

אלגוריתמים במולטימדיה ולמידת מכונה בסביבת פייתון

מס' זהות:

הוראות:

1. המבחן נעשה במחשב.
2. אין שימוש בחומר עזר.
3. מותר להשתמש במחשבון.
4. משך הבחינה - 3 שעות.
5. במבחן שלוש חלקים:
 - חלק ראשון חובה - יש לענות על כל השאלות.
 - חלק שני בחירה - יש לענות על שתי שאלות מתוך שלוש.
 - חלק שלישי בונים - לא חובה לענות על חלק זה.
6. אין לעזוב את המבחן לפני שהוגש כנדרש (אחרת הציון 0 אוטומטית).
7. שימו לב! לא ליצור את הפרויקט בקונן C (אם קיים קונן נוסף) אחרת יש סיכוי לעובדן מידע.
8. שימו לב! תוודאו שאתם עובדים בגרסת פייתון נכונה.
9. יש ליצור קובץ פונקציות אחד בלבד ובו לממש את כל הפונקציות הנדרשות.
10. כל פונקציה שאתם מתבקשים לכתוב חייבת להיות בעלת שם זהה לזה שבשאלה! כולל התייחסות לאותיות קטנות/גדולות. פונקציה שהשם שלה לא יהיה רשום במדויק, לא תקבל ניקוד.
11. אין להגיש שום סקריפט שלא התבקשתם לכתוב וזה כולל סקריפט הבדיקות שלכם.
12. כל קטע קוד שלא קשור לאחת הפונקציות שהתבקשתם לכתוב יגרור הורדה בציון וזה כולל פלטים שלא התבקשתם לעשות.
13. שימו לב! יש להתחשב במקרי הקצה שעלולים להיות. לדוגמה קלט לא תקין או כתובת שלא קיימת. במקרה של שגיאה יש להחזיר None
14. אתם יכולים להשתמש בכל פונקציה שתמצאו לפתרון המבחן.
15. במהלך המבחן אין גישה לאינטרנט.
16. מי שמסיים את המבחן מחכה במקומו עד להגעת אוסף הבחינות בכדי להגיש את הקוד.
17. יש לרשום את המספר ת.ז. על גבי טופס זה ולהגיש אותו בסוף המבחן.
18. לסקריפט שהנכם מגישים יש לקרוא בשם המתחיל באות "s" קטנה וללא רווח מס' הת.ז. שלכם. לדוגמה: s123456789
19. במידה וישנה פונקציה שאינכם יודעים מה יעודה, נסאו פקודה:
print FunName.__doc__

פונקציות עזר אפשריות

Basic functions:

isinstance, id, hex, ord, str, eval, enumerate, list, range, raw_input, tuple, type, zip, chr, cmp, in, not, or, and; **String operations:** count, endswith, find, isdecimal, isdigit, isnumeric, join, lower, upper, replace, split; **List operations:** append, extend, insert, remove, pop, count; **Dictionary operations:** has_key, items, keys, values; **Set operations:** issubset, union, intersection, difference, symmetric_difference, copy, add, remove, discard, pop

copy:

copy

os:

getcwd, listdir, path.isfile, path.exists, path.join, path.isdir

numbers:

Number

numpy:

zeros, ones, eye, tri, mean, median, sum, prod, diff, gradient, dot, transpose, trace, ceil, floor, round, max, min, argmax, argmin, sort, argsort, reshape, concatenate, all, any, pi, asarray, copy, logical_or, logical_and, logical_not, logical_xor, array_equal, isnan, linspace, arrange, mod, exp, log, sin, cos, tan, sinc, sinh, cosh, tanh, arcsin, arccos, arctan, arctan2, arcsinh, arccosh, arctanh, intp, int8, int16, int32, int64, uint8, uint16, uint32, uint64, float16, float32, float64, complex64, complex128, random.rand, random.randn, random.randint, random.random, random.shuffle, random.normal, random.uniform, ndarray.astype, linalg.eig, linalg.eigh, linalg.eigvals, linalg.norm, linalg.det, linalg.matrix_rank

cv2:

copyMakeBorder, BORDER_REPLICATE, BORDER_CONSTANT, BORDER_REFLECT, filter2D, split, merge, cvtColor, COLOR_BGR2GRAY, COLOR_GRAY2RGB, imread

matplotlib.pyplot:

figure, subplot, subplot2grid, plot, stem, scatter, show, bar, hist, hist2d, pie, psd, imshow, axis, grid close, colorbar, draw, gca, gcf, legend, loglog, semilogx, semiology, pause, title, xlabel, ylabel, xlim, ylim, xticks, yticks

חלק א' (חובה): 50%

שאלה 1 (5 נק.)

1. הגדירו פונקציה בשם: **myTriangularMatrix**: הפונקציה צריכה לקבל מספר טבעי n . הפונקציה צריכה להחזיר מטריצה ריבועית (מערך דו-מימדי), משולשית עליונה ממימד n .

לדוגמה:

$n = 4$
 $myMat = myTriangularMatrix(n)$

תוצר הפונקציה יהיה:

$$myMat = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

שאלה 2 (10 נק.)

2. הגדירו פונקציה בשם: **countMyChars**: הפונקציה צריכה לקבל מחרוזת. הפונקציה צריכה להחזיר מילון המכיל את כל התווים הקיימים במחרוזת ללא חשיבות לאותיות גדולות/קטנות בתור המפתחות של המילון (כל מפתח שהוא אות, יהי אות קטנה) וערך עבורו יהיה מספר הופעות של כל אחד מהתווים במחרוזת שהתקבלה.

לדוגמה:

$myStr = 'Hello Hello We Do Hi no No no we Donot Hey HE he heY'$
 $myDict = countMyChars(myStr)$

תוצר הפונקציה יהיה:

myDict :

$\text{' ' : 13, 'e' : 8, 'd' : 2, 'i' : 1, 'h' : 7, 'l' : 4,}$
 $\text{'o' : 8, 'n' : 4, 't' : 1, 'w' : 2, 'y' : 2}$

שאלה 3 (10 נק.)

3. הגדירו פונקציה בשם: **myListMin**: הפונקציה מקבלת רשימה המכילה מספרים ורשימות נוספות במבנה דומה לרשימה עצמה. הפונקציה צריכה להחזיר את הערך הקטן ביותר בין כל האיברים הנמצאים ברשימה שהתקבלה כולל כל תתי הרשימות.

לדוגמה:

$myList = [1, [2], 3, [1, -1, [2, 5, [1.0], 1], 2], -3, 1.0, [[0]]]$
 $myMin = myListMin(myList)$

תוצר הפונקציה יהיה:

$myMin = -3$

שאלה 4 (10 נק.)

1. הגדירו פונקציה בשם **myFileSearch**:
הפונקציה מקבלת כתובת של תיקייה כלשהי במחשב וסיומת של סוג קובץ כלשהו.
הפונקציה מחזירה רשימה של כל הקבצים בעלי סיומת המבוקשת אשר נמצאים בתיקיה המבוקשת.

לדוגמה:

```
myPath = 'D:\Settings'  
myExtention = 'pplx'  
myFileList = myFileSearch (myPath, myExtention)
```

שאלה 5 (15 נק.)

5. הגדירו פונקציה בשם: **myBorderCreation**

```
def myBorderCreation (img, borders):  
    if img.__class__ != np.ndarray or np.int64(borders) != borders:  
        return None  
  
    im = img.copy()  
    D = im.shape  
    if len(D) != 3:  
        return None  
  
    # -----  
    for n in range(D[1]):  
        if n < borders or n - D[1] > (-1 - borders):  
            im[:, n] = (np.mean(im[:, :, 0]), np.mean(im[:, :, 1]), np.mean(im[:, :, 2]))  
    # -----  
  
    return im
```

תחליפו את קטע הקוד שבין הקווים השבורים בשורת קוד אחת בלבד.

שאלה 6 (25 נק.)

6. הגדירו פונקציה בשם: **mySignalFiltering**
הפונקציה צריכה לקבל אות מוגדר על ידי מערך `ndarray`, מספר טבעי (סדר המסנן) ומספר נוסף (`T`) שערכו אפס או אחד.
הפונקציה צריכה להחזיר את האות אשר עבר סינון על ידי מסנן `MAF` סימטרי אם מספר `T` שווה לאפס או על ידי `Median Filter` אם מספר `T` שווה לאחד. עבור כל ערך `T` אחר יש להחזיר `None`. עוצמת הסינון צריכה להיות בהתאם לסדר המסנן.
יש לעשות ריפוד "המשכה". ריפוד אשר משכפל את האיבר שבקצה.
דוגמה:

$Sig = \text{mySignalFiltering}(\text{signal}, \text{order}, T)$

שאלה 7 (25 נק.)

7. הגדירו פונקציה בשם: **myKMeansIter**
הפונקציה צריכה לקבל מטריצת המידע (`Data`) בה כל שורה מייצגת וקטור מעפיינים (`Feature Vector`) ומטריצת המרכזים (`Cent`) בה כל שורה מייצגת וקטור מעפיינים ממוצע. הפונקציה צריכה להחזיר את מטריצת המרכזים המעודכנת לאחר איטרציה אחת בלבד של האלגוריתם `K-Means`. את המרחקים בין הוקטורים יש לחשב לפי מרחק אוקלידי (נורמה 2).

תזכורת:

מחשבים מרחק של כל נקודה לכל המרכזים. משייכים כל נקודה למרכז אליו היא הכי קרובה. מחשבים ממוצע של כלל הנקודות ששויכו למרכז מסויים ומגדירים את הממוצע להיות המרכז החדש שמחליף את זה שהנקודות שויכו אליו.

דוגמה:

$nCe = \text{myKMeansIter}(\text{Data}, \text{Cent})$

בדיקה אפשרית:

הנכם יכולים לבצע את בדיקת הקוד שכתבתם על ידי בחירה של שתי קבוצות בעלות 3-5 נקודות כל אחת כך שהמרחק המקסימלי בין הנקודות בתוך הקבוצות יהיה קטן מהמרחק המינימלי של הנקודות בין הקבוצות. יש לחשב ממוצע (מרכז) של כל קבוצה. בוחרים את המרכז לכל קבוצה כך שיהיה קרוב לממוצע של אותה הקבוצה, אך לא זהה לה. במידה וכל השלבים בוצעו נכון, לאחר ההרצה של הפונקציה הנכם תקבלו את הממוצעים של הקבוצות בתור המרכזים.
שימו לב! זהו אינו חלק מהשאלה, אין צורך לממשו, אלא אם ברצונכם לבדוק את עצמכם בשיטה שהוצעה לעיל. כמו גם, אין להגיש חלק זה בסוף המבחן.

שאלה 8 (25 נק.)

8. הגדירו פונקציה בשם: `myGradDescent`
 הפונקציה צריכה לקבל אות חד מימדי וזמן אשר מוגדרים על ידי מערכים של `ndarray`, נקודת התחלה שהיא ערך של זמן, מספר איטרציות וגודל צעד.
 הפונקציה צריכה להחזיר את ערך הזמן אליו הגיע האלגוריתם לאחר כמות האיטרציות שניתנה עם גודל צעד הנתון.
 יש להשתמש בגרדיאנט רובסטי. משמע, נגזרת מסדר ראשון אשר פחות רגישה לרעשים.

תזכורת:

$$t_{k+1} = t_k - \gamma_k \frac{\partial f(t)}{\partial t_k}$$

דוגמה:

$t_n = \text{myGradDescent}(\text{signal}, \text{time}, t_0, \text{iterNum}, \text{stepSize})$

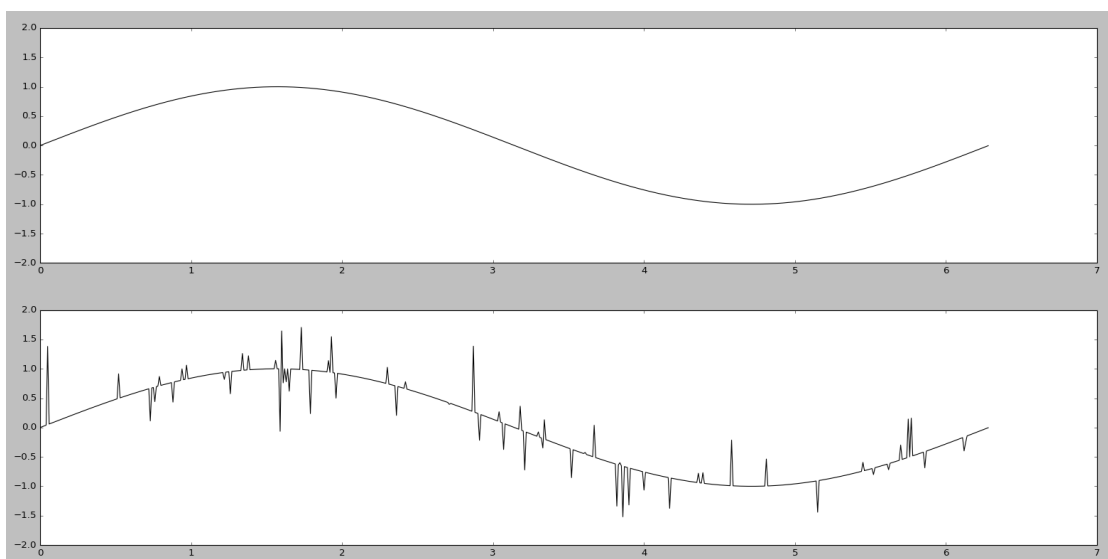
חלק ג' (בנוסף: 15%)

שאלה 9 (15 נק.)

9. הגדירו פונקציה בשם: `SaltAndPepperSignalNoise`
 הפונקציה צריכה לקבל אות חד מימדי (`ndarray`) ואחוז הדגימות הרועשות (מספר בין 0 ל-1).
 הפונקציה צריכה להחזיר את האות שהיא קיבלה עם תוספת רעש בדגימות אקראיות לפי התפלגות אחידה (מיקום הדגימות הרועשות מתפלג אחיד). כמות הדגימות היא בהתאם לאחוז הדגימות הרועשות מכמות הכוללת של הדגימות באות (במידה והערך לא שלם, יש לבצע עיגול). בכל אחד ממקומות בהם מתווסף הרעש, יש להוסיף רעש אקראי בהתפלגות נורמלית (גאוסיאנית) עם תוחלת 0 ושונות 0.5.

דוגמה:

`NoisySignal = SaltAndPepperSignalNoise(Signal, NoisePr)`



בהצלחה!