ליעד ברטלר, [redacted]

עדן ברמן, [redacted]

ביולוגיה חישובית תרגיל 3 – זיהוי תבניות באמצעות רשת נוירונים

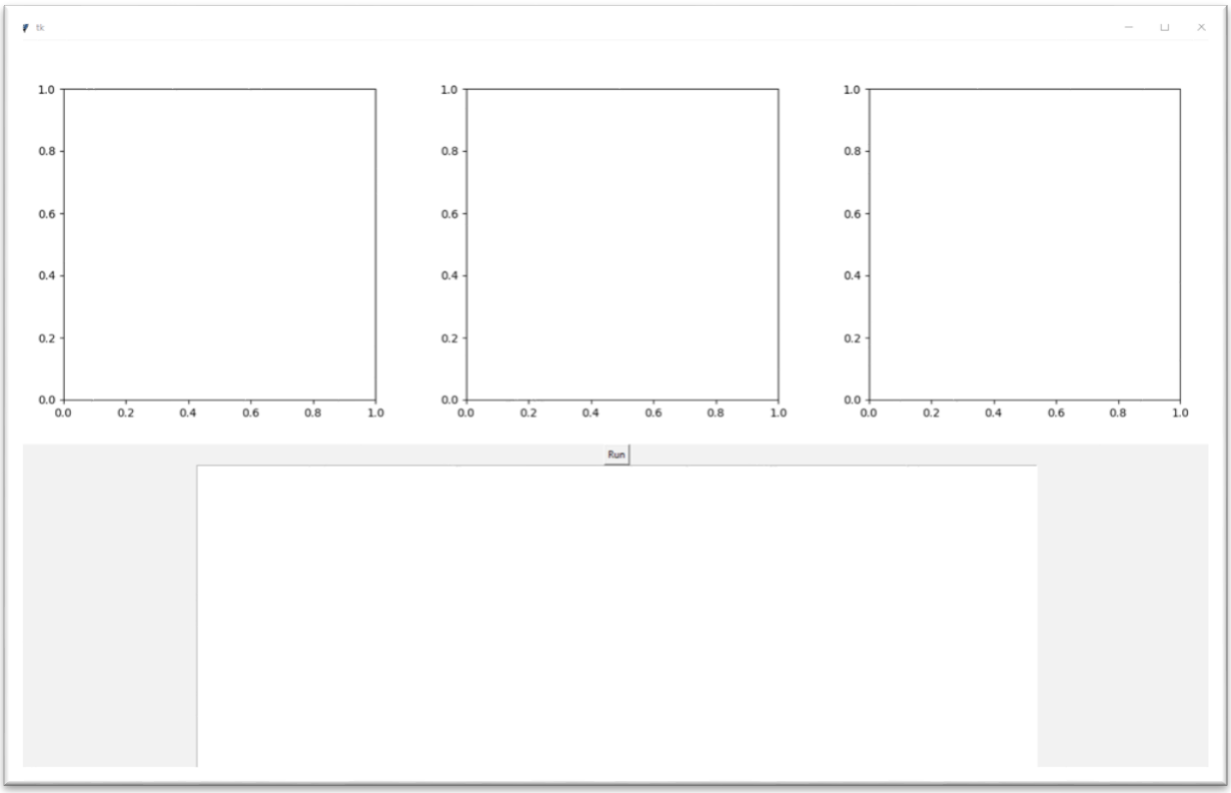
תרגיל זה מורכב משתי תוכניות עוקבות המפורטות להלן.

התוכנית הראשונה buildnet, בונה רשתות נוירונים בעזרת אלגוריתם גנטי אשר ילמדו לזהות תבניות וינבאו האם מחרוזת מתאימה לתבנית. התוכנית מקבלת כקלט שני קבצים; קובץ למידה וקובץ מבחן, המכילים מחרוזות בינאריות באורך 16 ספרות של 0 או 1. ככלל יש 20,000 מחרוזות המחולקות ביחס של 4:1 (למידה : מבחן). פלט התוכנית הינו קובץ בשם wnet המכיל את משקולות ואת הגדרות מבנה רשת הנוירונים הטוב ביותר.

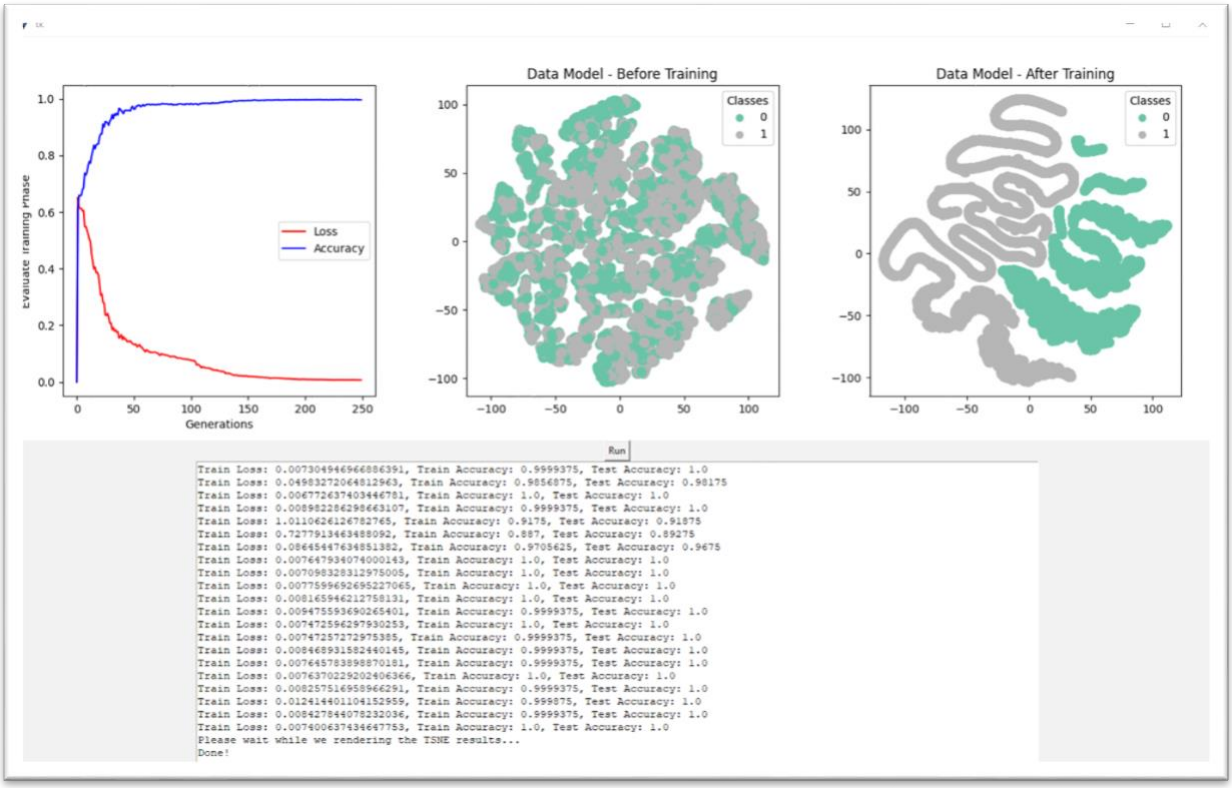
התוכנית השנייה runnet, מריצה את רשת הנוירונים הטובה ביותר מהחלק הקודם על דאטה חדש בעל אותה חוקיות, ומסווגת באופן בינארי את המחרוזות על פי השתייכותן לתבנית (1) או לא (0). חלק זה אינו מכיל אלגוריתם גנטי או תהליך למידה, אלא רק סיווג בינארי של המחרוזות באמצעות feedforward על הרשת. התוכנית מקבלת כקלט שני קבצים; קובץ wnet מהחלק הקודם, וקובץ דאטה המכיל 20,000 מחרוזות לא מסווגות. פלט התוכנית הינו קובץ המכיל את סיווגי המחרוזות.

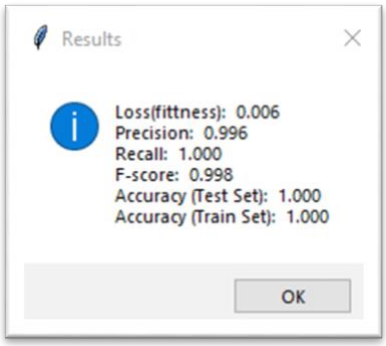
יתר על כן, עיצבנו ממשק גרפי להמחשת התקדמות התוכנית הראשונה buildnet ותוצאותיה. לאורך התוכנית מוצגת התקדמות האלגוריתם על פי ערך הדיוק וערך ה-loss באופן גרפי בחלון השמאלי, ובאופן נומרי בחלון התחתון. בכדי להציג את הדאטה הרב מימדי, נעזרנו באלגוריתם t-SNE להורדת הממדים לשניים וייצוג הדאטה. בחלון האמצעי מופיע גרף הממחיש את נראות הדאטה לפני שהמודל למד, ואילו בחלון הימני מופיע גרף הממחיש את נראות הדאטה לאחר הלמידה והקלסיפיקציה. בסוף ריצת התוכנית, יוצג תפריט סיכום של תוצאות ההרצה עבור רשת הנוירונים הטובה ביותר מסט המבחן, וייווצר קובץ המודל wnet.npz.

להלן תמונת תפריט ההתחלה:



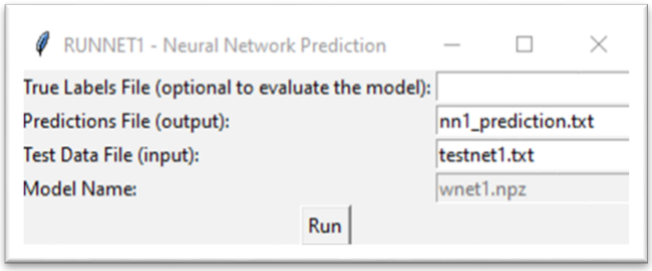
להלן דוגמא לתפריט הסיכום ותוצאותיה :



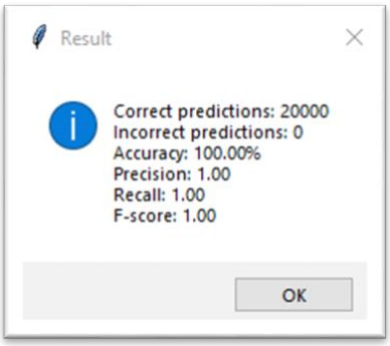


בנוסף, עיצבנו ממשק קלט עבור תוכנית runnet לנוחות הזנת הקבצים ; קובץ המודל מוזן ואינו ניתן לשינוי, קובץ הדאטה שיש לסווג, שם קובץ הפלט אשר יכיל את החיזויים, ובמידה וקיים, ניתן להוסיף קובץ המכיל את הסיווגים האמיתיים (בפורמט הפלט הנדרש, 20,000 שורות של התיוגים המתאימים). קובץ זה מאפשר חישוב מידת ההצלחה של המודל, אמנם אינו חובה להזינו. בסוף ריצת התוכנית, יוצג תפריט סיכום של תוצאות ההרצה של המודל, בנוסף לקובץ הפלט predictions.txt של הסיווגים.

להלן תמונת תפריט ההתחלה :



להלן דוגמא לתפריט הסיכום:



הוראות הפעלה לתוכנית א:

ההוראות זהות עבור הרצת התוכנית buildnet0 ו-buildnet1, עם שמות הקבצים בהתאמה.

1. כבה את מערכת האנטי-וירוס.
2. הורד את קובץ ה-zip מהלינק הבא וחלץ את הקבצים:

3. פתח את התיקייה buidnet0_exe, והחלף את קבצי האימון והמבחן בקבצים בעלי השמות train_nn0.txt ו-test_nn0.txt בהתאמה.
הרץ את קובץ ה-exe.
4. המתן לפתיחת התוכנה, ולחץ run.
5. בסיום הריצה יופיע מסך התוצאות, וקובץ הפלט wnet.npz ישמר לאותו נתיב.

הוראות הפעלה לתוכנית ב:

ההוראות זהות עבור הרצת התוכנית runnet0 ו-runnet1, עם שמות הקבצים בהתאמה.

1. כבה את מערכת האנטי-וירוס.
2. הורד את קובץ ה-zip מהלינק הבא וחלץ את הקבצים:

3. פתח את התיקייה runnet0_exe, והוסף את קובץ המחרוזות שברצונך לסווג בשם testnet0.txt.
קובץ המודל wnet0.npz כבר נמצא בתיקייה זו. לשם נוחות, ניתן להוסיף את קובץ הסיווגים האמיתיים בשם testnet0_labels.txt, או כל שם אחר עם סיומת txt.
לבסוף הרץ את קובץ ה-exe.
4. המתן לפתיחת התוכנה. בתפריט שנפתח ניתן להזין את קובץ הסיווגים האמיתיים אם קיים. יש לשים לב להקלדה תקינה של שם הקובץ יחד עם הסיומת txt.
לאחר מכן יש ללחוץ run.
5. בסיום הריצה יופיע מסך התוצאות במידה והוזן קובץ הסיווגים האמיתיים, וקובץ הפלט nn0_prediction.txt ישמר לאותו נתיב.

לנוחיותכם, ניתן לצפות בסרטון הסבר והרצה שיצרנו על הפעלת שתי התוכניות בלינק הבא:

תוכנית א' – בניית הרשת

ייצוג הדאטה:

באופן כללי, ייצגנו את הדאטה על ידי חלוקת המחרוזות והתיוגים לשני מבנים. את המחרוזות אכסנו במטריצה בגודל 16 x 20,000 כך שכל שורה מכילה מחרוזת וכל עמודה מכילה ספרה אחת של המחרוזת. את הסיווגים אכסנו במערך כך שתא ה-i תואם לשורה ה-i במטריצה.

בפרט, לאחר קביעת חלוקת הדאטה לסט אימון ולסט מבחן, התקבלו עבור סט האימון מטריצה בגודל 16 x 16,000 המכילה את הדוגמאות, ומערך בגודל 16,000 המכיל את הסיווגים. באופן דומה, עבור סט המבחן התקבלה מטריצה בגודל 16 x 4,000 לדוגמאות, ומערך בגודל 4,000 לסיווגים.

תיאור האלגוריתם הגנטי:

גודל האוכלוסייה מאותחל ל-200 פרטים כך שכל אחד מהווה רשת נוירונים עבור למידת התבנית וסיווג המחרוזות. החלטנו לקבע את גודל האוכלוסייה במספר זה כדי לשמור על אוכלוסייה יציבה אשר לא מושפעת משינויי גודל. מימשנו זאת כך שבכל דור בוחרים את 20% הפרטים בעלי ערך ה-loss הנמוך ביותר להשתתף בדור הבא, בעוד ש-80% הנותרים נכחדים מהאוכלוסייה. בין 20% הנבחרים מבצעים crossover עד ליצרית דור חדש בגודל 200 פרטים. לאחר מכן האלגוריתם ממשיך לפעול באמצעות פונקציות נוספות על פרטי הדור החדש בלבד וכך הלאה.

כל פרט מובדל בערכי המשקולות וה-bias שלו. בשלב האתחול, ה-bias מוגרל באופן רנדומלי בהתפלגות גאוסיאנית סביב האפס, כך ש-95% מההגרלות יהיו בין 2 ל- 2-. ערכי המשקולות מוגרלים באותו אופן, אמנם לבסוף מחלקים את הערכים בשורש גודל השכבה הקודמת. זאת בדומה לאתחול Xavier המסייע במניעת exploding gradient ובמניעת vanishing gradient.

כל פרט מקבל כקלט batch בגודל 512 מחרוזות מסט האימון ומבצע עליהן feedforward לקבלת ערך ההסתברות להשתייכות לתבנית. לאחר מכן מחשבים את ערך ה-loss על פי פונקציית ה- binary cross entropy. מבצעים תהליך זה באופן לינארי על כלל סט האימון, ומחשבים את ערך ה-fitness של הפרט על פי ממוצע ערכי ה-loss. בחרנו להגדיר את ה-fitness בצורה זו כדי למנוע overfitting באימון.

מאחר ולא ידוע אם הדאטה מאוזן מבחינת כמות הסיווגים שהם 1 וכמות הסיווגים שהם 0, ייתכן מצב של סיווגי False Positive (FP). מקרה זה נובע מכך שהמודל לומד מיהו הסיווג הדומיננטי ולפיו מתייג את הדגימות. זאת במקום ללמוד את התבנית הנסתרת ולסווג לפיה כנדרש. מכאן בחרנו בפונקציה cross entropy הנותנת משקל לטעויות FP על ידי פליטת ערך בהתאם למשקל הטעות. לכן קבענו כי ערך הכשירות יחושב על פי משקל הטעות, ה-loss. ככל שערך הכשירות נמוך יותר, כך למידת הפרט יעילה יותר. גישה זו נוגדת אינטואיציה, אמנם ננקטה בכדי להקל על החישוב המתמטי, ומדגישה את ההבדל בין הדיוק על סט האימון לבין הכשירות של כל פרט.

בכדי להכווין את התפתחות רשתות הנוירונים, אנו מעודדים הטרוגניות על ידי יצירת מוטציות, ושומרים על התקדמות האלגוריתם באמצעות הפרט הטוב ביותר. לכל פרט מגרילים באופן רנדומלי מספר בין 0 ל-1, במידה ומספר זה הינו קטן מ-0.5, מבצעים מוטציה נקודתית באחד מערכי המשקולות של אותו פרט. בכדי לשמר התקדמות, עבור כל דור שומרים את הפרט בעל ערך הכשירות הטוב ביותר. פרט זה מוחלף בפרט בעל ערך ה-loss הגדול ביותר בדור החדש (Elitism).

מכאן מחשבים את כשירות פרטי האוכלוסייה החדשה, והתהליך חוזר שוב למשך 250 דורות. במידה וה-test accuracy מעל 0.992 וה-loss מתחת ל-0.02, האלגוריתם מפסיק ללא תלות במספר הדורות שבוצעו. הפרט בעל ערך הכשירות הטוב ביותר ישמש כקלט עבור התוכנית השנייה. פלט התוכנית הינו בקובץ wnet.npz המכיל את משקולות ואת הגדרות מבנה רשת הנוירונים הטוב ביותר.

מבנה רשת נוירונים:

מבנה רשתות הנוירונים כולל שכבת קלט, שלוש שכבות חבויות (h1, h2, h3) ושכבת פלט. כל שכבה מחוברת במלואה לשכבה הבאה, כלומר כל נוירון בשכבה אחת מחובר לכל נוירון בשכבה האחרת. לכל רשת קיימת מטריצת משקולות ייחודי לחישוב ערכי השכבות למעט שכבת הקלט.

שכבת הקלט מורכבת מ-16 נוירונים כאורך מחרוזת. מפעילים פונקציית Elu על המכפלה הקרטזית בין ערכי הקלט והמשקולות לקבלת ערכי השכבה h1 המורכבת מ-8 נוירונים. על המכפלה הקרטזית בין תוצאות אלו וערכי המשקולות הבאים מפעילים פונקציית Relu ו-Batch Norm לקבלת ערכי השכבה h2 המורכבת מ-5 נוירונים. מכאן התוצאות מועברות יחד עם המשקולות הבאות לפונקציית Elu, ומבצעים dropout באופן רנדומלי לשניים מה-features לקבלת ערכי השכבה h3 המכילה 3 נוירונים. שלב ה-dropout נועד למנוע overfitting בסט האימון, אמנם לא מתקיים בסט המבחן כדי שהמודל ייקח בחשבון את כלל ה-features למען סיווג הדוגמאות. לבסוף התוצאות מועברות לפונקציית ה-Sigmoid לקבלת שכבת הפלט. פלט הרשת הינו ערך הסתברותי בין 0 ל-1 המעיד על סיכוי המחרוזת להיות חלק מהתבנית.

מבנה קובץ הפלט wnet.npz:

הקובץ מכיל את משקולות ואת הגדרות מבנה רשת הנוירונים הטוב ביותר שהתקבל במהלך ריצת התוכנית הראשונה. פורמט הקובץ מאפשר לשמור נתונים אלו בצורת מילון (key, value) אשר מייעל את השימוש לתוכנית השנייה.

- משקולות – לכל שתי שכבות עוקבות קיימת מטריצה המכילה את ערכי המשקולות, ומערך המכיל את ערכי ה-bias. סך הכל ישנם ארבעה אובייקטים כאלו המסודרים באופן סידורי.
- מבנה – לכל שכבה מצוין מספר הנוירונים שהיא מכילה לפי סדר הופעתן.

פירוט הפונקציות:

- *Selection* : פונקציה זו משווה בין ערכי ה-loss של כלל האוכלוסייה ושומרת את 20 הפרטים בעלי ה-loss הנמוך ביותר, בעוד ש-80 הפרטים הנותרים נכחדים מהאוכלוסייה. 20 פרטים אלו מאכלסים את הדור החדש עד הגעתו ל-100 פרטים. כך אנו מכוונים את האלגוריתם לעבר הפתרון האופטימלי.
- *Crossover* : פונקציה זו מקבלת שני פרטים שנבחרו באקראי מהאוכלוסייה שעברה סלקציה, ועל פיהם מייצרת שני פרטים חדשים עבור הדור החדש. מעתיקים את מטריצת המשקולות של שני ההורים ליצירת שני צאצאים הזהים להם. כעת עבור כל ערך משקולת מקבילה בין שני הצאצאים, משווים בין ערכיהם לקבלת טווח המוגדר לפי הערך הגדול המשמש כגבול עליון, ולפי הערך הקטן המשמש כגבול תחתון. לשני גבולות אלו מבצעים שינוי מתמטי כדי להעשיר את הטווח קלות, ומגרילים מספר רנדומלי בטווח עבור ערך המשקולת באינדקס זה לכל צאצא. אנו משערים כי הצאצאים יהיו יותר יעילים מהוריהם מאחר והם משלבים בין ערכיהם ומתכנסים לטווח ביניהם. אותו תהליך מתבצע עבור ערך ה-bias של הצאצאים. כך למעשה נוצרות שתי רשתות חדשות עבור הדור החדש מזוג רשתות מהדור הקודם.
- *Mutations* : פונקציה זו מייצרת שינו לוקאלי בפרטי האוכלוסייה באופן הסתברותי. תחילה לכל פרט מגרילים מספר אקראי בין 0 ל-1, ומשווים לשיעור הסף שנקבע 0.5. במידה והערך שהתקבל קטן יותר, אזי תתבצע מוטציה. אחרת, הפונקציה תחזיר את הרשת שהתקבלה ללא שינוי. המוטציה הינה נקודתית בה אחד מערכי המשקולות של הפרט מוחלף לערך רנדומלי חדש.

ביצועי התוכנית:

תוצאות התוכנית משתנות בהתאם לאתחול הרנדומלי של משקולות ו-bias רשתות הנוירונים. מאחר ורשתות הנוירונים מתאמנות על ידי אלגוריתם גנטי, התוכנית בעלת אופי לא דטרמיניסטי. לכן הפעלנו את התוכנית מספר רב של פעמים על שני סוגי קבצי הקלט בכדי לקבל סטטיסטיקה מובהקת יותר. עיקר הבדיקה הינה של ביצועי קובץ המבחן עבור רשת הנוירונים הטובה ביותר.

תוכנית buildnet0 על קבצי train_nn0.txt ו-test_nn0.txt :

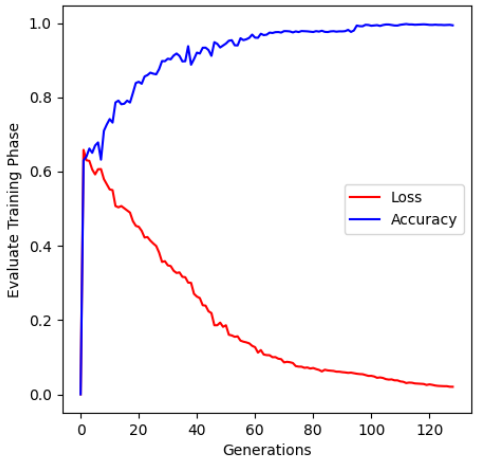
Train Accuracy	Test Accuracy	F1-Score	Recall	Precision	Loss/ Fitness	סימולציה
1	1	0.998	1	0.996	0.006	1
0.989	0.989	0.988	1	0.976	0.064	2
1	1	0.998	1	0.996	0.006	3
0.989	0.989	0.991	1	0.982	0.034	4
0.989	0.989	0.978	1	0.957	0.089	5
1	1	0.992	0.999	0.986	0.027	6
1	1	0.994	1	0.988	0.021	7
0.989	0.989	0.988	1	0.976	0.17	8
1	1	0.984	0.981	0.987	0.038	9
1	1	0.996	0.993	1	0.023	10

תוכנית buildnet1 על קבצי train_nn1.txt ו-test_nn1.txt :

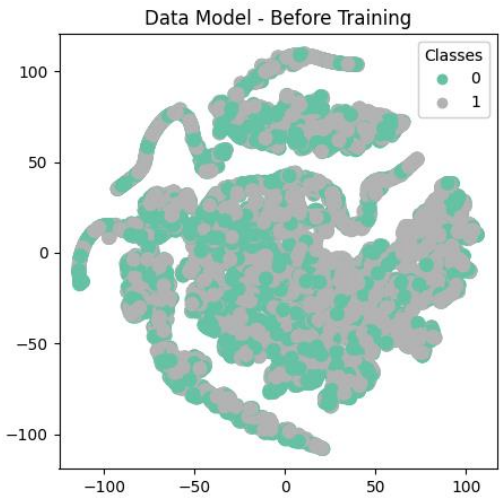
Train Accuracy	Test Accuracy	F1-Score	Recall	Precision	Loss/ Fitness	סימולציה
1	1	0.995	0.992	0.998	0.024	1
1	1	0.993	0.988	0.998	0.039	2
1	1	0.996	0.993	0.999	0.024	3
0.999	1	0.979	0.988	0.97	0.05	4
1	1	0.978	0.957	1	0.022	5
1	1	0.996	0.992	0.999	0.024	6
0.999	1	0.995	0.991	0.999	0.023	7
1	1	0.993	0.988	0.997	0.023	8
1	1	0.99	0.991	0.988	0.026	9
1	1	0.993	0.986	1	0.024	10

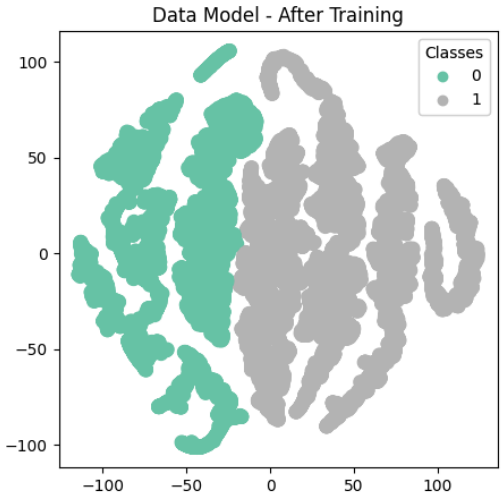
מתוצאות שני הקבצים, ניתן לראות כי כלל ערכי בדיקת הביצועים טובים למדי. ערך ה-Precision מעיד על איכות הסיווג, ואילו ערך ה-Recall מעיד על כמות הסיווג. ה-F1 score הינו מדד המשלב בין שני הערכים הקודמים. עבור הקובץ nn0, אחוז הדיוק הממוצע הינו 0.995 בסט המבחן, עם ערך loss ממוצע של 0.047. לעומת זאת בקובץ nn1, אחוז הדיוק הממוצע הינו 1 בסט המבחן, עם ערך loss ממוצע של 0.027.

- להלן גרפים הממחישים נתוני קובץ nn0, סימולציה 10 :



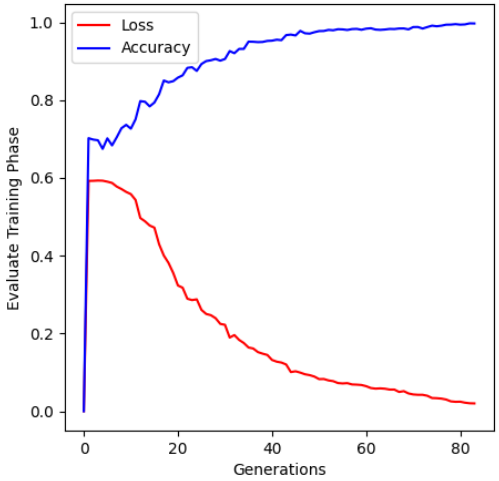
תוצאות שתי הפונקציות מאותחלות לאפס, ורק לאחר חישובן בהינתן הדור הראשון הן מקבלות ערך ממשי. בסימולציה זו הערך הינו סביב 0.6 מימנו הפונקציות מתפצלות לקטבים שונים. ניתן לראות כי הפונקציה הכחולה, המתארת את ה-Accuracy של האלגוריתם, עולה בהדרגה עד התכנסות ל-1. לעומת זאת הפונקציה האדומה המתארת את ערך ה-loss, דועכת בקצב מתון לעבר ה-0. המחשה זו מתארת את יכולת האלגוריתם לבנות רשת נוירונים מיטבית ללמידת התבנית.

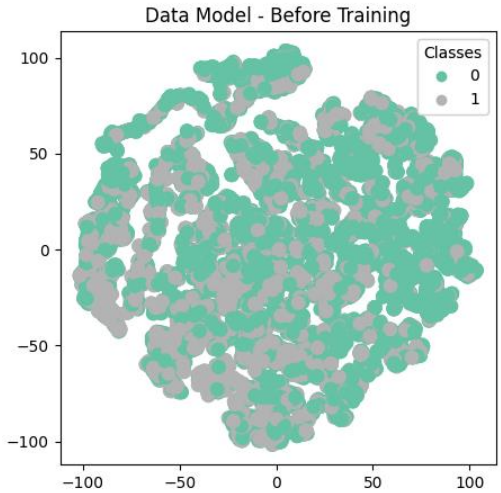


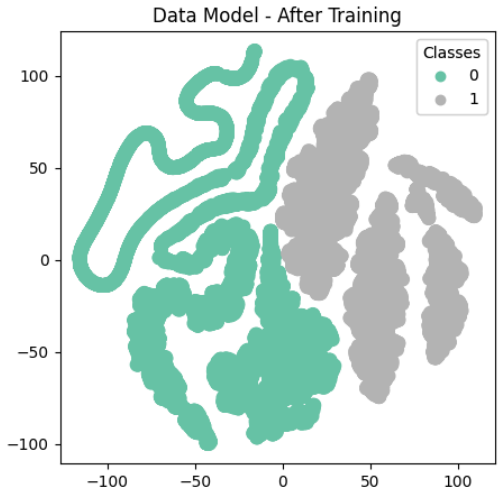


הצגת הדאטה בממד נמוך על ידי פונקציית ה-t-SNE ממחישה את פעילות רשת הנוירונים הטובה ביותר wnet. ניתן לראות כי לפני למידת המודל, כלל הדגימות מעורבבות זו בזו ללא הפרדה אפשרית ביניהן עבור קלסיפיקציה. לאחר שהמודל למד, ניתן לראות שני מקבצים מסודרים על פי סיווג המחרוזות. דבר זה מעיד על יכולת המודל להפריד את הדאטה בצורה לינארית על פי השתייכות המחרוזת לתבנית.

- להלן גרפים הממחישים נתוני קובץ nn1, סימולציה 3 :







ניתן לראות כי התוצאות דומות לאלו של הקובץ nn0.

תוכנית ב' – הרצת הרשת על דאטה חדש

תיאור האלגוריתם:

תוכנית זו מקבלת כקלט את קובץ רשת הנוירונים מהתוכנית הראשונה, וקובץ דאטה חדש שיש לסווג באופן בינארי. תחילה התוכנית מחלצת את מבנה ומאפייני הרשת לפי key-value, ויוצרת את האובייקט בהתאמה. את המחרוזות מאכסנים במערך, ועליהם מריצים את רשת הנוירונים ב-feedforward לקבלת הסיווגים. את התוצאות מעגלים מעלה או מטה בהתאמה, כך שסיווג 1 מעיד על השתייכות המחרוזת לתבנית, ואילו סיווג 0 מעיד על חוסר השתייכותה. פלט התוכנית הינו קובץ המכיל את סיווגי 20,000 המחרוזות על פי סדר הופעתן.

ביצועי התוכנית:

תוצאות התוכנית הינן דטרמיניסטיות על פי רשת הנוירונים שנבנתה. לפיכך הרצנו את כלל המודלים מהתוכניות buildnet0 ו-buildnet1 על קבצים שיצרנו עם אותה חוקיות, בכדי למצוא ולוודא את המודל הטוב ביותר עבור קלט התוכניות runnet0 ו-runnet1.

תוכנית runnet0 על פלטי תוכנית buildnet0 :

F1-Score	Recall	Precision	Accuracy	Incorrect Predictions	Correct Predictions	סימולציה
1	1	1	1	0	20,000	1
0.99	1	0.98	0.991	190	19,810	2
1	1	1	1	0	20,000	3
0.99	1	0.98	0.991	189	19,811	4
0.99	1	0.98	0.991	189	19,811	5
1	1	1	0.99	3	19,997	6
1	1	1	0.99	2	19,998	7
0.99	1	0.98	0.991	189	19,811	8
1	1	1	0.999	16	19,984	9
1	1	1	1	0	20,000	10

תוכנית runnet1 על פלטי תוכנית buildnet1 :

F1-Score	Recall	Precision	Accuracy	Incorrect Predictions	Correct Predictions	סימולציה
1	1	1	0.999	2	19,998	1
1	1	1	1	0	20,000	2
1	1	1	1	0	20,000	3
1	1	1	0.999	13	19,987	4
1	1	1	1	0	20,000	5
1	1	1	0.999	2	19,998	6
1	1	1	0.998	33	19,967	7
1	1	1	1	1	19,999	8
1	1	1	1	0	20,000	9
1	1	1	1	0	20,000	10

על פי טבלאות אלו ניתן לראות כי רשתות הנוירונים למדו את התבנית ומצליחות לסווג את המחרוזות באופן בינארי ואף מדויק. בחרנו את סימולציות 10 עבור שני סוגי הקבצים להיות המודלים האופטימליים בשביל סיווג דאטה חדש בעל תבנית דומה.

חוקיות התבניות:

לאחר התעניינות וחקירת המחרוזות, הגענו למסקנה כי התבניות הינן :

- קובץ nn0 – מספר האחדות נע בטווח בין 8 ל-12 (כולל הקצוות).
- קובץ nn1 – מספר האחדות קטן ממש ממספר האפסים, כלומר 7 ומטה.

בנוסף יצרנו סקריפט הבודק את חוקיות המחרוזות כדי לבסס את השערתנו. תוצאות הסקריפט תומכות בעמדה זו.