

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

Dirbtinis neuronas

Skaitmeninis intelektas ir sprendimų priėmimas

Užduoties ataskaita

Atliko: 4 kurso 4 grupės studentas
Vilius Puškunalis

Vilnius – 2023

TURINYS

1. ĮVADAS	2
2. KLASIFIKAVIMO DUOMENYS IR KLASĖS	3
3. PROGRAMOS KODAS SU KOMENTARAIŠ	4
4. SVORIŲ IR POSLINKIO SURADIMAS	6
4.1. Svarių ir poslinkio parinkimo strategija.....	6
4.2. Svarių ir poslinkio reikšmių rinkiniai	6
4.2.1. Naudojant slenkstinę aktyvacijos funkciją.....	6
4.2.2. Naudojant sigmoidinę aktyvacijos funkciją.....	7
5. NELYGYBIŲ SISTEMOS SUDARYMAS IR SPRENDIMAS	8
5.1. Nelygybių sistemos sudarymas	8
5.2. Nelygybių sistemos sprendimas	8
5.3. Sprendinių patikrinimas	10
6. REZULTATAI	11
ŠALTINIAI	12

1. Įvadas

Užduoties tikslas

Išanalizuoti dirbtinio neurono modelio veikimo principus.

Uždaviniai

- Parašyti programą, kurioje būtų realizuotas dirbtinis neuronas ir galimybė pasirinkti aktyvacijos funkciją (slenkstinę arba sigmoidinę).
- Surasti tokias poslinkio ir svorių reikšmes slenkstinei ir sigmoidinei funkcijai, kad duomenys būtų klasifikuojami teisingai.
- Duomenis ir jų klases surašyti į nelygybių sistemą ir ją išspręsti naudojant slenkstinę aktyvacijos funkciją, patikrinti sprendinius.

2. Klasifikavimo duomenys ir klasės

Turimi duomenys pavaizduoti 1 lentelėje, jie yra realaus skaičiaus tipo, susidaro iš dviejų taškų x_1 ir x_2 . Duomenys priklauso vienai iš dviejų klasių – 0 arba 1.

1 lentelė. Duomenys klasifikavimui

Duomenys		Klasė
x_1	x_2	t
-0,2	0,5	0
0,2	-0,7	0
0,8	-0,8	1
0,8	1	1

3. Programos kodas su komentarais

Dirbtinis neuronas įgyvendintas Python programavimo kalba. Apibrėžtos slenkstinė ir sigmoidinė aktyvacijos funkcijos, duomenų klasifikavimo funkcija ir duomenų rinkinio klasifikavimo funkcija, kuriai paduodama norima aktyvacijos funkcija ir poslinkis su svoriais.

```
1 import math
2 import random
3
4 # Slenkstine aktyvacijos funkcija
5 def binary_step_activation(a):
6     return 0 if a < 0 else 1
7
8 # Sigmoidine aktyvacijos funkcija
9 def sigmoid_activation(a):
10     return round(1 / (1 + math.exp(-a)))
11
12 # Klasifikavimo funkcija
13 def classify_data_point(f, w0, weights, points):
14     return f(w0 + sum(w * x for w, x in zip(weights, points)))
15
16 def classify_dataset(classification_data, activation_function, w0, w1,
17 w2):
18     # Aktyvacijos funkcija paduodama kaip simboliu eilute
19     if activation_function == "binary":
20         selected_activation_function = binary_step_activation
21     elif activation_function == "sigmoid":
22         selected_activation_function = sigmoid_activation
23     else:
24         print("Invalid activation function selected, must be 'binary'
25             or 'sigmoid'.")
26         return
27
28     classes = ()
29
30     # Klasifikavimas ir rezultatu spausdinimas
31     for x1, x2 in classification_data:
32         result = classify_data_point(selected_activation_function, w0,
33             (w1, w2), (x1, x2))
34         classes += (result,)
35
36     return classes
37
38 # Paleidimas: python task-1.py
```

```

36 if __name__ == "__main__":
37     # Programos argumentai
38     classification_data = [(-0.2, 0.5), (0.2, -0.7), (0.8, -0.8), (0.8,
39         1)]
40     data_classes = (0, 0, 1, 1)
41     activation_function = "sigmoid"
42     weights_interval = 100
43     weights_to_find = 10
44
45     # Rasti svoriu reiksmiu rinkinius
46     for _ in range(weights_to_find):
47         classes = []
48         while classes != data_classes:
49             # Sugeneruoti atsitiktina poslinki ir svorius
50             w0 = random.randint(-weights_interval, weights_interval)
51             w1 = random.randint(-weights_interval, weights_interval)
52             w2 = random.randint(-weights_interval, weights_interval)
53
54             classes = classify_dataset(classification_data,
55                 activation_function, w0, w1, w2)
56
57     print(f"Classes correct using {activation_function} activation
58         function with weights w0={w0}, w1={w1}, w2={w2}")

```

4. Sviurių ir poslinkio suradimas

4.1. Sviurių ir poslinkio parinkimo strategija

Sviuriai ir poslinkis buvo parenkami atsitiktiniu būdu. Atsitiktinai parenkami buvo tik sveikieji skaičiai intervale nuo -100 iki 100.

4.2. Sviurių ir poslinkio reikšmių rinkiniai

4.2.1. Naudojant slenkstinę aktyvacijos funkciją

Slenkstinė aktyvacijos funkcija grąžina 1 jei argumentas didesnis arba lygus 0, ir grąžina 0, jei argumentas mažesnis už 0 [Kur21]. Rastos reikšmės pavaizduotos 2 lentelėje.

$$f(a) = \begin{cases} 1, & \text{jei } a \geq 0 \\ 0, & \text{jei } a < 0 \end{cases}$$

2 lentelė. Sviurio ir poslinkių reikšmių rinkiniai teisingam duomenų klasifikavimui su slenkstine aktyvacijos funkcija

Poslinkis	Sviuriai	
w_0	w_1	w_2
-8	79	23
-36	92	22
-39	76	-9
-49	81	-5
1	59	20
-30	78	-10
-2	88	38
-28	99	31
-40	86	-21
-12	68	20

4.2.2. Naudojant sigmoidinę aktyvacijos funkciją

Sigmoidinė (logistinė) funkcija grąžina reikšmes nuo 0 iki 1, jos grafikui būdinga „S“ formos kreivė [Kur21]. Rastos reikšmės pavaizduotos 3 lentelėje.

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

3 lentelė. Svorio ir poslinkių reikšmių rinkiniai teisingam duomenų klasifikavimui su sigmoidine aktyvacijos funkcija

Poslinkis	Svoriai	
w_0	w_1	w_2
-37	57	8
-48	93	10
-38	92	-4
-22	71	16
-16	56	11
-42	67	4
-27	79	-16
-38	98	14
-29	59	18
-41	91	-25

5. Nelygybių sistemos sudarymas ir sprendimas

5.1. Nelygybių sistemos sudarymas

$$\begin{cases} w_0 - 0.2w_1 + 0.5w_2 \leq 0, \\ w_0 + 0.2w_1 - 0.7w_2 \leq 0, \\ w_0 + 0.8w_1 - 0.8w_2 > 0, \\ w_0 + 0.8w_1 + w_2 > 0. \end{cases} \quad (1)$$

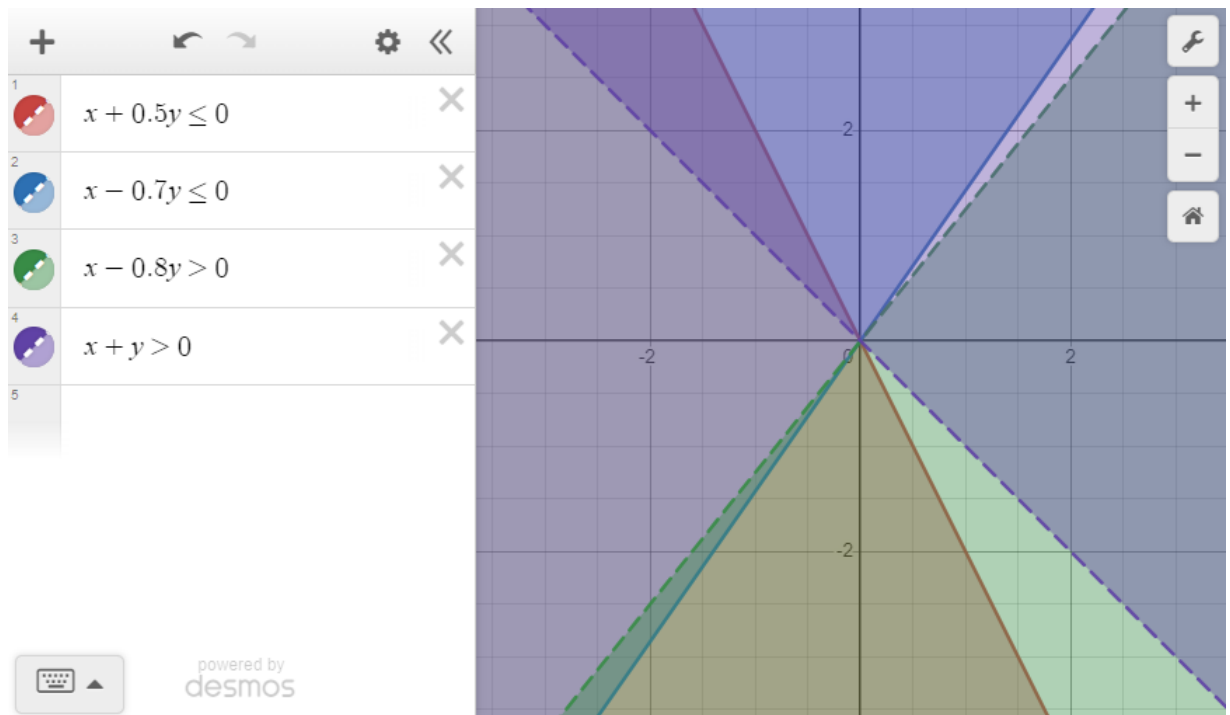
5.2. Nelygybių sistemos sprendimas

Tam, kad palengvinti sprendimą, vieną iš kintamųjų galima pakeisti į konstantą.

Grafinių sprendinių radimui naudojama Desmos [Des23] ir WolframAlpha [Alp23].

Pasirinkus $w_1 = 0$, pažymėjus $x = w_0$ ir $y = w_2$ gauname tokią sistemą, grafiškai pavaizduotą 1 pav.:

$$\begin{cases} x + 0.5y \leq 0, \\ x - 0.7y \leq 0, \\ x - 0.8y > 0, \\ x + y > 0. \end{cases} \quad (2)$$

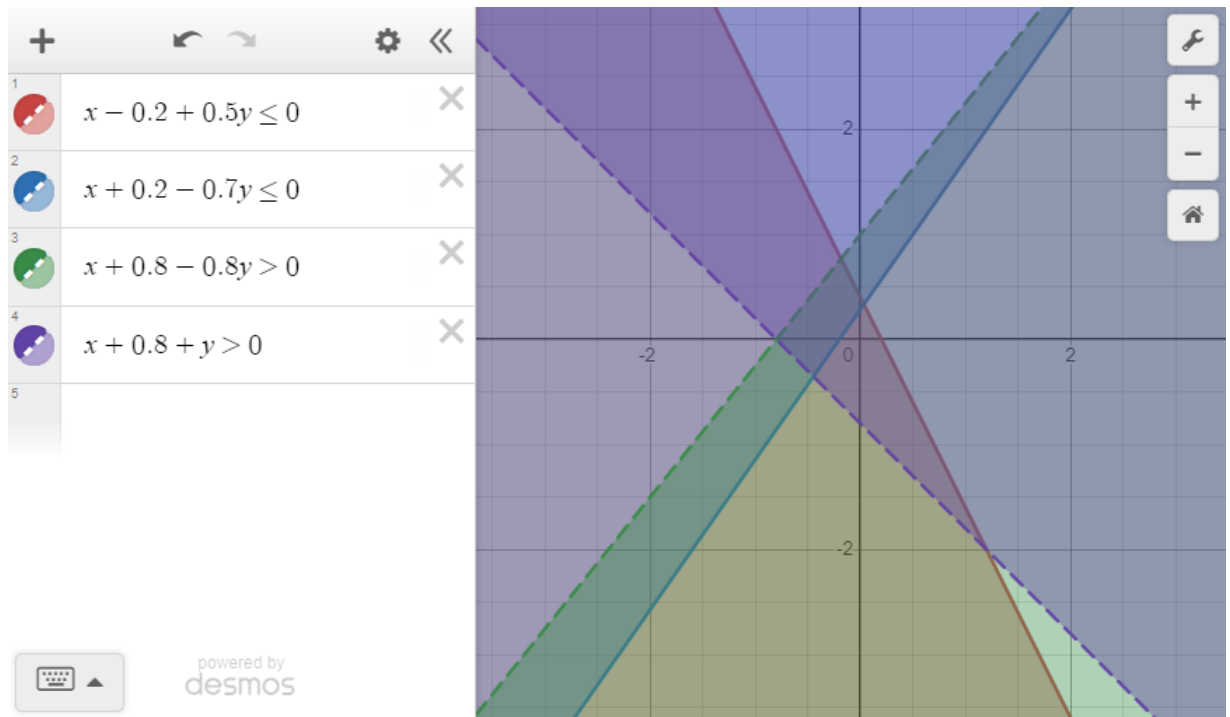


1 pav. 2 sistemos brėžinys

Iš 1 pav. matome, kad jokiaje vietoje nesusikerta visi keturi plotai, todėl su pasirinkta reikšme $w_1 = 0$ sprendinių nėra.

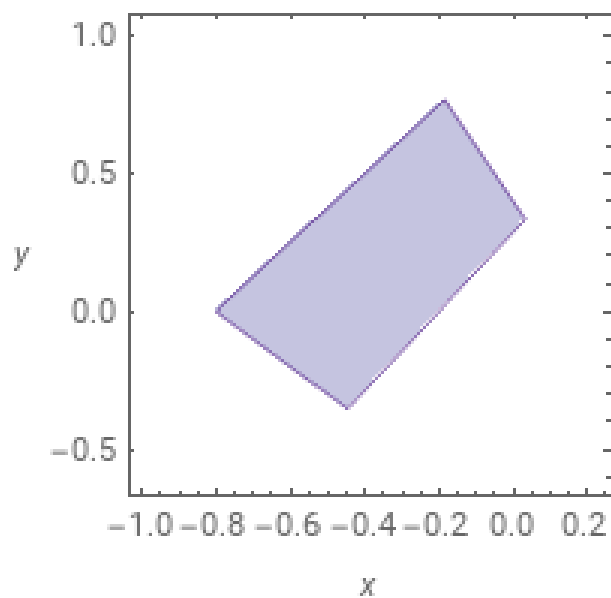
Pasirinkus $w_1 = 1$, pažymėjus $x = w_0$ ir $y = w_2$ gauname tokią sistemą:

$$\begin{cases} x - 0.2 + 0.5y \leq 0, \\ x + 0.2 - 0.7y \leq 0, \\ x + 0.8 - 0.8y > 0, \\ x + 0.8 + y > 0. \end{cases} \quad (3)$$



2 pav. 3 sistemos brėžinys

Iš 1 pav. matome, kad šiuo atveju yra vieta, kurioje visi keturi plotai susikerta, todėl su $w_1 = 1$ egzistuoja sprendinių aibė.



3 pav. 3 sistemos sprendiniai

WolframAlpha pagalba taip pat gauti tikslūs sprendiniai:

$$\frac{1}{5}(-5x - 4) < y < \frac{1}{4}(5x + 4), \text{ kai } -\frac{4}{5} < x \leq -\frac{38}{85}$$

$$\frac{2}{7}(5x + 1) \leq y < \frac{1}{4}(5x + 4), \text{ kai } -\frac{38}{85} < x \leq -\frac{12}{65}$$

$$\frac{2}{7}(5x + 1) \leq y \leq -\frac{2}{5}(5x - 1), \text{ kai } -\frac{12}{65} < x < \frac{1}{30}$$

5.3. Sprendinių patikrinimas

Sprendimą galima patikrinti į 1 sistemą įstatydami pasirinktą reikšmę $w_1 = 1$, taip pat iš 3 pav. paėmus reikšmės $x = w_0 = -0.4$, $y = w_2 = 0$.

$$\begin{cases} -0.4 & - & 0.2 & + & 0.5 \cdot 0 & \leq 0, \\ -0.4 & + & 0.2 & - & 0.7 \cdot 0 & \leq 0, \\ -0.4 & + & 0.8 & - & 0.8 \cdot 0 & > 0, \\ -0.4 & + & 0.8 & & & > 0. \end{cases} \quad (4)$$

Suprastinus šią sistemą gauname kitą sistemą:

$$\begin{cases} -0.6 & \leq 0, \\ -0.2 & \leq 0, \\ 0.4 & > 0, \\ 0.4 & > 0. \end{cases} \quad (5)$$

Visos nelygybės teisingos, taigi klasifikavimas teisingas su poslinkiu $w_0 = -0.4$ ir svoriais $w_1 = 1$, $w_2 = 0$.

6. Rezultatai

- Python kalba parašyta programa, kurioje realizuotas dirbtinis neuronas klasifikuojantis duomenis į dvi klases naudodamas slenkstinę arba sigmoidinę aktyvacijos funkcijas.
- Surastos tokios poslinkio ir svorių reikšmės naudojant slenkstinę ir sigmoidinę aktyvacijos funkcijas, kad visi duomenys yra klasifikuojami teisingai.
- Surastos poslinkio ir svorių reikšmės sudarius nelygybių sistemą, ją išsprendus grafiškai ir patikrinus rastus sprendinius.

Šaltiniai

- [Alp23] Wolfram Alpha. *Wolfram | Alpha: Computational Intelligence*. 2023.
- [Des23] Desmos. *Desmos | Graphing Calculator*. 2023.
- [Kur21] O. Kurasova. *Skaitmeninis intelektas ir sprendimų priėmimas*. 2021.