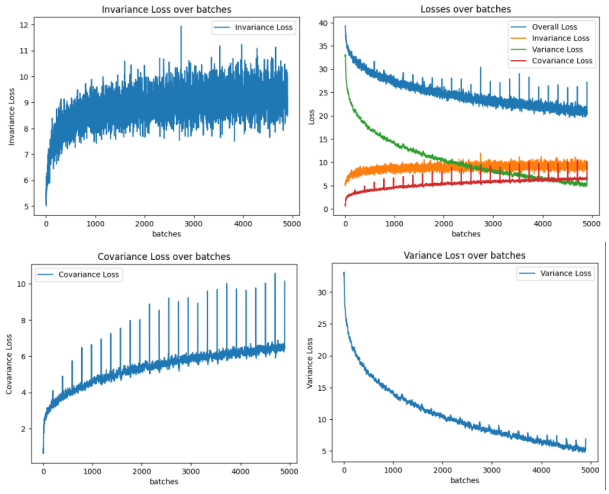
$AML - SSL$

18 במאי 2024

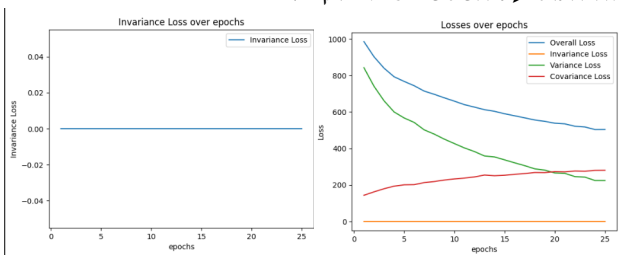
איתמר שכטר $itamar.sh$ 315092759

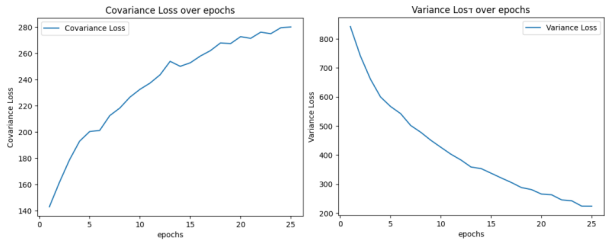
$Q1$: $training$-

זה הלוס על האימון ב25 איפוקים:



זה הלוס על הטסט ב25 אפוקים:

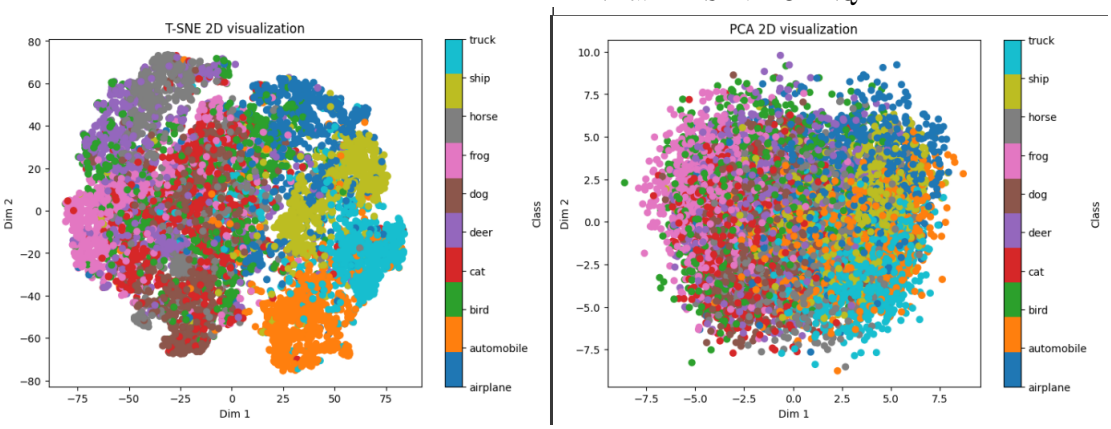




אפשר לראות את הניסיון של המודל להגיע למינימום ושיווי משקל בין הלוסים, כשמורידים בכיוון לוס אחד השניים האחרים לא תמיד מגיבים על זה טוב ואז צריך לתקן.

יש פה מטרה להוריד את הוריאנס תוך ניסיון לא לעלות א ה$covariance$ וה$invariance$ יותר מדי.

$Q2$: PCA וTSE ייצוגים:



לדעתי $T-SNE$ מצליח לבצע ויזואליציה הרבה יותר טובה, יש אזורים שממש נראים שייכים ל$class$ מסוים. לעומת זאת בpca כל המחלקות די סבוכות אחת בשנייה.

כן נגיד לטובת הpca שהוא שם מחלקות באזורים מסוימים ולא לגמרי מפזר אותם על כל השטח ולכן כן יש איזשהי הצגה של המיקום של מחלקות מסוימות במרחב.

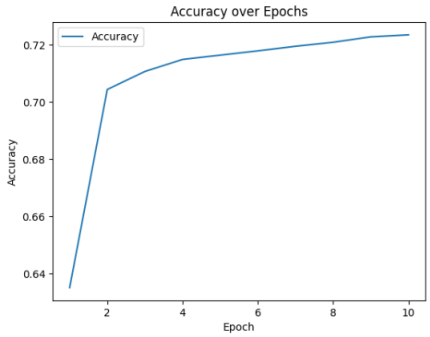
מה שיפה לראות זה ששתי השיטות שמות את אותן מחלקות על אותם אזורים של התמונה. לדוגמא מחלקות של $frog, dog, cat$ בצד שמאל בשניהם ו$truck, ship, airplabe$ בימין בשניהם.

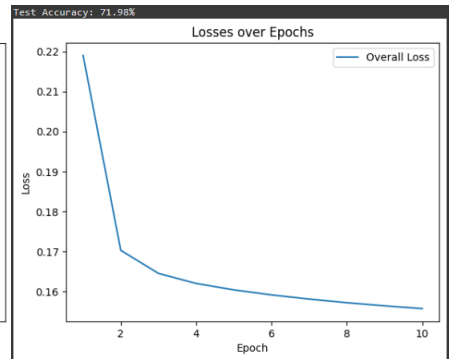
אצל הpca יש ערבוב בין dog וcatוקשה להפריד ביניהם וככה גם ב$ship$ ו$auto, obile$ בניגוד ל$T-SNE$ שמצליח להפריד גם בין מחלקות ששני השיטות מסכימות עליהן שהן דומות, או לפחות קרובות במובן מסוים.

כן נחמד לראות שמחלקות דומות כמו חיות וכלי רכב מקבלים אזורים שונים על גבי הויזואליזציה.

למה $T-SNE$ טובה יותר? אני משערך שבגלל זה שהיא לא לינארית בניגוד לPCA וזה מאפשר לה יותר אקספרסיביות, וגם בגלל זה שהיא נותנת שיערוך גם גלובלי וגם לוקאלי. כמובן שגם השיטה הזאת פיספה באזורים מסוימים אבל בגדול התוצאה שלה די טובה.

$Q3$:





הצלחנו להגיע ל72 אחוז דיוק. שזה די גבוה. אמנם לא לשימוש אמיתי אבל בהחלט תוצאות טובות מאוד. במיוחד כשאנחנו מדברים על 10 מחלקות, הקושי לזהות מחלקה מסויימת הוא יותר גדול ככה כי צריך לבחור אחת נכונה מתוך 10 אפשרויות.

התאמנתי פה על 10 אפוקים, מכיוון שהקלאסיפייר יחסית קטן (שכבת fc אחת) לא רציתי ליצור פה $overfit$ גדול מדי ואני כן רואה שהגרף של ה$accuracy$ כבר התחיל להתמתן.

זאת הייתה האינדיקציה הראשונה שלה להצלחת המודל, ונראה שעברנו אותה.

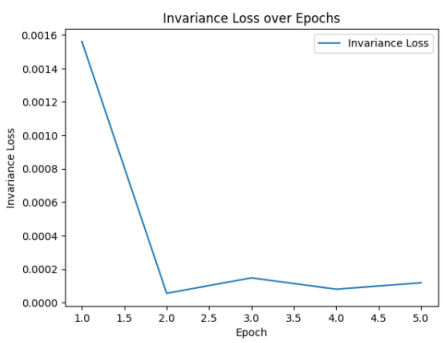
$4Q$:

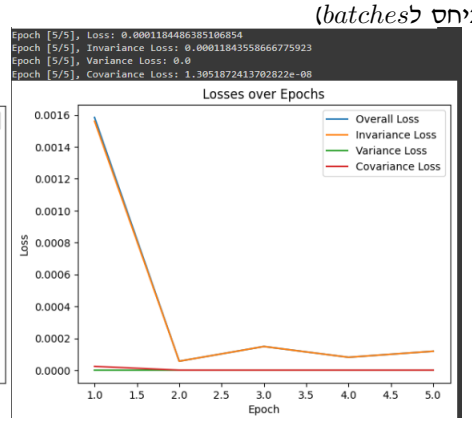
באופן מאוד מעניין בהתחלה התאמנתי על 25 אפוקים והתוצאות יצאו מאוד גרועות למרות שהלוס היה מאוד נמוך.

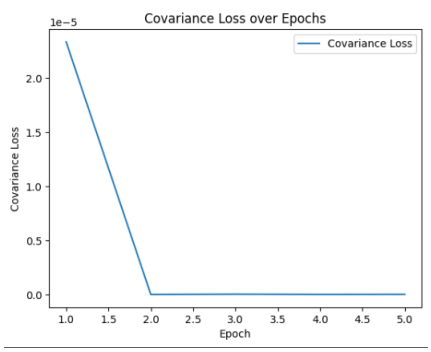
לאחר מכן התאמנתי על 5 אפוקים והלוס למרות שהיה קצת יותר גבוה נתן תוצאות טובות הרבה יותר.

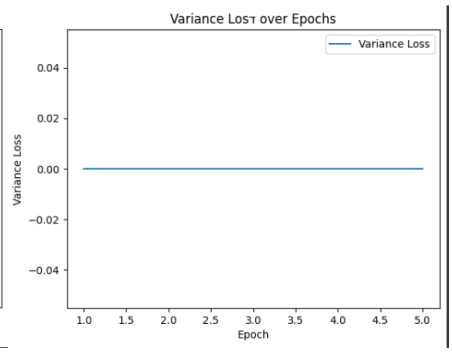
אז נראה את 2 האימונים:

האימון השני עם התוצאות הטובות עם ה5 אפוקים: (הלוסים מוצגים ביחס לאפוקים ולא ביחס ל$batches$)

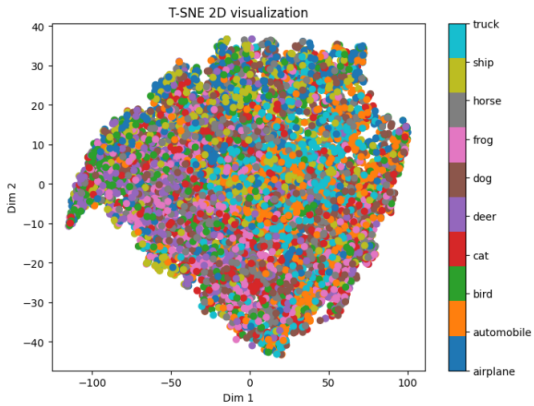


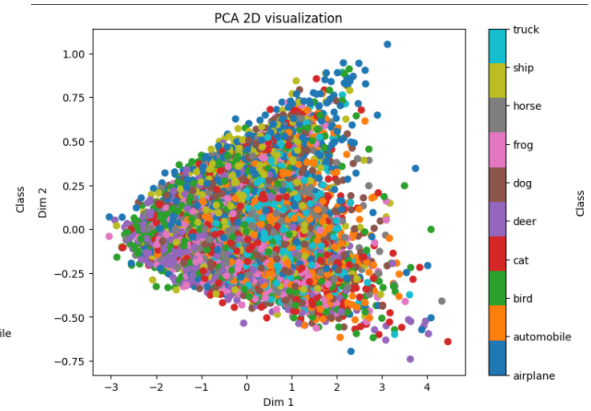




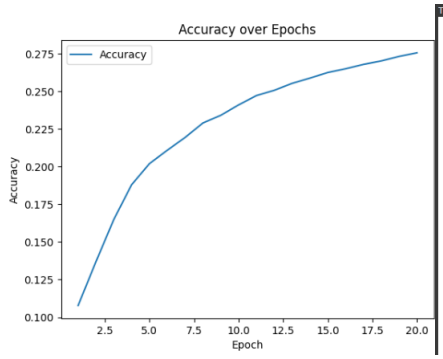


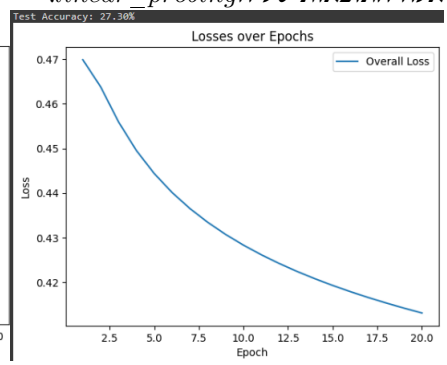
ואלה התוצאות של הויזואליזציה:





אלה התוצאות של ה$linear_probing$:



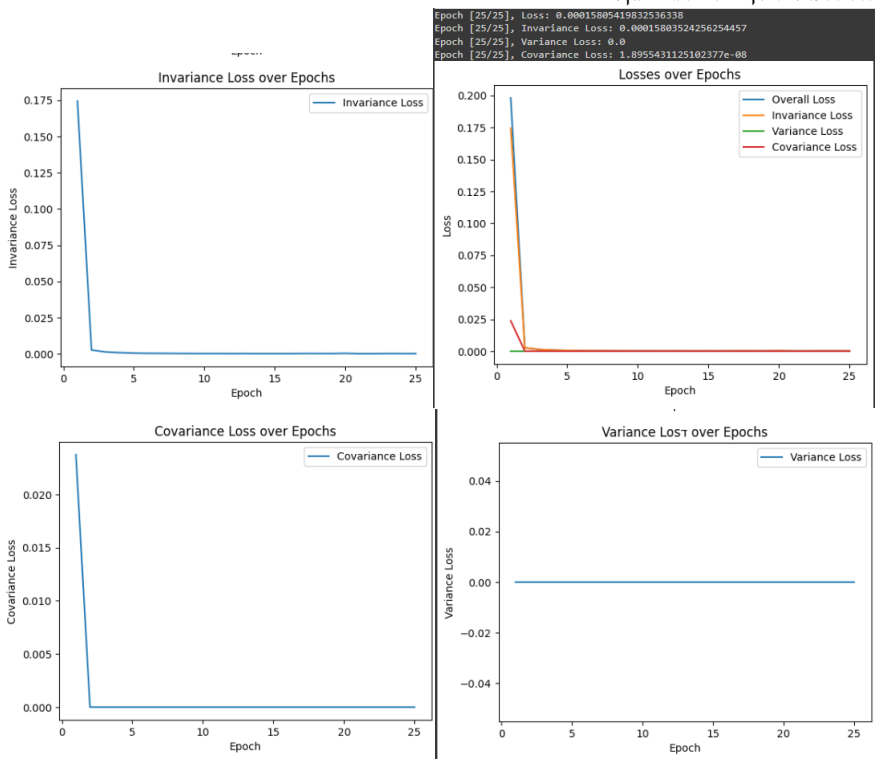


יש לנו בערך 27 אחוז דיוק, שזה לא באמת מודל עובד וכשיר אבל זה הרבה יותר טוב מרנדומלי. התוצאות של המודל הראשוני שלנו עם כל פונקציות הלוס היו הרבה יותר טובים.

אפשר לראות בויזואליזציה שאפילו ב$T-SNE$ שהמחלקות מאוד מפוזרות אחת מהשנייה, יכול להיות שזה שלא הייתה משמעות ללוס של ה$variance$ איפשר למודל לפזר יותר מדי ולהגדיל את ה$variance$, ככה כל מחלקה יצאה מפוזרת יותר וכשמדובר ב10 מחלקות אנחנו מקבלים קושי רציני לנחש את מ10 כשאחת נכנסת בתחום של השנייה.

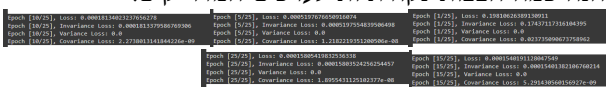
וזו הריצה השנייה עם 25 אפוקים ותוצאות לא יותר טובות מרנדומליות:

זה הלוס תוך כדי האימון:

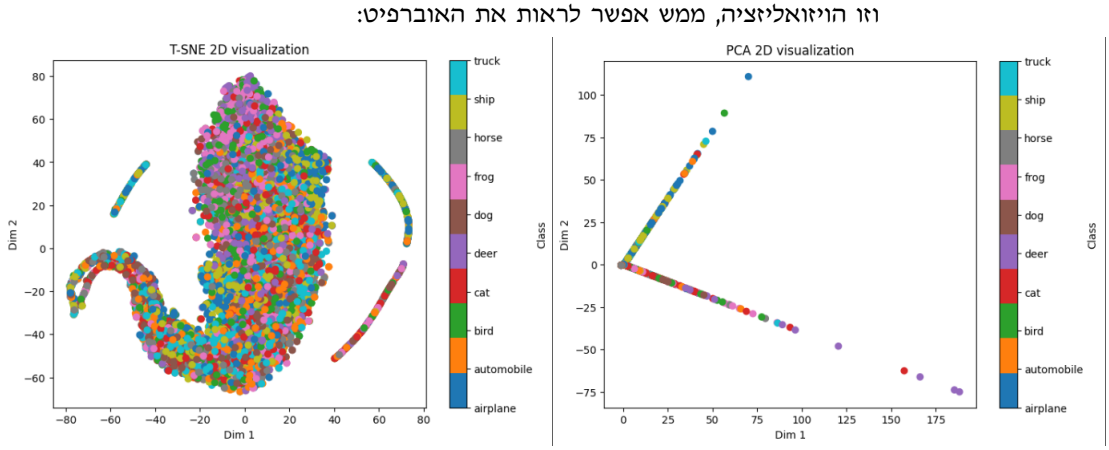


מכיוון שגם ה$covariance$ וגם ה$invariance$ מאוד קרובים לאפס אבל מתחילים קצת מעל אז קשה לראות את השינוי שלהם לאורך הזמן.

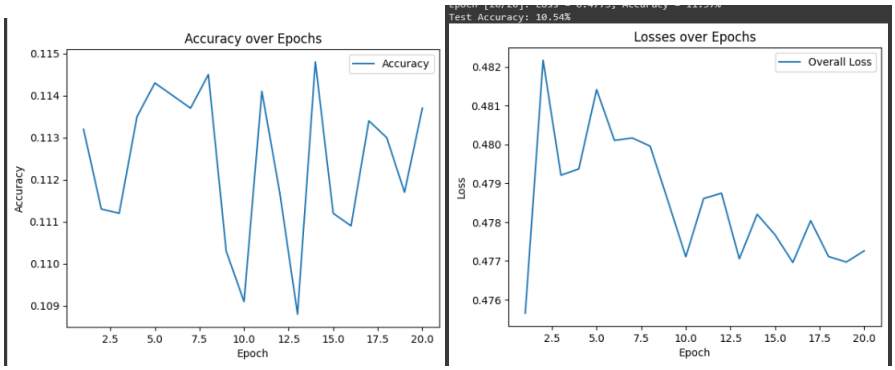
הנה כמה הצצות נקודתיות לערכים המדוייקים:



וזו הויזואליזציה, ממש אפשר לראות את האוברפיט:



וזה ה$linear_probing$: דיוק של 11 אחוז בערך, ממש רנדומלי.



אפשר לראות שהמודל פשוט לא מצליח ללמוד כלום. הוראינס כבר כל כך היה גבוה כנראה שאיבדנו משמעות לאינפורמציה המקורית ואנחנו כבר סתם מנחשים אותו מספר. יש פה כנראה כבר סוג של $overfit$.

$Q5$:

אם נשתמש בLO אנחנו לא נלמד מודל חדש שיצליח להכליל את ה$dataset$ ככה שיהיה אפשר להשתמש בו לדברים אחרים.

כל הכוח של ה$encoder$ זה שהוא לומד את המבנה של ה$dataset$ ומקבל איזשהי יכולת הכללה באמצעות הלוסים שאנחנו נותנים לו.

בנוסף נשמר לנו מודל שממש ניתן להשתמש בו אחר כך שיש לו איזשהי הכנה מוקדמת.

LO פשוט ישנה את ה$latent$ ככה שיתאים ללוסים שנבחר בלי הכוונה מוקדמת, וככה כל דגימה תצטרך לעבור מחדש תהליך של למידה במקום הסקה כשהתוצאה תיהיה תלויה אך ורק בו ובפונקציות הלוס שנלמד בלי הבנה מרחבית של שאר הדאטא סט.

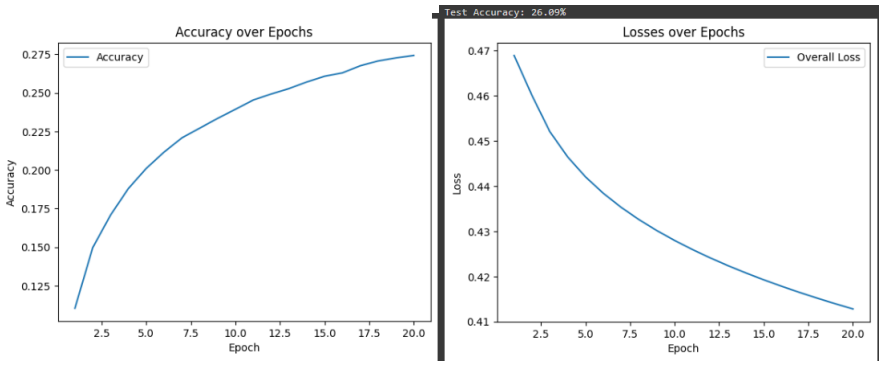
דבר ראשון שצריך להוסיף יהיה שימוש בLO עם האוגמנטציות השונות ופונקציות הלוס. (אולי זה היה התכנון המקורי בכלל, בכל מקרה נוסיך על זה נתבך)

אבל אפילו אם נלך לקראת ונגיד שבLO אנחנו נשתמש ב$views$ השונים שעברו אוגמנטציות, ונשתמש בפונקציות הלוס שהשתמשנו ב$encdoer$, כאלה שמצליחים לקבל איזה שהיא ראייה מרחבית לכל $batch$ זה עדיין לא מספיק כי הלמידה שלנו לא יודעת יותר מדי להכוין את ה$latent$ וקטור למקום שמכיל את התכונות הסמנטיות של התמונה.

אז הצעה שנייה היא להוסיף רכיב גנרטיבי שיודע ללמוד איך מייצרים $views$ שונים לתמונה עצמה ככה שה$view$ יכיל אוגמנטציה שכן יודעת במובן מסויים לשמור על הסמנטיקה של תכונות מסוימות בתמונה. ככה נוכל ללמוד אוגמנטציות שונות אבל חכמות ולנווט את ה$latent$ וקטור בכיוון שיכיל אינפורמציה רלוונטית על התמונה.

$Q6$:

אלה התוצאות: הדיוק הגיע ל26 אחוז.



אני חושב ששימוש בוריאציות של תמונות אחרות שאולי לא מספיקות קרובות לתמונה המקורית, ואז הוריאציה שהמודל לומד היא שונה מדי מהתמונה המקורית, וככה הוא לא באמת מצליח ללמוד הכללה טובה של התמונה.

ניסיתי להסתכל על הויזואליזציה של המודל הזה, יצא אפילו פחות טוב מהמקרה שמורידים את הוריאנס בפונקציות הלוס, כי הדאטא מעורבב בדחיסות יחסית.

נראלי שחסרים לנו כמה דברים - דבר ראשון דאטאסט יותר גדול. ככה יכולנו לקבל תמונות קרובות יותר אחת לשנייה.

דבר שני שהיחס בין הוריאציות של התמונה המקורית והתמונה השכנה יהיו בעדיפות לוריאציות בתמונה המקורית. לכן הייתי נותן אימון שלוקח כמה פעמים וריאציות רק של התמונה המקורית ואז פעם אחת וריאציה של תמונה קרובה. נגיד אפוק אחד בדיוק מה שעשינו בארכיטקטורה המקורית ואפוק שני מה שעשינו פה.

$Q7$:

נתחיל בזה שהתוצאה ב$Q2$ נראית הרבה יותר טובה, קודם כל יש אזורים ספציפיים למחלקות מסוימות ודבר שני המחלקות יחסית מקובצות באזורים ספציפיים והערבוב נעשה באזורים אחרים.

אני לא בטוח שזה לגמרי אומר שהתוצאה של $VICReg$ טובה יותר, אבל לדעתי אין סיבה לתת לשיטה של $Laplacian\ Eigenmaps$ קרדיט בלי סיבה.

בסופו של דבר יש ל$Laplacian\ Eigenmaps$ קושי בהתמודדות עם דאטא ממימד גבוה אבל מצד שני יש לה גם קושי להיות אינטרפטבילית.

אני חושב שהקושי נובע מכמה דברים ובין היתר מהניסיון שלה לשמר את הצורה של המידע ולא להמיר אותו למימד שונה לחלוטין בו יהיה קל יותר להפריד את הדאטא.

בנוסף $Laplacian\ Eigenmaps$ מנסה לשחק על הצורה הגיאומטרית של דברים ולקרב דברים דומים, ככה אנחנו דוחסים תמונות לשכנים שלהם ומאבדים את הוראינס.

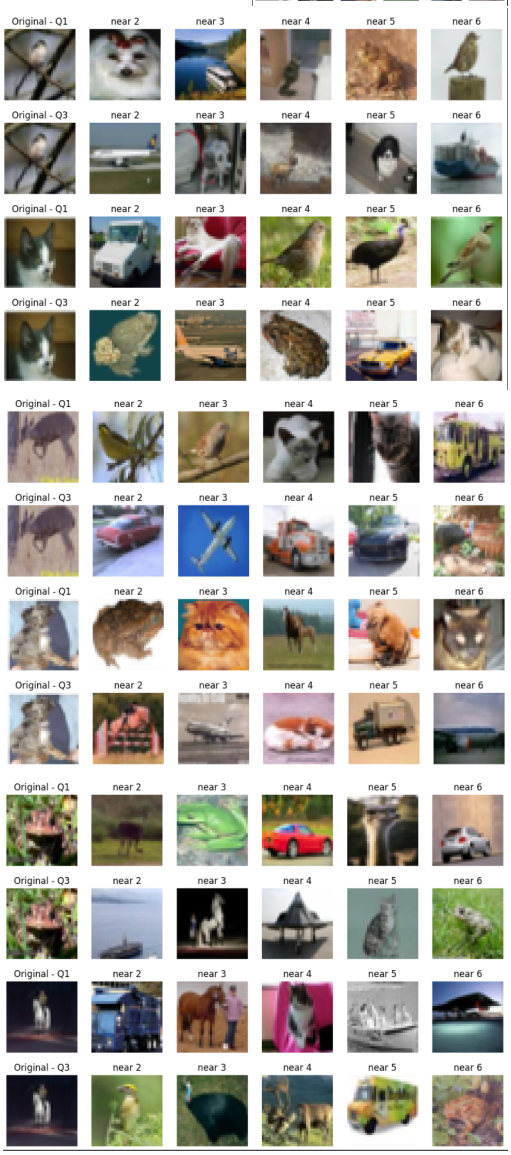
המטרה של $Laplacian\ Eigenmaps$ היא פחות לנסות יצור קלאסיפיקציה ויותר לשמר מימד גיאומטרי של הדאטאסט, ובכך אין לו ניסיון להבין סמנטיקה שדרושה לסיווג והפרדה בין מחלקות, לא פלא שכשהיא מורידה את המידע למימד נמוך ההפרש בין מחלקות שבנוי על סמנטיקות מסוימות לא יישמר כראוי.

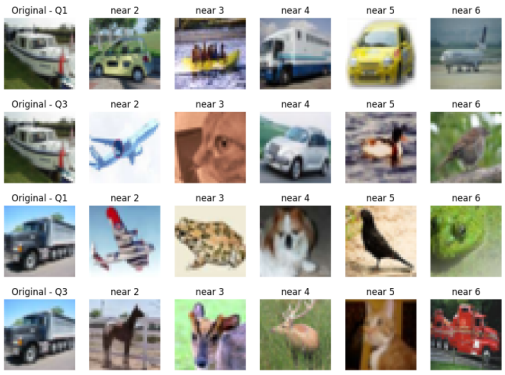
דבר אחרון שכבר הזכרנו בהקשר אחר זה הלינאריות, $Laplacian\ Eigenmaps$ גם מתבסס כמו PCA על הורדת לינארית של המימד ולכן מייצר פחות אקספרסיביות שדרושה לשמירת סמנטיקה של תמונות.

$Q8$:

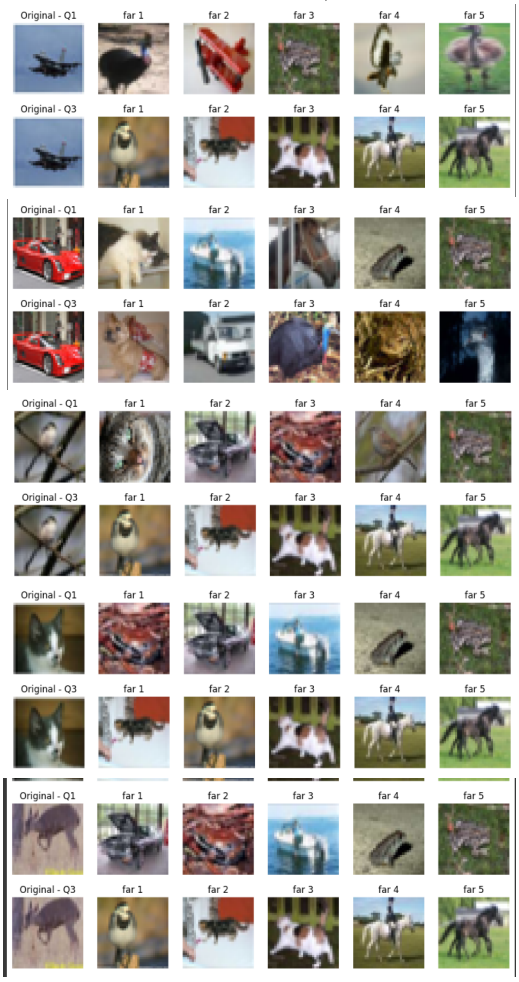
השכנים הקרובים ביותר לכל תמונה: (הכי שמאלי זה התמונה עצמה)

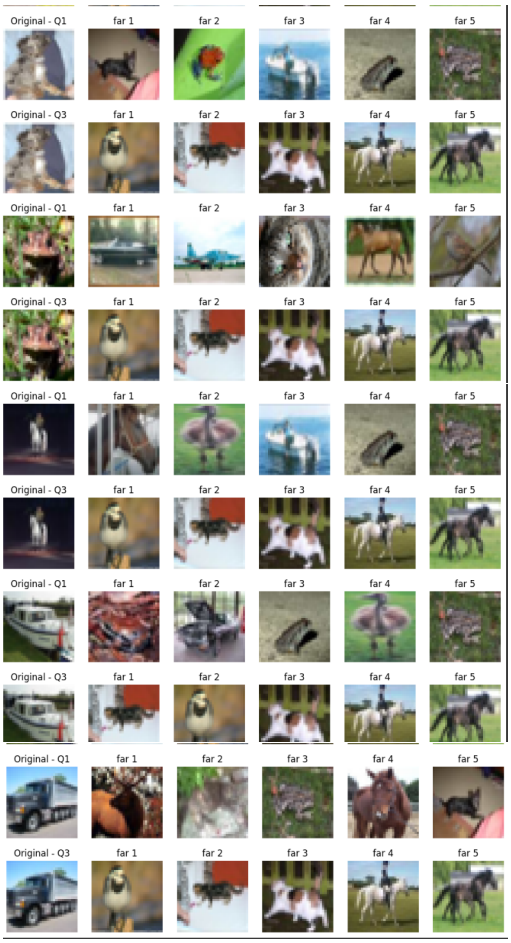
שורה ראשונה היא של המודל המקורי והשנייה של המודל $no\ generate\ neighbors$:





התמונות הכי רחוקות מכל תמונה: (שוב התמונה עצמה משמאל)





ננסה לנתח קצת את התוצאות. ראשית נגיד שציפיתי לתוצאות הרבה יותר דומות לתמונות המקור. הדיוק היה מאוד גבוה, במיוחד במודל הראשון.

אפשר לראות שהמודל הראשון מביא תוצאות טובות יותר מהמודל השני. בעיקר ביכולת להביא תמונה עם תוכן דומה ולא רק תכונות דומות.

המודל הראשון תפס טוב יותר את התכונות הסמנטיות שנדרשות לסיווג ואולם המודל השני התמקד בתכונות פחות חשובות כמו צבעים ומסגרות.

אפשר להבין את זה בכך שהמודל הראשון למד על אוגמנטציות של התמונה עצמה ובכך למד יותר טוב את התכונות שייחודיות לתמונה הזאת. לעומת זאת המודל השני שהתאמן גם אוגמנטציה של שכנים אז הוא למד עולם רחב יותר לכל תמונה ואיבד מעט את ההפרש בין התמונה עצמה לשאר התמונות.

מבחינת שמירת תמונות קרובות אחת לשנייה נדמה לי שהמודל הראשון תפס יותר טוב. וגם מבחינת שמירת תמונות רחוקות אפשר לראות שהמודל הראשון היה יותר טוב. אולי כי הוא הבין יותר את הדאטאסט הכללי ובכך איפשר לעצמו ליצור מרחק גדול יותר בין 2 תמונות לא דומות.

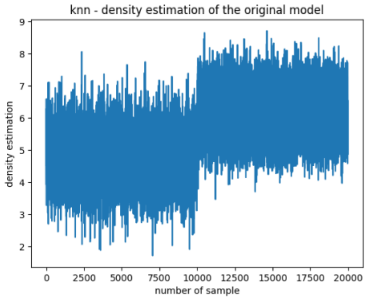
נעיר שיש תמונות שחוזרות על עצמן בתור אלו שרחוקות מכל שאר התמונות, זאת

תופעה שדיברנו עליה בשיעור בהקשר של תמונות שכולן חושבות שהן קרובות אליהן, רק בצורה הפוכה. כנראה שמשהו בלמידה יצר בידול לתמונות הללו בכך שהן שונות מאיך שהמודלים תפסו את מה שהן מייצגות בתמונה ונוצר דיסוננס שיש חתול בצורה וברקע שמבחינת המודל ממש לא אמור להיות שם תכונות סמנטיות של חתול ואז הן מתרחקות מכולן מבחינה סמנטית. התופעה קוראת בעיקר במודל השני, שוב אולי בגלל שהוא למד פחות טוב את הדאטאסט.

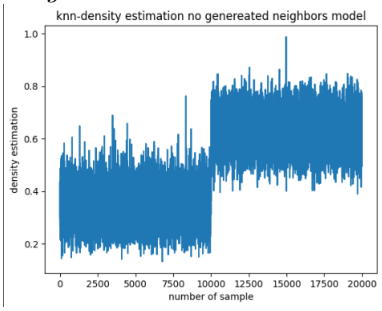
$Section\ 4$

$Q1$:

בשביל המודל הראשון:



בשביל המודל השני של ה$no\ generated\ neighbors$:



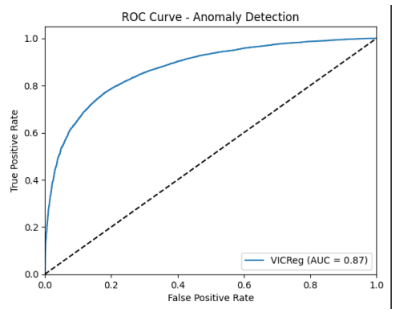
אפשר לראות שהמודל הראשון מביא תוצאות עם $inverse\ density$ יותר גדול, כלומר המרחק בין דגימות לשכנים הוא יותר גדול.

אפשר גם לראות שהמרחק בין תמונות אנומליות לתמונות לא אנומליות ($MNIST$ ו$CIFAR$) הוא גדול יותר ב2 המודלים. שניהם נתנו מרחק גדול יותר לתמונות מהדאטאסט הלא נכון וזה מה שציפינו.

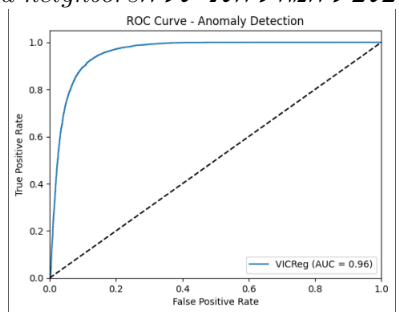
מבחינת מימוש היה צריך לבצע אדפטציה קטנה לדאטא של $MNIST$ ועשיתי פשוט $resize$ והכפלה של המימדים ב3.

$Q2$:

בשביל המודל הראשון:



בשביל המודל השני של ה$no\ generated\ neighbors$:



באופן מעניין המודל השני $No\ generate\ neighbor$ יצר $ROC\ curve$ טוב יותר. כלומר נוכל לבחור $threshold$ קטן יותר ועדיין לקבל סבירות גבוהה יותר לTP ביחס לכמות הFP שלנו.

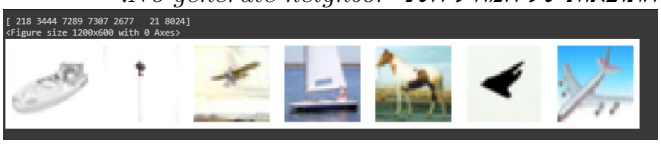
גם בערכי הAUC הערך של המודל הראשון 0.87 ושל השני 0.96 שזה ממש מוצלח.

בעיקרון היוכלת של מודל להיות קרוב לפינה שמאלית העליונה וכמות הAUC שלו מעידים על יחס טוב בין הTPR לFPR כשאנחנו רוצים את הTPR גבוה כי זה אומר שעל כל T שלנו אנחנו צודקים יותר ואת הFPR נמוך כי זה אומר שטעינו פחות.

$Q3$:

התוצאות של המודל הראשון:

התוצאות של המודל השני $No\ generate\ neighbor$:



אפשר לראות שהמודל הראשון ממש זיהה כמו שצריך שיש פה דאטאסט שונה לגמרי עם תמונות שונות. לעומת זאת המודל השני $No\ generate\ neighbor$ לא זיהה כלל והביא תמונות יחסית שונות מהדאטסט הרגיל. כלומר מבחינת הראשון השוני בגדול בין התמונות של המספרים זה נחשב אנומליה.

לעומת זאת המודל השני $No\ generate\ neighbor$ הגדיר אנומליה כתמונה שדומה לדאטסט אבל ייחודית יחסית. לדוגמא המטוס שנופל, או הציפור (?) השחורה עם הרקע

הלבן, למעשה יש שם 2 תמונות שאני לא כל כך מבין מה הן וזה לבד מעיד על אנומליה מסוימת.

אם אני אצטרך לבחור איזה מודל יותר טוב, אני אגיד שזה תלוי מאוד במשימה מולנו. אבל לא בהכרח שהראשון טוב מהשני.

יכול להיות שנרצה את המודל הראשון כי הוא זיהה חריגה גדולה, אבל אם אנחנו נצטרך שימוש כמו פס ייצור שרוצה חריגה קטנה אז יכול להיות שדווקא המודל השני יהיה לנו טוב יותר.

במיוחד כשהROC שלו יצא טוב יותר.