

# Sztuczna inteligencja

Neuron nieliniowy i sieć neuronowa

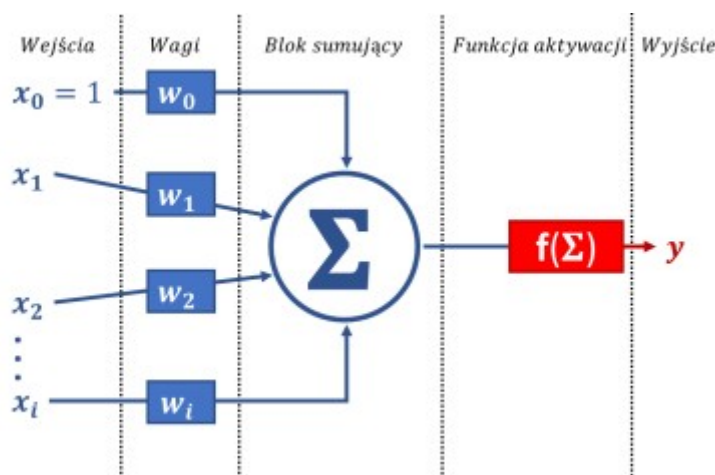
Marcin Wojtas  
Nr. albumu: s153915  
III rok Niestacjonarnie  
Grupa: L3

# 1. Cel

Celem było stworzenie programów opartych o sieci neuronowe w tym neuron nieliniowy w oparciu o wzory na funkcję progową unipolarną i funkcję sigmoidalną oraz prosty program oparty o sieć neuronową mający na celu po podaniu danych przez użytkownika jakie gatunek ma na myśli (ryba, ptak, ssak).

## 2. Wykorzystane pojęcia oraz wzory

Model sztucznego neuronu



Wzór na blok sumujący

$$S = x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_nw_n$$

Wzór na funkcję sigmoidalną

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

Gdzie  $\lambda n$  w naszym przypadku wynosi 1

Wzór na funkcję progową unipolarną

$$f(n) = \begin{cases} 1 & \text{jeśli } n \geq 0 \\ 0 & \text{wpp.} \end{cases}$$

### 3. Kod źródłowy programów

Zadanie 1. Neuron nieliniowy

```
1  import random
2  import math
3
4  w1 = random.random() * (1 - (-1)) + (-1)
5  w2 = random.random() * (1 - (-1)) + (-1)
6  w3 = random.random() * (1 - (-1)) + (-1)
7
8  def progowa_unipolarna(x1, x2, x3):
9      sw = x1*w1 + x2*w2 + x3*w3
10     if sw > 0:
11         return 1
12     else:
13         return 0
14
15     def sigmaidalna(x1,x2,x3):
16         sw = x1 * w1 + x2 * w2 + x3 * w3
17         return 1/(1 + math.exp(-sw))
18
19     print(progowa_unipolarna(4, 5, 2))
20     print(sigmaidalna(4, 5, 2))
21     |
```

Wykorzystane biblioteki: „math”, „random”.

1. w1, w2, w3 korzystając z funkcji random losują dla siebie różną liczbę z przedziału (-1, 1)
2. funkcja progowa unipolarna pobiera 3 podane wartości x1, x2, x3
  - 2.1 sw oblicza blok sumujący za pomocą wzoru na blok sumujący
  - 2.2 jeśli sw >= 0 funkcja zwraca wartość 1

- 2.3 jeśli wartość  $sw < 0$  funkcja zwraca wartość 0
- 3. funkcja sigmoidalną pobiera 3 podane wartości  $x_1, x_2, x_3$ 
  - 3.1  $sw$  oblicza blok sumujący za pomocą wzoru na blok sumujący
  - 3.2 funkcja zwraca wartość liczbową obliczoną za pomocą wzoru na funkcję sigmoidalną

## Zadanie 2. Sieć neuronową

```
1  def animal(x1, x2, x3, x4, x5):
2      ssak = [4, 0.01, 0.01, -1, -1.5]
3      ptak = [2, -1, 2, 2.5, 2]
4      ryba = [-1, 3.5, 0.01, -2, 1.5]
5
6      sSsak = ssak[0]*x1 + ssak[1]*x2 + ssak[2]*x3 + ssak[3]*x4 + ssak[4]*x5
7      sPtak = ptak[0]*x1 + ptak[1]*x2 + ptak[2]*x3 + ptak[3]*x4 + ptak[4]*x5
8      sRyba = ryba[0]*x1 + ryba[1]*x2 + ryba[2]*x3 + ryba[3]*x4 + ryba[4]*x5
9
10     if(sSsak > sRyba and sSsak > sPtak):
11         return "Ssak"
12     elif(sRyba > sPtak):
13         return "Ryba"
14     else:
15         return "Ptak"
16
17
18     d1 = float(input("Ile ma nóg? Liczba: "))
19     d2 = float(input("Czy żyje w wodzie? Liczba: "))
20     d3 = float(input("Czy umie latać? Liczba: "))
21     d4 = float(input("Czy jest pokryte piórami? Liczba: "))
22     d5 = float(input("Czy rodzi się z jaj? Liczba: "))
23
24     print(animal(d1, d2, d3, d4, d5))
25
```

- 1. użytkownik podaje dane liczbowe dla:  $d_1, d_2, d_3, d_4, d_5$
- 2. funkcja animal pobiera dane podane przez użytkownika
  - 2.1 tabele ssak, ptak i ryba przechowują odpowiednie wartości wagowe dla określonej gromady
  - 2.2  $sSsak, sPtak, sRyba$  oblicza dane dla własnej zmiennej za pomocą wzoru na blok sumujący
  - 2.3 funkcja sprawdza która z określonych zmiennych jest największa i zwraca odpowiedni typ gromady

## 4. Podsumowanie

Programy wykorzystujące sieć neuronową można zastosować do określenia prawdopodobieństwa wystąpienia danego elementu przez wcześniejsze dostarczenie odpowiednich wag.

Funkcje aktywacji mogą być różne w zależności od potrzebnego zastosowania, mogą też zwracać określone wartości.

## Bibliografia

<https://batmaja.com/sztuczny-neuron/> - wykorzystana grafika modelu neuronu