**בינה מלאכותית תרגיל 3**

חלק ב':

שאלה 3:

2. חשוב לשמור על עקביות כדי שכל הניסויים יבוצעו על אותן קבוצות מבחן ונוכל ככה לבדוק את טיב המסווגים.

שאלה 5:

3. התקבלו הערכים הטובים ביותר עבור k=3.

4. ניתן להבחין בירידה אחרי k=3, אבל כולם דיוק גבוה מאוד. ערכי המקסימום הם 0.954 עבור k=3 והמינימום הוא 0.935 עבור k=13. ניתן להסביר זאת על ידי התייעצות עם שכנים יותר "רחוקים" שפוגעת בדמיון לבעיה שיש לנו-הדוגמאות לא מספיק דומות.

שאלה 7:

4. הצליח הכי טוב הKNN עבור k=3.

חלק ג':

המסווג שבחרנו פועל בצורה הבאה:

חלק א': עיבוד הנתונים:

משתמשים בFFT לצורך הוספת פיצ'רים חדשים למידע המתקבל.

בשלב הראשון אז עושים FFT לכל אחת מהדגימות. מוצאים את נקודות הקיצון בFFT(מקומיות-פיקים) ולוקחים את 30 הערכי הקיצונים הראשונים ושמים את הערכים עצמם והאינדקסים שלהם במערך זמני.

לאחר מכן מנרמלים את הערכים.

הרעיון שעומד מאחוריו הוא שאנחנו רוצים לזהות דגימות בעלי קצב לא תקין(קצב לב מהיר או איטי וכו') ולכן זה מתבקש להשתמש בFFT שנותן תדרים בsignal. נבחרים הערכים הראשונים כדי להתמקד בתדרים הנמוכים וממבט בגרף שראינו שרוב ערכי הקיצון ממוקמים בהתחלה.

חלק ב'-בחירת פיצ'רים חשובים:

השתמשו בrandom forest classifier ומחלקה של sklearn שנקראת selectFromModel כדי לבחור את 50 הפיצ'רים החשובים מהדגימה המקורית. חיברנו מערך 50 הפיצ'רים החשובים עם המערך מהFFT ועליו עושים את האלגוריתם עצמו.

האלגוריתם עצמו:

And לוגי בין שני האלגוריתמים הבאים:

KNN עם k=3

Random\_forest\_classifier עם מאה עצים רנדומיים, ללא גיזום.

ביצענו(בין היתר) את הניסויים הבאים, עם corss validation של 2-3(משתנה) בכל מיני קלטים אקראיים(ביחס הנכון בין חולים ובריאים):

* Id3 מול Random\_forest -נראה שRandom עובד יותר טוב באחוזים אז בחרנו אותו.
* מספר העצים האקראיים-ראינו שמספר של 100 נותן תוצאות טובות ושמעבר אליו אין שיפור ניכר (עד 200) ושפחות מזה נותן תוצאות פחות טובות.
* גיזום-נראה שגיזום מכל סוג שהוא רק פגע בדיוק אז שלא לבצע גיזום.
* כמה פיצ'רים מובילים לבחור-ראינו שמספרים בין 50 ל60(בדקנו בקפיצות של 10) עובדים בצורה הטובה ביותר. על ידי הרצות cross validation עם מסווגים שונים ומספר פיצ'רים שונה וראינו ש50 מגיע תמיד לתוצאות טובות.
* משרטוט הפונקציות ראינו שהפורמט שלהן עשוי להתאים לFFT. שימוש בrandom forest ללא FFT הגיע לדיוק נמוך משמעותית מאשר KNN "נאיבי", עם זאת KNN בעצמו לא הושפע.
* KNN הגיע לרוב לתוצאות טובות יותר מאשר עץ(מכל סוג), אבל ביצוע and בין שתי ההחלטות הביאה לתוצאה הטובה ביותר. לעומת זאת or לוגי הביא לתוצאות איומות.
* שימוש בKNN הצליח בצורה הטובה ביותר עם K=3 וללא משקול המרחקים(שרק פגעו בביצועים), ניסיונות למשקל או לשנות את K פגעו בדיוק.
* ניסינו להשתמש לKNN בנורמת מנהטן במקום אוקלידית-זה שיפר ביצועים עבור KNN על הנתונים המקוריים אבל אחרי FFT לא היה שיפור.