מערכות לומדות תשפ"ד - תרגיל 3

למידה לא מפוקחת

בתרגיל זה תשתמשו בשיטות של למידה לא מפוקחת לחילוץ מאפיינים ולהפחתת ממדים.
הערה: בכל סעיף בו מופיע @ עליכם לרשום את התשובה\התוצאות במסמך התשובות.

נושא 1 – חילוץ מאפיינים אוטומטי

בתרגיל הקודם פיתחתם ומימשתם מאפיינים של ספרות הכתובות בכתב יד, המרתם את הקלט למרחב המאפיינים וביצעתם בו סיווג. כעת תשתמשו במודל למידה לא מפוקחת למציאה אוטומטית של מאפיינים מן הקלט הלא מסווג, כשלב מקדים לסיווג.

(שאלה 1)

תחילה הריצו את הקוד מן הדף הבא:
http://scikit-learn.org/stable/auto_examples/neural_networks/plot_rbm_logistic_classification.html
(הורידו בעזרת כפתור ההורדה של plot_rbm_logistic_classification.py)

בקוד זה, מודל הנקרא Bernoulli Restricted Boltzmann machine. זהו סוג של רשת נוירונים מלאכותית המהווה מודל גנרטיבי הלומד את ההתפלגות של הקלט. הקוד משתמש במודל לביצוע חילוץ מאפיינים אוטומטי. אחר כך הנתונים מומרים למרחב המאפיינים ועליו מתבצע סיווג בעזרת logistic regression. לצורך השוואה מתבצע גם סיווג (בצורה בלתי תלויה) בעזרת logistic regression על הקלט המקורי. בחנו את הקוד והבינו את פעולתו. שימו לב במיוחד לנקודות הבאות:

- הגדלת סט הנתונים על ידי הזזות של פיקסל יחיד לארבעה כיוונים. פעולה זו גם מגדילה את כמות הנתונים המתויגים וגם מאפשרת "אדישות" להזזה (לפחות ברמה בסיסית).
  הבינו כיצד מתבצעת ההזה: השימוש בפונקציה shift, השימוש בקונבלוציה עם גרעיני ההזזה, שכפול התיוג הנכון לכל דוגמה מוזזת. השימוש במשתנה "_" ליצירת ערכי ה Y (התיוג) המשוכפלים. ראו גם ההסבר כאן
  https://betterprogramming.pub/how-to-use-underscore-properly-in-python-37df5e05ba4c

a) השתמשו ב shape לברר מה גודל סט הנתונים (כולל התיוגים) לפני פעולת ההגדלה, ואחריה. רשמו את הגדלים @.

b) מדוע משתמשים בפעולת x.reshape((8,8)) בתוך הקריאה לקונבולוציה @?

c) מה תעשה פעולת הקונבולוציה המוצגת בקוד, עבור הגרעין הזה ? @:

[[0,0.5,0],
[0,0,0],
[0,0,0]]

- נורמליזצית הקלט.

d) הסבירו את השימוש ב minmax_scale , ובפרט ביחס למה היא מנרמלת את הנתונים. האם ביחס למינימום או למקסימום של כל תמונה בפני עצמה? משהו אחר?@.

- חלוקה לסט אימון ולסט מבחן.

e) מדוע משתמשים ב random_state=0 בקריאה ל train_test_split @?

- המודלים לשימוש.

f) מה גורם השימוש ב verbose=True @ ?

- השימוש ב Pipeline הכולל שני שלבים: הראשון BernoulliRBM, והשני LogisticRegression. קיראו על המחלקה Pipeline בתיעוד.

- האימון של המודל המשולב.

- g) מדוע משתמשים בקוד ב clone @ ?

- האימון של מודל logisticRegression נפרד.
- הצגת ביצועי המסווגים על סט המבחן, והתוצאות עצמן.
- הצגת המרכיבים (components_) שנמצאו על ידי BernoulliRBM.

בשימוש עצמאי ב BernoulliRBM ההמרה למרחב המאפיינים מתבצעת על ידי המתודה transform של BernoulliRBM. אך כאשר המודל הזה נמצא כחלק מ Pipeline יש שימוש במתודה fit_transform, המבצעת את ה fit המחשב את המרכיבים, ואחר כך transform הממיר את הקלט למרחב המאפיינים. התוצאה היא טרנספורמציה לא לינארית (הפונקציה הלוגיסטית) של הקלט בעזרת המרכיבים.

במהלך הלמידה, הרשת מנסה לשחזר את הקלט וכך מהווה מודל הלומד את התפלגות הקלט. המסגרת הכללית בו משחזרים את הקלט נקרא autoencoder ו Restricted Boltzmann machine מהווה אחת השיטות למימוש מסגרת זו.

למתעניינים:
קיראו כאן הסבר פשוט על סוג הרשת הזו ושימושיה
https://pathmind.com/wiki/restricted-boltzmann-machine

וזה מאמר המשתמש ב Restricted Boltzmann machine כשלב עיבוד מקדים ללימוד דמיון בין קטעי מוזיקה
http://mirg.city.ac.uk/blog/wp-content/uploads/2013/09/rbm-features-for-music-similarity.pdf

(שאלה 2)

פרמטר חשוב של המודל הוא מספר היחידות הנסתרות ברשת, או במונחי הקוד שלנו מספר הרכיבים (_components) הנלמדים. מספר זה הוא הממד של מרחב המאפיינים החדש, ובשאלה זו נחקור את השפעתו על תוצאות הסיווג.
נשים לב שממד הקלט המקורי הוא (8x8=64), **כך שאם נבחר מספר קטן מזה נבצע הפחתת ממדים**.
הריצו את הקוד מספר פעמים (פעם אחת לכל ממד של מרחב המאפיינים) עבור ערכי _rbm.n_components שהם ריבועי המספרים מ 2 ועד 20 (כלומר 400,... ,4,9,16).

עבור כל הרצה שימרו את ערך ה precision הממוצע, ואת הזמן בשניות שלקח בכל ההרצה **חלק האימון של ה pipeline בלבד**. העזרו ב ()time.perf_counter למדידת זמנים.

עבור כל הרצה הציגו את _rbm.n_components בצורה גרפית, על ידי שינוי הקוד המקורי כך שיוצגו ... ,2x2, 3x3, 4x4 (בעזרת subplot). הנה התוצאה הרצויה עבור 10x10:

בסיום ההרצות הציגו שני גרפים:

- ה precision הממוצע (macro avg) כנגד מספר הרכיבים. הציגו גם את ערך ה precision הממוצע עבור מסווג הlogistic regression הפועל על ה raw pixels (0.78) בתור קו אופקי, כך שאפשר יהיה להשוות אליו.
- הזמן בשניות לכל הרצה כנגד מספר הרכיבים.

להגשה בשאלה 2:

א. הקוד המלא בקובץ ex3_2.py.
ב. התצוגה הגרפית עבור 20x20 @.
ג. שני הגרפים @. הקפידו על כותרות וטקסט לצירים.
ד. ניתוח שלכם לתוצאות:
1. האם הפחתת ממדים בעזרת RBM מועילה? מה המחיר לביצוע הפעולה?
2. האם העלאת הממד בעזרת RBM מועילה? מה המחיר לביצוע הפעולה?
3. האם לדעתכם כדאי להעלות לממד יותר גבוה ממה שביצעתם בתרגיל? מדוע?

נושא 2 – הפחתת ממדים

בחלק זה תכירו ותשתמשו ב PCA (Principal Component Analysis). שיטה זו מורידה את הממד של הקלט על ידי מציאת סט צירים המותאמים לקלט, ובחירה k מתוכם כאשר k קטן (בהרבה) מן הממד המקורי n.

תחילה הורידו והריצו את הקוד הבא:
http://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html

הקוד הזה מסווג תמונות פנים של אנשים מפורסמים ל 7 מחלקות (7 מפורסמים). הדוגמה משתמשת ב PCA כשלב של מציאת מאפיינים. הקוד מבצע את השלבים האלה:

- מוצא 150 צירי PCA שהם 150 הצירים המתאימים ביותר לקלט. ממד המרחב המקורי הוא (50x37 = 1850), כך שזו הפחתה משמעותית. כל ציר הוא "תמונת בסיס" או "פרצוף בסיס" ומספר האיברים בו כממד המרחב המקורי.

- ממיר את הנתונים (סט האימון וסט הבחינה) לצירים אלה. כל תמונה מיוצגת על ידי וקטור קואורדינטות שלה בצירים החדשים (כל תמונה מהווה צירוף לינארי של

הצירים, כאשר מקדמי הצירוף הם הקואורדינטות שלה). אורך כל וקטור הוא 150. וקטורי הקואורדינטות מהווים את מרחב המאפיינים להמשך.

- מבצע סיווג על ידי SVM על וקטורי הקואורדינטות של התמונות. ראשית אימון על סט האימון, ולאחריו בחינה על סט הבחינה.

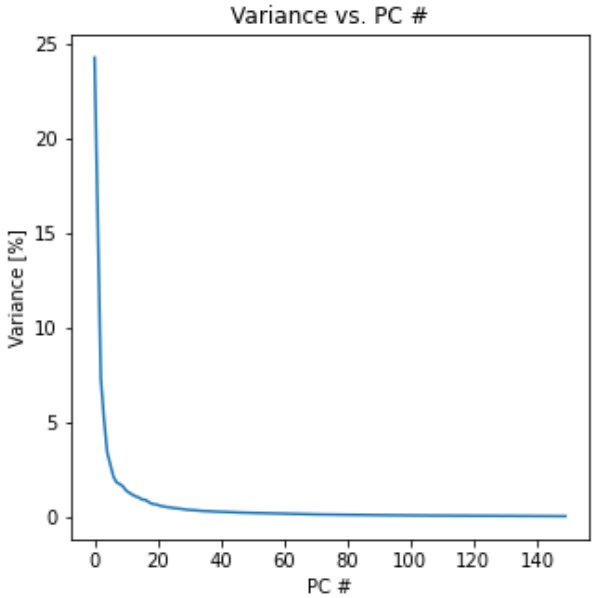
- מציג את ביצועי המסווג על סט המבחן.

- ולבסוף מציג שני חלונות. בראשון דוגמאות של התמונות וסיווגן, ובשני אוסף של פרצופי בסיס (Eigen faces).

בחנו את הקוד היטב וראו כי אתם מבינים הכול.
שימו לב לשימוש ב RandomizedSearchCV לחיפוש הפרמטרים המתאימים למסווג SVM.

(שאלה 3)
השתמשו בקוד והוסיפו לו \ שנו אותו כך:

א. אחרי השורה המחשבת את ה eigenfaces הוסיפו קטע קוד להצגת השונות המוסברת על ידי וקטורי הבסיס החדשים. השתמשו ב pca.explained_variance_ratio_. הציגו את ערכו כפול 100 בגרף. הוסיפו כותרות וכיתוב לצירים. התוצאה צריכה להראות כך:



ב. מדדו כמה זמן לוקח האימון (**fit**) בעזרת RandomizedSearchCV.
דווחו על איכות הסיווג (השורה של ה macro avg) ועל הזמן בקובץ התשובות @.

ג. שנו את התצוגה של תמונות הפנים ושל פרצופי הבסיס כך שתכיל 7 x 7 תמונות.
השתמשו בקוד הזה במקום ההגדרות הקיימות (או שנו לפי בחירתכם לקבלת הצגה טובה)

```
plt.figure(figsize=(1.45 * n_col, 1.5 * n_row))
plt.subplots_adjust(bottom=0.03, left=.01, right=.99, top=.93,
hspace=.36)
```

ד. התוצאות צריכות להראות כך:

ה. הוסיפו לקוד מסווג SVM שיאומן על סט האימון **המקורי** לפני המרתו למרחב PCA, ויבחן על סט המבחן **המקורי** לפני המרתו למרחב PCA.
בצעו חיפוש בעזרת RandomizedSearchCV אחר המסווג הטוב ביותר לנתונים אלה (כולל **kernel 'linear' ולא רק 'rbf'**).
מדדו כמה זמן לוקחת האימון (**fit**) בעזרת RandomizedSearchCV.
דווחו על איכות הסיווג (השורה של ה macro avg) ועל הזמן בקובץ התשובות @.

להגשה בשאלה 3:

- הקוד המלא בקובץ ex3_3.py.
- התשובות לשאלות 3ב, 3ה @.

הגשת התרגיל

א. תאריך הגשה: עד יום ראשון, 25.2.24, בשעה 23:55.
ב. ניתן להגיש בזוגות.
ג. יש לכתוב **שם \ שמות + ת"ז** בראשית כל מסמך מוגש (**כולל בקבצי הקוד**).
ד. כל מגיש (ביחיד או בזוג) צריך לדעת להסביר כל מה שנעשה בפתרון המוגש. חלק מן המגישים ידרשו להסביר את הפתרון שלהם למרצה.
ה. יש להגיש מסמך Word המכיל את כל התשובות לתרגיל. שם מסמך זה יהיה **ex3.docx**.
הקפידו שמספור סעיפי התשובות שלכם יהיה זהה למספור סעיפי השאלות.
ו. לכל פונקציה צריך להיות תיעוד.
ז. יש להגיש את כל הקוד לתרגיל בשני קבצים לפי ההנחיות למעלה. בתחילת כל קובץ יבואו הגדרות כל הפונקציות. בהמשך הקובץ יבוא חלק ההרצה. חלק זה **יופרד על ידי הערות לכל אחד מסעיפי השאלות.**
ח. שלושת הקבצים ישכנו בתוך תיקייה הכוללת את שמכם.
שם התיקיה למגיש יחיד:
EX3FamilyName
שם התיקיה לשני מגישים:
EX3Family1Family2
התיקיה תארז לקובץ **zip** בעל אותו שם כשל התיקיה.