

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMOS

**Skaitmeninis intelektas ir sprendimų priėmimas**  
**Dirbtinis neuronas**

**Darbą atliko:**  
Pijus Zlatkus

Vilnius  
2023

## **Turiny**

<b>1. Užduoties tikslas.....</b>	<b>3</b>
<b>2. Klasifikavimo duomenys ir klasė.....</b>	<b>3</b>
<b>3. Dirbtinio neurono realizacija.....</b>	<b>3</b>
<b>4. Nelygybių sistemos sprendimas .....</b>	<b>7</b>
<b>5. Šaltiniai .....</b>	<b>8</b>

## 1. Užduoties tikslas

Šios užduoties tikslas yra ištirti ir išanalizuoti kaip veikia dirbtinio neurono – perceptrono modelis. Jo ištyrimui reikalinga programa, kuri realizuotų jo veikimą ir leistų pasirinkti aktyvacijos funkciją: slenkstinę arba sigmoidinę. Kartu su programa taip pat tikslas yra išspręsti uždavinį grafiškai.

## 2. Klasifikavimo duomenys ir klasė

Dirbtinio neurono realizacijai atlikti bus naudojami klasifikavimo duomenys, pagal kuriuos bus nustatomos svorių ir poslinkio reikšmės.

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

Error! Reference source not found. pateiktos požymių reikšmės  $x_1$  ir  $x_2$  atitinka įėjtis kurios bus naudojamos klasifikavimo uždaviniui spręsti. Kadangi yra tik du požymiai, tai dirbtinis neuronas turės tik du įėjimus ir poslinkį.

## 3. Dirbtinio neurono realizacija

Šiai užduočiai yra realizuoti vienas dirbtinis neuronas su skirtingomis pasirenkamomis aktyvacijos funkcijomis: slenkstine ir sigmoidine. Programos rašymui pasirinkta naudoti Python kalba, kuri jau turi daug naudingų bibliotekų skirtų dirbti su matematiniais duomenimis ir neuroniniais tinklais.

Programose naudojami kintamieji  $w_1$ ,  $w_2$ , reiškia svorius, o  $w_0$  reiškia poslinkį. Šie kintamieji yra naudojami atrasti tokias reikšmes, kad *Error! Reference source not found.* pateikti duomenys būtų tinkamai klasifikuoti – priskirti klasei 0 arba 1. Naudojant dirbtinį neuroną pasirinktame intervale tam tikru žingsniu perrinkamos svorių ir poslinkio reikšmės ir ieškomi pirmi 5 svorių ir poslinkio rinkiniai su kuriais būtų galima teisingai klasifikuoti visus pateiktus duomenis lentelėje.

```
[1] import numpy as np

class ArtificialNeuron:
    # Klasės konstruktorius priskiriantis svorius ir aktyvacijos funkciją
    def __init__(self, w0, w1, w2, activation_fn):
        self.w0 = w0
        self.w1 = w1
        self.w2 = w2
        self.activation_fn = activation_fn

    # Apskaičiuojame prognozę naudojant x1 ir x2 įvestis
    def predict(self, x1, x2):
        z = self.w1 * x1 + self.w2 * x2 + self.w0
        prediction = self.activation_fn(z)
        if self.activation_fn == self.sigmoid_activation:
            return round(prediction)
        return prediction

    # Aktyvacijos funkcijos, kurios pažymėtos kaip statinės (t. y. galima pasiekti nesukuriant objekto)
    @staticmethod
    def threshold_activation(a):
        return 1 if a >= 0 else 0

    @staticmethod
    def sigmoid_activation(a):
        return 1 / (1 + np.exp(-a))
```

**1 pav.** Dirbtinio neurono realizacija.

Kode esančiame **1 pav.** Dirbtinio neurono realizacija. pateikta dirbtinio neurono realizacija naudojant klases, kurioje mes laikome reikalingą informaciją ir funkcijas, reikalingas dirbti su dirbtiniu neuronu ir atlikti reikalingus veiksmus kaip prognozavimas ar aktyvacijos funkcijos skaičiavimas. Dirbti su skaičiais buvo panaudota *numpy* biblioteka. Kadangi prieš darant prognozes su dirbtiniu neuronu turi būti galimybė pasirinkti vieną iš aktyvacijos funkcijų, tai joms pritaikyta *@staticmethod* anotacija, kuri nurodo kad šios funkcijos gali būti iškviestos ir nesukuriant klasės objekto.

```

# Įvertinama, ar prognozuotos išvestys atitinka tikėtinas išvestis
def evaluate_model(weights, inputs, expected_outputs, activation_fn):
    neuron = ArtificialNeuron(weights[0], weights[1], weights[2], activation_fn)
    predictions = [neuron.predict(x1, x2) for [x1, x2] in inputs]
    return np.array_equal(predictions, expected_outputs)

def find_weights(input_x, output_class, activation_fn):
    best_weights = []

    # Nustatomi svorių ir poslinkio intervalai, per kuriuos bus ieškoma geriausių kombinacijų
    w0_range = np.arange(-0.6, 1, 0.1)
    w1_range = np.arange(-1, 1, 0.1)
    w2_range = np.arange(-1, 1, 0.1)

    # Ieškoma geriausių svorių kombinacijų, kurios leidžia modeliui teisingai atpažinti duomenis

    for w0 in w0_range:
        for w1 in w1_range:
            for w2 in w2_range:
                if evaluate_model([w0, w1, w2], input_x, output_class, activation_fn):
                    best_weights.append((w0, w1, w2))
                    if len(best_weights) == 5:
                        return best_weights
    return best_weights

```

**2 pav.** Pagalbinės funkcijos.

**2 pav.** Pagalbinės funkcijos. esančios pagalbinės funkcijos padeda įvertinti dirbtinio neurono modelį ir rasti geriausias 5 svorių ir poslinkio kombinacijas su kuriomis būtų galima teisingai išspręsti klasifikavimo uždavinį iš .

```

for activation_function in [ArtificialNeuron.threshold_activation, ArtificialNeuron.sigmoid_activation]:
    best_weights = find_weights(input_x, output_class, activation_function)

    print(f"\nBest weights and bias with {activation_function.__name__}:")
    print("w0    w1    w2")
    for (w0, w1, w2) in best_weights:
        print(f"{w0:.2f} {w1:.2f} {w2:.2f}")

```

**3 pav.** Svių ir poslinkio kombinacijų ieškojimas.

Kodas esantis **3 pav.** Svių ir poslinkio kombinacijų ieškojimas. naudojamas atspausdinti rastus geriausius svorius ir poslinkius., ir pateikiantis juos lentelės formatu.

**2 lentelė.** Geriausi svoriai su slenkstine aktyvacijos funkcija.

<b>w<sub>0</sub></b>	<b>w<sub>1</sub></b>	<b>w<sub>2</sub></b>
-0,70	0,90	0
-0,6	0,8	0
-0,6	0,9	-0,1
-0,6	0,9	0
-0,6	0,9	1,0

**2 lentelė.** Geriausi svoriai su slenkstine aktyvacijos funkcija. atvaizduojami geriausi svoriai ir poslinkis pritaikant slenkstinę aktyvacijos funkciją. Pagal pateiktus duomenis galima pastebėti imant visoms reikšmėms skaičių intervalą -1–1 tai geriausius svorius, su kuriais galima teisingai suklasifikuoti duomenis, priskyre su -0,7 ir -0,6 poslinkiais. Galima pastebėti kad tos reikšmės mums užtenka. Ši

**3 lentelė.** Geriausi svoriai su sigmoidine aktyvacijos funkcija.

<b>w<sub>0</sub></b>	<b>w<sub>1</sub></b>	<b>w<sub>2</sub></b>
-0,70	0,90	0
-0,6	0,8	0
-0,6	0,9	-0,1
-0,6	0,9	0
-0,6	0,9	1,0

**3 lentelė.** Geriausi svoriai su sigmoidine aktyvacijos funkcija. pateikti duomenys naudojant sigmoidinę funkciją taip pat buvo paimami intervale -1–1. Nesunkiai galima pastebėti kad visi duomenys yra šiuo atveju vienodi su **2 lentelė.** Geriausi svoriai su slenkstine aktyvacijos funkcija. pateiktais duomenimis.

Iš šių testavimo duomenų galima spręsti, kad sigmoidinė ir slenkstinės funkcijos yra labai panašios ir didelės įtakos nedaro. Tačiau, negalime to teigti netestuoiant šių funkcijų su didesne duomenų aibe.

## 4. Nelygybių sistemos sprendimas

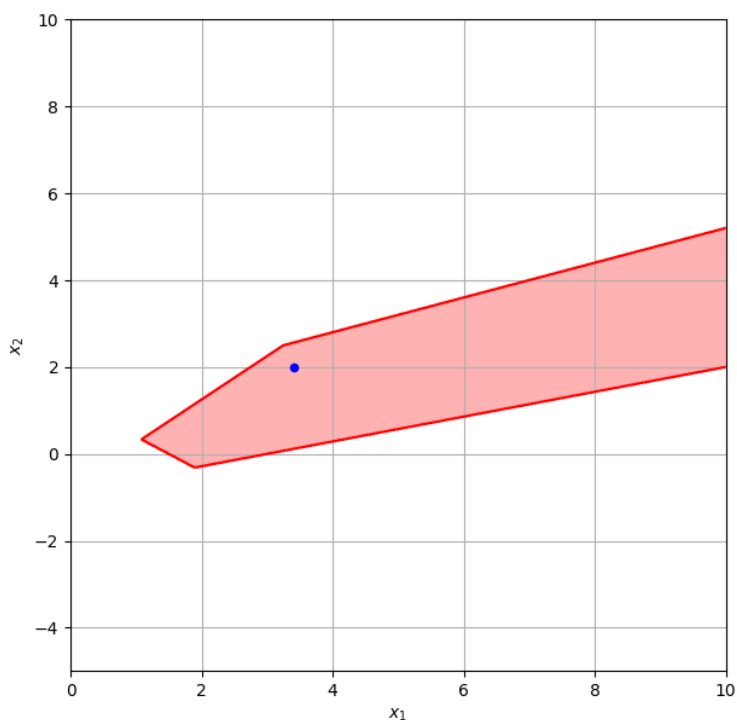
Kaip praeitame skyriuje matėme, kad naudojant slenkstinę aktyvacijos funkciją paimant poslinkį lygų  $-0,6$  jau galima parinkti tinkamus svorius prasidedančius apytiksliai nuo 1.

Kad susidarytume geresnį vaizdą ir pamatytume tai grafiškai mums reikia pirmiausia išspręsti nelygybių sistemą.

Šiuo atveju  $b = -slenkstis$ , nes poslinkį galima sutapatinti su neigiama slenksčio reikšme.

$$\begin{cases} -0,2x_1 + 0,5x_2 + b < 0 \\ 0,2x_1 - 0,7x_2 + b < 0 \\ 0,8x_1 - 0,8x_2 + b \geq 0 \\ 0,8x_1 + x_2 + b \geq 0 \end{cases} \rightarrow \begin{cases} x_2 < \frac{0,2x - b}{0,5} \\ x_2 > \frac{-0,2x - b}{-0,7} \\ x_2 \leq \frac{-0,8x - b}{-0,8} \\ x_2 \geq -0,8x - b \end{cases}$$

Kad galėtume nelygybių sistemą atvaizduoti grafiškai reikia pirmiausia ją pertvarkyti taip kad vienoje nelygybės pusėje liktu tik svoris  $x_2$ . Pertvarkius šią lygčių sistemą galima ją išspręsti grafiškai. Ją atvaizduoti buvo panaudota Python programavimo kalbos *matplotlib* biblioteka.



4 pav. Nelygybių sistemos sprendinys

Pagal **4 pav.** Nelygybių sistemos sprendinys pavaizduotą grafiškai atvaizduoją nelygybių sistemos sprendinį galima pastebėti, kad raudonai nuspalvintame plote galima parinkti svorius kurie galėtų teisingai suklasifikuoti anksčiau pateiktus duomenis.

Kad patikrintume ar tikrai gauti sprendiniai yra nelygybių sistemos sprendiniai, reikia paimti vieną tašką iš raudonai pažymėto ploto. Šiam atvejui paimsime mėlyną tašką atvaizduotą sprendinį, kuris lygus  $x_1 = 3,4$ ;  $x_2 = 2,0$ .

$$\begin{cases} -0,2 \cdot 3,4 + 0,5 \cdot 2 - 0,6 < 0 \\ 0,2 \cdot 3,4 - 0,7 \cdot 2 - 0,6 < 0 \\ 0,8 \cdot 3,4 - 0,8 \cdot 2 - 0,6 \geq 0 \\ 0,8 \cdot 3,4 + 2 - 0,6 \geq 0 \end{cases} \rightarrow \begin{cases} -0,28 < 0 \\ -1,32 < 0 \\ 0,52 \geq 0 \\ 4,12 \geq 0 \end{cases}$$

Patikrinus sprendinį galime įsitikinti, kad sprendinys paimtas iš grafiniu būdu gautų sprendinių aibės yra nelygybių sistemos sprendinys.

## 5. Šaltiniai

- Skaitmeninis intelektas ir sprendimų priėmimo skaidrės „Dirbtinis neuronas – perceptronas“
- [Programos kodas](#)