מכון טכנולוגי חולון
Holon Institute of Technology

סמסטר א – תשע"ט, מועד ג'
תאריך: 24.05.19
המרצה: דמיטרי פטשוב
קבוצות הרצאה: 0-21210-1/2

אלגוריתמים במולטימדיה ולמידת מכונה בסביבת פייתון

מס' זהות:

הוראות:

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

מכון טכנולוגי חולון
Holon Institute of Technology

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, 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

מכון טכנולוגי חולון
Holon Institute of Technology

חלק א' (חובה): 50%

שאלה 1 (5 נק.)

1. הגדירו פונקציה בשם : **myChessMatrix**
הפונקציה צריכה לקבל מספר טבעי n.
הפונקציה צריכה להחזיר מטריצה ריבועית (מערך דו-מימדי) ממימד 2n בה יש אחדים ואפסים המופיעים לסירוגין כן לאורך העמודות וכן לאורך השורות. איבר במיקום (0,0) חייב להיות 0.

דוגמה :

n = 2
myMat = **myChessMatrix** (n)

תוצר הפונקציה יהיה :

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

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 עבורן 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 :

‘ ‘: 13,	‘e’ : 8,	‘d’ : 2,	‘i’ : 1,	‘h’ : 7,	‘l’ : 4,
‘o’ : 8,	‘n’ : 4,	‘t’ : 1,	‘w’ : 2,	‘y’ : 2	

שאלה 3 (15 נק.)

3. הגדירו פונקציה בשם :**myListHist**
הפונקציה מקבלת רשימה המכילה מספרים טבעים ורשימות נוספות במבנה דומה לרשימה עצמה.
הפונקציה צריכה להחזיר מערך חד מימדי מסוג ndarray שהוא ההיסטוגרמה של הרשימה. מיקום 0 במערך מסמל כמה ערכים השווים ל- 0 יש ברשימה (כולל תתי רשימות).

לדוגמה :

myList = [1, [2], 3, [11, 1, [2, 3, [1.0]], 2], 3, 1.0, [[0]]]
myHist = **myListHist (myList)**

תוצר הפונקציה יהיה :

myHist = [1, 4, 3, 3, 0, 0, 0, 0, 0, 0, 0, 1]

מכון טכנולוגי חולון
Holon Institute of Technology

שאלה 4 (10 נק.)

4. הגדירו פונקציה בשם **myKNearest:**

הפונקציה מקבלת מספר טבעי, k, מספר ממשי, x ומערך חד-מימדי מסוג ndarray המכיל מספרים בעלי יצוג float64.

הפונקציה צריכה להחזיר מערך חד-מימדי מסוג ndarray המכיל k איברים הקרובים ביותר בגודלם למספר הממשי x. האיברים צריכים להיות מסודרים כך שהאיבר במיקום הראשון, הוא האיבר הקרוב ביותר ל- x והאיבר במיקום האחרון, הוא האיבר הרחוק ביותר מ- x מבין k האיברים הקרובים ביותר.

לדוגמה:

```
k = 3
x = -1.2
myArray = np.array([-3.2, 8, 1, 0, -1.1, 0.1, 12])
myKN = myKNearest(k, x, myArray)
```

תוצר הפונקציה יהיה:

```
myKN = [-1.1, 0, 0.1]
```

שאלה 5 (10 נק.)

5. הגדירו פונקציה בשם: **myFastToneReplacement**

להלן הפונקציה:

```
def myFastToneReplacement(img, fromA, toB, equalsC):
    if img.__class__ != np.ndarray:
        return None
    myImg = img.copy()
    # ------------------------------------------------------------------
    D = myImg.shape
    if len(D) == 2:
        for i in range(D[0]):
            for j in range(D[1]):
                if myImg[i, j] >= fromA and myImg[i, j] <= toB:
                    myImg[i, j] = equalsC
    elif len(D) == 3:
        for m in range(D[0]):
            for n in range(D[1]):
                for k in range(D[2]):
                    if myImg[m, n, k] >= fromA and myImg[m, n, k] <= toB:
                        myImg[m, n, k] = equalsC
    else:
        return None
    # ------------------------------------------------------------------
    return myImg
```

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

מכון טכנולוגי חולון
Holon Institute of Technology

חלק ב' (בחירה): 50%

שאלה 6 (25 נק.)

6. הגדירו פונקציה בשם : **myLRcycles**

הפונקציה צריכה לקבל מטריצת המידע (ndarray), וקטור עמודה של תיוגים (ndarray), מטריצת משקלים W ההתחלתית (ndarray), פרמטר גמא (מספר חיובי), ומספר איטרציות (מספר טבעי).

הפונקציה צריכה להחזיר את מטריצת ה- W לאחר התהליך של רגרסיה לוגיסטית בעלת מספר חזרות מוגדר מראש.

תזכורת :

עדכון מטריצת W :

$$\boldsymbol{w_{k+1} = w_k + \gamma_k \frac{\partial l(w)}{\partial w_k}}$$

קירוב לנגזרת החלקית :

$$\boldsymbol{\frac{\partial l(w)}{\partial w} = \sum_{n=1}^{N} \left[x_n \cdot \left(y_n - g(x_n w) \right) \right]}$$

הגדרת הסיגמואיד :

$$\boldsymbol{g(z) = sig(z) = \frac{1}{1 + e^{-z}}}$$

W = **myLRcycles(**TrainData, TrainLabel, W, gamma, iterNum**)**

שאלה 7 (25 נק.)

7. 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 :

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

דוגמה :

$t_n =$ **myGradDescent(**$signal, time, t_0, iterNum, stepSize$ **)**

מכון טכנולוגי חולון
Holon Institute of Technology

שאלה 8 (25 נק.)

8. 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 = **myKMeansIter** (Data, 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.

חלק ג' (בונוס): 15%

שאלה 9 (15 נק.)

9. הגדירו פונקציה בשם : **myPeakDetect**

הפונקציה צריכה לקבל אות חד מימדי (ndarray).

הפונקציה צריכה להחזיר מערך ndarray המכיל את אוסף האינדקסים בהם יש נקודות מקסימום באות שהתקבל.
הנכם יכולים להניח שאין מקטעים שטוחים (אמפליטודה קבועה) באות.

דוגמה :

locs = **myPeakDetect** (Signal)

בהצלחה!