

2. Rad s bibliotekama Numpy i Matplotlib

2.1 Cilj Vježbe

Upoznati se s načinom korištenja Python biblioteka Numpy i Matplotlib.

2.2 Teorijska pozadina

Za Python je napisan velik broj biblioteka koje olakšavaju programiranje. U ovoj vježbi studenti se upoznaju s `numpy` bibliotekom koja olakšava numeričke proračune i `matplotlib` bibliotekom za grafički prikaz podataka.

2.2.1 Numpy

Numpy¹ je open source Python biblioteka za numeričke proračune. Sadrži niz brzih unaprijed prevedenih funkcija za različite numeričke rutine. Ovo je važno kada se razmatraju problemi strojnog učenja u kojima se provode različite optimizacije na temelju raspoloživih podataka. Prednosti korištenja Numpy biblioteke u odnosu na osnovni Python i liste su efikasno (brzo) matrično računanje te efikasna implementacija polja (engl. *array*). Osnovni objekt Numpy biblioteke je je višedimenzionalno polje `ndarray` za pohranu podataka koji su istog tipa (uobičajeno cijelobrojni brojevi ili brojevi s pomičnim zarezom). Koristi se za predstavljanje vektora (1D polje), matrica (2D polja) ili višedimenzionalnih polja. Naredbom `import numpy` učitava se numpy biblioteka. Tada se raspoloživim funkcijama pristupa na način `numpy.ime_funkcije`. Stoga se često zbog kraćeg zapisa koristi *alias*:

```
import numpy as np  
pa se metodama pristupa na način np.ime_funkcije.
```

Numpy polje definira se pozivanjem funkcije `array`. Ovaj objekt tada sadrži sljedeće atribute:

- `ndim` - broj dimenzija.
- `shape` - dimenzija polja. Za matricu dimenzija NxM, shape attribute je tuple (N, M).

¹numpy.org

- `size` - ukupni broj elemenata u polju.
- `dtype` - tip elemenata u polju.

Elementi numpy polja indeksirani su pozitivnim cijelim brojevima. Iz polja je moguće izdvajati dijelove polja pomoću operatora : na isti način kao i kod Python lista. Nadalje, postoje funkcije za kreiranje polja koje sadrži nule (`zeros`), jedinice (`ones`), konstantu (`full`), jediničnu matricu (`eye`) i sl. Neki primjeri korištenja numpy biblioteke dani su u primjeru 2.1.

■ Primjer 2.1

```
import numpy as np

a = np.array([6, 2, 9])      #napravi polje od tri elementa
print(type(a))              #prikazi tip polja
print(a.shape)               #koliko redaka ima vektor
print(a[0], a[1], a[2])      #prikazi prvi, drugi i treci element
a[1] = 5                    #promjeni vrijednost polja na drugom mjestu
print(a)                     #prikazi cijeli a
print(a[1:2])               #izdvajanje
print(a[1:-1])              #izdvajanje

b = np.array([[3,7,1],[4,5,6]]) #napravi 2 dimenzionalno polje (matricu)
print(b.shape)                #ispisi dimenzije polja
print(b)                      #ispisi cijelo polje b
print(b[0, 2], b[0, 1], b[1, 1]) #ispisi neke elemente polja
print(b[0:2,0:1])            #izdvajanje
print(b[:,0])                 #izdvajanje

c = np.zeros((4,2))          #polje sa svim elementima jednakim 0
d = np.ones((3,2))           #polje sa svim elementima jednakim 1
e = np.full((1,2),5)         #polje sa svim elementima jednakim 5
f = np.eye(2)                #jedinicna matrica 2x2
g = np.array([1, 2, 3], np.float32)
duljina = len(g)
print(duljina)
h = g.tolist()
print(h)
c = g.transpose()
print(g)
np.concatenate((a, g,))
```

Unutar Numpy biblioteke dostupne su matematičke operacije s numpy poljima poput zbrajanja, množenja i ostalih uobičajenih matematičkih operacija, zatim funkcija za sumiranje polja, traženje maksimalne/minimalne vrijednosti unutar polja, sortiranje i sl. kao što je prikazano primjerom 2.2

■ Primjer 2.2

```
import numpy as np

a = np.array([3,1,5], float)
b = np.array([2,4,8], float)
print(a+b)
print(a-b)
print(a*b)
print(a/b)

print(a.min())
```

```

print(a.argmin())
print(a.maxn())
print(a.argmax())
print(a.sum())
print(a.mean())

print(np.mean(a))
print(np.max(a))
print(np.sum(a))

a.sort()
print(a)

```

Unutar Numpy biblioteke nalaze se osnovne „statističke“ funkcije poput srednje vrijednosti, varijance i standardne devijacije. Također je na raspolaganju generator slučajnih brojeva (različite distribucije). Neke od funkcija dane su u primjeru 2.3

■ Primjer 2.3

```

import numpy as np

np.random.seed(56)          #postavi seed generatora brojeva
rNumbers = np.random.rand(10)    #generiraj 10 slučajnih brojeva
print(rNumbers)
print(rNumbers.mean())

```

2.2.2 Matplotlib

Matplotlib² biblioteka sadrži različite funkcije za kreiranje statičkih i dinamičkih vizualizacija. `matplotlib.pyplot` je sučelje koje omogućava izradu različitih prikaza. Ovo sučelje kreira sliku (engl. *figure*) koje se ujedno ponaša kao i grafičko sučelje. Primjer crtanja sinus i kosinus funkcije u rasponu od 0 do 6 dan je u primjeru 2.4 i na slici 2.1. Funkcija `.plt()` prima dva numpy polja (vrijednosti x i y osi) te spaja točke ravnim linijama. Dodatnim funkcijama moguće je označiti osi, dati naslov slici i sl.

■ Primjer 2.4 Vizualizacija

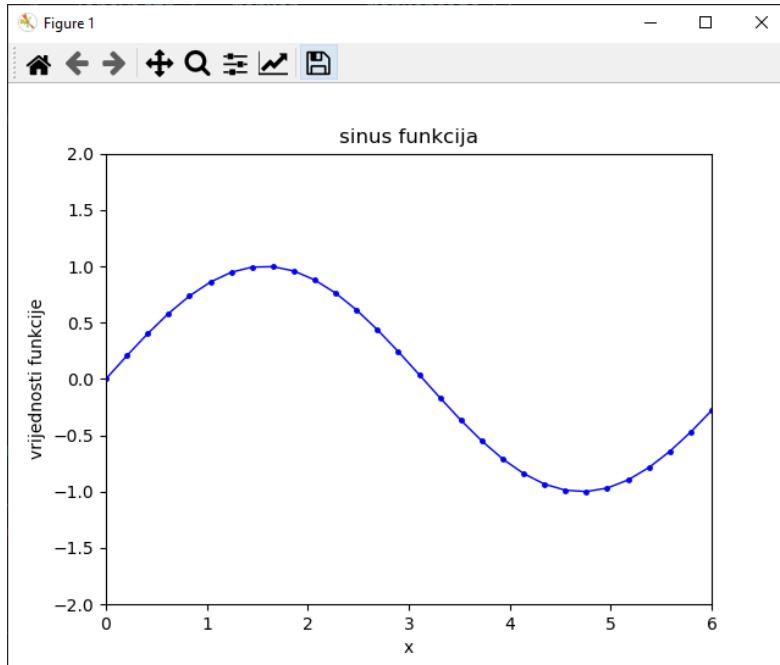
```

import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 6, num=30)
y = np.sin(x)
plt.plot(x, y, 'b', linewidth=1, marker=".", markersize=5)
plt.axis([0, 6, -2, 2])
plt.xlabel('x')
plt.ylabel('vrijednosti funkcije')
plt.title('sinus funkcija')
plt.show()

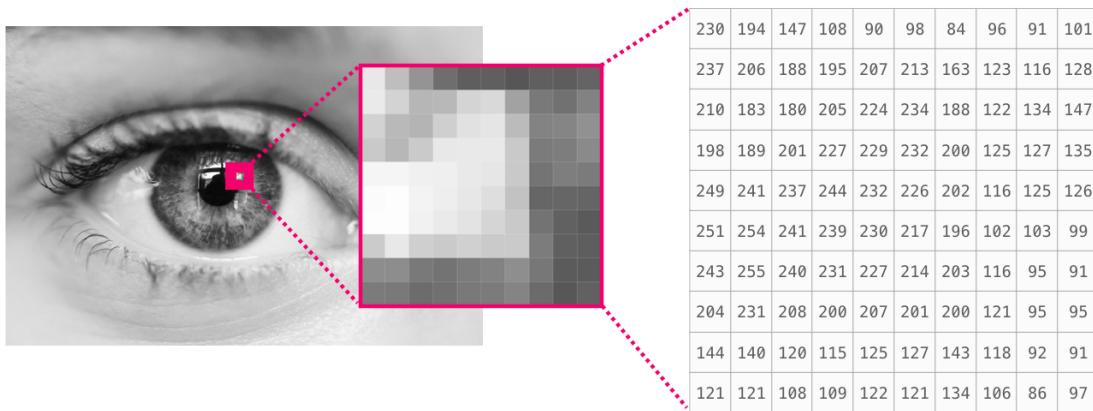
```

²matplotlib.org



Slika 2.1: matplotlib.pyplot slika

Digitalna slika sastoji se od osnovnih elemenata koji se nazivaju pikseli. Digitalna slika se u Pythonu predstavlja numpy poljem pri čemu svaki element numpy polja odgovara jednom pikselu na slici. Vrijednosti elemenata se kreću od 0 (bijela boja) do 255 (crna boja) kao što je prikazano na slici 2.2.



Slika 2.2: Primjer slike u sivim tonovima i način predstavljanja slike kao 2D polja.

Na sličan način se u memoriju računala pohranjuju slike u boji. Najčešći način zapisivanja slike u boji je u RGB formatu. Tada je slika u memoriji prikazana s tri kanala (matrice) pri čemu prva matrica predstavlja udio crvene, druga matrica udio zelene, a treća matrica udio plave boje za svaki

piksel.

Učitavanje slika s medija za pohranu (npr. hardisk) radi se pomoću `matplotlib.pyplot.imread` funkcije. Potrebno je predati putanju do slike u obliku stringa, a funkcija vraća numpy polje koje predstavlja sliku. Slike se učitavaju u 3D dimenzionalno polje bez obzira radi li se o slici u boji ili slici u sivim tonovima. U slučaju slika u sivim tonovima, sva tri kanala će imati jednake vrijednosti pa je dovoljno koristiti samo jedan. Primjer 2.5 prikazuje kako se učitava i prikazuje slika 'road.jpg' koja je slika u sivim tonovima. Iz dobivenog 3D polja odabire se samo prvi kanal. Slika je tada predstavljena 2D numpy poljem čiji shape je jednak rezoluciji slike. Učitana slika se može prikazati pomoću funkcije `matplotlib.pyplot.imshow` na način da joj se preda numpy polje koje odgovara slici.

■ Primjer 2.5

```
import numpy as np
import matplotlib.pyplot as plt

img = plt.imread("road.jpg")
img = img[:, :, 0].copy()
print(img.shape)
print(img.dtype)

plt.figure()
plt.imshow(img, cmap="gray")
plt.show()
```

2.3 Priprema za vježbu

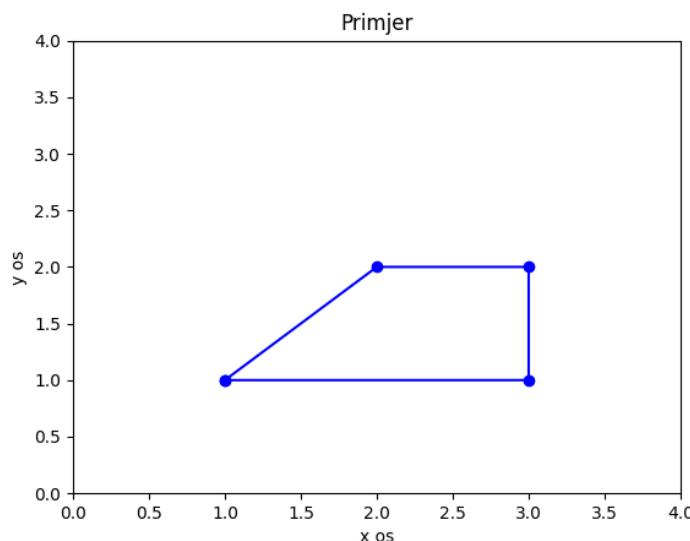
1. Proučite primjere 2.1 do 2.4. Možete li opisati što je rezultat izvođenja svake linije koda u navedenim primjerima?
2. Proučite kako se pohranjuje slika u sivim tonovima u memoriju računala, a kako slika u boji. Pogledajte primjer 2.5 koji učitava i prikazuje sliku pomoću Matplotlib biblioteke.

2.4 Rad na vježbi

1. Isprobajte Python primjere iz poglavlja 2.2. Python u Visual Studio Code IDE. Razmislite o svakoj liniji programske koda i što je njen rezultat. Pokrenite primjere u Debug modu i pogledajte u Explorer-u kako izgleda svaka od varijabli u danim primjerima.
2. Riješite dane zadatke.

Zadatak 2.4.1 Pomoću funkcija `numpy.array` i `matplotlib.pyplot` pokušajte nacrtati sliku 2.3 u okviru skripte `zadatak_1.py`. Igrajte se sa slikom, promijenite boju linija, debljinu linije i sl.

Zadatak 2.4.2 Datoteka `data.csv` sadrži mjerenja visine i mase provedena na muškarcima i ženama. Skripta `zadatak_2.py` učitava dane podatke u obliku numpy polja `data` pri čemu je u prvom stupcu polja oznaka spola (1 muško, 0 žensko), drugi stupac polja je masa u kg, a treći stupac polja je visina u cm.



Slika 2.3: Zadatak 2.4.1

- a) Na temelju veličine numpy polja data, na koliko osoba su izvršena mjerena?
- b) Prikažite odnos visine i mase osobe pomoću naredbe `matplotlib.pyplot.scatter`.
- c) Ponovite prethodni zadatak, ali prikažite mjerena za svaku pedesetu osobu na slici.
- d) Izračunajte i ispišite u terminal minimalnu, maksimalnu i srednju vrijednost visine u ovom podatkovnom skupu.
- e) Ponovite zadatak pod d), ali samo za muškarce, odnosno žene. Npr. kako biste izdvjijili muškarce, stvorite polje koje zadrži bool vrijednosti i njega koristite kao indeks retka.
`ind = (data[:,0] == 1)`

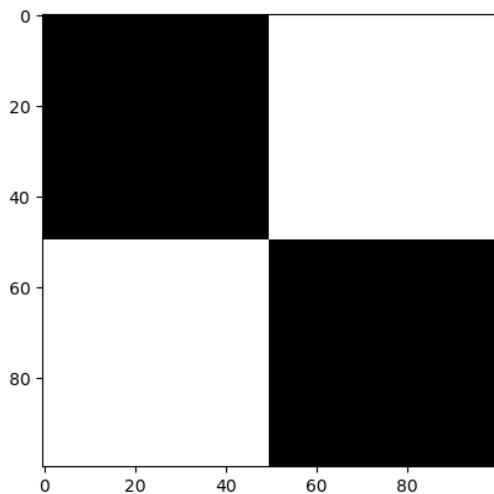
Zadatak 2.4.3 Skripta `zadatak_3.py` učitava sliku 'road.jpg'. Manipulacijom odgovarajuće numpy matrice pokušajte:

- a) posvijetliti sliku,
- b) prikazati samo drugu četvrtinu slike po širini,
- c) zarotirati sliku za 90 stupnjeva u smjeru kazaljke na satu,
- d) zrcaliti sliku.

Zadatak 2.4.4 Napišite program koji će kreirati sliku koja sadrži četiri kvadrata crne odnosno bijele boje (vidi primjer slike 2.4 ispod). Za kreiranje ove funkcije koristite numpy funkcije `zeros` i `ones` kako biste kreirali crna i bijela polja dimenzija 50x50 piksela. Kako biste ih složili u odgovarajući oblik koristite numpy funkcije `hstack` i `vstack`.

2.5 Izvještaj s vježbe

Kao izvještaj s vježbe prihvata se web link na repozitorij pod nazivom OSU_LV.



Slika 2.4: Zadatak 2.4.4

Literatura

[1] <https://www.kaggle.com/datasets/mustafaali96/weight-height>