

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS

**Praktinė užduotis Nr. 5**  
**Saviorganizuojantys neuroniniai tinklai**

Atliko:

Programų sistemų 4 k. 1 gr. stud. Rokas Petrauskas

VILNIUS, 2023

## Turiny

Tikslas .....	3
Naudojami duomenys .....	3
Programos kodas .....	3
Rezultatai .....	4
Išvados .....	5

## Tikslas

Suprogramuoti saviorganizuojančio neuroninio tinklo (žemėlapiu, SOM) mokymo algoritmą, apmokyti jį naudojant pasirinktus duomenis.

## Naudojami duomenys

Duomenims analizuoti naudoju „Iris“ duomenų rinkinį: <https://archive.ics.uci.edu/dataset/53/iris>

Duomenų rinkinį sudaro 150 įrašų po 4 požymius ir klasės identifikatorių. Analizei naudoju visus 4 požymius – sepalwidth, sepalwidth, petalwidth, petalwidth. Yra po 50 visų 3 klasių įrašų, todėl visus 150 įrašų ir naudoju.

## Programos kodas

```
import numpy as np
from minisom import MiniSom
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder

# Duomenų skaitymas ir paruošimas
data = pd.read_csv('data\iris.data', header=None)
x = data.iloc[:, :-1].values
y = data.iloc[:, -1].values

le = LabelEncoder()
y_encoded = le.fit_transform(y)

# SOM parametrai
som_size = 10
som = MiniSom(som_size, som_size, len(x[0]), sigma=1.0, learning_rate=0.5)
som.train_random(x, 1000)

# SOM atvaizdavimas
plt.figure(figsize=(som_size, som_size))
for i, xx in enumerate(x):
    w = som.winner(xx)
    plt.text(w[0], w[1], str(y_encoded[i]), color=plt.cm.rainbow(y_encoded[i] /
len(set(y_encoded))), ha='center', va='center',
            bbox=dict(facecolor='white', alpha=0.5, lw=0))

plt.axis([0, som_size, 0, som_size])
plt.show()

# Spausdinami SOM neuronų svoriai, neuronai nugalėtojai ir jiems priskirti duomenų objektai
winners = []
for i, xx in enumerate(x):
    w = som.winner(xx)
    winners.append(w)
```

```

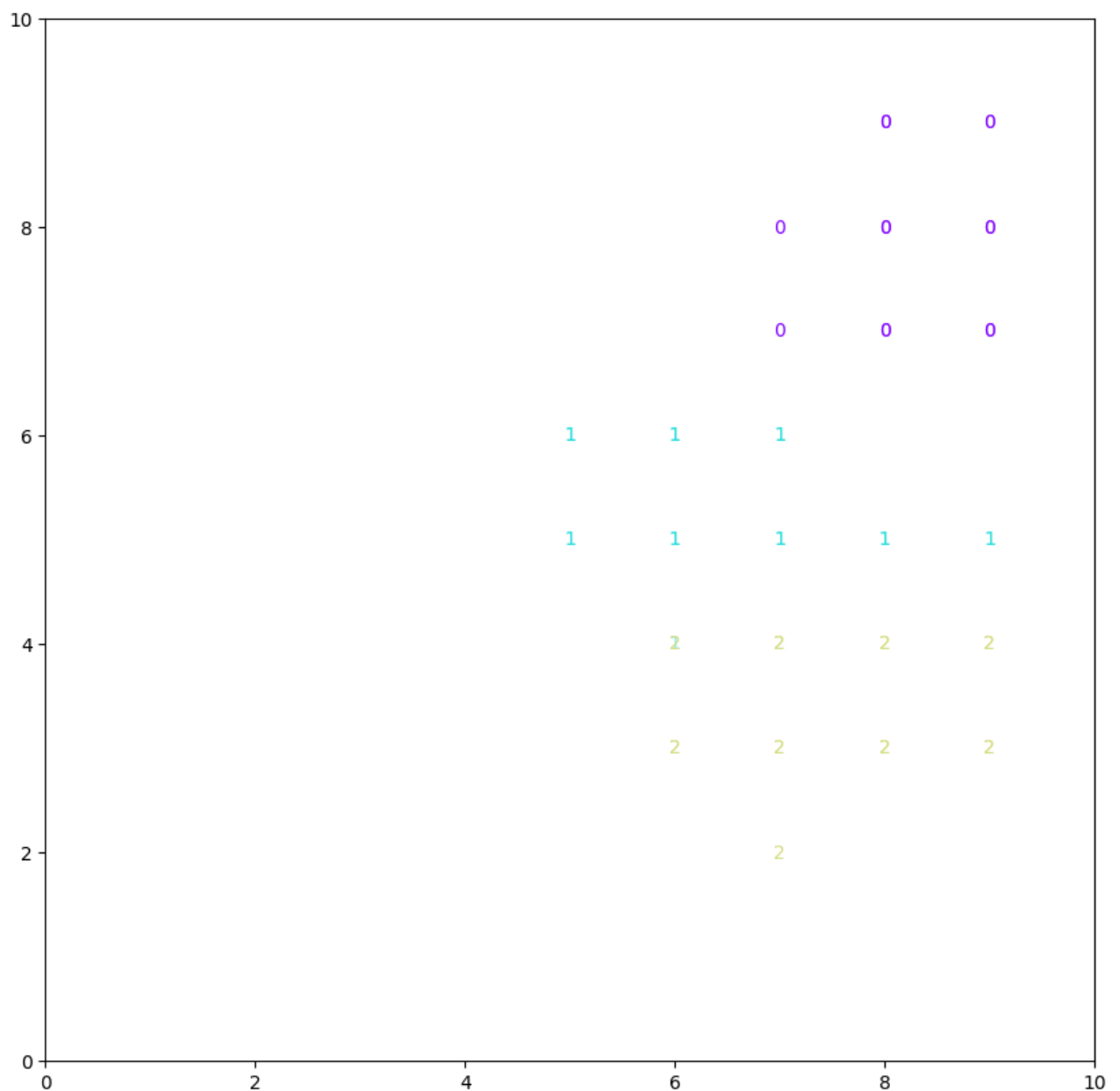
    print(f'Duomenų objekto {i+1} numeris: {w}, neuronas-nugalėtojas:
{som.get_weights()[w[0], w[1]]}')

# Atspausdinama kvantavimo paklaidos reikšmė
print(f'Kvantavimo paklaidos reikšmė: {som.quantization_error(x)}')

```

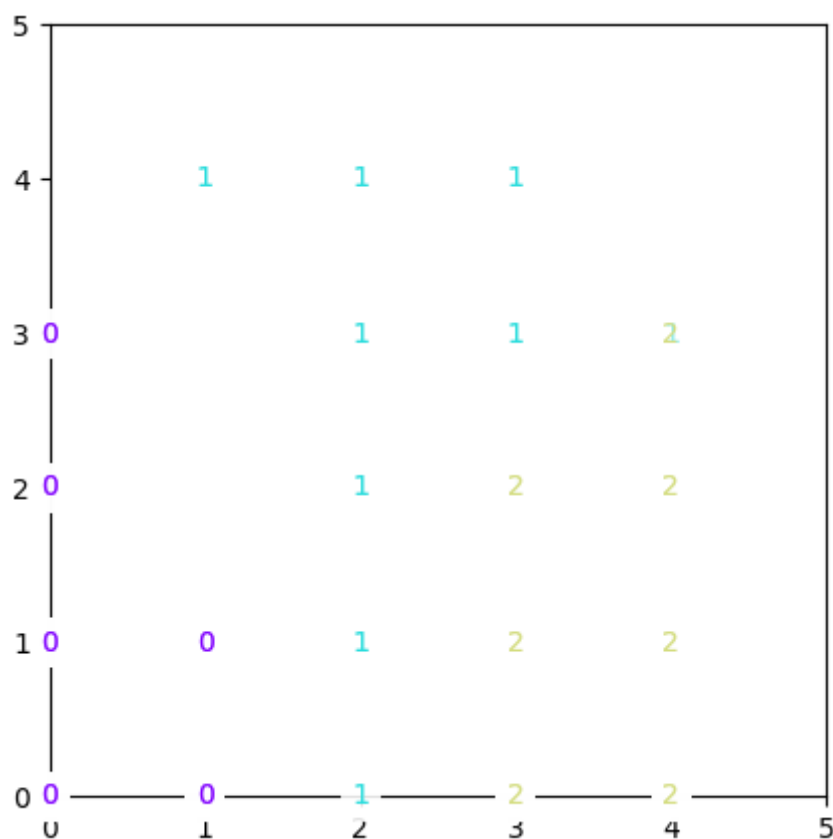
## Rezultatai

Pirmame paveikslėlyje matomas SOM žemėlapis, kai jo dydis 10x10. Naudota 500 epochų, mokymo greitis 0,5. Kvantavimo paklaidos reikšmė: 0,3049016819296261.



**1 pav.** 10x10 SOM žemėlapis.

Antrame paveikslėlyje matomas SOM žemėlapis, kai jo dydis 5x5. Naudota 500 epochų, mokymo greitis 0,5. Kvantavimo paklaidos reikšmė: 0,3203535662114577.



**2 pav.** 5x5 SOM žemėlapis.

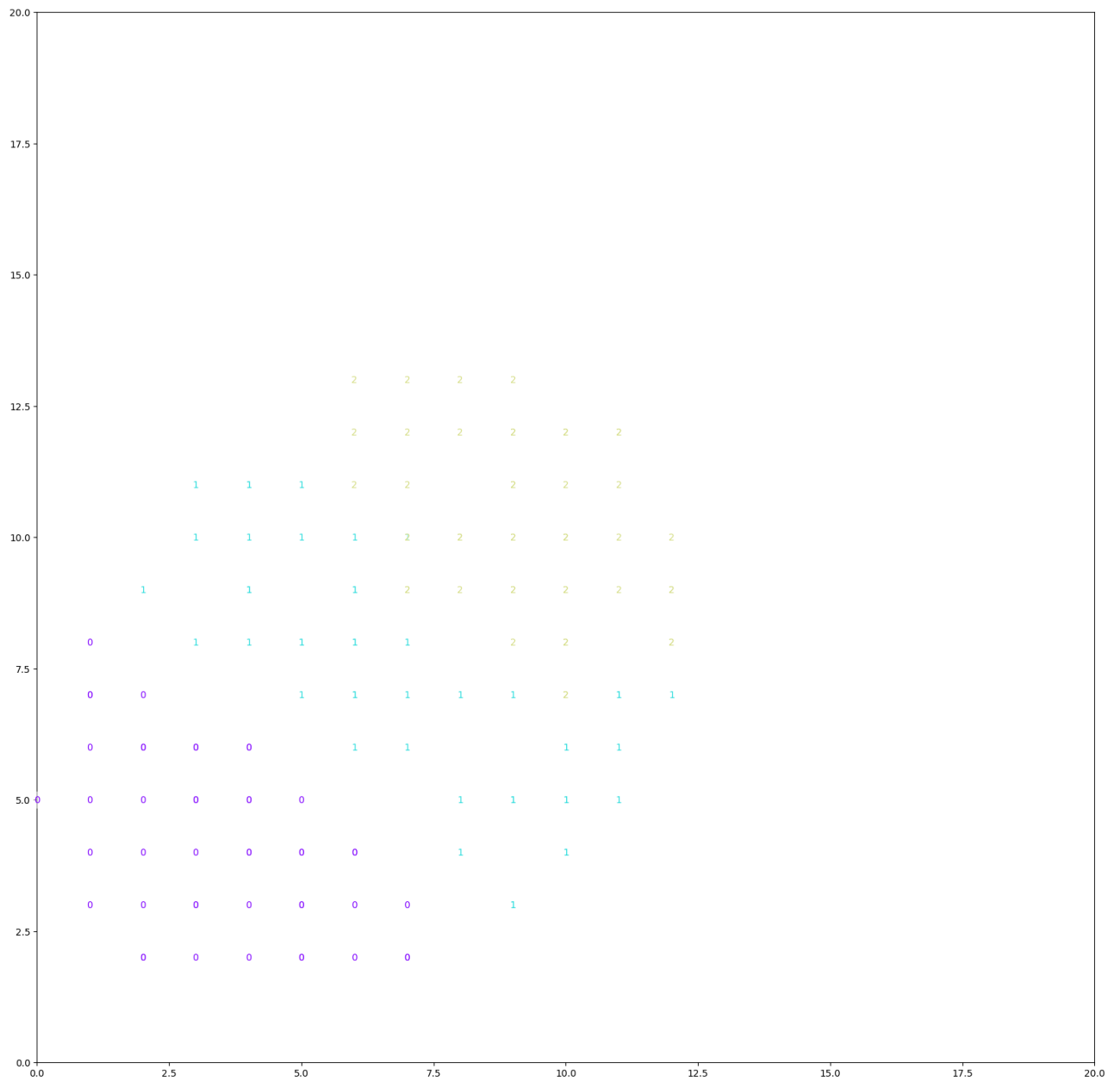
## Išvados

Remiantis pirmu ir antru paveikslėliu, matyti, kad abejais atvejais 0 klasė (Iris-setosa) atsiskiria labiau, tuo tarpu 1 (Iris-versicolor) ir 2 klasė (Iris-virginica) klasterizuojamos arčiau viena kitos, todėl galima teigti, kad 1 ir 2 klasės tarpusavyje panašesnės atributų atžvilgiu. Taip pat visais atvejais 0 ir 2 klasė neturi sąlyčio taškų, todėl galima suprasti, kad tarpusavyje šios 2 klasės yra nepanašiausios iš visų 3.

Modelis buvo leistas bandant skirtingą epochų skaičių. Kai epochų skaičius 100, taškų kiekis per mažas suprasti tendencijai, kai epochų skaičius 1000, taškų per daug, kadangi užpildomas visas SOM, tokiomis atvejais galima didinti SOM žemėlapij, tačiau jį analizuoti darosi sunkiau. Priešingai, mažas SOM leidžia lengviau analizuoti tendencijas, tačiau dėl mažos erdvės, rezultatai gali prarasti kai kurias tendencijas.

Išbandžius skirtingus mokymo greičius matyti, kad ši parametrai didinant, matomas didesnis duomenų kiekis, panašiai, kaip ir didinant epochų skaičių. Kai SOM didelis, prasminga didinti ir mokymo greitį, taip pat pastebėta, kad esant didesniam mokymo greičiui, šiuo konkrečiu atveju kvantavimo paklaidos reikšmė artesnė 0.

Pritaikęs šias išvadas modelį leidau su tokiais parametrais: SOM dydis – 20x20, epochų sk. – 1500, mokymosi greitis – 1. Gaunamas 3 paveikslėlyje matomas SOM, tendencijos čia matomos labai aiškiai, tuo tarpu kvantavimo paklaidos reikšmė 0,09378375896300448.



**3 pav.** 20x20 SOM žemėlapis.