

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

## **Dirbtinis neuronas**

Laboratorinis darbas

Atliko: 4 kurso 1 grupės studentas

Paulius Minajevs

Vilnius – 2024

# Turinys

ĮVADAS .....	3
1. KLASIFIKAVIMO DUOMENYS IR METODIKA .....	4
1.1. Klasifikavimo duomenys .....	4
1.2. Metodika.....	4
1.2.1. Intervalo perėjimas nustatytu žingsniu metodas .....	4
1.2.2. Atsitiktinio generavimo iš intervalo metodas .....	5
2. PROGRAMOS REALIZACIJA IR REZULTATAI .....	6
2.1. Programos kodas .....	6
2.1.1. Dirbtinio neurono modelio realizacija .....	6
2.1.2. Intervalo perėjimas nustatytu žingsniu metodo realizacija .....	7
2.1.3. Atsitiktinio generavimo iš intervalo metodo realizacija.....	8
2.2. Svorio ir poslinkio reikšmių rinkiniai.....	8
3. NELYGYBIŲ SISTEMOS SPRENDIMAS IR ANALIZĖ .....	10
3.1. Nelygybių sistema .....	10
3.2. Grafinis sprendimas .....	10
IŠVADOS .....	12

## **Įvadas**

Užduoties tikslas – išanalizuoti dirbtinio neurono modelio veikimo principus. Ši analizė apima dirbtinio neurono modelio sukūrimą, paiešką tinkamų neurono įeities svorių skirtingomis metodikomis, nagrinėjimą dviejų aktyvacijos funkcijų – slenkstinės ir sigmoidinės, bei įeities duomenų analizę naudojant grafinį būdą.

# 1. Klasifikavimo duomenys ir metodika

## 1.1. Klasifikavimo duomenys

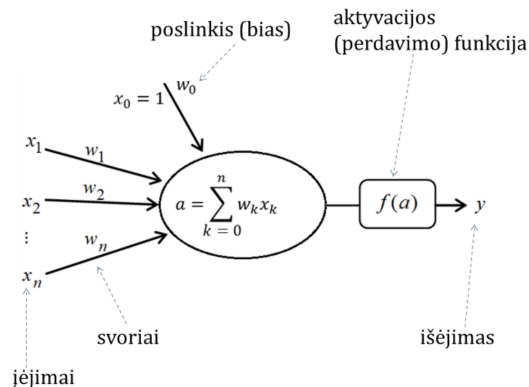
Sukurto dirbtinio neuroninio modelio tikrinimui bei grafinei analizei bus naudojami 1 lentelėje pateikti duomenys. Požymių  $x_1$  ir  $x_2$  reikšmės bus pateikiamos į neurono įėjimus, o neurono rezultatas po aktyvacijos funkcijos lyginamas su  $t$  reikšme.

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

## 1.2. Metodika

Darbo metu reikalinga nustatyti dirbtinio neurono svorių rinkinį pavaizduotą 1 paveikslėlyje žymėjimais:  $w_1, w_2, w_0$ . Svorių rinkinys turi būti toks, kad teisingai klasifikuotų kiekvieną 1 lentelėje pavaizduotą klasės reikšmę iš įeities duomenų  $x_1, x_2$  paskaičiavus neurono reikšmę ir panaudojus aktyvacijos funkciją.



1 pav. Dirbtinio neurono schema

Tinkamiems svorio rinkiniams nustatyti pasirinkti du metodai – intervalo perėjimas nustatytu žingsniu bei atsitiktinis generavimas iš intervalo.

### 1.2.1. Intervalo perėjimas nustatytu žingsniu metodas

Pasirinktas intervalo  $[-1; 1]$  perėjimas su žingsnio dydžiu 0,1. Rezultate reikalingas svorio priskyrimas trims skirtingiems kintamiesiems, todėl naudojami trys lygiagretus intervalų perėjimai ir kiekvieno intervalo išrinkta reikšmė priskiriama atitinkamai  $w_1, w_2, w_0$  reikšmėms.

### 1.2.2. Atsitiktinio generavimo iš intervalo metodas

Reikšmės generuojamos atsitiktinai intervale  $[-1, 1]$  tris kartus per kiekvieną iteracijos žingsnį<sup>1</sup> ir atitinkamai priskiriamos svorių rinkiniui  $(w_1, w_2, w_0)$ .

---

<sup>1</sup>Iteracijos žingsnis – vieno svorių rinkinio sukūrimas ir patikrinimas naudojant klasifikacijos duomenis.

## 2. Programos realizacija ir rezultatai

### 2.1. Programos kodas

Analizė atlikta naudojantis „Python” kalba. Tolesni poskyriai išskirti pagal esmines kodo realizacijos grupes.

#### 2.1.1. Dirbtinio neurono modelio realizacija

Žemiau pateikiama dirbtinio neurono implementacija, aprašanti neurono inicializaciją, aktyvacijos funkcijos pritaikymą bei spėjimo apskaičiavimą.

```
import numpy as np

class Neuron:
    def __init__(self, weights, bias, activation_function):
        self.weights = np.array(weights)
        self.bias = bias
        self.activation_function = activation_function

    def activation(self, x):
        if self.activation_function == 'threshold':
            return 1 if x >= 0 else 0
        elif self.activation_function == 'sigmoid':
            return 1 / (1 + np.exp(-x))
        else:
            raise ValueError("Invalid activation function")

    def predict(self, inputs):
        total_input = np.dot(self.weights, inputs) + self.bias
        output = self.activation(total_input)
        if self.activation_function == 'sigmoid':
            output = round(output)
        return output
```

Klasės „Neuron” metodas „\_\_init\_\_” atlieka inicializacijos darbą iš parametrų „weights” (svorių masyvas  $[w_1, w_2]$ ), „bias” ( $w_0$ ), bei „activation\_function” (galimos reikšmės: „sigmoid” arba „threshold”, atitinkančios sigmoidinę arba slenkstinę aktyvacijos funkcijas).

Klasės „Neuron” metodas „activation” pritaiko sigmoidinę arba slenkstinę aktyvacijos funkciją. Sigmoidinės funkcijos matematinė išraiška:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Slenkstinės funkcijos matematinė išraiška:

$$\theta(x) = \begin{cases} 0 & , \text{ kai } x < 0, \\ 1 & , \text{ kai } x \geq 0 \end{cases}$$

Klasės „Neuron“ metodas „predict“ pagal gautą parametą „inputs“ (neurono įeities duomenų rinkinys  $(x_1, x_2)$ ), atlieką spėjimą naudodamas neurono formulę:

$$a = \sum_{k=0}^n w_k x_k$$

, bei pritaiko aktyvacijos funkciją gauti spėjamą klasifikacijos reikšmę (0 arba 1).

### 2.1.2. Intervalo perėjimas nustatytu žingsniu metodo realizacija

Skyriuje 1.2.1. aprašytos metodikos realizacija, ieškanti tinkamų svorių bei poslinkio iš trijų intervalo perėjimo su numatytu fiksuotu žingsniu.

```
def find_weights_manual(data , function , min_val = -1, max_val =
    1, step = .1):
    for w1 in np.arange(min_val * 10, (max_val + 1) * 10, step *
        10)/10:
        for w2 in np.arange(min_val * 10, (max_val + 1) * 10,
            step * 10)/10:
            for b in np.arange(min_val * 10, (max_val + 1) * 10,
                step * 10)/10:
                neuron = Neuron([w1, w2], b, activation_function=
                    function)
                if all(neuron.predict([x1, x2]) == t for x1, x2,
                    t in data):
                    print(f"Manual weight search. Activation: {
                        function}. Suitable weights w1={w1}, w2={
                            w2}, bias={b}")
                    return
            print(f"Manual weight search. Activation: {function}. No
                suitable weights and bias found. Increase ranges or
                decrease step")
    return
```

Metodas „find\_weights\_manual“ pagal įeities duomenis „data“ (duomenys, aprašyti 1 lentelėje), „function“ (aktyvacijos funkcijos pasirinkimas), „min\_val“ (apatinė intervalo dalis), „max\_val“ (viršutinė intervalo dalis), „step“ (žingsnis, per kurį parenkamos naujos reikšmės), generuoja

galimų svorių bei poslinkio rinkinius bei patikriną ar gautas rinkinys atitinką neurono tikslą – tinkamai klasifikuoti įeities duomenis.

### 2.1.3. Atsitiktinio generavimo iš intervalo metodo realizacija

Skyriuje 1.2.2. aprašytos metodikos realizacija, ieškanti tinkamų svorių bei poslinkio atsitiktiniu būdu.

```
def find_weights_random(data, function, trials=1000, min_val =
    -1, max_val = 1):
    for _ in range(trials):
        w1 = round(random.uniform(min_val, max_val), 1)
        w2 = round(random.uniform(min_val, max_val), 1)
        b = round(random.uniform(min_val, max_val), 1)
        neuron = Neuron([w1, w2], b, activation_function=function)
        if all(neuron.predict([x1, x2]) == t for x1, x2, t in
            data):
            print(f"Random weight search. Activation: {function}.
                Suitable weights w1={w1}, w2={w2}, bias={b}")
            return
    print(f"Random weight search. Activation: {function}. No
        suitable weights and bias found. Increase trials")
    return
```

Metodas „find\_weights\_random“ pagal įeities duomenis „data“ (duomenys, aprašyti 1 lentelėje), „function“ (aktyvacijos funkcijos pasirinkimas), „trials“ (maksimalus bandymų skaičius), „min\_val“ (apatinė intervalo dalis), „max\_val“ (viršutinė intervalo dalis), generuoja galimų svorių bei poslinkio rinkinius bei patikriną ar gautas rinkinys atitinką neurono tikslą – tinkamai klasifikuoti įeities duomenis.

## 2.2. Svorio ir poslinkio reikšmių rinkiniai

Atlikus eksperimentus, viso surinkta 20 svorio bei poslinkio duomenų rinkinių (su pasikartojimais), su kuriais dirbtinis neurono modelis sėkmingai klasifikuoja duomenis (rinkinių yra ir daugiau, tačiau, išrinkti pirmieji rasti). Toliau lentelėse pateikiama iš skirtingų metodikų ir su skirtingomis aktyvacijos funkcijomis gauti rinkiniai.



2 lentelė. Intervalo perėjimo metodu su slenkstine aktyvacija

$w_0$	$w_1$	$w_2$
0,2	0	-0,1
0,3	0	-0,2
0,3	0	-0,1
0,3	0,1	-0,1
0,4	-0,1	-0,2

3 lentelė. Intervalo perėjimo metodu su sigmoidine aktyvacija

$w_0$	$w_1$	$w_2$
0,2	0	-0,1
0,3	0	-0,2
0,3	0	-0,1
0,3	0,1	-0,1
0,4	-0,1	-0,2

4 lentelė. Atsitiktiniu metodu su slenkstine aktyvacija

$w_0$	$w_1$	$w_2$
0,6	-0,1	-0,2
0,8	0	-0,6
0,4	0,1	-0,1
0,9	0,3	-0,1
0,4	0	-0,2

5 lentelė. Atsitiktiniu metodu su sigmoidine aktyvacija

$w_0$	$w_1$	$w_2$
0,8	-0,1	-0,5
0,9	0,4	-0,3
1	0,6	-0,2
0,6	-0,1	-0,3
0,8	0,4	-0,3

### 3. Nelygybių sistemos sprendimas ir analizė

#### 3.1. Nelygybių sistema

Tinkamus svorius bei poslinkį galime rasti ir kitu būdu – susidarant ir sprendžiant lygčių sistemą. Naudojant duotus įeities duomenis galima susidaryti lygčių sistemą, kurios sprendiniai būtų svoriai, tinkami modeliui teisingai klasifikuoti reikšmes.

Tarkime, naudotume slenkstinę aktyvacijos funkciją, tuomet pagal 1 lentelės duomenis, žinotume, kad neurono paskaičiuota reikšmė pirmai ir antrai eilutei turėtų būti  $<0$ , o likusioms  $\geq 0$ , pagal slenkstinę funkciją:

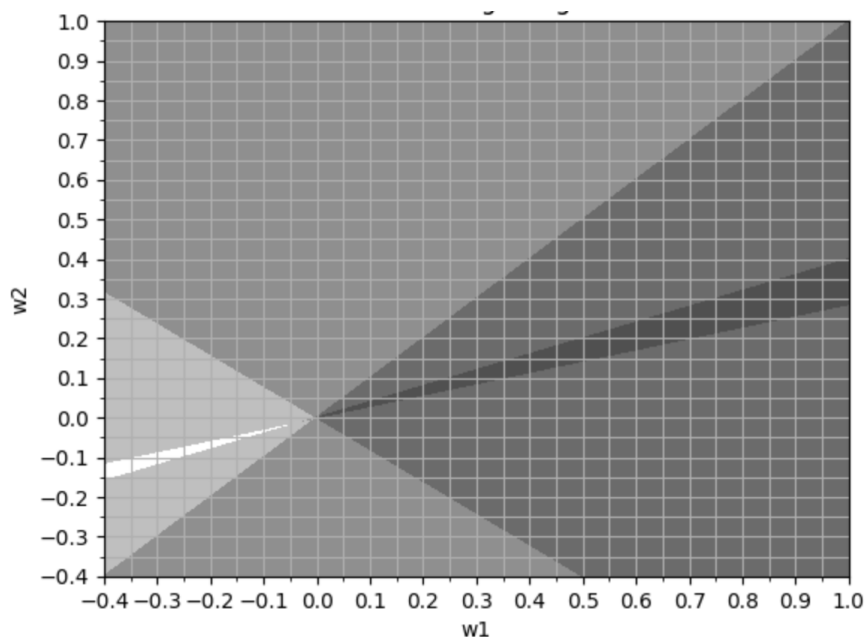
$$\theta(x) = \begin{cases} 0 & , \text{ kai } x < 0, \\ 1 & , \text{ kai } x \geq 0 \end{cases}$$

Taip pat, dėl paprastumo, paimkime vieną iš argumentų kaip konstantą, tam, kad sumažėtų lygčių sistemos argumentų, bei sistemą galėtume spręsti dviejų dimensijų grafike. Tarkime, kad  $w_0 = b = 0$ , tuomet gauname tokią lygčių sistemą:

$$\begin{cases} -0,2 \cdot w_1 + 0,5 \cdot w_2 < 0 \\ 0,2 \cdot w_1 - 0,7 \cdot w_2 < 0 \\ 0,8 \cdot w_1 - 0,8 \cdot w_2 \geq 0 \\ 0,8 \cdot w_1 + w_2 \geq 0 \end{cases}$$

#### 3.2. Grafinis sprendimas

Nubraižius 3.1. skyriuje gautą lygčių sistemą ir nuspalvinus nelygybės sprendinių ribojamus plotus gauname grafiką, parodyta 2 paveikslėlyje.



2 pav. Nelygybių sistemos sprendiniai

Grafike, tamsiausiai pažymėtas plotas yra lygčių sistemos sprendinių regionas. Jame esančios  $w_1$  ir  $w_2$  reikšmės tenkina visas lygčių sistemos nelygybes.

Patikrinimui, paimkime vieną tašką iš pavaizduoto regiono, pvz. ,  $w_1 = 0,3$  ir  $w_2 = 0,1$  ir įstačius reikšmes, į lygčių sistemą gauname:

$$\begin{cases} -0,2 \cdot 0,3 + 0,5 \cdot 0,1 < 0 \\ 0,2 \cdot 0,3 - 0,7 \cdot 0,1 < 0 \\ 0,8 \cdot 0,3 - 0,8 \cdot 0,1 \geq 0 \\ 0,8 \cdot 0,3 + 0,1 \geq 0 \end{cases} \implies \begin{cases} -0,01 < 0 \\ -0,01 < 0 \\ 0,16 \geq 0 \\ 0,34 \geq 0 \end{cases}$$

## **Išvados**

1. Buvo sėkmingai sukurtas ir išbandytas dirbtinio neurono modelio pritaikymas ant duotų duomenų.
2. Abi svorio radimo metodikos – intervalo perėjimo nustatytu žingsniu ir atsitiktinio generavimo, tinkamos svoriams ir poslinkiui rasti, kurie užtikrintų tinkamą modelio klasifikavimo rezultatą su duotais duomenimis.
3. Nelygybių sistemos grafinė analizė parodė, kad kartais reikiamus neurono svorius bei poslinkį gali padėti surasti ir primityvus būdai.
4. Šiuo atveju, aktyvacijos funkcijos pasirinkimas neturėjo įtakos modelio galimybėms teisingai klasifikuoti binarinę klasę, kadangi abiem atvejais buvo pasiektas 100% tikslumas.