



**Vilniaus
universitetas**

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

Praktinė užduotis nr. 1 DIRBTINIS NEURONAS

Ataskaita

Parengė: Duomenų mokslo 3k. 2gr.
studentė Simona Gelžinytė

Vilnius, 2022

Turinys

IVADAS	3
DIRBTINIO NEURONO MODELIO ANALIZĖ.....	3
Užduoties tikslas	3
Naudotos formulės ir funkcijos	3
Formulė ir funkcijos	3
Naudoti duomenys ir jų klasės	4
Slenkščio ir svorių radimas	4
Gauti rinkiniai	5
NELYGIBIŲ SISTEMA	6
Nelygių sistemos sprendimas grafiniu būdu	7
IŠVADOS	9
PROGRAMOS KODAS	10

IVADAS

Šio darbo tikslas yra aprašyti ir išanalizuoti sukurtą dirbtinio neurono modelį bei jo veikimo principus. Darbe bus pateikiamos modeliui sukurti naudotos formulės, įeities duomenys, aprašyta slenksčio bei svorių, tinkamų įeities duomenims, parinkimo eiga bei pateikti sugeneruotų tinkamų reikšmių pavyzdžiai. Darbe taip pat, bus išanalizuota nelygybių sistema, kurią reikia spręsti, norint teisingai parinkti svorių ir slenksčio reikšmes slenkstinės ir sigmoidinės aktyvavimo funkcijos naudojimo atveju ir pateiktas dirbtinio neurono modelio programos kodas.

Programos kodas parašytas „Python“ programavimo kalba. Programoje neurono modelis nebuvo mokomas, o reikšmės buvo atrinktos perrinkimo būdu.

DIRBTINIO NEURONO MODELIO ANALIZĖ

Užduoties tikslas

Aprašomos užduoties tikslas yra sukurti dirbtinio neurono modelį ir aprašyti jo veikimo principus.

Naudotos formulės ir funkcijos

Kuriant dirbtinio neurono modelį buvo naudojama formulė, skirta skaičiuoti įėjimo reikšmių ir svorių sandaugų sumai, taip pat slenkstinė ir sigmoidinė aktyvacijos funkcijos, neurono išėjimo reikšmei nustatyti.

Pateikus įėjimo reikšmes ir parinkus atsitiktinius svorius, buvo skaičiuojama įėjimo reikšmių ir svorių sandaugų suma. Buvo įvestas numanomas nulinis įėjimas, lygus vienetui, dėl kurio poslinkio reikšmė, tapo nuliniu svoriu w_0 ir buvo naudojamas įėjimo reikšmių ir svorių sandaugų sumos skaičiavimo metu. Vėliau gauta suma buvo naudojama slenkstinėje ir sigmoidinėje aktyvacijos funkcijose, ieškant neurono išėjimo reikšmės.

Sigmoidinės funkcijos naudojimo atveju, buvo pasirinkta laikyti, jog klasė bus lygi 1, kai $f(a) > 0,5$ ir lygi 0, kai $f(a) < 0,5$.

Formulė ir funkcijos

1. Įėjimo reikšmių ir svorių sandaugų sumos formulė

$$a = w_0 \cdot x_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n = \sum_{k=1}^n w_k \cdot x_k$$

2. Slenkstinė funkcija

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

3. Sigmoidinė funkcija

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

Naudoti duomenys ir jų klasės

Duomenys pateikti 1-oje lentelėje buvo naudojami ieškant tinkamų svorių (w_1, w_2) ir poslinkio (bias) (w_0) reikšmių, t.y. kad pateikti duomenys būtų tinkamai klasifikuojami, priskirti 0 arba 1 klasei.

1 lentelė. Duomenys klasifikavimui

<i>Duomenys</i>		<i>Klasė</i>
<i>x_1</i>	<i>x_2</i>	<i>t</i>
-0,3	0,6	0
0,3	-0,6	0
1,2	-1,2	1
-1,2	1,2	1

Slenksčio ir svorių radimas

Programos vykdymo metu įeities (x_1, x_2) ir klasės (t) duomenys buvo naudojami rasti tinkamoms slenksčio ir svorių reikšmėms.

Programoje pirma buvo įvedami turimi duomenys, tuomet apsirašomos slenkstinė bei sigmoidinė funkcijos. Svoriai ir poslinkis buvo randami atsitiktinai generuojant jų reikšmes (w_0, w_1, w_2) intervale $[-10, 10)$, ieškant tinkamos kombinacijos. Poslinkis w_0 buvo interpretuojamas kaip vienas iš svorių.

Naudojant slenkstinę ir sigmoidinę funkciją, pirma, buvo skaičiuojama įėjimo reikšmių ir svorių sandaugų suma (a). Gautoji suma buvo tikrinama, kuriai klasei priklauso:

- Slenkstinė funkcija – jeigu a mažiau už 0 – priklauso 0 klasei, jeigu daugiau arba lygi 0 – priklauso 1 klasei.
- Sigmoidinė funkcija – jeigu pritaikius sigmoidinės aktyvacijos funkciją $f(a)$ mažiau už 0,5 – priklauso 0 klasei, jeigu daugiau už 0 – priklauso 1 klasei.

Skaičiai atsitiktinai buvo generuojami 1000 kartų, po kiekvieno generavimo, gautos svorių reikšmės buvo įsistatomas į aprašytas slenkstinę bei sigmoidinę funkcijas. Jeigu su atsitiktinai sugeneruotais skaičiais buvo galima tinkamai klasifikuoti turimus duomenis, jie buvo išvedami į ekraną.

Gauti rinkiniai

Slenkstinės funkcijos

- 1) $w_0 = -1,13, w_1 = 6,27, w_2 = 3,59$
- 2) $w_0 = -5,23, w_1 = 8,77, w_2 = 1,44$
- 3) $w_0 = -7,77, w_1 = 7,43, w_2 = -0,16$
- 4) $w_0 = -2,6, w_1 = 5,06, w_2 = -0,44$

Sigmoidinės funkcijos

- 1) $w_0 = -3,53, w_1 = 6,65, w_2 = 1,18$
- 2) $w_0 = -5,41, w_1 = 6,21, w_2 = 6,56$
- 3) $w_0 = -3,09, w_1 = 6,99, w_2 = 2,35$
- 4) $w_0 = -5,47, w_1 = 8,07, w_2 = -0,43$

NELYGYBIŲ SISTEMA

$$-0,3w_1 + 0,6w_2 + w_0 < 0$$

$$0,3w_1 - 0,6w_2 + w_0 < 0$$

$$1,2w_1 - 1,2w_2 + w_0 \geq 0$$

$$1,2w_1 + 1,2w_2 + w_0 \geq 0$$

Kadangi sprendime w_2 gauname 0, galime pašalinti iš nelygybių.

$$-0,3w_1 + w_0 < 0$$

$$0,3w_1 + w_0 < 0$$

$$1,2w_1 + w_0 \geq 0$$

$$1,2w_1 + w_0 \geq 0$$

Kadangi trečioji ir ketvirtoji nelygybės sutampa, galime vieną iš jų pašalinti.

$$-0,3w_1 + w_0 < 0$$

$$0,3w_1 + w_0 < 0$$

$$1,2w_1 + w_0 \geq 0$$

Galime perkelti viską su w_1 į dešinę pusę.

$$w_0 < 0,3w_1$$

$$w_0 < -0,3w_1$$

$$w_0 \geq -1,2w_1$$

Iš čia galime įrodyti, kad w_1 visą laiką turi būti teigiamas, nes su neigiamu w_1 , gausime iš pirmos lygties, kad w_0 turi būti neigiamas skaičius, o iš trečios lygties, kad w_0 - teigiamas skaičius, tad intervalai niekada nepersikirs.

Kadangi žinome, kad $w_1 > 0$, galime perdėlioti nelygybes palikdami tik antrą ir trečią nelygybes (pirmoji ir taip yra patenkinama antrosios nelygybės, žinant kad $w_1 > 0$).

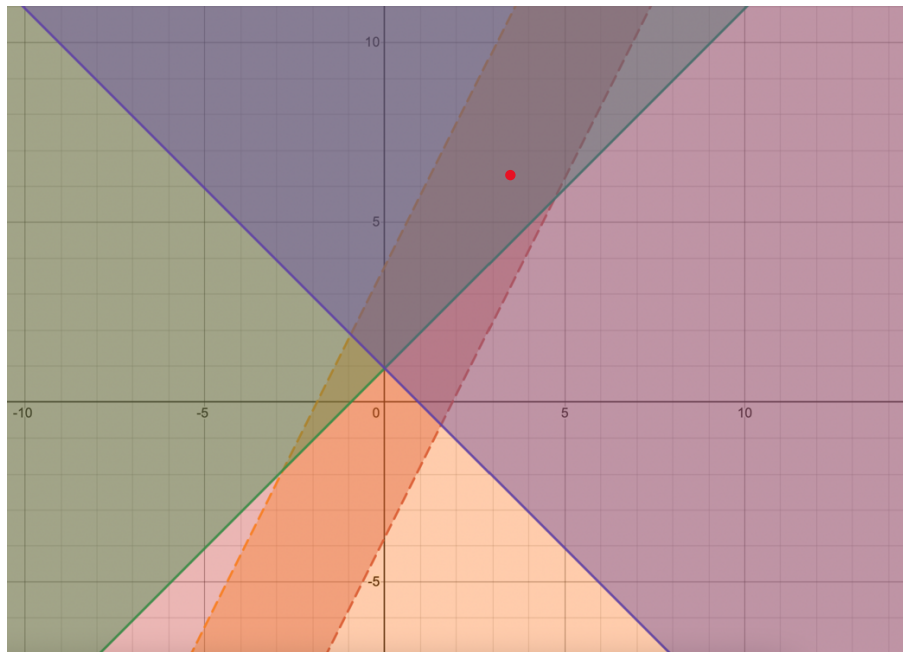
$$-1,2w_1 \leq w_0 < -0,3w_1$$

Nelygybių sistemos sprendimas grafiniu būdu

Sudarytuose grafikuose w_1 skaitomas kaip y, o w_2 kaip x. Grafikai buvo braižomi naudojant kompiuterinę grafinę skaičiuotuvą „Desmos“.

Pirmojo pavyzdžio duomenų testavimas

$$w_0 = -1,13, w_1 = 6,27, w_2 = 3,59$$

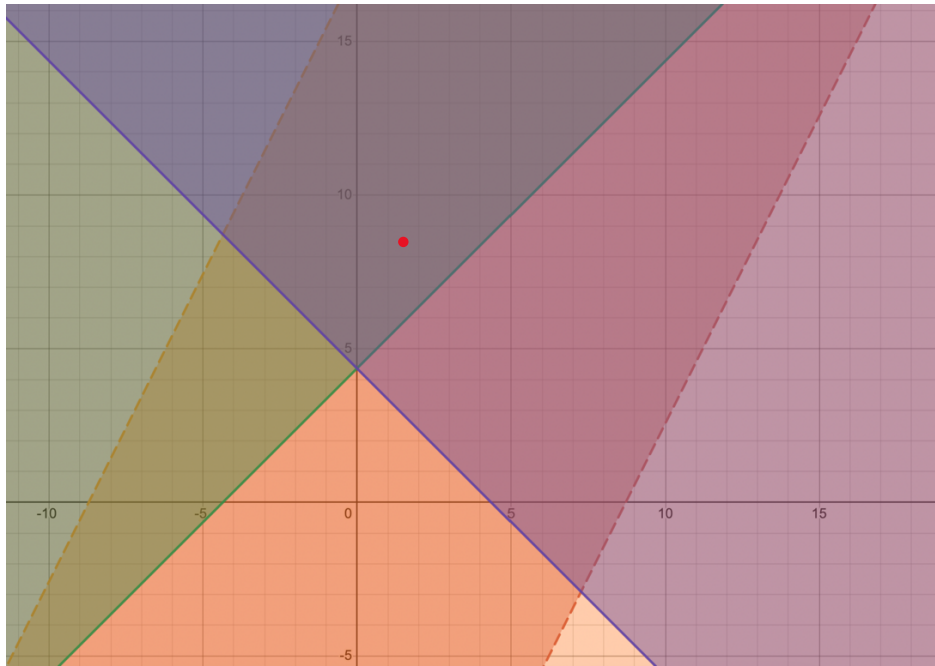


1 pav. Pirmojo pavyzdžio grafikas

Grafike matosi, kad svoriai yra duotos sistemos sprendiniai, nes taškas (3,59; 6,27) priklauso nuspalvintam plotui, kur kertasi visos nelygybės (jis pažymėtas raudonai).

Antrojo pavyzdžio duomenų testavimas

$$w_0 = -5,23, w_1 = 8,77, w_2 = 1,44$$

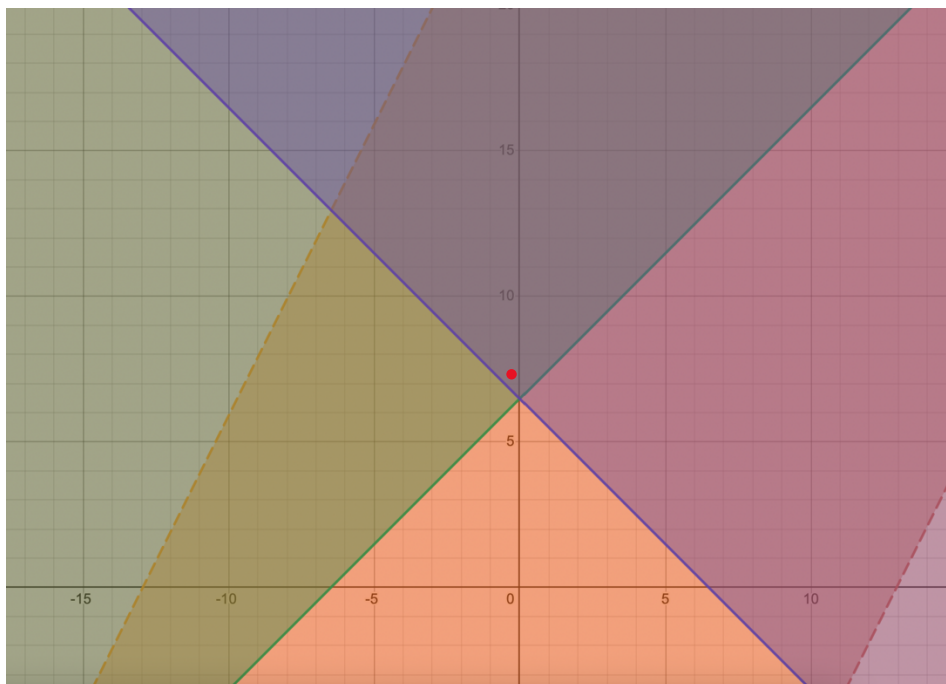


2 pav. Antrojo pavyzdžio grafikas

Grafike matosi, kad svoriai yra duotos sistemos sprendiniai, nes taškas $(1,44; 8,77)$ priklauso nuspaltintam plotui, kur kertasi visos nelygybės (jis pažymėtas raudonai).

Trečiojo pavyzdžio duomenų testavimas

$$w_0 = -7,77, w_1 = 7,43, w_2 = -0,16$$



3 pav. Trečiojo pavyzdžio grafikas

Grafike matosi, kad svoriai yra duotos sistemos sprendiniai, nes taškas $(-0,16; 7,43)$ priklauso nuspaltintam plotui, kur kertasi visos nelygybės (jis pažymėtas raudonai).

IŠVADOS

Užduoties atlikimo metu buvo suprastas dirbtinio neurono modelio veikimo principas ir parašytas kodas, skaičiuojantis išėjimo reikšmę su slenkstine bei sigmoidine aktyvacijos funkcijomis. Buvo suprasta, jog naudojant slenkstinę aktyvacijos funkciją, gaunamos reikšmės iškart yra tikslios ir tinkamos klasifikavimui, o sigmoidinės funkcijos naudojimo atveju, reikia apsibrėžti ribas, kuriose rezultatas priklausys norimai klasei.

PROGRAMOS KODAS

```
import random
import math

#surasome turimus duomenis
duomenys = [[-0.3, 0.6, 0],
             [0.3, -0.6, 0],
             [1.2, -1.2, 1],
             [1.2, 1.2, 1]]

#Slenkstine aktyvacijos funkcija
def slenkstine_fja(w0, w1, x1, w2, x2):
    a = w1 * x1 + w2 * x2 + w0
    if a >= 0:
        return 1
    return 0

#Sigmoidine aktyvacijos funkcija
def sigmoidine_fja(w0, w1, x1, w2, x2):
    a = w1 * x1 + w2 * x2 + w0
    fa = 1 / (1 + math.exp(-a))
    if fa >= 0.5:
        return 1
    return 0

#atsitiktiniu skaiciu generavimas
def skaiciu_generavimas():
    w0 = round(random.uniform(-10, 10), 2)
    w1 = round(random.uniform(-10, 10), 2)
    w2 = round(random.uniform(-10, 10), 2)
    return w0, w1, w2

#atsitiktinai generuojame skaiciu 1000 kartu
for n in range(1, 1000):
    slenkstinei_fjai = []
    sigmoidinei_fjai = []
    geri_slenkst = []
    w0, w1, w2 = skaiciu_generavimas()
    for i in duomenys:
        a_slenkst = slenkstine_fja(w0, w1, i[0], w2, i[1])
        slenkstinei_fjai.append(a_slenkst)
        #tikriname gautos klases sutampa, jeigu taip - isvedami gautos
        #svorius i ekrana
        if slenkstinei_fjai == [duomenys[0][2], duomenys[1][2],
                                duomenys[2][2], duomenys[3][2]]:
            print(f"Slenkstines fjos gauti svoriai {w0}, {w1}, {w2}")
    for i in duomenys:
        a_sigmoid = sigmoidine_fja(w0, w1, i[0], w2, i[2])
        sigmoidinei_fjai.append(a_sigmoid)
        if sigmoidinei_fjai == [duomenys[0][2], duomenys[1][2],
                                duomenys[2][2], duomenys[3][2]]:
            print(f"Sigmoidines fjos gauti svoriai {w0}, {w1}, {w2}")
```

