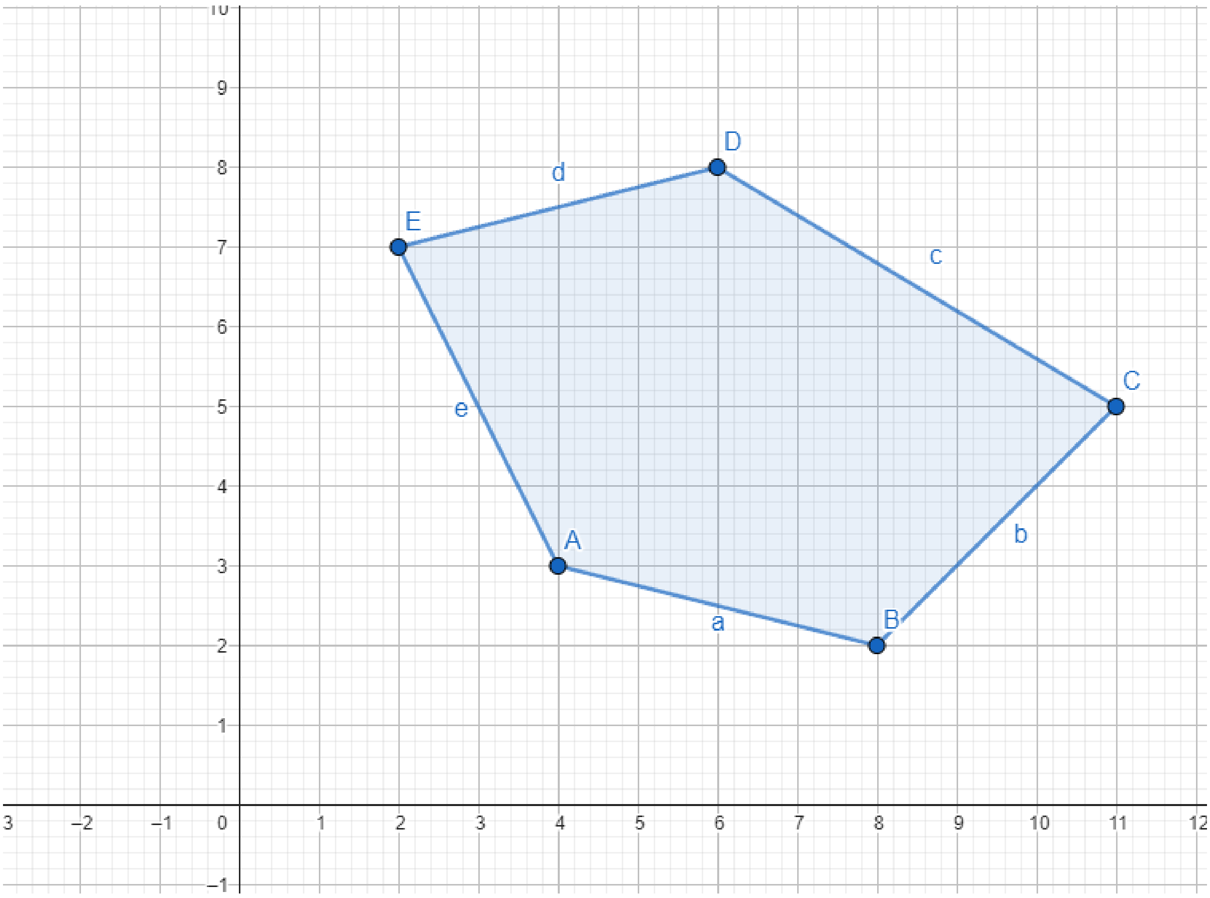
4. Sieci neuronowe

Wstęp

Skonstruuję sieć neuronową która sprawdzać będzie czy dany punkt (x, y) znajduje się wewnątrz pięciokąta.

Pięciokąt



Punkty:
A(4, 3)
B(8, 2)
C(11, 5)
D(6, 8)
E(2, 7)

Wyznaczenie prostych

W celu wyznaczenia warunków które punkt musi spełniać aby znajdować się w tym pięciokącie potrzebujemy prostych, które zawierają jego boki - a, b, c, d i e - na wykresie. Każda prosta zawiera w sobie dwa punkty więc w łatwy sposób można policzyć współczynniki tych prostych. **Obliczenia:**

a) Punkty $A(4,3)$ $B(8,2)$ $\qquad \boxed{y = ax+b}$

$$-\begin{cases} 3 = 4a+b \\ 2 = 8a+b \end{cases}$$

$1 = -4a$

$a = -1/4$

$b = 4$

$y = -\frac{1}{4}x + 4$

$\boxed{a:\ x+4y-16 = 0}$

b) Punkty $B(8,2)$ $C(11,5)$

$$-\begin{cases} 2 = 8a+b \\ 5 = 11a+b \end{cases}$$

$-3 = -3a$

$a = 1 \quad b = -6$

$y = x-6$

$\boxed{b:\ x-y-6 = 0}$

c) Punkty $C(11,5)$ $D(6,8)$

$$-\begin{cases} 5 = 11a+b \\ 8 = 6a+b \end{cases}$$

$-3 = 5a$

$a = \frac{-3}{5} \quad b = \frac{58}{3}$

$y = \frac{-3}{5}x + \frac{58}{5}$

$\boxed{c:\ 3x+5y-58 = 0}$

d) Punkty $D(6,8)$ $E(2,7)$

$$-\begin{cases} 8 = 6a+b \\ 7 = 2a+b \end{cases}$$

$1 = 4a$

$a = 1/4 \quad b = \frac{13}{2}$

$y = \frac{1}{4}x + \frac{13}{2}$

$\boxed{d:\ x-4y+26 = 0}$

e) Punkty $E(2,7)$ $A(4,3)$

$$-\begin{cases} 7 = 2a+b \\ 3 = 4a+b \end{cases}$$

$4 = -2a$

$a = -2 \quad b = 11$

$y = -2x+11$

$\boxed{e:\ 2x+y-11 = 0}$

Przykładowy punkt

Teraz na podstawie przykładowegu punktu F(6, 6), który znajduje się wewnątrz wielokąta sprawdzę jakie powinny być warunki dla wyznaczonych równań a-e (mniejsze lub większe od zera)

$F(6,6)$

a) $6 + 6 \cdot 4 - 16 = 14 > 0$

b) $6 - 6 - 6 = -6 < 0$

c) $3 \cdot 6 + 5 \cdot 6 - 58 = -10 < 0$

d) $6 - 4 \cdot 6 + 26 = 8 > 0$

e) $2 \cdot 6 + 6 - 11 = 7 > 0$

Dla uproszczenia działań w neuronach możemy zamienić znaki w równaniu b i c. Wtedy wszystkie neurony będą na wyjściu z neuronu dawały 1 jeśli wynik działania jest większy od 0.

Otrzymujemy wtedy następujące równania:

a: $x + 4y - 16 = 0$

b: $-x + y + 6 = 0$

c: $-3x - 5y + 58 = 0$

d: $x - 4y + 26 = 0$

e: $2x + y - 11 = 0$

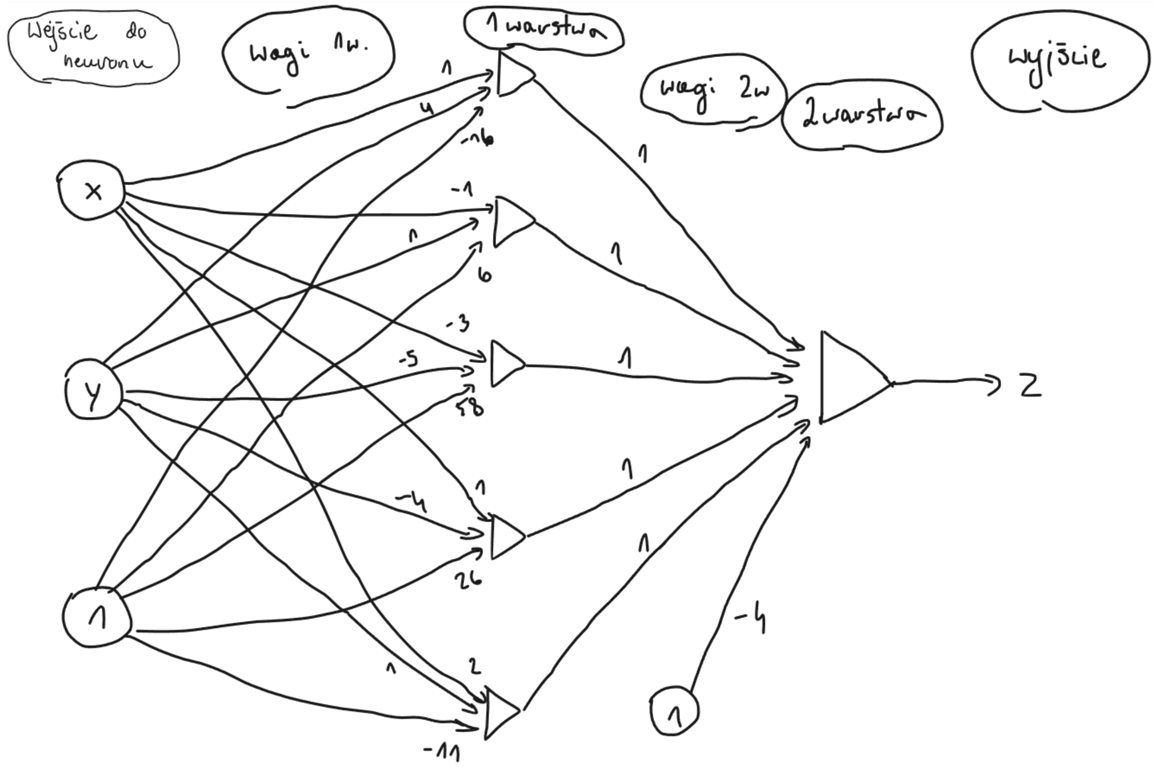
Neuron

Pojedynczy neuron pierwszej warstwy ma postać:

$$z = w_1 x + w_2 y + w_3$$

równanie\waga	w1	w2	w3
a [1]	1	4	-16
b [2]	-1	1	6
c [3]	-3	-5	58
d [4]	1	-4	26
e [5]	2	1	-11

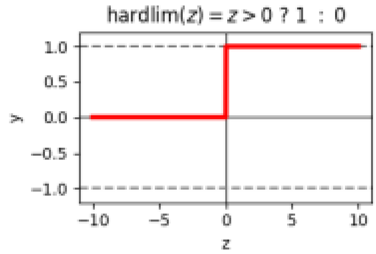
Na wejściu do neuronu podajemy x, y i 1 (w celu dodania w3). Tak wygląda sieć neuronowa:



Druga warstwa tej sieci przyjmuje 6 wejść - 5 wyjść z poprzedniej warstwy (0 lub 1) z wagami 1 oraz stała 1 o wadze -4. W ten sposób jeśli wszystkie warunki są spełnione (5 jedynek na wejściu z pierwszej warstwy) na wyjściu neuronu znajdzie się 1 a jeśli chociaż jeden warunek nie jest spełniony to 0 lub mniej.

Funkcja aktywacji

Dla pierwszej warstwy zostanie użyta funkcja hardlim(z).



jeśli z będzie > 0 to na wyjściu pojawi się 1 w przeciwnym przypadku będzie to 0.

Dla drugiej warstwy wykorzystam tę samą funkcję.

Test

W Pythonie wygenerowałam 1000 punktów z zakresu x(0, 14) i y(0,10) i podałam je na wejściu sieci neuronowej. Na zielono zaznaczone są granice pięciokąta, na czerwono punkty które po podaniu do sieci zwróciły 0, a na niebiesko te dla których sieć neuronowa na wyjściu zwróciła 1. Potwierdza to działanie obliczonych wag.