METODY I ZASTOSOWANIA SZTUCZNEJ INTELIGENCJI

LABORATORIUM nr 01

Temat: **PERCEPTRON**

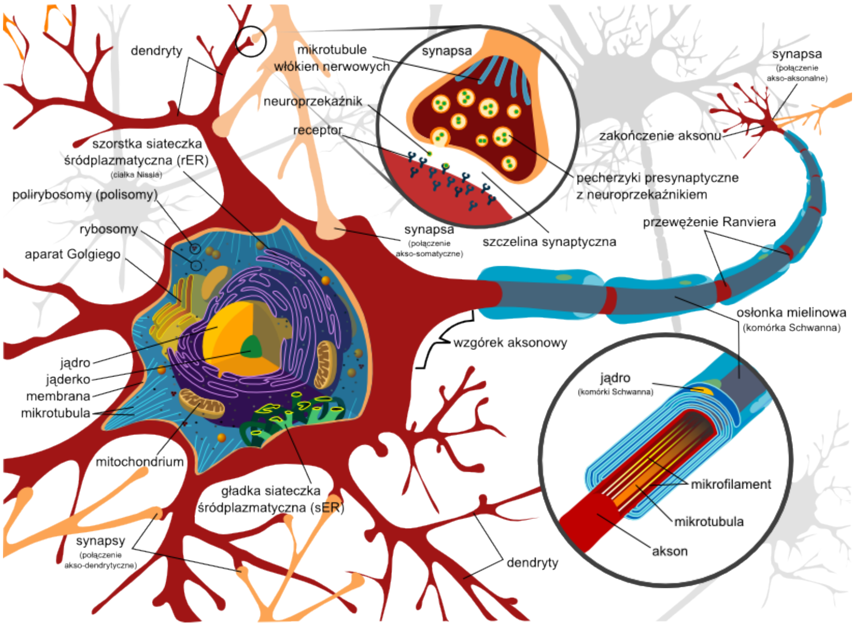
dr inż. Robert Tomkowski

pok. 118 bud. C
robert.tomkowski@tu.koszalin.pl
tel. 94 3178 251

SPIS TREŚCI

1. Wprowadzenie

Za podstawowy element systemu nerwowego uważany jest neuron. Neuron jest złożoną komórką, której ciało (*soma*) jest otoczona przez dwa rodzaje wypustek: *dendryty* – wprowadzające informacje do neuronu (są silnie rozgałęzione i mogą stanowić nawet 90 % powierzchni neuronu) oraz *aksony* – wyprowadzające informacje z neuronu (neurony posiadają zazwyczaj jeden akson). Jeden neuron przekazuje informację w postaci pobudzając inne neurony poprzez złącza nerwowe zwane *synapsami* (są to miejsca komunikacji neuronu z innym neuronem lub komórką efektorową, czyli mięśniową lub gruczołową). Pobudzenie neuronu poprzez synapsy może mieć charakter wzmocnienia lub osłabienia. Neuron sumuje wszystkie sygnały, zarówno te pobudzające jak i hamujące i jeżeli ich suma algebraiczna przekroczy pewną wartość progową to sygnał wychodzący z neuronu zostaje przesłany poprzez akson do innego neuronu. Na rysunku 1 przedstawiono złożoną budowę neuronu.



Rys. 1. Budowa szczegółowa neuronu, połączenia i synapsa (źródło: http://pl.wikipedia.org/wiki/Neuron)

2. Model neuronu

Za początek sztucznej inteligencji uważa się określenie modelu pojedynczego neuronu, czyli jego matematycznego opisu. Działanie neuronu zilustrowane na rysunku 2, może zostać opisane wzorem 1.

$$y = f(s), \text{ gdzie } s = b + \sum_{i=1}^{n} w_i x_i \tag{1}$$

dla

$$f(s) = \begin{cases} 1, & gdy\ s > 0 \\ 0, & gdy\ s \leq 0 \end{cases}$$

$$w_0 = \vartheta, x_0 = -1$$

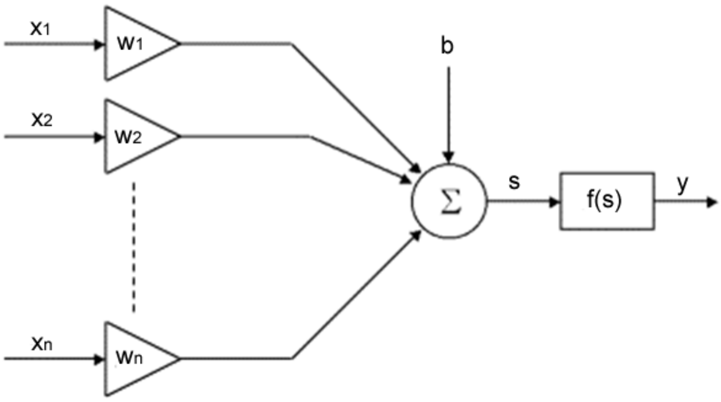
gdzie:

n	– liczba wejść w neuronie
$x_i, \cdots, x_n$	– sygnały wejściowe
$w_i, \cdots, w_n$	– wagi synaptyczne
y	– sygnał wyjściowy (wartość wyjściowa neuronu)
b	– wartość progowa
f	– funkcja aktywacji

Model ten można również zapisać w postaci (2).

$$y = \begin{cases} 1, & gdy\ \sum_{i=1}^{n} w_i x_i > \vartheta \\ 0, & gdy\ \sum_{i=1}^{n} w_i x_i \le \vartheta \end{cases} \tag{2}$$

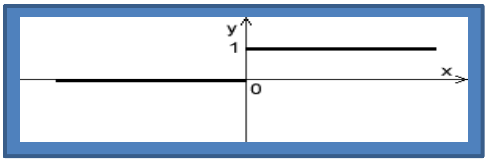
Poniższy rysunek (rys. 2) przedstawia model neuronu.



Rys. 2. Model neuronu

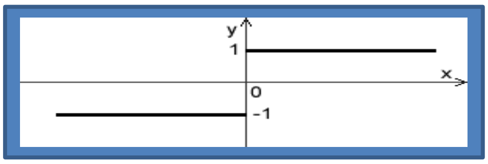
Można wyróżnić kilka funkcji aktywacji neuronu, należą do nich:

Progowa unipolarna



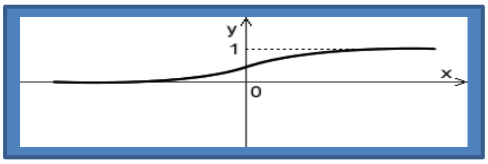
$$f(s) = \begin{cases} 1, & gdy\ s > 0 \\ 0, & gdy\ s \le 0 \end{cases}$$

Progowa bipolarna



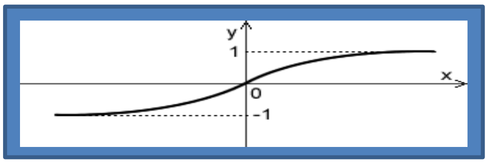
$$f(s) = \begin{cases} 1, & gdy\ s > 0 \\ -1, & gdy\ s \le 0 \end{cases}$$

Sigmoidalna unipolarna



$$f(s) = \frac{1}{1 + e^{-\beta s}} \quad dla\ \beta > 0$$

Tangens hiperboliczny



$$f(s) = tgh\left(\frac{\alpha s}{2}\right) = \frac{1 - e^{-\alpha s}}{1 + e^{-\alpha s}}$$

dla $\alpha > 0$

3. Perceptron

Model neuronu został opisany przez McCullocha i Pittsa w 1943 roku i był punktem wyjścia do tworzenia sieci neuronowych. Na przełomie lat 50 i 60 dwudziestego wieku Rosenblatt, korzystając z wcześniej opisanego modelu neuronu zaproponował najprostszą sieć jednokierunkową zwaną *perceptronem*. Jako funkcję aktywacji w perceptronie przyjęto funkcję bipolarną.

Podstawowym zadaniem perceptronu jest klasyfikacja wektora **X** do jednej z dwóch klas oznaczonych jako $\mathbf{L_1}$ i $\mathbf{L_2}$. Perceptron dokonuje klasyfikacji wektora **X** do klasy $\mathbf{L_1}$, jeżeli sygnał wyjściowy $y = 1$ oraz do klasy $\mathbf{L_2}$ jeżeli $y = -1$.

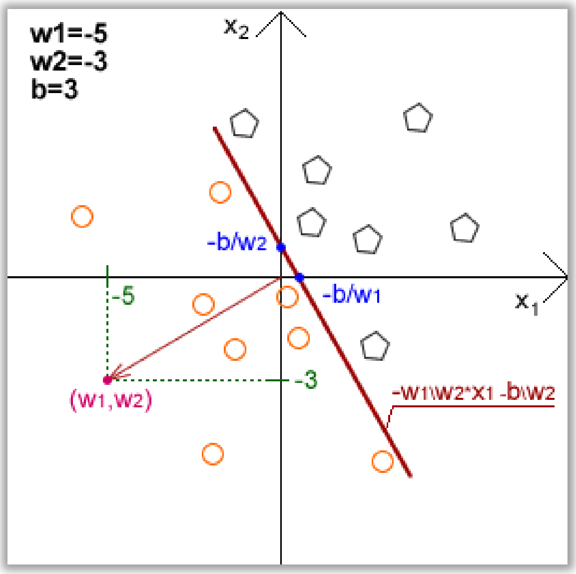
Przykład

W przypadku dwóch wejść x1 i x2 perceptron dzieli płaszczyznę na dwie części.

Podział ten wyznacza prosta o równaniu:

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

$$x_2 = -\frac{w_1}{w_2} \cdot x_1 - \frac{b}{w_2}$$



3.1. Uczenie perceptronu

Załóżmy , że nie są znane **wagi wi**, i=1,2,... oraz **biast b.**

Co możemy zrobić?

- Nieznane wagi wyznaczymy w procesie uczenia. Jest to tzw. uczenie „***nadzorowane***”, lub inaczej „***z nauczycielem***”.
- Uczenie tego typu polega na podaniu na wejście perceptronu sygnałów **x(t)**=[x1(t), x2(t), ...], t=1,2,..., dla których znamy prawidłowe wartości sygnałów wyj. **ywzor(t)**, t=1,2,..., zwanych **sygnałami wzorcowymi**.
- Zbiór próbek wejściowych wraz z odpowiadającymi im wartościami sygnałów wzorcowych nazywamy **ciągiem uczącym.**

3.2. Algorytm

Klasyfikacja z wykorzystaniem perceptronu przebiega w następujący sposób:

1. wybieramy w sposób losowy wagi początkowe **w** i bias **b**,
2. na wejścia neuronu podajemy wektor uczący **X**,
3. obliczamy wartość wyjściową perceptronu **y** zgodnie ze wzorem:

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

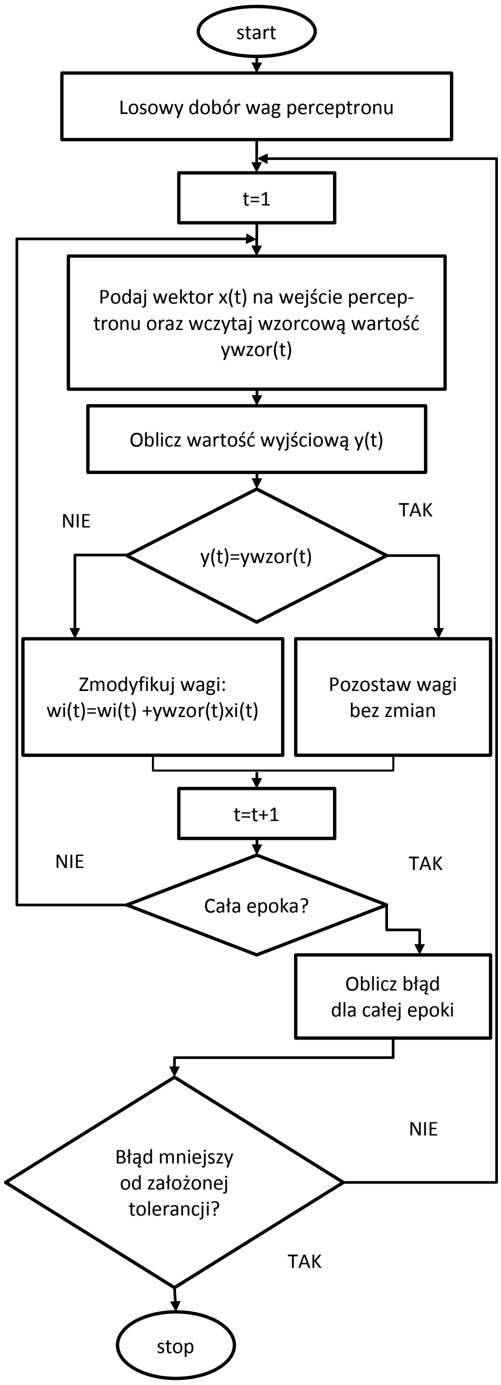
4. porównujemy wartość wyjściową **y** z wartością wzorcową $\mathbf{y_{wzor}}$
5. dokonujemy modyfikacji wag według zależności:

$$y(X) \neq y_{wzor}(X), \;\; to \;\; w_i = w_i + y_{wzor}(X) \cdot x_i$$

$$y(X) = y_{wzor}(X), \;\; to \;\; w_i = w_i \text{ wagi pozostają bez zmian}$$

6. wracamy do punktu 2.

Algorytmy powtarza się tak długo, aż błąd na wyjściu będzie mniejszy od założonej tolerancji.



Przykład uczenia perceptronu

			WAGI
w1	2	-1	
w2	3	1	
b	3	2	

	t=1	t=2	t=3	t=4	CIĄG UCZĄCY
x1	1	3	3	4	
x2	1	2	5	1	
ywzor	1	-1	1	-1	

Epoka I

t=1: s=b+x1(t)·w1+x2(t)·w2 = 3 + 1 ·2 + 1 ·3 = 8 → (8 > 0) → **f(s) = y = 1**

y = ywzor(t) Pozostaw wagi bez zmian!

t=2: s=b+x1(t)·w1+x2(t)·w2 = 3 + 3 ·2 + 2 ·3 = 15 → (15 > 0) → **f(s) = y = 1**

y ≠ ywzor(t) Zmodyfikuj wagi!

w1=w1+ ywzor(t) · x1(t)=2+(-1)*3=-1 → **w1=-1**

w2=w2+ ywzor(t) · x2(t)=3+(-1)*2=1 → **w2=1**

b=b+ ywzor(t) =3+(-1)=2 → **b=2**

t=3: s=b+x1(t)·w1+x2(t)·w2 = 2 + 3 ·(-1) + 5 ·1 = 4 → (4> 0) → **f(s) = y = 1**

y = ywzor(t) Pozostaw wagi bez zmian!

t=4: s=b+x1(t)·w1+x2(t)·w2 = 2 + 4 ·(-1) + 1 ·1 = -1 → (-1 < 0) → **f(s) = y = -1**

y = ywzor(t) Pozostaw wagi bez zmian!

WARTOŚĆ BŁĘDU: s(t) =b+ x1(t) ·w1 + x2(t) ·w2;

$$y(t) = \begin{cases} 1 & \text{gdy } s(t) > 0 \\ -1 & \text{gdy } s(t) \le 0 \end{cases}$$

$$\varepsilon = \sum_{i=1}^{t} \left| y_{wzor\,(i)} - y_{(i)} \right|$$

t=1: s(1) = 2+1·(-1)+1·1=2 ; **t=2:** s(2)=1; **t=3:** s(3)=4 ; **t=4:** s(3)=-1 → y = [1 1 1 -1]

ε = |1-1| + |-1-1| + |1-1| +| (-1)- (-1)| = 0+2+0+0 = 2 → ε > 0 kolejna epoka!

						WAGI
w1	2	-1	-4	-1	5	
w2	3	1	-1	4	3	
b	3	2	1	2	1	

	t=1	t=2	t=3	t=4	CIĄG UCZĄCY
x1	1	3	3	4	
x2	1	2	5	1	
ywzor	1	-1	1	-1	

Epoka II

t=1: s=b+x1(t)·w1+x2(t)·w2 = 2 + 1 ·(-1) + 1 ·1 = 2 → (2 > 0) → **f(s) = y = 1**

y = ywzor(t) Pozostaw wagi bez zmian!

t=2: s=b+x1(t)·w1+x2(t)·w2 = 2 + 3 · (-1) + 2 ·1 = 1 → (1 > 0) → **f(s) = y = 1**

y ≠ ywzor(t) Zmodyfikuj wagi!

w1=w1+ ywzor(t) · x1(t)=(-1)+(-1)*3=**-4** → **w1=-4**

w2=w2+ ywzor(t) · x2(t)=1+(-1)*2=-1 → **w2=-1**

b=b+ ywzor(t) =2+(-1)=1 → **b=1**

t=3: s=b+x1(t)·w1+x2(t)·w2 = 1 + 3 ·(-4) + 5 ·(-1) = -6 → (-6 < 0) → **f(s) = y = -1**

y ≠ ywzor(t) Zmodyfikuj wagi!

w1=w1+ ywzor(t) · x1(t)=(-4)+1*3=**-1** → **w1=-1**

w2=w2+ ywzor(t) · x2(t)=(-1)+1*5=4 → **w2=4**

b=b+ ywzor(t) =1+1=2 → **b=2**

t=4: s=b+x1(t)·w1+x2(t)·w2 = 2 + 4 ·(-1) + 1 ·4 = 2 → (2 > 0) → **f(s) = y = 1**

y ≠ ywzor(t) Zmodyfikuj wagi!

w1=w1+ ywzor(t) · x1(t)=(-1)+(-1)*4=**-5** → **w1=-5**

w2=w2+ ywzor(t) · x2(t)=4+(-1)*1=3 → **w2=3**

b=b+ ywzor(t) =2+(-1)=1 → **b=1**

WARTOŚĆ BŁĘDU: ε > 0 kolejna epoka!

Epoka III

…

itd.

3.3. Przykład w Matlabie

Dwa klastry danych należące do dwóch klas, zdefiniowano w przestrzeni dwuwymiarowej. Klasy są separowalne liniowo. Dokonać klasyfikacji danych z zastosowaniem perceptronu.

Rozwiązanie:

```
close all, clear all, clc, format compact

% liczba próbek dla każdej z klas
n = 30;

% definiowanie wejść i wyjść
offset = 5; % przesuniecie w drugiej klasie
x = [randn(2,n) randn(2,n)+offset]; % wejścia
y = [zeros(1,n) ones(1,n)];         % wyjścia

% narysuj wejścia sieci na wykresie dwuwymiarowym korzystając
% z funkcji PLOTPV (wektory wejść i wyjść perceptronu)
figure(1)
plotpv(x,y);


% utwórz sieć jednokierunkową
net = perceptron; % utworzenie perceptronu z domyślnymi ustawieniami
net = train(net, x, y); % uczenie sieci z nauczycielem
view(net) % widok struktury perceptronu


% narysuj odpowiedź sieci
figure(1)
plotpc(net.IW{1}, net.b{1});
```

3.4. Zadanie domowe

Napisać funkcję implementującą działanie perceptronu. Funkcja za argument powinna przyjmować wektor dwóch danych wejściowych, wektor wag, wartość progową i wagę zerową w_0,. Wygenerować dane w przestrzeni $\mathbb{R}^2$ przynajmniej dwoma metodami. Wyświetlić wygenerowane dane. Narysować prostą separującą.

Przykład wprowadzania danych:

```
x = [1 2 3 4 5];
w = [1,2,3,4,5];
b = 5;
w0= 1;
odp = my_perc(x,w,b,w0)
```

Wszystkie etapy przedstawić w formie sprawozdania (kod wykresy i opisy poszczególnych etapów). Wykresy można zapisywać w Matlabie poprzez funkcję *save* lub *export setup* w menu *File* okna wykresu.