

# Naučno izračunavanje, Jun1 rok, 22. jun 2020.

Na Desktop-u u arhivi sa imenom *NI\_jun1\_2020\_materijali.zip* nalaze se Jupyter sveske i materijali potrebni za rad. Raspakovati arhivu pa dobijeni direktorijum preimenovati tako da odgovara vašim podacima u formi *NI\_jun1\_2020\_ImePrezime\_BrojIndeksa*. Zatim mu pristupiti iz terminala pokretanjem komande *jupyter notebook*.

Na Desktop-u se nalazi i direktorijum sa imenom *docs* u kojem se nalazi dokumentacija.

## 1. (7 poena)

Data je funkcija  $f(x) = 2x + 3\sin(x)$ .

- Aproksimirati funkciju  $f(x)$  u srednjekvadratnom smislu polinomom drugog stepena na ekvidistantnoj mreži 10 tačaka intervala  $[-3, 3]$ .
- Aproksimirati funkciju  $f(x)$  u srednjekvadratnom smislu polinomom trećeg stepena na ekvidistantnoj mreži 10 tačaka intervala  $[-3, 3]$ .
- Nacrtati dobijene polinome i njihove vrednosti u tačkama mreže.
- Koji polinom u tački  $-1$  daje bolju aproksimaciju? Zaključak numerički obrazložiti.

## 2. (7 poena)

- Učitati i prikazati sliku *art.png*, a zatim je transformisati u crno-beli mod.
- Napisati funkciju *downsample* koja modifikuje sliku koja se zadaje kao argument tako što izbacuje svaki drugi piksel po vrsti i koloni.
- Napisati funkciju *apply\_gauss\_kernel* koja vrši primenu Gausovog filtera određenog kernelom

$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

nad slikom koja se zadaje kao argument funkcije.

- Prikazati prva 4 sloja takozvane Gausove piramide slike *art.png*: originalnu sliku, sliku koja se dobija primenom Gausovog filtera nad originalnom slikom za kojom sledi redukovanje dimenzije, a potom i sledeće dve slike koje se dobijaju primenom opisanog postupka nad prethodno dobijenim slikama.

## 3. (7 poena)

U datoteci *leaves.csv* nalaze se imena različitih vrsta listova i prateće metrike.

- Učitati podatke koje se nalaze u datoteci *leaves.csv*.
- Izdvojiti numeričke podatke i izvršiti njihovu standardizaciju.
- Odrediti najmanji broj glavnih komponenti kojima se objašnjava barem 85% varijanse učitano skupa podataka, a potom i izvršiti redukciju dimenzionalnosti skupa na dobijenu vrednost. Za vrednost *random\_state* parametra odabrati vrednost 7.
- Koristeći kosinusnu sličnost pronaći par najbližijih listova u skupu smanjene dimenzije.

## 4. (9 poena)

Jedna od modifikacija osnovne metode gradijentnog spusta je *Barzilai-Borvejn* metoda u kojoj se korak gradijentnog spusta izračunava na osnovu vrednosti gradijenata u dvema tačkama  $x_n$  i  $x_{n-1}$  po formuli

$$\gamma_n = \frac{(x_n - x_{n-1})^T (\nabla f(x_n) - \nabla f(x_{n-1}))}{\|\nabla f(x_n) - \nabla f(x_{n-1})\|^2}$$

za  $n \geq 2$ , a sa namerom da se aproksimira Njutnova metoda i ubrza ceo proces konvergencije.

- Implementirati Barzilai-Borvejn metodu koja za zadatu funkciju  $f$  dveju promenljivih, njen gradijent  $\nabla f$ , početnu tačku  $x_0$  i vrednost koraka  $\gamma_0$  koji se koristi za izračunavanje tačke  $x_1$  standardnom gradijentnom iteracijom izračunava minimum funkcije  $f$ . Algoritam zaustaviti ukoliko je broj iteracija veći od zadatog ograničenja *max\_iterations* ili ukoliko je norma gradijenta u tekućoj tački manja od zadate tačnosti  $\epsilon$ .

b) Primeniti implementiranu metodu na funkciju

$$f(a, b) = (1 - a)^2 + 100(b - a^2)^2$$

Za početnu tačku uzeti  $(2.1, 1.3)$ , za vrednost koraka  $\gamma_0$  u prvoj iteraciji 0.01, za maksimalan broj iteracija 100, a za tačnost epsilon  $10^{-8}$ .

c) Uporediti ovako dobijeno rešenje sa rešenjem neke od funkcija biblioteke *scipy.optimize*.