דו"ח אלגוריתמי סיווג

תומר שי

29 בנובמבר 2021

Part I

KNN

המימוש בקצרה

1. צור מערך ריק `test_labels` (שיכיל את ה-predictions של כל הנקודות ב-`test_set`).
2. לכל נקודה `test_point` ב-`test_set`:
   (א) אתחול מערך שכנים בגודל $k = 5$.
   (ב) לכל נקודה `train_point` ב-`train_set`:
      i. חשב את המרחק בין `train_point` ל-`test_point` לתוך המשתנה `curr_dist`.
      ii. חשב את המרחק המקסימלי בין ה-`test_point` לבין הנקודות במערך השכנים.
      iii. אם `curr_dist` קטן מהמרחק המקסימלי:
         א'. הכנס את `train_point` במקום הנקודה המקסימלית.
   (ג) מצא את ה-labels של כל הנקודות במערך השכנים, ובדוק איזה label מופיע הכי הרבה.
   (ד) הכנס את ה-label שמופיע הכי הרבה לתוך `test_labels`.
3. הוצא את `test_labels` כפלט.

בחירת k

לאחר המימוש, הרצתי את האלגוריתם עם validation set שנלקח מה-training set לכל $1 \leq k \leq 100$. חישבתי את אחוזי הטעות מול ה-validate labels וקיבלתי כי עבור $k = 5, 10$ אחוזי הטעות מינימליים (כ-96.666% הצלחה). בחרתי באופן שרירותי ב-$k = 5$.

שיפורים נוספים

בנוסף, ניסיתי לבצע שני שינויים על הקלט על-מנת לנסות לשפר את הביצועים.

שינוי 1. לבצע נרמול של הקלט. ניסיתי לנרמל את הקלט לפי minmax, שהוביל לתוצאות גרועות, ולפי נרמול z score, שהוריד את אחוזי ההצלחה ל-95%. לכן, לא השתמשתי בנרמול כלל על הקלט האלגוריתם זה.

שינוי 2. לבצע feature selection, כלומר לנסות להוריד את כמות ה-features בקלט, לצורך שיפור ביצועים. ניסיתי להוריד כל אחד מה-features וגיליתי כי feature 3 (ה-feature באינדקס 2) הוביל לתוצאות הטובות ביותר מכל הניסיונות, רק שהוא עדיין לא הוביל לשינוי באחוזי הדיוק הכלליים.

Part II

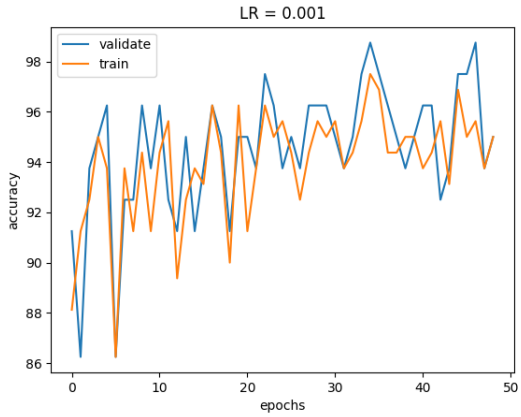
Perceptron

המימוש בקצרה

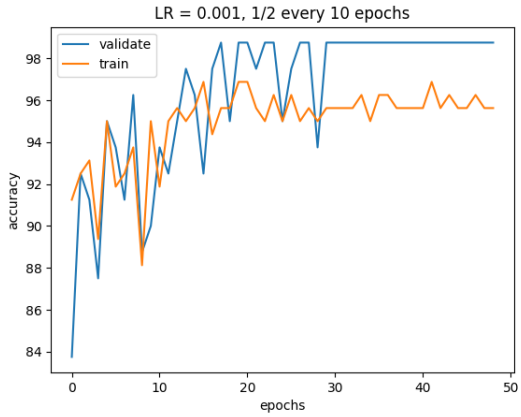
הגדרתי שלושה מערכי משקולות באורך 6 המאותחלים ל-0, וחילקתי את ה-data set ל-training set ו-validation set באופן אקראי (כ-$1/2$ לכל אחד). לאחר מכן ביצעתי 50 epochs באופן הבא: ערבבתי את ה-training set, ולכל נקודה ב-training set הוספתי עוד מימד (עבור ה-bias). חישבתי את $\hat{y}$ וכן אם היה צורך אז עדכנתי את המשקולות. לאחר כל העדכונים, עברתי על ה-validation set וחישבתי אחוזי הצלחה. בכל פעם שמרתי את המשקולות שהביאו לאחוזי הצלחה מקסימליים ב-validation set ואיתם ביצעתי את ה-prediction של ה-test set.

בחירת η

את ה-learning rate בחרתי באופן הבא: את אחוזי ההצלחה בכל פעם שמרתי במערך ויצרתי גרף. בכל פעם שיניתי את ה-learning rate עד שהגרף יראה עולה (ניתן לראות את איור 1). לאחר מכן ניסיתי להוריד את ה-learning rate בכל כמה epochs, וראיתי כי התוצאות הטובות ביותר מתקבלות עבור הכפלת ה-learning rate ב-$\frac{1}{10}$ (חלוקה ב-10) בכל 10 epochs. ניתן לראות את התוצאה באיור 2. בחרתי ב-50 epochs כיוון שראיתי ששיעור ההצלחה אינו משתנה מהותית (אם בכלל) לאחר מספר זה.



איור 1: perceptron with learning rate 0.001

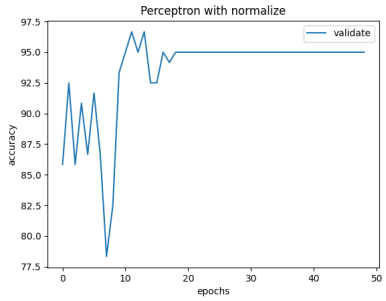


איור 2: perceptron with learning rate 0.001 that changes every 10 epochs

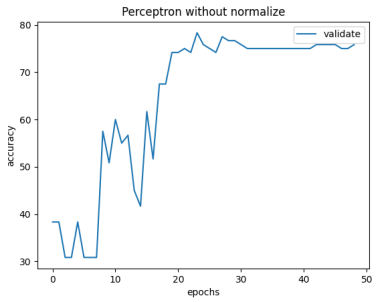
שיפורים נוספים

בנוסף, ניסיתי לבצע שני שינויים על הקלט על-מנת לנסות לשפר את הביצועים.

שינוי 1. לבצע נרמול של הקלט. ניסיתי לנרמל את הקלט לפי minmax, שהוביל לתוצאות גרועות, ולפי נרמול z score, שהגדיל משמעותית את אחוזי הדיוק. ניתן לראות כיצד השימוש בנרמול השפיע באיורים 3 ו-4.

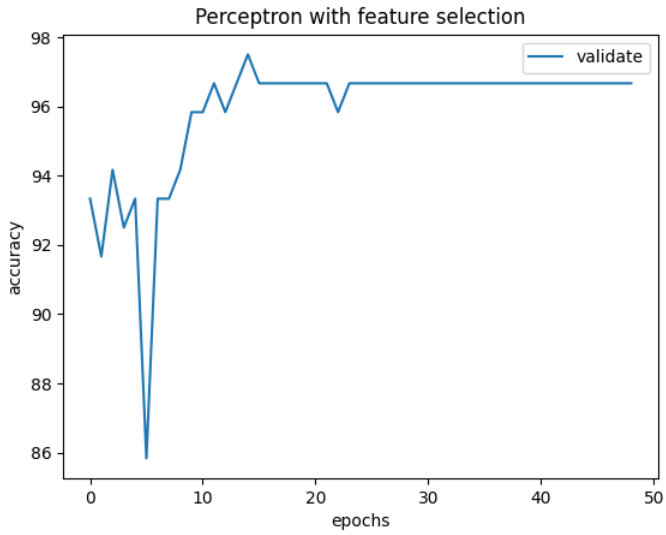


איור 3: perceptron with normalize

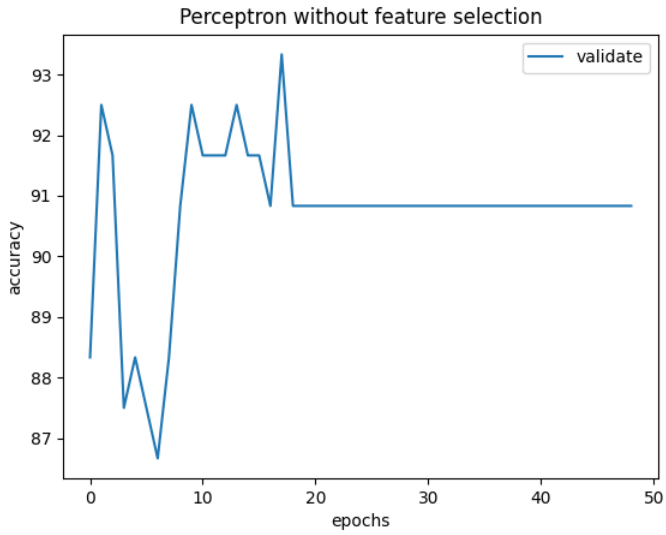


איור 4: perceptron without normalize

שינוי 2. לבצע feature selection, כלומר לנסות להוריד את כמות ה-features בקלט, לצורך שיפור ביצועים. ניסיתי להוריד כל אחד מה-features וגיליתי כי feature 3 (ה-feature באינדקס 2) הוביל לתוצאות הטובות ביותר מכל הניסיונות. ניתן לראות כיצד השימוש ב-feature selection השפיע באיורים 5 ו-6 (כאשר ה-feature selection הוא הורדה באינדקס 2).



איור 5: perceptron with feature selection



איור 6: perceptron without feature selection

Part III

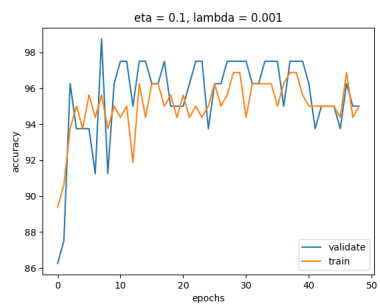
SVM

המימוש בקצרה

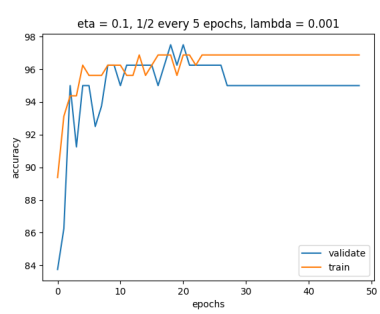
הגדרתי שלושה מערכי משקולות באורך 6 המאותחלים ל-0, וחילקתי את ה-data set ל-training set ו-validation set באופן אקראי (כ-$1/2$ לכל אחד). לאחר מכן ביצעתי 50 epochs באופן הבא: ערבבתי את ה-training set, ולכל נקודה ב-training set ביצעתי נרמול (לפי zscore) והוספתי עוד מימד (עבור ה-bias). חישבתי את r וכן עדכנתי את המשקולות. לאחר כל העדכונים, עברתי על ה-validation set וחישבתי אחוזי הצלחה. בכל פעם שמרתי את המשקולות שהביאו לאחוזי הצלחה מקסימליים ב-validation set ואיתם ביצעתי את ה-prediction של ה-test set.

בחירת η ו-λ

תחילה שיחקתי עם ערכי ה-η וה-λ לערכים כלשהם בטווח $[1, 0.0001]$. לאחר שראיתי שהגרף עולה מהר יחסית לאחר מכן נע בין ערכים גבוהים לקטנים (איור 7), התחלתי לנסות להוריד את ה-learning rate (η) בכל כמה epochs, וראיתי כי התוצאות הטובות ביותר מתקבלות עבור הכפלת ה-learning rate ב-$\frac{1}{2}$ (חלוקה ב-2) בכל 5 epochs. ניתן לראות את התוצאה באיור 8. בחרתי ב-50 epochs כיוון שראיתי ששיעור ההצלחה אינו משתנה מהותית (אם בכלל) לאחר מספר זה.



איור 7: svm with learning rate 0.1 and $\lambda = 0.001$

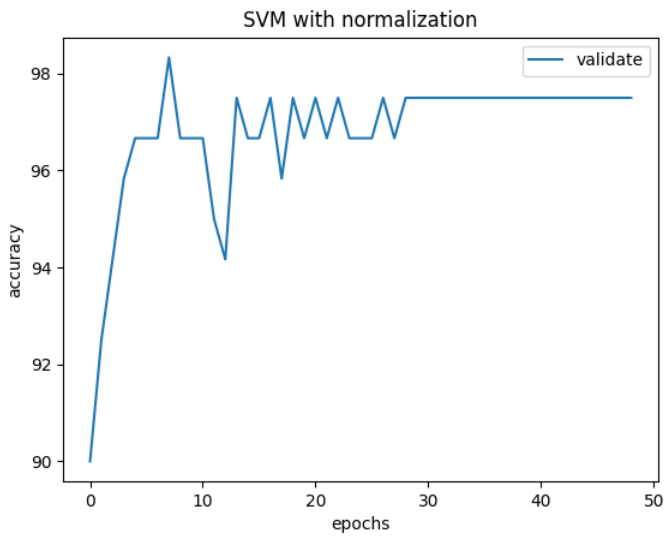


איור 8: svm with $\lambda = 0.001$ and learning rate 0.1 that changes every 5 epochs

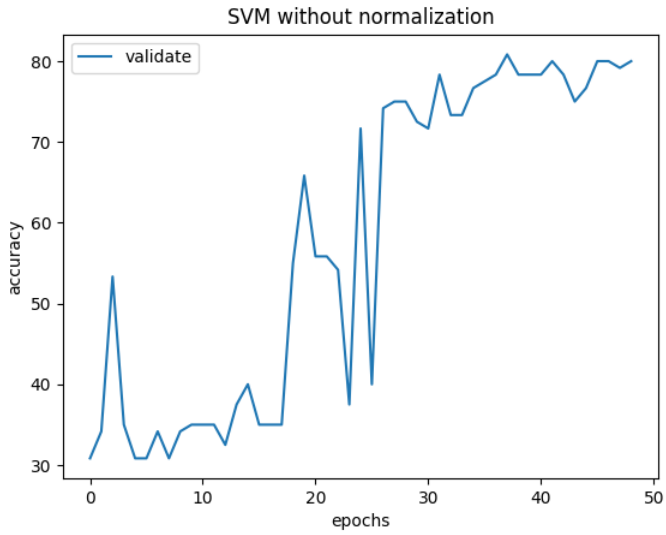
שיפורים נוספים

בנוסף, ניסיתי לבצע שני שינויים על הקלט על-מנת לנסות לשפר את הביצועים.

שינוי 1. לבצע נרמול של הקלט. ניסיתי לנרמל את הקלט לפי minmax, שהוביל לתוצאות גרועות, ולפי נרמול z score, שהגדיל משמעותית את אחוזי הדיוק. ניתן לראות כיצד השימוש בנרמול השפיע באיורים 9 ו-10.

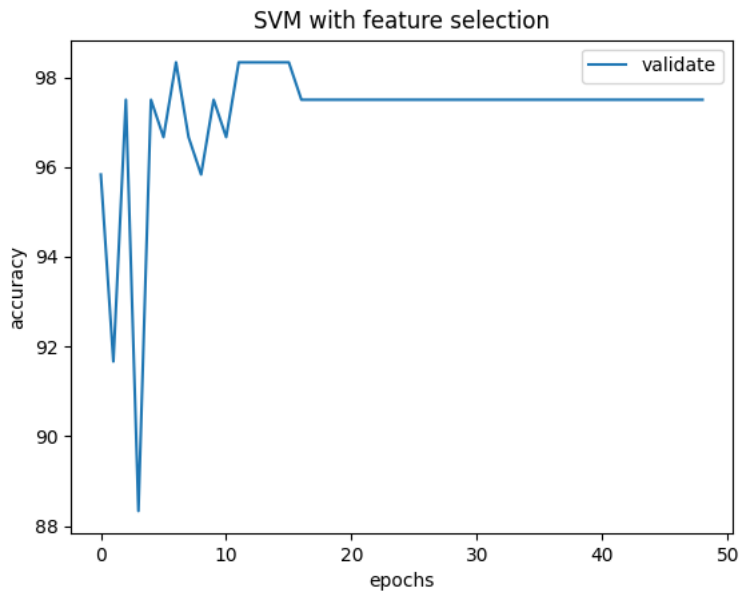


איור 9: SVM with normalize

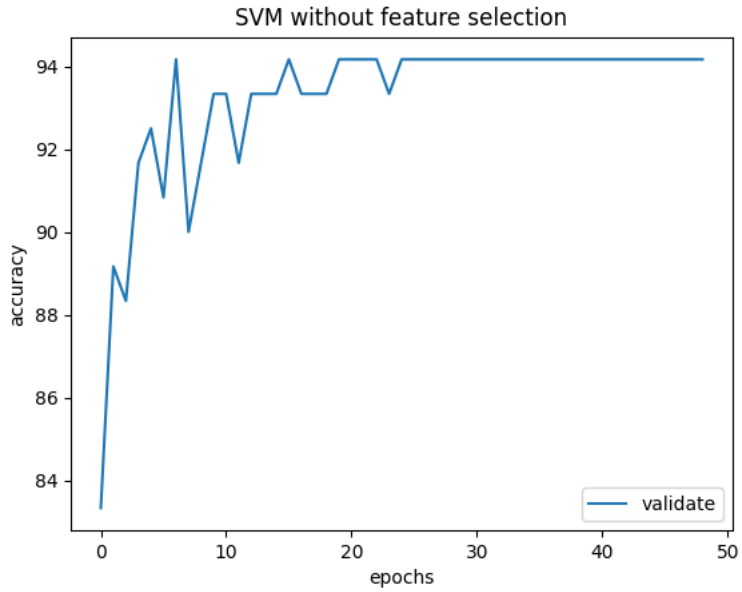


איור 10: SVM without normalize

שינוי 2. לבצע feature selection, כלומר לנסות להוריד את כמות ה-features בקלט, לצורך שיפור ביצועים. ניסיתי להוריד כל אחד מה-features וגיליתי כי feature 3 (ה-feature באינדקס 2) הוביל לתוצאות הטובות ביותר מכל הניסיונות. ניתן לראות כיצד השימוש ב-feature selection השפיע באיורים 11 ו-12 (כאשר ה-feature selection הוא הורדה באינדקס 2).



איור 11: SVM with feature selection



איור 12: SVM without feature selection

Part IV

Passive Aggressive

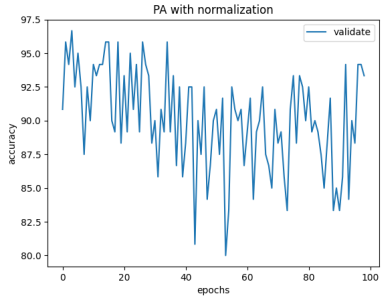
המימוש בקצרה

גדרתי שלושה מערכי משקולות באורך 6 המאותחלים ל-0, וחילקתי את ה-data set ל-training set ו-validation set באופן אקראי (כ-$1/2$ לכל אחד). לאחר מכן ביצעתי 100 epochs באופן הבא: ערבבתי את ה-training set, ולכל נקודה ב-training set ביצעתי נרמול (לפי zscore) והוספתי עוד מימד (עבור ה-bias). חישבתי את $\hat{y}$ וכן עדכנתי את המשקולות. לאחר כל העדכונים, עברתי על ה-validation set וחישבתי אחוזי הצלחה. בכל פעם שמרתי את המשקולות שהביאו לאחוזי הצלחה מקסימליים ב-validation set ואיתם ביצעתי את ה-prediction של ה-test set.

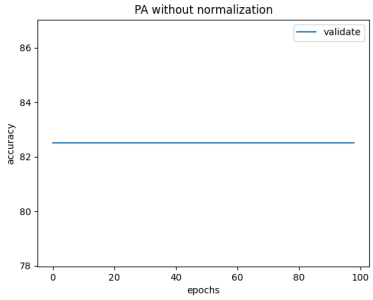
שיפורים נוספים

בנוסף, ניסיתי לבצע שני שינויים על הקלט על-מנת לנסות לשפר את הביצועים.

שינוי 1. לבצע נרמול של הקלט. ניסיתי לנרמל את הקלט לפי minmax, שהוביל לתוצאות גרועות, ולפי נרמול z score, שהגדיל משמעותית את אחוזי הדיוק. ניתן לראות כיצד השימוש בנרמול השפיע באיורים 13 ו-14.

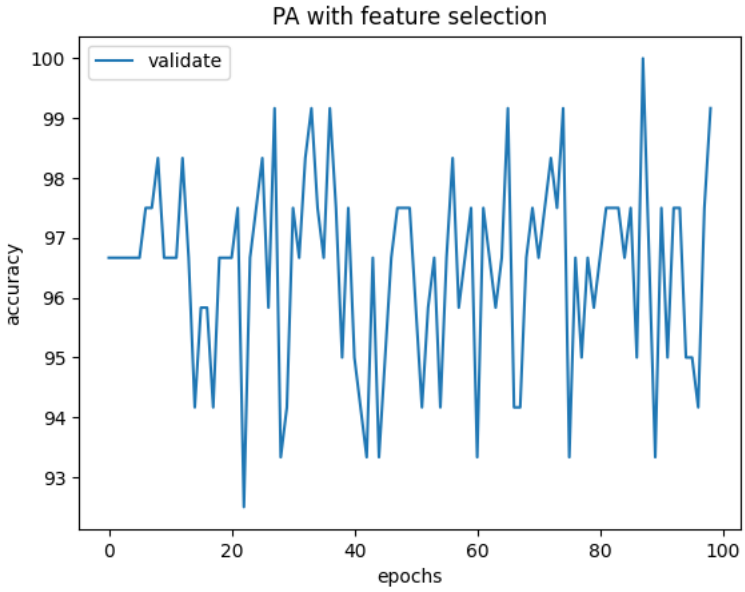


איור 13: PA with normalize

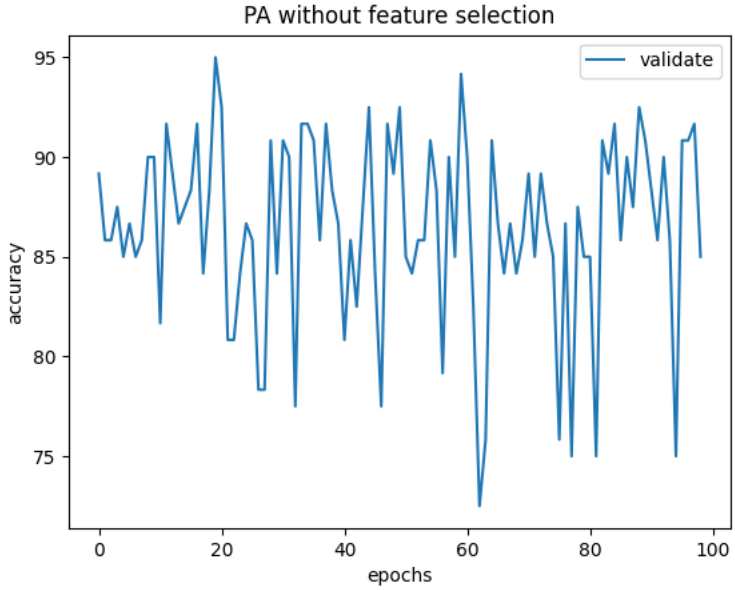


איור 14: PA without normalize

שינוי 2. לבצע feature selection, כלומר לנסות להוריד את כמות ה-features בקלט, לצורך שיפור ביצועים. ניסיתי להוריד כל אחד מה-features וגיליתי כי feature 3 (ה-feature באינדקס 2) הוביל לתוצאות הטובות ביותר מכל הניסיונות. ניתן לראות כיצד השימוש ב-feature selection השפיע באיורים 15 ו-16 (כאשר ה-feature selection הוא הורדה באינדקס 2).



איור 15: PA with feature selection



איור 16: PA without feature selection