Vilniaus Universitetas

Dirbtinio intelekto pagrindai

1 užduotis. Dirbtinis neuronas

Darbą atliko:

Vilniaus universiteto matematikos ir informatikos fakulteto

duomenų mokslo 3 kurso 2 grupės studentas

Matas Amšiejus

Vilnius, 2021 09 28

Turinys

Užduoties tikslas 2
Pirmas punktas. Dirbtinis neuronas 3
Kodo aprašymas 3
Kodas 3
Antras punktas. Slenkstinė funkcija 3
Išėjimo parinkimas 3
Kodo aprašymas 3
Kodas 3
Trečias punktas. Sigmoidinė funkcija 4
Išėjimo pasirinkimas 4
Kodo aprašymas 4
Kodas 4
Ketvirtas punktas 5
Išvados 6
Priedai 7
1 priedas 7
2 priedas. 9
3 priedas 9

Užduoties tikslas

Sukurti programą, kuri pagal pateiktą duomenų lentelę gali apmokyti dirbtinį neuroną – parinkti tinkamus svorius, kurie tenkintų duotas sąlygas, kai pasirenkama slenkstinė arba sigmoidinė aktyvacijos funkcija.

1 lentelė. Sąlygoje duota lentelė.

x_1	x_2	**Norima reikšmė t (klasė)**
–0,2	0,5	0
0,2	–0,5	0
0,8	–0,8	1
0,8	0,8	1

Pirmas punktas. Dirbtinis neuronas

Kodo aprašymas

Naudodamas „Python“ programavimo kalbą sukūriau dirbtinį neuroną – funkciją, į kurią paduodami svoriai, įėjimų reikšmės bei aktyvacijos funkcijos pavadinimas. Pagal duotą sąlygą (*1 lentelė*. Sąlygoje duota lentelė.), mes paduodame tris svorius (w_0 – slenkstis, w_1, w_2 – svoriai) ir dvi įėjimo reikšmes (x_1, x_2). Kintamuoju *aktyv* mes nurodome kokią aktyvacijos funkciją norėsime naudoti – slenkstinę (*slen*) arba sigmoidinę (kitais atvejais). *r1* ir *r2* nurodo ribas, kurios žymi grupavimą sigmoidinei funkcijai (kai vykdoma slenkstinė funkcija, ribos ignoruojamos).

Kodas

```
def neuronas(w0, w1, w2, x1, x2, aktyv, r1, r2):
    a = 1*w0 + x1*w1 + x2*w2
    if aktyv == "slen":
        if a>=0:    return 1
        else:       return 0
    else:
        f = 1/(1+math.e**(-a))
        if f <= r1: return 0
        if f >= r2: return 1
        else:       return "gr neegz"
```

Antras punktas. Slenkstinė funkcija

Išėjimo parinkimas

Naudojant slenkstinę funkciją turėsime tik du atsakymų variantus: jei a (suma) < 0, tai y (išėjimas) $= 0$, jei $a \geq 0$, tai $y = 1$.

Kodo aprašymas

Tam, kad keisčiau svorių reikšmes, susikūriau potencialių reikšmių masyvą *gal_svor* nuo –1 iki 1 kas 0,1. Tada sukūriau tris *for* ciklus, kurie iteruoja per *gal_svor* masyvą ir stato turimus w_0, w_1, w_2 bei įėjimus x_1, x_2 į jau sukurtą neuroną (funkcija slenkstinė). Tam, kad patikrinčiau visas keturias sąlygas, duotas *1 lentelė*. Sąlygoje duota lentelė.*je*, sudarau dar vieną *for* ciklą. Jis taip pat leidžia gautą *y* reikšmę (išėjimą) iš neurono palyginti su sąlygoje duota klase. Jei klasės nesutampa, ciklas stabdomas ir pereinama prie kito svorio. Tačiau jei sąlyga yra tenkinama, *counter* padidėja per vienetą, o jei pasiekia keturis, tai leidžia suvokti, kad visos keturios lygtys buvo teisingos ir svoriai yra tinkami sąlygai. Juos atspausdinu ir pridedu į masyvą (jei reikėtų ateičiai). **Kodo atsakymai***: 1 priedas.*

Kodas

```
#Antras punktas
#sukuriame galimu reiksmiu vektoriu (nuo -1 iki 1 kas 0.1)
gal_svor = np.round(np.arange(-1,1.1,0.1), 1)
#sukuriame lentele, kuri duota uzduotyje (naudosime ir treciame punkte)
reiks_mat = np.array([[-0.2,0.5,0],[0.2,-0.5,0],[0.8,-0.8,1],[0.8,0.8,1]])
#sukuriame atsakymu vektoriu prie kurio lipdysime tinkamus svorius
atsakymai1 = np.empty([1,3], dtype=np.uint16)

#iteruojame per nustatyta svoriu intervala
for w0 in gal_svor:
    for w1 in gal_svor:
        for w2 in gal_svor:
```

```
            #sioje vietoje mes jau turime svorius kaip kostantas
            #dabar reikia imti x is lenteles ir tikrinti visus
            #sal[0] yra x1, sal[1] yra x2, sal[3] yra neurono isejimo reiksme
            counter = 0
            #imame viena lenteles eilute (viena salyga)
            for sal in reiks_mat:
                #skaiciuojame suma
                y = neuronas(w0,w1,w2,sal[0],sal[1],"slen",0,0)#gale 0, nes sigmoidine mums
dabar nesvarbu
                #kreipiames i aktyvacijos fja, jei salygoje duota klase nesutampa su
isejimu, nustojame tikrinti svoriu kombinacija ir einame prie kitos
                if y != sal[2]:
                    break
                #counter padeda sekti kiek salygu buvo patikrintos
                counter += 1
            #jei visos 4 salygos patvirtintos, musu svoriai tinkami pagal salyga ir yra
itraukiami i galimu atsakymu masyva
            if counter == 4:
                print("w0 = ", w0, ", w1 = ", w1, ", w2 = ", w2)
                atsakymai1 = np.append(atsakymai1, [[w0,w1,w2]], axis = 0)

atsakymai1 = np.delete(atsakymai1, 0, axis = 0)
```

Trečias punktas. Sigmoidinė funkcija

Išėjimo pasirinkimas

Kadangi sigmoidinė funkcija yra intervale $(0,1)$, tai iš anksto reikia nustatyti sąlygas, kada bus klasė 1, kada 0. Aš pasirinkau, kad visi $f(a) \leq 0{,}2$ būtų priskiriami 0 klasei ($y = 0$), o visi $f(a) \geq 0{,}8$ – 1 klasei ($y = 1$). Intervale $0{,}2 < f(a) < 0{,}8$ reikšmės nebus priskiriamos jokiai klasei.

Kodo aprašymas

Pirma sukuriu naują galimų svorių vektorių *gal_svor2* nuo -5 iki 5 kas 0,1, nes sigmoidinėje funkcijoje atkris daugiau reikšmių dėl jos siauresnių intervalų. Sukuriu *r1* ir *r2*, kurie žymi intervalų ribas, tinkančias mūsų uždaviniui. Šiame pavyzdyje pasirenku ribas tarp $(0, 0{,}2]$ ir $[0{,}8\ , 1)$. Kaip ir su slenkstine funkcija, leidžiu tris ciklus, naudoju tą patį *counter* bei spausdinimo metodą. Skirtumas tarp slenkstinės funkcijos yra tai, kad kreipiuosi į funkciją *neuronas* nenaudodamas žodžio „*slen*“. Dėl to skaičiuojama sigmoidinė aktyvacijos funkcija, naudojamos duotos ribos *r1* ir *r2*. **Kodo atsakymai**: *2 priedas.*

Kodas

```
gal_svor2 = np.round(np.arange(-5,5.1,0.1), 1)
atsakymai2 = np.empty([1,3], dtype=np.uint16)

#sukurkime norimas ribas musu sigmoidinei funkcijai atrinkti. Paimkime iki 0.1 (imtinai) 0ai
klasei ir nuo 0.9 1ai klasei
r1 = 0.2
r2 = 0.8

for w0 in gal_svor2:
    for w1 in gal_svor2:
        for w2 in gal_svor2:
            counter = 0
            #imame viena lenteles eilute (viena salyga)
            for sal in reiks_mat:
```

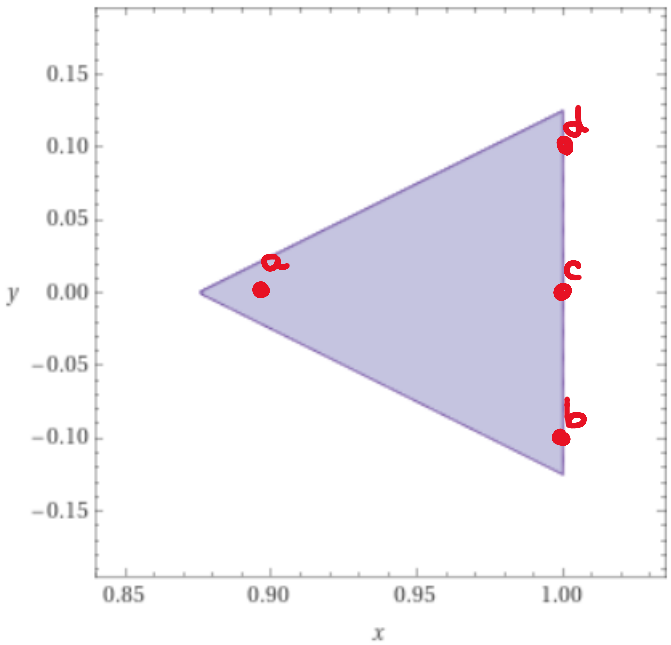
```
                #kreipiamasi i neurona, kad gautu isejima
                y = neuronas(w0,w1,w2,sal[0],sal[1],"sig",r1,r2)
                if y != sal[2]:
                    break
                #counter padeda sekti kiek salygu buvo patikrintos
                counter += 1
            #jei visos 4 salygos patvirtintos, musu svoriai tinkami pagal salyga ir yra
itraukiami i galimu atsakymu masyva
            if counter == 4:
                print("w0 = ", w0, ", w1 = ", w1, ", w2 = ", w2)
                atsakymai2 = np.append(atsakymai2, [[w0,w1,w2]], axis = 0)

atsakymai2 = np.delete(atsakymai2, 0, axis = 0)
```

Ketvirtas punktas

Norint rasti svorius ir slenkstį neuronui naudojant slenkstinę aktyvacijos funkciją, reikia išspręsti nelygybių sistemą:

$$\begin{cases} w_0 - 0{,}2w_1 + 0{,}5w_2 < 0 \\ w_0 + 0{,}2w_1 - 0{,}5w_2 < 0 \\ w_0 + 0{,}8w_1 - 0{,}8w_2 \geq 0 \\ w_0 + 0{,}8w_1 + 0{,}8w_2 \geq 0 \end{cases}$$

Tam, kad palengvinčiau nelygybių sprendimą, slenkstį w_0 prilyginu –0,7 (su šiuo skaičiumi pagal kodą yra tik 4 sprendiniai, todėl visus galėsiu patikrinti), taip pat x intervalą imu kaip ir kode: $(-1{,}1)$. Šią sistemą išsprendus grafiniu būdu (naudojant www.wolframalpha.com skaičiuotuvą *3 priedas*), gauname:



1 pav. w_1 (čia x) ir w_2 (čia y) reikšmės ($w_0 = -0{,}7$)

Įstačius reikšmes (reikšmės įstatytos rankiniu būdu (sužymėtos alfabetišku eiliškumu), gali turėti minimalią paklaidą) matome, kad kodas veikia gerai ir visos reikšmės tenkina nelygybių sistemos sprendinius.

Išvados

Šioje užduotyje sukūriau dirbtinį neuroną, kuris geba grąžinti išėjimą y pasitelkiant arba slenkstinę, arba sigmoidinę funkciją. Taip pat sudariau ciklus, kurie padeda atrinkti svorius (įskaičiuojant slenkstį), tenkinančius sąlygoje nurodytas lygybes. Tai nebuvo tikrasis neurono mokymas, tačiau tai, kas buvo reikalauta užduotyje, buvo įvykdyta. Grafinis nelygybių sistemos pavaizdavimas padeda geriau įsivaizduoti kompiuterio atrinktus rezultatus, tačiau tik tada, kai yra du kintamieji. Jei jų yra daugiau, grafinis sprendimas tampa sunkiai interpretuojamas ir suprantamas.

Priedai

1 priedas

Sąlygą tenkinantys svoriai naudojant slenkstinę funkciją (iš kairės į apačią kas puslapį):

w0 = -0.8 , w1 = 1.0 , w2 = -0.0
w0 = -0.7 , w1 = 0.9 , w2 = -0.0
w0 = -0.7 , w1 = 1.0 , w2 = -0.1
w0 = -0.7 , w1 = 1.0 , w2 = -0.0
w0 = -0.7 , w1 = 1.0 , w2 = 0.1
w0 = -0.6 , w1 = 0.8 , w2 = -0.0
w0 = -0.6 , w1 = 0.9 , w2 = -0.1
w0 = -0.6 , w1 = 0.9 , w2 = -0.0
w0 = -0.6 , w1 = 0.9 , w2 = 0.1
w0 = -0.6 , w1 = 1.0 , w2 = -0.2
w0 = -0.6 , w1 = 1.0 , w2 = -0.1
w0 = -0.6 , w1 = 1.0 , w2 = -0.0
w0 = -0.6 , w1 = 1.0 , w2 = 0.1
w0 = -0.6 , w1 = 1.0 , w2 = 0.2
w0 = -0.5 , w1 = 0.7 , w2 = -0.0
w0 = -0.5 , w1 = 0.8 , w2 = -0.1
w0 = -0.5 , w1 = 0.8 , w2 = -0.0
w0 = -0.5 , w1 = 0.8 , w2 = 0.1
w0 = -0.5 , w1 = 0.9 , w2 = -0.2
w0 = -0.5 , w1 = 0.9 , w2 = -0.1
w0 = -0.5 , w1 = 0.9 , w2 = -0.0
w0 = -0.5 , w1 = 0.9 , w2 = 0.1
w0 = -0.5 , w1 = 0.9 , w2 = 0.2
w0 = -0.5 , w1 = 1.0 , w2 = -0.3
w0 = -0.5 , w1 = 1.0 , w2 = -0.2
w0 = -0.5 , w1 = 1.0 , w2 = -0.1
w0 = -0.5 , w1 = 1.0 , w2 = -0.0
w0 = -0.5 , w1 = 1.0 , w2 = 0.1
w0 = -0.5 , w1 = 1.0 , w2 = 0.2
w0 = -0.4 , w1 = 0.5 , w2 = -0.0
w0 = -0.4 , w1 = 0.6 , w2 = -0.0
w0 = -0.4 , w1 = 0.7 , w2 = -0.1
w0 = -0.4 , w1 = 0.7 , w2 = -0.0
w0 = -0.4 , w1 = 0.7 , w2 = 0.1
w0 = -0.4 , w1 = 0.8 , w2 = -0.3
w0 = -0.4 , w1 = 0.8 , w2 = -0.2
w0 = -0.4 , w1 = 0.8 , w2 = -0.1
w0 = -0.4 , w1 = 0.8 , w2 = -0.0
w0 = -0.4 , w1 = 0.8 , w2 = 0.1
w0 = -0.4 , w1 = 0.8 , w2 = 0.2
w0 = -0.4 , w1 = 0.8 , w2 = 0.3
w0 = -0.4 , w1 = 0.9 , w2 = -0.4
w0 = -0.4 , w1 = 0.9 , w2 = -0.3
w0 = -0.4 , w1 = 0.9 , w2 = -0.2
w0 = -0.4 , w1 = 0.9 , w2 = -0.1
w0 = -0.4 , w1 = 0.9 , w2 = -0.0
w0 = -0.4 , w1 = 0.9 , w2 = 0.1
w0 = -0.4 , w1 = 0.9 , w2 = 0.2
w0 = -0.4 , w1 = 0.9 , w2 = 0.3
w0 = -0.4 , w1 = 0.9 , w2 = 0.4
w0 = -0.4 , w1 = 1.0 , w2 = -0.3
w0 = -0.4 , w1 = 1.0 , w2 = -0.2
w0 = -0.4 , w1 = 1.0 , w2 = -0.1
w0 = -0.4 , w1 = 1.0 , w2 = -0.0
w0 = -0.4 , w1 = 1.0 , w2 = 0.1
w0 = -0.4 , w1 = 1.0 , w2 = 0.2
w0 = -0.3 , w1 = 0.4 , w2 = -0.0
w0 = -0.3 , w1 = 0.5 , w2 = -0.1
w0 = -0.3 , w1 = 0.5 , w2 = -0.0
w0 = -0.3 , w1 = 0.5 , w2 = 0.1
w0 = -0.3 , w1 = 0.6 , w2 = -0.2
w0 = -0.3 , w1 = 0.6 , w2 = -0.1
w0 = -0.3 , w1 = 0.6 , w2 = -0.0
w0 = -0.3 , w1 = 0.6 , w2 = 0.1
w0 = -0.3 , w1 = 0.6 , w2 = 0.2
w0 = -0.3 , w1 = 0.7 , w2 = -0.3
w0 = -0.3 , w1 = 0.7 , w2 = -0.2
w0 = -0.3 , w1 = 0.7 , w2 = -0.1
w0 = -0.3 , w1 = 0.7 , w2 = -0.0
w0 = -0.3 , w1 = 0.7 , w2 = 0.1
w0 = -0.3 , w1 = 0.7 , w2 = 0.2
w0 = -0.3 , w1 = 0.7 , w2 = 0.3
w0 = -0.3 , w1 = 0.8 , w2 = -0.2
w0 = -0.3 , w1 = 0.8 , w2 = -0.1
w0 = -0.3 , w1 = 0.8 , w2 = -0.0
w0 = -0.3 , w1 = 0.8 , w2 = 0.1
w0 = -0.3 , w1 = 0.8 , w2 = 0.2
w0 = -0.3 , w1 = 0.8 , w2 = 0.3
w0 = -0.3 , w1 = 0.8 , w2 = 0.4
w0 = -0.3 , w1 = 0.9 , w2 = -0.2

w0 = -0.3 , w1 = 0.9 , w2 = -0.1
w0 = -0.3 , w1 = 0.9 , w2 = -0.0
w0 = -0.3 , w1 = 0.9 , w2 = 0.1
w0 = -0.3 , w1 = 0.9 , w2 = 0.2
w0 = -0.3 , w1 = 0.9 , w2 = 0.3
w0 = -0.3 , w1 = 0.9 , w2 = 0.4
w0 = -0.3 , w1 = 0.9 , w2 = 0.5
w0 = -0.3 , w1 = 1.0 , w2 = -0.1
w0 = -0.3 , w1 = 1.0 , w2 = -0.0
w0 = -0.3 , w1 = 1.0 , w2 = 0.1
w0 = -0.3 , w1 = 1.0 , w2 = 0.2
w0 = -0.3 , w1 = 1.0 , w2 = 0.3
w0 = -0.3 , w1 = 1.0 , w2 = 0.4
w0 = -0.3 , w1 = 1.0 , w2 = 0.5
w0 = -0.3 , w1 = 1.0 , w2 = 0.6
w0 = -0.2 , w1 = 0.3 , w2 = -0.0
w0 = -0.2 , w1 = 0.4 , w2 = -0.1
w0 = -0.2 , w1 = 0.4 , w2 = -0.0
w0 = -0.2 , w1 = 0.4 , w2 = 0.1
w0 = -0.2 , w1 = 0.5 , w2 = -0.1
w0 = -0.2 , w1 = 0.5 , w2 = -0.0
w0 = -0.2 , w1 = 0.5 , w2 = 0.1
w0 = -0.2 , w1 = 0.5 , w2 = 0.2
w0 = -0.2 , w1 = 0.6 , w2 = -0.1
w0 = -0.2 , w1 = 0.6 , w2 = -0.0
w0 = -0.2 , w1 = 0.6 , w2 = 0.1
w0 = -0.2 , w1 = 0.6 , w2 = 0.2
w0 = -0.2 , w1 = 0.6 , w2 = 0.3
w0 = -0.2 , w1 = 0.7 , w2 = -0.1
w0 = -0.2 , w1 = 0.7 , w2 = -0.0
w0 = -0.2 , w1 = 0.7 , w2 = 0.1
w0 = -0.2 , w1 = 0.7 , w2 = 0.2
w0 = -0.2 , w1 = 0.7 , w2 = 0.3
w0 = -0.2 , w1 = 0.7 , w2 = 0.4
w0 = -0.2 , w1 = 0.8 , w2 = -0.0
w0 = -0.2 , w1 = 0.8 , w2 = 0.1
w0 = -0.2 , w1 = 0.8 , w2 = 0.2
w0 = -0.2 , w1 = 0.8 , w2 = 0.3
w0 = -0.2 , w1 = 0.8 , w2 = 0.4
w0 = -0.2 , w1 = 0.8 , w2 = 0.5
w0 = -0.2 , w1 = 0.9 , w2 = -0.0
w0 = -0.2 , w1 = 0.9 , w2 = 0.1
w0 = -0.2 , w1 = 0.9 , w2 = 0.2
w0 = -0.2 , w1 = 0.9 , w2 = 0.3
w0 = -0.2 , w1 = 0.9 , w2 = 0.4
w0 = -0.2 , w1 = 0.9 , w2 = 0.5
w0 = -0.2 , w1 = 0.9 , w2 = 0.6
w0 = -0.2 , w1 = 1.0 , w2 = 0.1
w0 = -0.2 , w1 = 1.0 , w2 = 0.2
w0 = -0.2 , w1 = 1.0 , w2 = 0.3
w0 = -0.2 , w1 = 1.0 , w2 = 0.4
w0 = -0.2 , w1 = 1.0 , w2 = 0.5
w0 = -0.2 , w1 = 1.0 , w2 = 0.6
w0 = -0.2 , w1 = 1.0 , w2 = 0.7
w0 = -0.1 , w1 = 0.2 , w2 = -0.0
w0 = -0.1 , w1 = 0.3 , w2 = -0.0
w0 = -0.1 , w1 = 0.3 , w2 = 0.1
w0 = -0.1 , w1 = 0.4 , w2 = -0.0
w0 = -0.1 , w1 = 0.4 , w2 = 0.1
w0 = -0.1 , w1 = 0.4 , w2 = 0.2
w0 = -0.1 , w1 = 0.5 , w2 = 0.1
w0 = -0.1 , w1 = 0.5 , w2 = 0.2
w0 = -0.1 , w1 = 0.5 , w2 = 0.3
w0 = -0.1 , w1 = 0.6 , w2 = 0.1
w0 = -0.1 , w1 = 0.6 , w2 = 0.2
w0 = -0.1 , w1 = 0.6 , w2 = 0.3
w0 = -0.1 , w1 = 0.6 , w2 = 0.4
w0 = -0.1 , w1 = 0.7 , w2 = 0.1
w0 = -0.1 , w1 = 0.7 , w2 = 0.2
w0 = -0.1 , w1 = 0.7 , w2 = 0.3
w0 = -0.1 , w1 = 0.7 , w2 = 0.4
w0 = -0.1 , w1 = 0.8 , w2 = 0.2
w0 = -0.1 , w1 = 0.8 , w2 = 0.3
w0 = -0.1 , w1 = 0.8 , w2 = 0.4
w0 = -0.1 , w1 = 0.8 , w2 = 0.5
w0 = -0.1 , w1 = 0.9 , w2 = 0.2
w0 = -0.1 , w1 = 0.9 , w2 = 0.3
w0 = -0.1 , w1 = 0.9 , w2 = 0.4
w0 = -0.1 , w1 = 0.9 , w2 = 0.5
w0 = -0.1 , w1 = 1.0 , w2 = 0.3
w0 = -0.1 , w1 = 1.0 , w2 = 0.4
w0 = -0.1 , w1 = 1.0 , w2 = 0.5
w0 = -0.1 , w1 = 1.0 , w2 = 0.6

2 priedas.

Sąlygą tenkinantys svoriai naudojant sigmoidinę funkciją (iš kairės į apačią):

w0 = -2.6 , w1 = 5.0 , w2 = -0.0

w0 = -2.5 , w1 = 4.9 , w2 = -0.0

w0 = -2.5 , w1 = 5.0 , w2 = -0.1

w0 = -2.5 , w1 = 5.0 , w2 = -0.0

w0 = -2.5 , w1 = 5.0 , w2 = 0.1

w0 = -2.4 , w1 = 4.8 , w2 = -0.0

w0 = -2.4 , w1 = 4.9 , w2 = -0.0

w0 = -2.4 , w1 = 4.9 , w2 = 0.1

w0 = -2.4 , w1 = 5.0 , w2 = -0.0

w0 = -2.4 , w1 = 5.0 , w2 = 0.1

w0 = -2.4 , w1 = 5.0 , w2 = 0.2

w0 = -2.3 , w1 = 4.8 , w2 = 0.1

w0 = -2.3 , w1 = 4.9 , w2 = 0.2

w0 = -2.3 , w1 = 5.0 , w2 = 0.2

w0 = -2.3 , w1 = 5.0 , w2 = 0.3

w0 = -2.2 , w1 = 4.8 , w2 = 0.3

w0 = -2.2 , w1 = 4.9 , w2 = 0.4

w0 = -2.2 , w1 = 5.0 , w2 = 0.4

w0 = -2.2 , w1 = 5.0 , w2 = 0.5

w0 = -2.1 , w1 = 5.0 , w2 = 0.6

3 priedas

Nuoroda į grafiką:

plot -0.7-0.2x+0.5y<0 and -0.7+0.2x-(-0.5)y<0 and -0.7+0.8x-0.8y>=0 and -0.7+0.8x+0.8y>=0, x = -1 to 1, y = -1 to 1 - Wolfram|Alpha (wolframalpha.com)