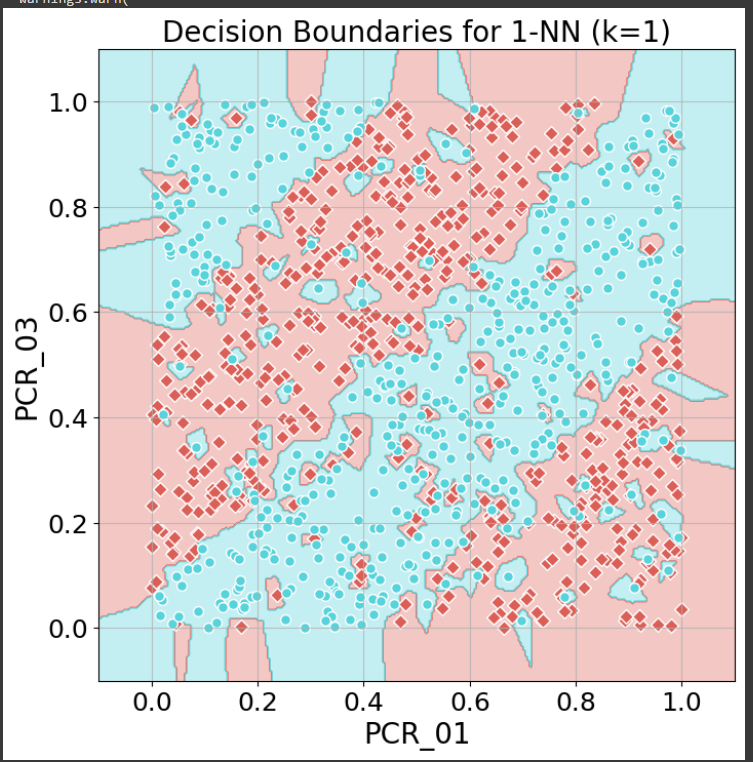
Visualization and basic analysis

(Q1)



(Q2)

A graph with blue and orange lines

Description automatically generated

העקומה הכתומה מציגה לנו את את התוצאה של הדיוק (הממוצעת לפי VC) על הvalidation set עבור כל מודל knn (עבור k שונים).

אנחנו רוצים לבחור את הk שנתן לנו את הדיוק הכי טוב על הvalidation set, לכן ניקח את הנקודה המקסימלית על העקומה הכתומה ונמצא את הk שנותן לנו את הדיוק המקסימלי.

עשינו את זה כמובן בעזרת קוד פייתון וקיבלנו :

A screenshot of a computer

Description automatically generated

ראינו בשיעורי בית הקודמים שK נמוכים בKNN גורמים לoverfitting כי יש משקל גדול יותר למספר קטן של השכנים הקרובים ביותר ולכן ההשפעה של כל שכן קרוב גדולה יותר ואם יש חריגים בtraining set זה עלול ליצור איזור סיווג שגוי באיזור שלהם.

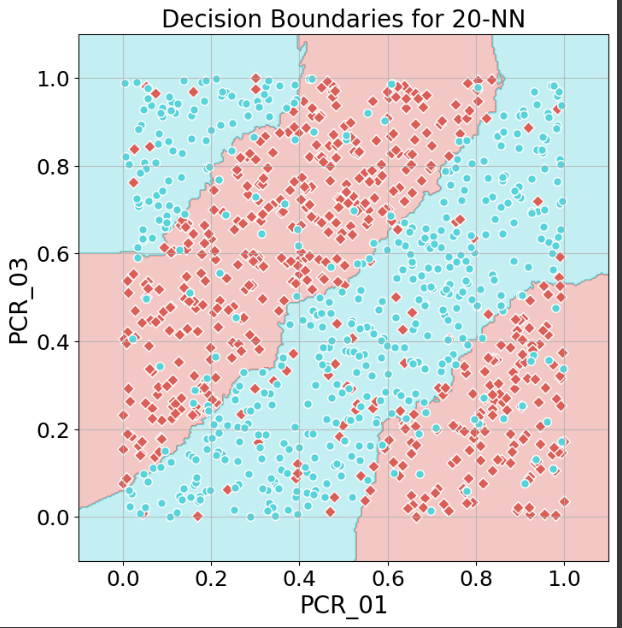
ראינו שהדיוק מאוד טוב על הtraining set לkים קטנים אבל לא בהכרח טוב בהכללה שלנו כלומר על test set שמייצג לנו את הכללה. במקרה שלנו בcv הtest הוא הvalidation data וניתן לראות שלkים קטנים יש דיוק גבוה על הtraining set ודיוק נמוך יותר על הvalidation set שבמקרה שלנו הוא הtest set.

עבור kים גדולים נוצר underfitting כי נילקחים בחשבון יותר ויותר שכנים עד שנקלחים גם שכנים רחוקים שלא רלוונטים לסיווג של אותה הנקודה, זה יוצר מצב שבוא הסיווג של אותה נקודה לוקחת נקודות לא רלוונטיות שמעפילות באיזהו שלב על הנקודות הקרובות ביותר שלהם יש חשיבות בסיווג ולכן נוצר מצב של חוסר התאמה של המודל לtraining set וגם לtest set כי גם המודל בזמן האימון לוקח בחשבון נקודות לא רלוונטיות וככה יפעל גם המודל על הtest set.

ניתן לראות גם בעקומה שלנו שככל שk גדל אנחנו מקבלים שהדיוק הולך וקטן והעקומות של שניהם מתכנסים לאותה נקודה של דיוק נמוך מאוד.

נשים לב שבמקרה של ערכי k גדולים, ההתכנסות של העקומות זהה ,הסיבה לכך היא שהמודל מתייחס למספר גדול של שכנים, כולל אלה שנמצאים רחוקים ופחות רלוונטים וכתוצאה מכך המודל נעשה כללי מדי ואינו מסתגל היטב לניואנסים של מערך האימונים, מכיוון שהוא לא מתאים התחזיות של המודל אינן מושפעות במידה רבה מהמאפיינים של נתוני האימון, מה שמוביל לציוני דיוק דומים ונמוכים.

(Q3)

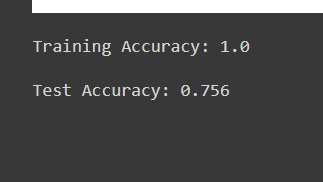


A black background with white text

Description automatically generated

(Q4)

התוצאות של המודל 1NN בסעיף 1 הן :



ניתן לראות שהדיוק לtraining set מאוד גבוה, בעצם יש פה overfitiing, כי המודל לא עושה הכללה טובה ולכן הדיוק שלו בtest set הוא נמוך. בסעיף 3, ניסינו למצוא את הk המתאים ביותר לאמן את המודל כדי לקבל הכללה טוב יותר וניתן לראות שהדיוק בtraining set קטן יותר אבל הדיוק על הtest set הוא גבוה יותר ולכן זה מחזק את התוצאות שלנו בבחירת הפרמטר הטוב ביותר למודל שלנו.

ניתן לראות בתוצאה האחרונה שאזורי הסיווג יותר "חלקים" לעומת התוצאה של סעיף 1 שמתייחס גם לאיזורים עם רעש ולכן עובד בדיוק נמוך יותר על הtest set, על data שהוא לא התאמן עליו ודייק את עצמו אליו.

(Q5)

A grey background with white text

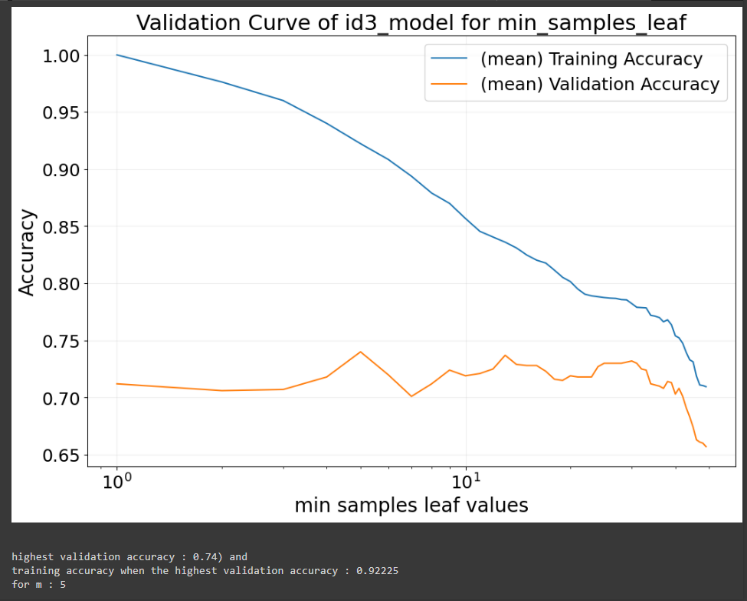
Description automatically generated

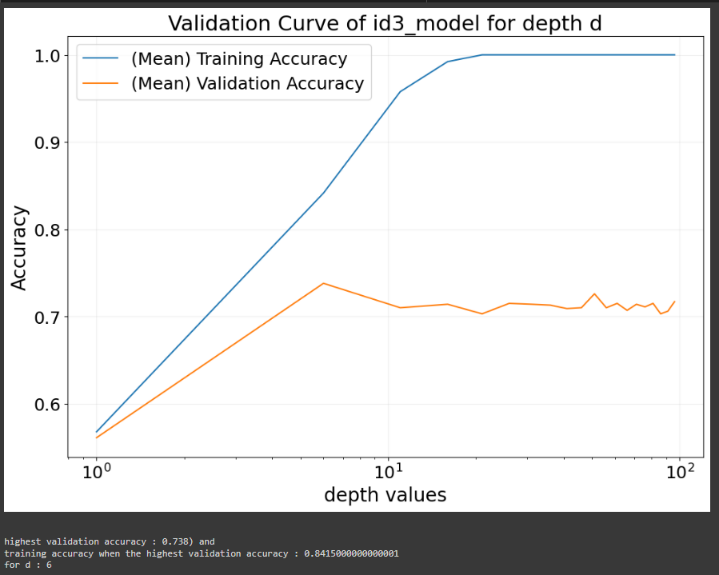
A diagram of a diagram

Description automatically generated with medium confidence

(Q6)

בהתחלה התייחסנו לפרמטרים כבלתי תלויים אחד בשני ובדקנו כל פרמטר בניפרד בשביל לבדוק לכל פרמטר בניפרד מתי מתקבל דיוק מקסימלי כמו שעשינו בשאלה 2 וזה כדי להבין פחות או יותר הטווחים שכדי להשתמש בהם.

התוצאה עבור max\_depth, והתוצאה עבור min\_samples\_leaf:



A screen shot of a computer

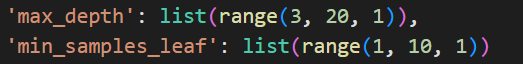
Description automatically generatedA screenshot of a computer

Description automatically generated

אפשר לראות כmin\_samples\_leaf גדול מ20 אז מתחילה להיות ירידה בדיוק,

ועבור max\_depth מעומק 5 הוא נשאר יציב פחות או יותר.

1. בחרנו שני טווחים מתאימים :



ב+ג.

A screenshot of a graph

Description automatically generated

ד+ו.

underfitting יתקבל כאשר הפרמטרים לא יאפשרו למודל להתאמן מספיק טוב על הtraining set, מצב בו הדיוק גם על הtraining set וגם על הtest set נמוך.

כאשר העומק של העץ קטן מאוד אז המודל לא יוכל לבצע מספיק החלטות כדי שהוא יוכל לחזות משהו, באותו אופן, כאשר המספר המינמלי שעלה יכול להחזיק הוא גדול מאוד זה אומר שהסיווג ינתן על קבוצות מאוד גדולות ולא יאפשר לפיצולים נוספים הדרושים לשם חיזוי שונה לנקודות ממחלקות שונות.

בהתאם לטווחים שלנו :

Max\_depth=3

Min\_samples\_leaf=9

(\*נשים לב, שערך קטן יותר לmax\_depth וערך גדול יותר לmin\_samples\_leaf כנראה יתן לנו overfitting גדול יותר, כלומר דיוק קטן הרבה יותר, בגלל זה ערכים אלה לא כלולים בטווחים שלנו כי לנו צורך בהם לצורך מציאת הsweet spot) .

ה+ו.

Overfitiing מתקבל כאשר המודל מתאמן יותר מידי טוב על הtraining set ומתאים את עצמו לנקודות אלה כך שהמודל לא מתאים לחזות נקודות שהוא לא יתאמן עליהם.

מצב זה יתאפשר כאשר הmax\_depth מספיק גדול שמאפשר לקבל מספיק החלטות עד לקבל החיזוי וmin\_samles\_leaf מספיק קטן כדי שנוכל לעשות מספיק פיצולים על נקודות ממחלקות שונות.

בהתאם לטווחים שלנו :

Max\_depth=19

Min\_samples\_leaf=1

(\*נשים גם פה לב שאם Max\_depth גדל אז המודל יתאים את עצמו יותר טוב על הtraining set ויקבל דיוק 1 כמו שקיבלנו במקרה ש Max\_depth=19 אבל הדיוק על הvalidation set יהיה קטן יותר, שוב אין לנו צורך בנקודות האלה למציאת הsweet spot ולכן לא בטווח הנבחר)

(Q7)

מספר הקובינציות שנבדקו הוא כמספר הערכים האפשריים לכל פרמטר,

לפרמטר Max\_depth הטווח הוא מ-3 ל-20 (לא כולל )ולכן יש 17 ערכים אפשריים.

לפרמטר Min\_samples\_leaf הטווח הוא מ1 ל10 (לא כולל) ולכן יש 9 ערכים אפשריים.

סה"כ יש 153 קומבינציות אפשריות (17\*9).

הוספת פרמטר נוסף תגדיל את העלות החישובית ואת הזמן הנדרש למציאת השילוב האופטימלי, בנוסף פרטמר נוסף יכול להיות בעל טווחים אפשריים גדולים ואף להגדיל את הטווחים של הפרמטרים האחרים(בגלל התלות שלהם) ולכן זה לא רק להכפיל את מספר הקומבינציות האפשרי של שני הפרמטרים שלנו בטווח המספרים של הפרטמטר החדש, אלא לבדוק מחדש אלו טווחים יהיו מתאימים יותר כדי לקבל דיוק גבוה יותר בcross validation.

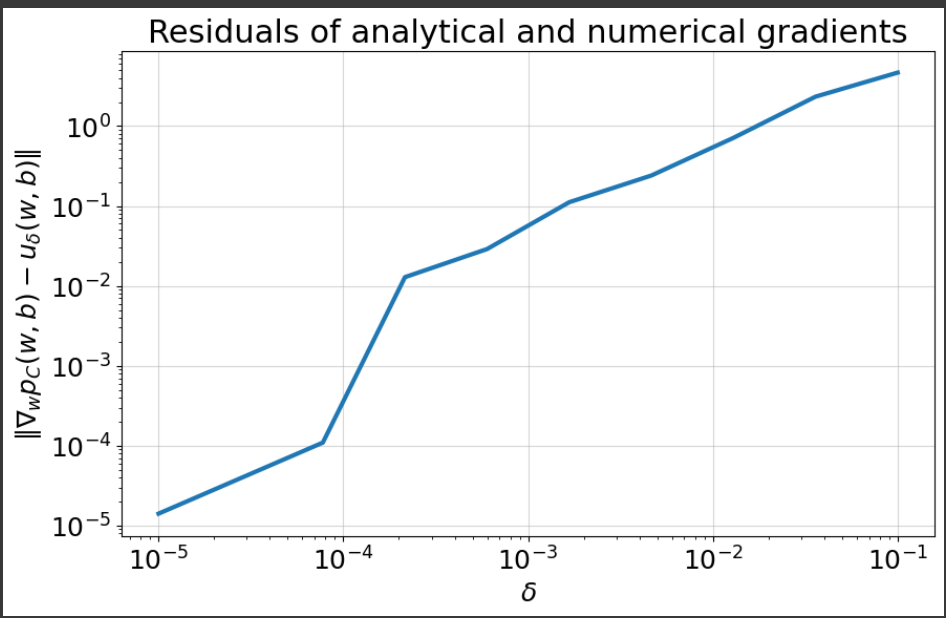
(Q8)

קיבלנו test accuracy of 0.772



(Q9)

הגדרת הנגזרת כמו שלמדנו בקורסים קודמים משאיפה את לאפס (), החישוב הנומרי שהוצג לנו הוא קירוב לנגזרת החלקית לפי וככל ש קטן נצפה שהחישוב הנומרי יהיה יותר מדויק ולכן החישוב האנליטי והנומרי יהיו קרובים יותר למרות הרנדומליות של .



העקומה הבאה מראה ששתי השיטות של החישוב קרובות ככל ש קטן, מה שמאפשר את נכונות המימוש של הגראדיאנט שלנו.

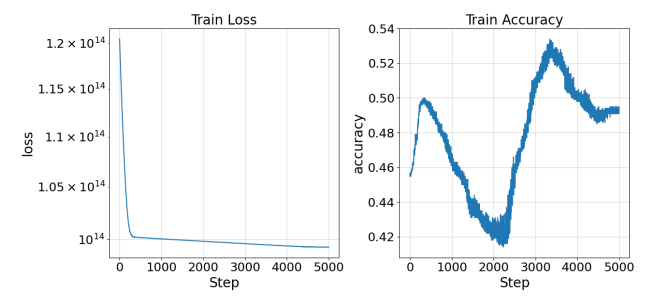
ככל ש- גדל אנחנו רואים שהקירוב בין הגראדיאנט האנליטי והנומרי גדל כי הדיוק של הנומרי קטן ולכן אנחנו מצפים שהקירוב ביניהם יגדל.

נשים לב שאנחנו מקבלים בכל הרצה תשובות שונות כי אנחנו לוקחים דוגמאות באופן רנדומלי ובודקים את הערכים שמתקבלים ולכן בכל פעם יכולים להתקבל ערכים שונים,

מה שחשוב שבכל ההרצות אנחנו מקבלים שככל קטן הרנדומליות של לא משפיעה ואנחנו מקבלים קירוב טוב בין הגרדיאנט שלנו האנליטי לבין הנומרי.

יכול להיות שיש גם רעשים מהחילוק במספרים אינפינטימליים במהלך תהליך הקירוב אבל אנחנו מסתכלים על ההתנהגות הכללית של העקומות שקיבלנו בשביל לאמת את החישוב האנליטי שלנו לגרדיאנט - ככל קטן הresiduals קטן.

(Q10)



אנחנו מנסים למצוא את ה המינמלי שיביא את למינימום את הביטוי של soft-SVM formulation ואנחנו מבצעים זאת על ידי שימוש בSGD.

SGD עובד באיטרציות(steps), בכל איטרציה לוקח כמות(batch) דגימות מסויימת (ושונה) ובודק את הgradient של הביטוי על פי ומעדכן את שנותנים תוצאה מינמלית יותר לביטוי.

מבצע זאת steps פעמים עד שהוא מגיע ל המינמלים האפשריים בהתאם למספר הצעדים ולגודל כל צעד.

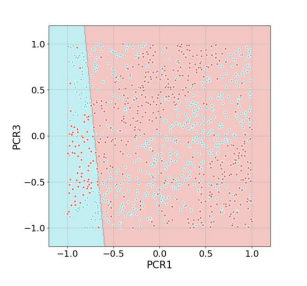
מספר צעדים נמוך לא יאפשר להגיע ל מינמלי ואנחנו רואים זאת בגרף הloss שרק ממספר צעדים מסויים מגיעים לאיזהו loss מינמלי, בנוסף ניתן לראות שמדובר על loss מינמלי בהתאם לבעיה הנתונה, בגלל שC מאוד גדול, אז יש משקל גדול על הhinge loss ובגלל שהdata לא פריד אז לא ניתן להגיע לloss נמוך והגרך מציג loss יחסית גבוהים.

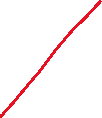
לפי הגרף של הloss אנחנו יכולים לראות שמתבצע מינמיזציה בכל שלב באלגוריתם כי הגרף יורד וככל שעושים יותר צעדים אז הloss יורד, ניתן גם לשים לב שממספר צעדים מסויים העקומה יורדת בצורה איטית הרבה פחות כי אחרי מספר מסויים של צעדים ההגראדיאנט קטן ולכן קצב ההתכנסות לw האופטימלי קטן.

אחרי נקודה מסויימת (בערך 300 איטרציות) מגיעים ל loss מסויים והוא נשאר יחסית יציב וזה מתאים לציפיות שלנו כי הdata לא פריד ולכן לא משנה כמה איטרציות נמשיך לעשות לא נוכל לקבל loss ששואף לאפס כמו שהיינו רוצים כי בבעיה הנ"ל C מאוד גדול ונותן משקל גדול יותר על הhinge loss.

בגרף של הaccuracy, אנחנו רואים עקומה תלולה ואנחנו יכולים להסיק שזה בגלל שהdata לא פריד וקשה למצוא hyperplane שיתן דיוק טוב, רואים שהדיוק הוא מאוד נמוך גם בנקודות הגבוהות ביותר. התלילות של העקומה נובעת מכך שאנחנו ממשיכים לעשות עוד איטרציות של הSGD עם משקל גדול יותר על הhinge loss ולכן הדיוק יורד כאשר אנחנו עושים יותר צעדים ממה שצריך, העקומה יורדת ועולה כי הdata פריד לסירוגין והדיוק משתנה בהתאם.

ניתן לראות את הקשר בין שני הגרפים : כאשר מגיעים לנקודה יציבה בגרף של הloss אנחנו מגיעים לנקודת מקסימום הראשונה בגרף של הaccuracy וזה כי הגענו למספר צעדים שנותן דיוק יחסית טוב (לבעיה) אבל בגלל שאנחנו ממשיכים לבצע עוד איטרציות בSGD אנחנו מתחילים לקבל ירידה בaccuracy כי אנחנו משנים את w עוד ועוד. באיזשהו נקודה אנחנו מקבלים עלייה נוספת בגרף הaccuracy וניתן להסיק אולי שהכיוון של הw עשה שינוי של כמעט 90 מעלות ולפי הdata כנראה הw ההתחלתי היה בכיוון כזה שנותן דיוק פחות או יותר של 0.5:

**

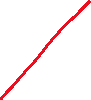


*ולאט לאט הוא שינה את הכיוון שלו כי הוא נתן משקל גדול על נקודות "חריגות"* לכן הaccuracy קטן עד

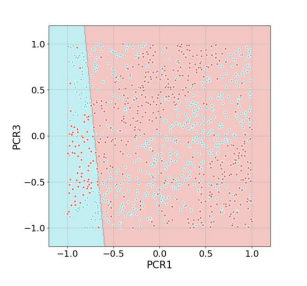
*שהוא הגיע לנקודת דיוק מאוד נמוכה:*

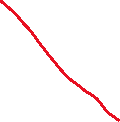
*A diagram of a graph

Description automatically generated*



ואז יש עלייה עד לנקודה עם דיוק גבוהה שוב 0.53:

**שניתן לראות שזה פחות או יותר גם חצי נקודות מסווגות נכון וחצי לא.



ושוב יש ירידה כי יש עוד איטרציות בSVD עד שהוא מגיע לw כמו בתמונה בשאלה ואכן ניתן לראות שהדיוק יורד מ0.5.

זה מתאים לציפיות שלנו כי C מאוד גדול וזה נותן משקל לנקודות "חריגות" בloss hinge והניסיון לבצע אופטימזציה על זה ישנה את w בלי סוף כי אין w מינמלי אחד.

היינו מצפים שכאשר הדיוק יורד הloss גם יעלה, אב למה זה לא קורה ?

הינו יכולים לחשוב שזה מגדול הצעד אבל אנחנו רואים שגודל הצעד יחסית קטן ולכן זה לא מה שגורם להתנהגות הזאת, מה שגורם להתנהגות הזאת היא שאנחנו מחשבים את בעיה האופטימזציה על קירוב לo/1 loss בעזרת הhinge loss, שגם כאשר הסיווג הוא נכון הוא נותן משקל לנקודות שנמצאות בקרבת הmargin ולכן הloss ממשיך לרדת כאשר הדיוק יורד.

מסקנה זאת גם מתאימה לתמונה של הdata שאנחנו רואים - הנקודות מאוד צפופות ולכן הרבה נקודות ימצאו בקרבת הmargin.

(קרבת הmargin היא לא 1 ואז זה אומר שלכל הנקודות שהמסווגות נכון יש משקל על פי המרחק ? )

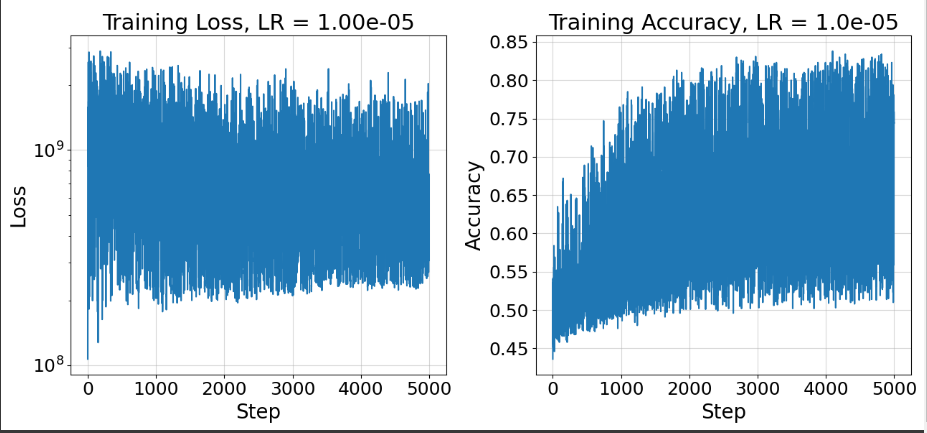
(Q11)

A graph of a training curve

Description automatically generated with medium confidence

A graph of steps and steps

Description automatically generated



\*\*\*\*\*\*\*\*\*\*תשובה של גל \*\*\*\*\*\*\*\*\*\*\*\*

(Q12)