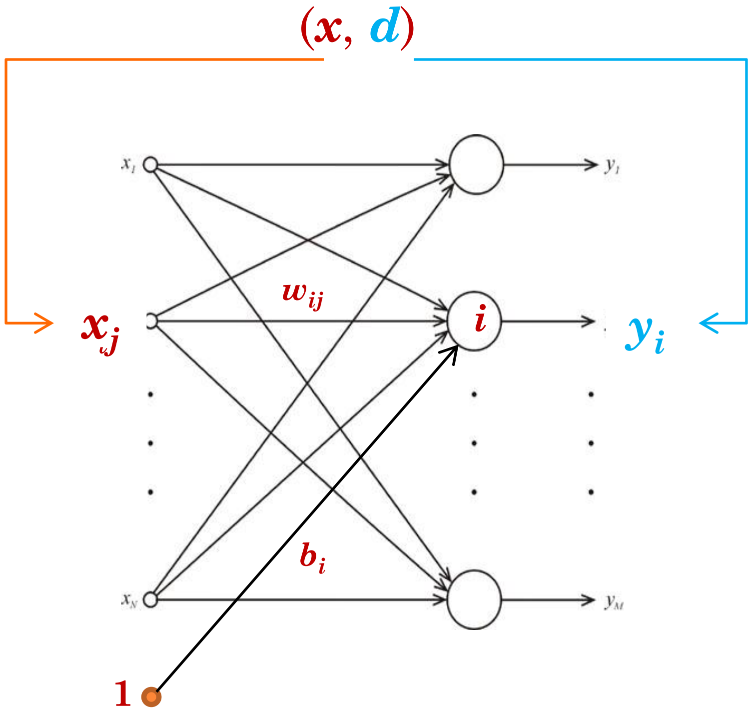
η : współczynnik uczenia

- **Reguła Delta**

(dla neuronów dyskretnych):

$$\Delta w_{ij} = \eta.(d_i - y_i).f'(net_i)\, x_j$$
$$b_j = \eta.(d_i - y_i)\, f'(net_i)$$

$f'(net_i)$: pochodna funkcji aktywacji i-tego neuronu



Algorytm uczenia perceptronu (reguła perceptronowa)

1. Inicjalizuj wagi sieci jako niewielkie liczby losowe;
2. Dla kolejnego przykładu uczącego ***x***
   - oblicz wektor sygnałów wyjściowych ***y***
   - wartość błędu

$$\delta_k = d_k - y_k$$

3. Oblicz zmianę wag Δ**w:**

$$\Delta \mathbf{w}_{\boldsymbol{k}}(t) = \eta \delta_k \boldsymbol{x}$$

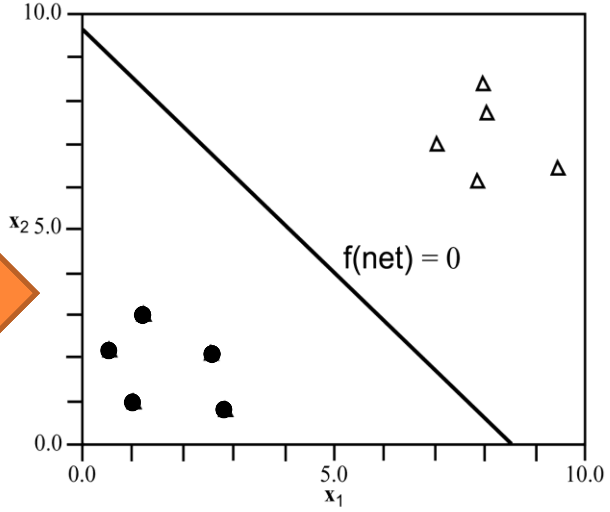
4. Uaktualnij wektor wag **w**(t+1):

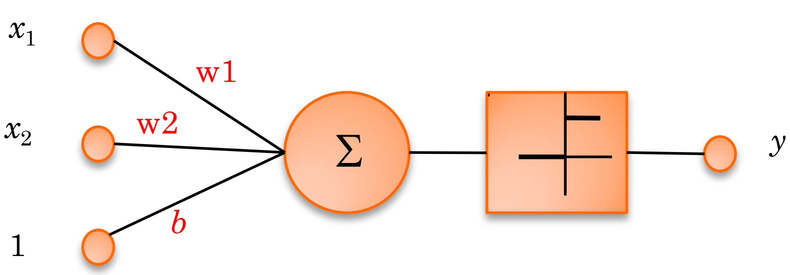
$$\mathbf{w}_{\boldsymbol{k}}(t+1) = \mathbf{w}(t) + \Delta \mathbf{w}(t);$$

5. Powtarzaj kroki 1-4 dopóki błąd nie osiągnie akceptowalnej wartości.

Algorytm uczenia perceptron - Przykład

x_1	x_2	Output
1.0	1.0	1
9.4	6.4	−1
2.5	2.1	1
8.0	7.7	−1
0.5	2.2	1
7.9	8.4	−1
7.0	7.0	−1
2.8	0.8	1
1.2	3.0	1
7.8	6.1	−1

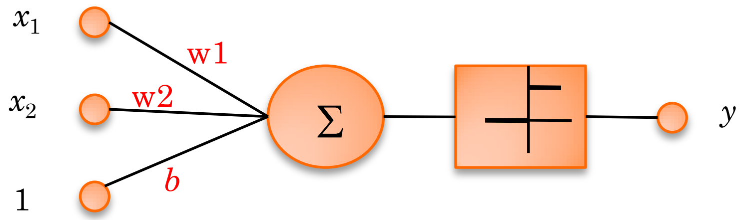




Funkcja aktywacji:

$$y = \begin{cases} 1 \leftrightarrow & \sum w_i x_i + b \geq 0 \\ -1 & \textit{otherwise} \end{cases}$$

UCZENIA PERCEPTRON -PRZYKŁAD



- Niech początkowe wagi: $\mathbf{w_1=1, w_2=1}, \boldsymbol{b} = \mathbf{1}$
- $net_1 = x_1w_1+x_2w_2 + b = 1+1+1=3 \Rightarrow$
  $y_1=f(net) = 1$
- Pierwszy przykład jest dobry, ale drugi nie jest, bo
  $net_2 = x_1w_1+x_2w_2 + b = 9.4+6.4+1=16.8 \Rightarrow$
  $y_2=f(net_2) = 1$
- Uaktualizuj wagi: ($\eta = 1$)
  $\boldsymbol{w_1 := w_1 + \eta(-1-1)\ 9.4 = 1-18.8 = -17.8}$
  $\boldsymbol{w_2 := w_2 + \eta(-1-1)\ 6.4 = 1-12.8 = -11.8}$
  $\boldsymbol{b := b + \eta(-1-1) = 1-2 = -1}$
- Zatem
  $\mathbf{w_1 = -17.8.}$
  $\mathbf{w_2 = -11.8}$
  $\boldsymbol{b = -1}$
- Drugi przykład jest dobry, sprawdź dalej trzeci przykład…

x_1	x_2	Output
1.0	1.0	1
9.4	6.4	−1
2.5	2.1	1
8.0	7.7	−1
0.5	2.2	1
7.9	8.4	−1
7.0	7.0	−1
2.8	0.8	1
1.2	3.0	1
7.8	6.1	−1

Pytania sprawdzające:

1. Czym się charakteryzuje prosty perceptron?
2. Gdzie jest realizowana funkcja aktywacji i jakie są typy funkcji aktywacji?
3. Reguła perceptronowa (delta) do uczenia dyskretnych (ciągłych) perceptronów