

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Dirbtinis neuronas

1 užduotis

Atliko: 4 kurso 1 grupės studentė
Rosita Raišuitytė

September 20, 2023

TURINYS

| | |
|---|---|
| 1. ĮVADAS | 2 |
| 1.1. Tikslas | 2 |
| 1.2. Uždaviniai | 2 |
| 2. DUOMENYS | 2 |
| 3. PROGRAMOS KODAS | 3 |
| 4. SVORIŲ IR POSLINKIO PASIRINKIMAS | 4 |
| 4.1. Pasirinkimo strategija | 4 |
| 4.2. Rezultatai..... | 4 |
| 4.2.1. Su slenkstine aktyvacijos funkcija | 4 |
| 4.2.2. Su sigmoidine aktyvacijos funkcija | 5 |
| 5. NELYGYBIŲ SISTEMA, KAI AKTYVACIJOS FUNKCIJA SLENKSTINĖ | 5 |
| 5.1. Nelygybių sistema ir sprendinys | 5 |
| 5.2. Nelygybių sistemos sprendinio patikrinimas | 6 |

1. Įvadas

1.1. Tikslas

Užduoties tikslas – išanalizuoti dirbtinio neurono modelį, jo veikimo principus, teisingai suklasifikuoti duomenis į klases.

1.2. Uždaviniai

- Parašyti programą, kuri realizuotų dirbtinio neurono veikimo principą bei rastų svorių ir poslinkių reikšmes naudojant slenkstinę ir sigmoidinę aktyvacijos funkcijas.
- Grafiniu būtu rasti poslinkių ir svorių reikšmes naudojant slenkstinę aktyvacijos funkciją.
- Patiktinti grafiškai gautus sprendinius.

2. Duomenys

Užduotyje naudojami duomenys iš 1 lentelės, kuriems ieškomas poslinkis ir svoriai, norint teisingai suklasifikuoti duomenis į klases.

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

Programos kodą galite rasti paspaudę nuorodą [čia](#).

```
import numpy as np

# susirašom duomenis pateiktus 1 lentelėje
input = [[-0.2, 0.5, 0],
         [0.2, -0.7, 0],
         [0.8, -0.8, 1],
         [0.8, 1, 1]]

# funkcija, kuri sugeneruoja visas galimas svorių kombinacijas (w0, w1, w2),
# kur w0 yra poslinkis, nuo intervale nuo -10 iki 10 imtinai su žingsniu 1
def generateWeightsAndBias():
    weights = []

    for i in range(-10, 11):
        for j in range(-10, 11):
            for k in range(-10, 11):
                weights.append([i, j, k])

    return weights

# apibrėžta slenkstinė aktyvacijos funkcija su slenksčiu 0
def threshold(x):
    if x >= 0:
        return 1
    else:
        return 0

# apibrėžta sigmoidinė aktyvacijos funkcija, kadangi gražinama reikšmė turi būti 1 arba 0,
# apskaičiuotą sigmoidinės funkcijos reikšmę turime suapvalinti arba pritaikyti slenkstį
def sigmoid(x):
    return round(1 / (1 + np.exp(-x)))

# apibrėžta funkcija, kuris skaičiuoja įėjimo reikšmių ir svorių sandaugų
# sumą, prie kurios dar pridėdamos poslinkis
# weights[0] - w0 arba poslinkis
# weights[1] - w1
# weights[2] - w2
# input[0] - x1
# input[1] - x2
def calculateX(input, weights):
    return weights[0]+input[0]*weights[1]+input[1]*weights[2]

# apibrėžta funkcija, kuri atrenka tinkamus svorius pagal nurodytą aktyvacijos funkciją.
# Veikimas: ciklo pagalba einama per visas svorių iteracijas, kiekvienai iteracijai
# sukamas papildomas ciklas, kuris tikrina ar suskaičiuota dirbtinio neurono modelio
# išeitis atitinka pradinuose duomenyse pateiktą klasę.
# Jei visiems duomenims svoriai gražino reikiamą atsakymą (skaičiuojamas count),
# jie pridėdami prie teisingų svorių sąrašo.
def findWeights(input, weights, activation):
    correctWeights = []
    for weight in weights:
        count = 0
        for i in range(np.shape(input)[0]):
            if activation(calculateX(input[i], weight)) == input[i][2]:
                count = count + 1
        if count == np.shape(input)[0]:
            correctWeights.append(weight)

    return correctWeights
```

1 pav. Programos kodas 1 dalis

```

# apibrėžta funkcija, kuri atrinka tinkamus svorius pagal nurodytą aktyvacijos funkciją.
# Veikimas: ciklo pagalba einama per visas svorių iteracijas, kiekvienai iteracijai
# sukamas papildomas ciklas, kuris tikrina ar suskaičiuota dirbtinio neurono modelio
# išeitis atitinka pradiniose duomenyse pateiktą klasę.
# Jei visiems duomenims svoriai gražino reikiamą atsakymą (suskaičiuojamas count),
# jie pridedami prie teisingų svorių sąrašo.
def findWeights(input, weights, activation):
    correctWeights = []
    for weight in weights:
        count = 0
        for i in range(np.shape(input)[0]):
            if activation(calculateX(input[i], weight)) == input[i][2]:
                count = count + 1
        if count == np.shape(input)[0]:
            correctWeights.append(weight)
    return correctWeights

# Susigeneruojam visas galimas svorių iteracijas intervale [-10,10],
# naudodamiesi funkcija findWeights randama tinkamus svorius aktyvacijos funkcijoms:
# slenkstinei, sigmoidinei.
weights = generateWeightsAndBias()

correctWeightsThreshold = findWeights(input, weights, threshold)
correctWeightsSigmoid = findWeights(input, weights, sigmoid)

print('Iš', np.shape(generateWeightsAndBias())[0], 'svorių iteracijų tinkamos buvo: ' )
print('    slenkstinei aktyvacijos funkcijai:', np.shape(correctWeightsThreshold)[0])
print('    sigmoidinei aktyvacijos funkcijai:', np.shape(correctWeightsSigmoid)[0])

```

2 pav. Programos kodas 2 dalis

4. Svorių ir poslinkio pasirinkimas

4.1. Pasirinkimo strategija

Šiame darbe naudojau strategiją, kurios metu pasirinktame intervale tam tikru žingsniu buvo perrinkti svorių ir poslinkio reikšmės ieškant tinkamų kombinacijų. Programos kode 1 aprašyta funkcija generateWeightsAndBias(), kuri sugeneruoja visas galimas svorių ir poslinkio kombinacijas intervale [-10;10] su žingsniu 1. Toliau naudojantis funkcija findWeights(input, weights, activation) buvo ieškoma tinkamų kombinacijų priklausomai nuo aktyvacijos funkcijos. Ši funkcija pereina visas sugeneruotas svorių ir poslinkio kombinacijas ir tikrina, ar pritaikius aktyvacijos funkciją suskaičiuotai dirbtinio neurono modelio išeičiai, gaunami teisingi rezultatai (klasės) kaip nurodyta 1 lentelėje.

4.2. Rezultatai

Iš viso buvo sugeneruota 9261 svorių ir poslinkio kombinacijų. Slenkstinei aktyvacijos funkcijai buvo surasta 167 tinkamos kombinacijos, sigmoidinei – 164.

4.2.1. Su slenkstine aktyvacijos funkcija

Iš 167 poslinkio ir svorių kombinacijų, su kuriomis slenkstinė aktyvacijos funkcija pateikė teisingus rezultatus, atsitiktinai išrinktos kelios, kurios pateikiamos 2 lentelėje.

2 lentelė. Galimi svorių ir poslinkių reikšmių rinkiniai

| w_0 | w_1 | w_2 |
|-------|-------|-------|
| -8 | 10 | 0 |
| -5 | 9 | -2 |
| -4 | 7 | 2 |
| -2 | 3 | 0 |
| -1 | 10 | 2 |
| 0 | 10 | 3 |

4.2.2. Su sigmoidine aktyvacijos funkcija

Iš 164 poslinkio ir svorių kombinacijų, su kuriomis sigmoidine aktyvacijos funkcija pateikė teisingus rezultatus, atsitiktinai išrinktos kelios, kurios pateikiamos 3 lentelėje.

3 lentelė. Galimi svorių ir poslinkių reikšmių rinkiniai

| w_0 | w_1 | w_2 |
|-------|-------|-------|
| -7 | 9 | 0 |
| -6 | 10 | 2 |
| -4 | 6 | 1 |
| -3 | 4 | 0 |
| -2 | 4 | -1 |
| 0 | 10 | 4 |

5. Nelygybių sistema, kai aktyvacijos funkcija slenkstinė

5.1. Nelygybių sistema ir sprendinys

Kai aktyvacijos funkcija slenkstinė, pateiktiems duomenims [1 lentelė], norint teisingai parinkti svorių ir poslinkio reikšmes, reikia išspręsti šią nelygybių sistemą:

$$w_0 - 0,2 \cdot w_1 + 0,5 \cdot w_2 < 0$$

$$w_0 + 0,2 \cdot w_1 - 0,7 \cdot w_2 < 0$$

$$w_0 + 0,8 \cdot w_1 - 0,8 \cdot w_2 \geq 0$$

$$w_0 + 0,8 \cdot w_1 + 1 \cdot w_2 \geq 0$$

Kadangi nelygybių sistemą spręsimė grafiškai ir nenorime naudoti 3d erdvės, tenka vieną iš kintamųjų apibrėžti kaip konstantą. Šiuo atveju pasirinkime w_0 ir priskirsime jam reikšmę $w_0 = -2$. Tada gausime nelygybių sistemą:

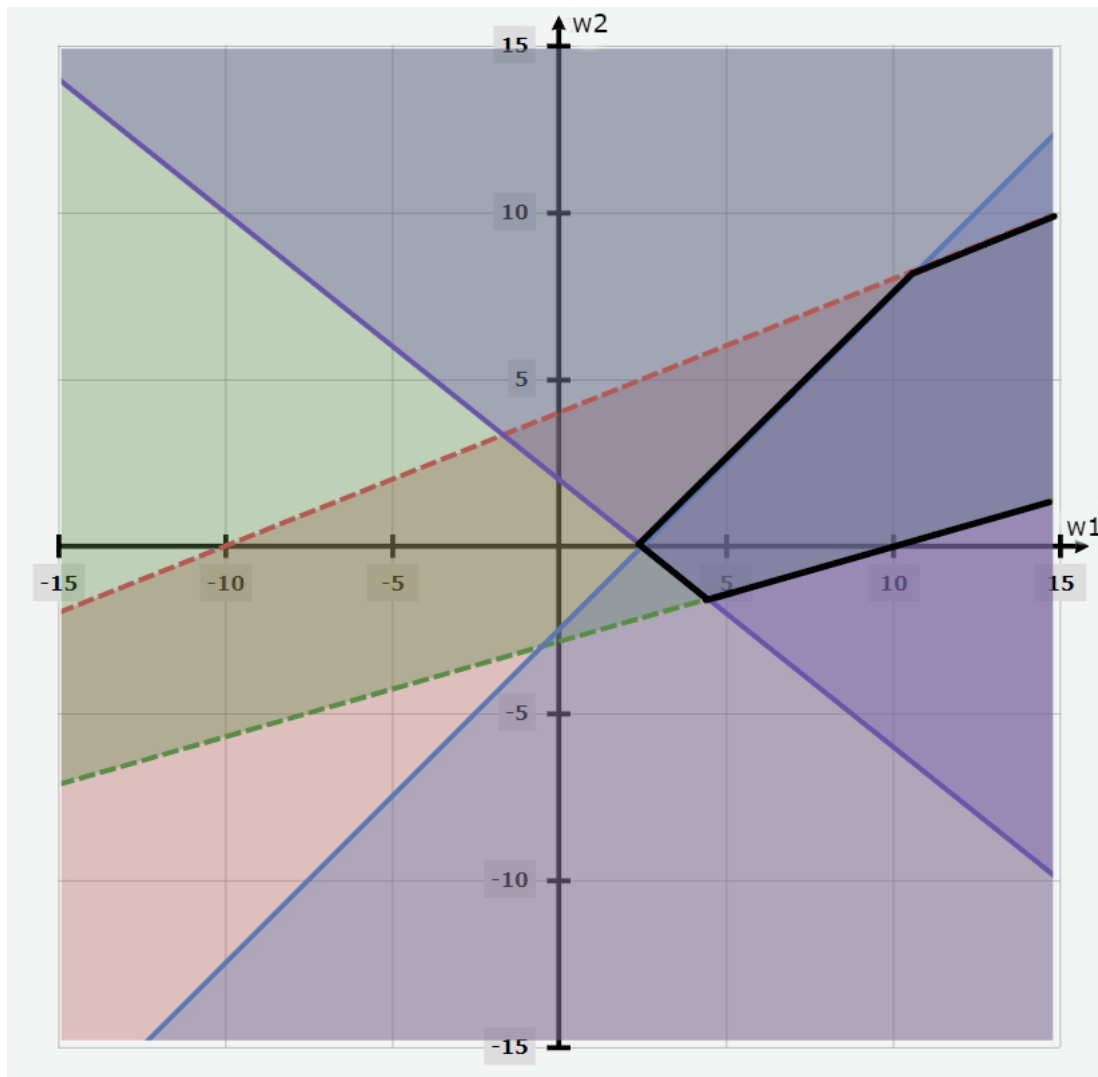
$$-2 - 0,2 \cdot w_1 + 0,5 \cdot w_2 < 0$$

$$-2 + 0,2 \cdot w_1 - 0,7 \cdot w_2 < 0$$

$$-2 + 0,8 \cdot w_1 - 0,8 \cdot w_2 \geq 0$$

$$-2 + 0,8 \cdot w_1 + 1 \cdot w_2 \geq 0$$

Šios nelygybių sistemos sprendinys matomas 3 pav. juodai apibrėžtame plote. Grafike w_1 ašis atitinka kintamąjį w_1 , w_2 ašis atitinka w_2 . Grafike matomos keturios spalvos: raudona, žalia, mėlyna, violetinė. Jos atpindi skirtingas nelygybių sistemos nelygybes. Raudona vaizduoja pirmosios sistemos nelygybės sprendinį, žalia – antrosios, mėlyna – trečiosios, violetinė – paskutinės, ketvirtosios nelygybės. Visų šių sprendinių, spalvų susikirtimo plotas yra visos nelygybių sistemos sprendinys, kuris apibrėžtas juodai.



3 pav. Grafinis nelygybių sistemos sprendinys

5.2. Nelygybių sistemos sprendinio patikrinimas

Kad patikrintumėme 3 pav. pateiktą nelygybių sistemos sprendinį išsirinksim du taškus, kurie turėtų tenkinti nelygybių sistemą. Vieną sprendinį pasirinksim, kad w_2 būtų žemiau nulio, o kitą, kad aukščiau.

Pirmasis sprendinys $(w_0, w_1, w_2) = (-2, 5, -1)$:

$$-2 - 0,2 \cdot 5 + 0,5 \cdot (-1) < 0 \quad -3,5 < 0$$

$$-2 + 0,2 \cdot 5 - 0,7 \cdot (-1) < 0 \quad -0,3 < 0$$

$$-2 + 0,8 \cdot 5 - 0,8 \cdot (-1) \geq 0 \quad 2,8 \geq 0$$

$$-2 + 0,8 \cdot 5 + 1 \cdot (-1) \geq 0 \quad 1 \geq 0$$

Matome, kad ištačius pirmąjį pasirinktą sprendinį, jis tenkina nelygybių sistemą.

Antras sprendinys $(w_0, w_1, w_2) = (-2, 12, 6)$:

$$-2 - 0,2 \cdot 12 + 0,5 \cdot 6 < 0 \quad -1,4 < 0$$

$$-2 + 0,2 \cdot 12 - 0,7 \cdot 6 < 0 \quad -3,8 < 0$$

$$-2 + 0,8 \cdot 12 - 0,8 \cdot 6 \geq 0 \quad 2,8 \geq 0$$

$$-2 + 0,8 \cdot 12 + 1 \cdot 6 \geq 0 \quad 13,6 \geq 0$$

Matome, kad ir antrasis sistemos nelygybių sprendinys tenkina sistemą, vadinasi grafinis nelygybių sistemos sprendinys yra teisingas.