

מתימטיקה שימושית ומבוא למיחשוב מדעי - תרגיל בית מספר 2

בתרגיל זה נעסוק בהיבטים שונים של נושא המכפלה הפנימית, הנורמה והמרחק.

1. (10 נקודות) יהי V מרחב מכפלה פנימית ויהיו $u, v \in V$ שני וקטורים כך ש:

$$\|u\| = 2, \|u + v\| = 10, \quad \|u - v\| = 5$$

חשבו למה שווה $\operatorname{Re}\langle u, v \rangle$

2. (15 נקודות) נרמלו את הוקטורים הבאים:

א. $v \in C^5, v = [1, i, 0, 2, 1 - i]$ עם המכפלה הפנימית הסטנדרטית.

ב. $f(x) = \cos(2x), f(x) \in C[-\pi, \pi]$ עם המכפלה הפנימית הסטנדרטית.

ג. $A \in M_{3 \times 3}, A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & k \end{pmatrix}$ (מעל לממשיים) עם המכפלה הפנימית:

$\langle A, B \rangle = \operatorname{tr}(B^t A)$ כאשר tr מסמן את העקבה (trace) של המטריצה, כלומר סכום איברי האלכסון הראשי.

3. (5 נקודות) יהי P_2 מרחב כל הפולינומים ממעלה קטנה או שווה ל-2.

$$\langle f, g \rangle = \int_0^\infty f(x)g(x)e^{-x}dx \quad \text{לכל } f, g \in P_2 \text{ נגדיר:}$$

א. הוכיחו כי זוהי מכפלה פנימית על P_2 .

ב. הראו שהקבוצה $\left\{1, 1 - x, 1 - 2x + \frac{1}{2}x^2\right\}$ היא מערכת אורתוגונלית ביחס

למכפלה פנימית זו.

4. (10 נקודות) עבור המרחב $C[0,1]$ כלומר קבוצת כל הפונקציות הרציפות המוגדרות על

הקטע

$[0,1]$ ומקבלות ערכים קומפלכסיים, נתונות הפונקציות הבאות:

$$f_0(x) = 1, \quad f_1(x) = x + a, \quad f_2(x) = x^2 + bx + c$$

נגדיר את המכפלה הפנימית הסטנדרטית, כלומר

$$\langle f, g \rangle = \int_0^1 f(x)\overline{g(x)}dx$$

מצאו a, b, c כך שהפונקציות תהיינה מערכת אורתוגונלית.

5. (10 נקודות) עבור המרחב $C[0,1]$ עם המכפלה הפנימית הסטנדרטית המתאימה, האם

הפונקציות $f_0(x) = 1, f_1(x) = x, f_2(x) = x^2$ הן מערכת אורתונורמלית?

הוכיחו או הפריכו.

אם זוהי לא מערכת אורתונורמלית, יצרו מערכת אורתונורמלית הפורשת את אותו תת-

מרחב כמו הפונקציות $\{1, x, x^2\}$.

חלק ב' תרגיל python:

i. (15 נקודות) יהי $V = C^n$. כתבו פונקציית python לחישוב נורמה טבעית השתמשו בדוגמא הבאה והשלימו את הפונקציה:

```
import numpy as np

def normip(v,p):
    """
    function to compute the natural norm of an input vector.
    Inputs: v - a numpy array (n dim vector)
    Outputs: natural norm of v
    """
    # your code here
```

א. חשבו באמצעות הפונקציה שכתבתם נורמה טבעית עבור הוקטור:

$v = \text{np.array}([1, 2i, -3, 1, 7])$

ב. מצאו באמצעות הפונקציה וקטור יחידה בכוון $v = (1, 5, 3i, -1+i, 2)$

ב. חשבו באמצעות הפונקציה מהו המרחק בין הוקטורים:
 $u = (2, 1, -2i, -3, 8)$ ו- $v = (6, 7, i, 2i, 7)$

ii. (10 נקודות) מטרת התרגיל היא ליצור סדרה של פונקציות קוסינוס דגומות באמצעות מערכי numpy, להציג אותן באופן גרפי, ולהשמיע אותן תוך שימוש בחבילת sounddevice. א. בסעיף זה ניצור מערך numpy של קוסינוס עם 100 מחזורים על קטע t בין 0 ל-1. כתבו פונקציית python המקבלת כקלט את תחום ההגדרה של הקוסינוס ואת מספר המחזורים ביחידה ("התדירות"), מייצרת את המערך x של ערכי הקוסינוס המתאימים לציר ה-t, ומחזירה את x. השתמשו בדוגמא הבאה כדי ליצור את פונקציית ה-python לפי ההוראות.

```
import numpy as np
import matplotlib.pyplot as plt
import sounddevice as sd
import time
fs = 10000
Ts = 1/fs
f0 = 100
```

```

t = np.arange(0, 1, Ts)
x = np.cos(2*np.pi*f0* t)
sd.play(x, fs)
plt.plot(t,x)
plt.grid(axis = 'both')
stitle = 'cosine function with freq = '+ str(f0)
plt.title(stitle)
plt.show()
time.sleep(1)

```

אם מוצגת שגיאה עבור `import sounddevice as sd` יש להתקין את חבילת `sounddevice`.
לצורך כך כתבו `pip install sounddevice` ב-console.

השימוש ב- `sd.play(x, fs)` מאפשר לשמוע את סדרת הקוסינוס, בהנחה שציר ה- t מייצג את ציר הזמן, כך שהמערך t (מקבל ערכים בין 0 ל-1) מייצג זמן של שניה אחת, והמרווח בין כל שתי נקודות על ציר ה- t מייצג מרווח זמן של $1/10000$ שניה.

ב. כתבו פונקציית `python` המשתמשת בפונקציה הקודמת כדי ליצור באמצעות לולאת `for` סדרה של פונקציות קוסינוס דגומות כמו בסעיף א', עם תדירויות (התדירות מיוצגת על-ידי f_0) מ-500 עד 20000 בקפיצות של 500. הפונקציה תשמיד את הצליל באמצעות `sd.play(x, fs)` ותצייר בכל מעבר של הלולאה את הקוסינוס, כמו בדוגמא. כדי לראות את המחזוריות באופן ברור נשתמש עבור הציוור ב-
`plt.plot(t[:100], x[:100])`

כתבו עד איזה תדירות הצלחתם לשמוע את הצליל.

ג. כתבו פונקציית `python` כמו בסעיף ב', אך עתה מספר המחזורים ההתחלתי הוא 440, ובכל מעבר דרך הלולאה מספר המחזורים ליחידה או התדירות של הקוסינוס יעלו פי $2^{1/12}$. כלומר במעבר הראשון הערך יהיה 440, בשני $440 \cdot 2^{1/12}$, וכן הלאה עד לתדירות של 20000. רצף הצלילים הנוצר נקרא סולם כרומטי, ורוב כלי הנגינה במוסיקה המערבית מיוצרים כדי לייצר צלילים לפי סולם זה.

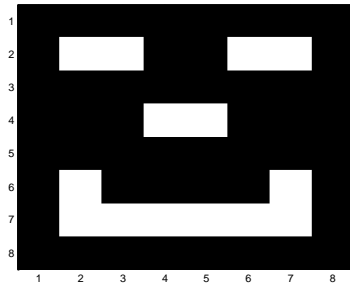
iii. (25 נקודות) תרגיל מעבדה : זיהוי פנים

מטרת התרגיל: שימוש במכפלה פנימית למציאת דמיון בין מטריצות בינאריות, התמצאות במערכי `numpy` ובמטריצות, שימוש בפקודה `imshow`.

מבוא :

"מערכת זיהוי תווי פנים היא אפליקציית מחשב אשר מסוגלת לזהות באופן אוטומטי או לאמת את זהותו של אדם על בסיס תצלום דיגיטלי או מקור וידאו. אחת הדרכים לעשות זאת היא באמצעות השוואת תכונות תווי הפנים בתמונה לתמונות המצויות במאגר נתונים. כיום מערכות זיהוי הפנים משמשות בעיקר מערכות אבטחה ופועלות לעתים רבות יחד עם מערכות זיהוי ביומטריות נוספות כגון זיהוי [טביעות אצבע](#) וזיהוי [קשתית](#) העין." (מתוך ויקיפדיה, האנציקלופדיה החופשית).

בתרגיל זה נעסוק בפנים סכימטיות כמו בציוור 1, ונפתח מערכת פשוטה לזיהוי "פנים" המבוססת על מדד דימיון (ראו בהמשך).



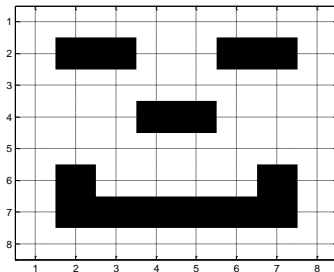
ציור 1 : פנים סכימטיות באמצעות מטריצה.

א. בסעיף זה ניצור פנים סכימטיות כמו בציור 1.

יצרו מטריצה בינארית של 8x8 באמצעות הפקודה np.zeros וציירו את המטריצה באמצעות הפקודה imshow. הדרכה : כתבו את הקוד הבא :

```
X = np.zeros([8,8])
X[ 1, 1:3 ] = 1
X[ 1, 5:7 ] = 1
X[ 3, 3:5 ] = 1
X[ 5:7, 1::5 ] = 1
X[6,2:6] = 1
plt.imshow(X, cmap = 'gray')
plt.show()
```

עתה צרו פנים בהם הרקע בהיר, ואיברי הפנים ('עיניים', 'אף', 'פה') כהים, לפי הציור.



ב. כתבו פונקציה שתקבל בכניסה שתי מטריצות בינאריות ('פנים') ותחשב את מקדם דימיון ביניהן באמצעות המכפלה הפנימית הבאה :

$$\langle A, B \rangle = \text{tr}(\overline{A^t} B) = \sum_{i=1}^n \sum_{j=1}^n \overline{a_{ij}} b_{ij} =$$

$$\overline{a_{11}} b_{11} + \overline{a_{12}} b_{12} + \dots + \overline{a_{1n}} b_{1n} \dots$$

$$+ \overline{a_{21}} b_{21} + \overline{a_{22}} b_{22} + \dots + \overline{a_{2n}} b_{2n} + \dots$$

$$+ \overline{a_{n1}} b_{n1} + \overline{a_{n2}} b_{n2} + \dots + \overline{a_{nn}} b_{nn}$$

מכפלה פנימית זו נקראת Frobenius inner product. אם המטריצות הן ממשיות, כמו בתרגיל זה, אין צורך בצמוד.

האם תוכלו להציע דרך לחשב מקדם דימיון מנורמל?

כלומר אם כל הערכים בשתי המטריצות זהים מקדם הדימיון המנורמל יהיה 1,

אם כולם שונים זה מזה הערך יהיה 0, ואם 50% מהערכים זהים הערך צריך

להיות 0.5, ובאופן כללי אם נסמן את מקדם הדימיון המנורמל ב- ρ , $0 \leq \rho \leq 1$.

כתבו פונקציה המחשבת את מקדם הדימיון המנורמל.
הדרכה : מקדם הדימיון בין שני וקטורים במרחב מכפלה פנימית כלשהו הוא :

$$\rho = \frac{\langle x, y \rangle}{\|x\| \cdot \|y\|}$$

הראו כי $-1 \leq \rho \leq 1$ (רמז : באמצעות אי-שוויון Cauchy-Schwartz).

ג. נניח כי קיימת גישה רק ל'אנשים' עם 'פנים' הדומות למטריצה X הנתונה בסעיף א'.

כתבו script עם לולאת while המדמה את שער הכניסה האוטומטי של חברת "עסק הוגן" המחשב את מקדם הדימיון המנורמל בין ה"פנים" X (מטריצת התבנית, המטריצה הראשונה בסעיף א') לבין מטריצה בינארית המייצגת "פנים" של אנשי החברה או אחרים המנסים להיכנס. חברת "טרופ ובלוע" מנסה להיכנס דרך השער כדי לבצע ריגול תעשייתי, ומציגה סדרה של מטריצות אקראיות ("פנים") באותו גודל של מטריצת התבנית X . עבור כל מטריצה אקראית כזאת ערכי 1 יכולים להופיע רק בחלק הפנימי, כלומר בשורות ובעמודות 1:6.

ה- script יציג את מטריצת התבנית X ואת מטריצת המבחן X_{test} בכל איטרציה על המסך באמצעות פקודת imshow, ויחשב את מקדם הדימיון המנורמל. המקדם יוצג כחלק מכותרת הציור של X_{test} .

אם מקדם הדימיון עולה על ערך של 0.7 הלולאה עוצרת (השער נפתח), ומוצגת ההודעה "access permitted". בכל מקרה אחר, כלומר אם מקדם הדימיון עבור המטריצה הנבחנת לא גדול או שווה לערך הסף מוצגת ההודעה "access denied". הדפיסו את מספר הפעמים של ניסיונות כניסה ל"שער" עד להצלחה.

