

ישראל כהן – 205812290 ת.ב. 2

עבור כולם ביצעתי נורמליזציה של MAX-MIN לערכים 0-1, וחילקתי לולידציה ואימון ביחס של 80 20 אחוז.
לפני החלוקה הזו (שהיא עבור ה train בלבד) ביצעתי shuffle בתוך לולאה..
כמו כן, ביצעתי השמה בסוף כל שורה עבור 0 W ועבור 1 R.

:KNN

- כדי למצוא את K:

בקצרה:

יצרתי מערך דו ממדי המכיל את ערכי האימון, ונתון נוסף לאחר מכן של class בכל שורה. (התאמתי את ה class לשורת האימון המתאימה)

ביצעתי מספר פעמים 1000 הרצות וכדומה, כדי לוודא את התוצאה, ראיתי שא $K = 10$ הוא הטוב ביותר שאני מקבל (קצת יותר טוב מ $K = 8$),
אשר ממנו והלאה הערכים מתחילים לרדת, והם רק עולים מ $K = 1$ עד אזור $K = 5, 6$.

בדקתי עבור K רץ מ 1-15 (לפעמים יותר מ 15, ראיתי שזה לא רלוונטי בפועל).

כדי לוודא את התוצאות, ביצעתי נורמליזציה נוספת בשילוב והיעדר של MAX-MIN וקיבלתי אותן התוצאות.

בפירוט: כחלק מהפתרון, עבור כל נקודה אני מחזיק מערך דו ממדי (מתעדכן - זמני)
המכיל מרחק אוקלידי (מהנקודות של ה train – 284 נק') class i עבור אותה הנקודה שמצאנו את מרחקה האוקלידי.

כעת עבור כל נקודה, אני ממיין לפי מרחק אוקלידי נמוך ביותר, בוחר ערכי K שונים (מהנמוך ועד $K = 15$), מבצע השמה של המידע הזה עבור אותה נקודה למטריצה המכילה 2 שורות.
בשורה הראשונה 15 ערכי מרחקים מינימאליים, ושורה שניה מתחת בהתאמה – ה class עבור כל מרחק.

כעת, אני מבצע בדיקה מהם רוב ה classes של ה K שבחרתי, זה ה class עבור הנקודה, בודק אם צדקתי או טעיתי, אם צדקתי, כותב 1 במיקום ה K (במטריצה חדשה בגודל מספר הנקודות הנבדקות) אם טעיתי – 0 (כך עבור כל ה 15 K-ים) ועובר לנקודה הבאה.

בסוף התהליך אני מחזיק טבלה (מטריצה אחרת) שכל אינדקס של עמודה שלה מבטא מספרי הצלחה עבור K השווה לגודל האינדקס+1 (אינדקס עמודה) ובדיקה של הסכומים של ההצלחה של כל עמודה תבחר את ה K האופטימאלי. (הסכום הגבוה מכולם מבטא הצלחה גבוהה ביותר)

בצורה הזו בסט האימון אני בעצם בודק את 71 הנקודות האחרונות (לפי החלוקה וה shuffle), ובוחר K אופטימאלי.

זה מתבצע כך שאני משווה עבור כל 71 הנקודות את ה class החזוי לפי כל אחד מה K, עבור הצלחה אני מקבל 1+ בנקודות, ה K שיתן לי את הסכום הגבוה ביותר, הוא המוצלח מכולם, הוא בעצם המספר הקרוב ביותר לאמת.

מצאתי כי $K = 10$, נותן לי בממוצע אחוז דיוק של כ 79.5.

לאחר מציאת ה K אני בודק את ה TEST בצורה דומה ומוציא פלט של ה class שלו לפי המרחק האוקלידי מהנקודות של ה training set ב K הנבחר.

לשני האלגוריתמים האחרונים (בעיקר לPerceptron), שמתי לב ש W לפעמים קופץ בגלל צורת הלמידה הזו, לכן ביצעתי תנאי של לקיחת משקלים (לטובת test) במצב דיוק אופטימלי כהגדרתי הנמצא עבור epoch נוכחי.

:Perceptron

מחלק ל 20 80, בכל epoch ערך ה ETA שהתחיל מ 0.2 יורד כל 50 epoch-ים בעשירית מגודלו, כך ישפיע פחות בהנחה שהמשקלים מתכנסים. מספר ה eta נבחר כיון שביצעתי נורמליזציה לערכים 1-2, ונשמע לי ראלי, וגם אני מוריד אותו בגודלו גם אם הוא גדול מדי, אז אני מרגיש עם זה בסדר 😊

מריץ 1000 epoch, (אצלי הקוד רץ 10 שניות) ה W נבחר החל מדיוק של val ו train לפי החלוקה לעיל מעל 70 אחוז דיוק בערך, עם אופציה ל W עם אחוזי דיוק גבוהים יותר אם במקרה יש נתונים טובים, והנ"ל חסום באזור 90 אחוז דיוק לטובת מניעת מצב שהתוכנית נעולה על הערכים יותר מדי (אני רוצה לאפשר מרווח נשימה לנקודות, ולא חסם הדוק מדי גרפי, כי יש מצב טוב שלא הכל נמצא בדיוק בקו הגבול של ה TRAIN שלי).

בעזרת ה W המתקבל אני מוצא את classn של ה testn.

:PE

כמו perceptron, במקום eta יש tau, הרצתי 1000 epoch-ים לשם דיוק, כי גם כך ה TAU לא זז בפראות יותר מדי כי זה כל הרעיון שלו – דיוק ושיפור לא מסיבי. בחרתי W בצורה דומה לPerceptron.