דו"ח תרגיל 3

מגישים: אלינוי עמר 318532132

אלעד וויצנבליט 315944611

- שם קבצי הריצה של הtrain וtest:

לקבצי הtrain קוראים:
nn1.txt ,nn0.txt

וקבצי הtest בהתאמה:
nn_test1.txt ,nn_test0.txt

- מפעילים אותו אלגוריתם לnn0 ולnn1 מכיוון ועשינו את הבונוס.

נתונים על אימון הרשת

1) את הדאטא מהקובץ nn0.txt\nn1.txt חילקנו ביחס של 20:80. כלומר 80% שומשו לtrain בעוד 20% שומשו לtest.

```
split_test_index = round(len(X) * 0.8)
build_inputs = np.array(X[0:split_test_index])
test_inputs = np.array(X[split_test_index:])
build_labels = np.array([Y[:split_test_index]])
test_labels = np.array([Y[split_test_index:]])
```

כמו כן שמרנו אותם במערכים מסוג np.

מערך ראשון ל-input של הtrain

מערך שני ל-input של הtest

מערך שלישי ה-labels של הtrain

מערך רביעי ה- labels של הtest.

2) את ארכיטקטורת הרשת קבענו על ידי אלגוריתם גנטי

- הפרמטרים ההתחלתיים שהכנסנו לאלגוריתם הגנטי הם:

גודל אוכלוסייה (רשתות נוירונים): 200

כמות דורות מקסימלית: 300

סף הצלחה שממנו נעצור את התוכנית: 99%

נתינה של הנתוני אימון (input,label) שלהם

(network הוא אובייקט ריק שממנו נבנה את הרשתות נוירונים שלנו.)

```
agent = ga.execute(200,300,0.99,build_inputs,build_labels,network)
```

<u>מבנה הרשת (יישום של הבונוס):</u>

כתחילה ניצור 200 רשתות נוירונים.

אופן היצירה של כל אחת מ-200 הרשתות הוא:

ראשית, מגרילים מספר רנדומי בין 1-4 שהוא יהיה מספר השכבות (שכבת הפלט לא כלולה במספר זה)

לאחר מכן, השכבה הראשונה היא בגודל 16 מכיוון והקלט הוא 16 ביטים.

לכל hidden layer אנחנו מגרילים מספר בין 1-10 שהוא יהיה מספר הנוירונים בשכבה.

לאחר כל שכבה הוספה פונקצית האקטיבציה sigmoid על מנת לשבור את הלינאריות.

וכמובן בסוף שכבת output בגודל נוירון אחד עם sigmoid מכיוון ואנו רוצים פלט של 1 או 0 כאשר ערך גדול מחצי יהיה 1 וקטן מחצי יהיה 0.

- כל המשקלים ההתחלתיים נקבעים בצורה רנדומלית

<u>אופן פעולת האלגוריתם הגנטי:</u>

כעת האלגוריתם הגנטי מחפש את הרשת הכי טובה בצורה הבאה:

- כתחילה עבור כל רשת מחשב את הfitness שלה על ידי הרצת כל הקלטים ברשת. לצורך חישוב הfitness השתמשנו בפונקצית הMSE loss. והערך עצמו מייצג את אחוז הדגימות שצדקנו עליהן.

- כעת כשאנו יודעים את הfitness של כל רשת, האלגוריתם לוקח את 40% הרשתות הכי טובות, ומעביר אותם לדור הבא כמו שהן.

- את ה60% הנותרים אנו נבנה באמצעות crossover שעובד כך:

א. מתוך האוכלוסייה של ה40% שאיתה המשכנו לדור הבא, בוחרים 2 רשתות כ-parents

ב. הרשת בעלת הfitness הגבוה ביותר היא זו שתשמש כארכיטקטורה של הרשת.

ג. כעת עלינו להחליט על המשקלים. עבור כל שכבה נבחר רנדומלית אינדקס i.
עבור child1 ניקח את המשקולות מparent1 עבור אינדקס 0 ועד ל-i. ומparent2 מi+1 ואילך.
עבור child2 נעשה הפוך. כלומר, ניקח את המשקולות מ-parent2 עבור אינדקס 0 ועד ל-i. ומ-parent1 מi+1 ואילך.

במידה וכמות המשקולות חורגת מparent1\parent2 (מכיוון והארכיטקטורות שונות אז ייתכן כי לכל אחד יש כמות שונה של נוירונים בכל שכבה). ניקח את המשקולות הקיימות מהparent שאכן יש לו את זה (כלומר הparent שבו השתמשנו לארכיטקטורה)

ד. עבור כ- 50% מהאוכלוסייה מבצעים MUTATE (מגרילים מספר בין 0 ל-1 ואם הוא גדול מ0.5 מבצעים MUTATE, 0.5 זה היפר-פרמטר שהביא תוצאות טובות בשם MUTATE_RATE).
הMUTATE מבוצע בצורה הבאה:
מגרילים מהתפלגות יוניפורמית מספר בין 0 ל-1.
אם הוא קטן מ0.8(היפר-פרמטר בשם RESET_RATE):
מגרילים בצורה יוניפורמית משקולת כלשהי, ומשנים את משקלה לערך מהתפלגות סטנדרטית.

אם הערך גדול מ0.8:
מאפסים רנדומלית 20% מהמשקולות של הרשת. (כלומר 20% מהמשקולות מקבלים ערך 0)

וכך האלגוריתם פועל למשך 300 דורות, במידה ורשת כלשהי הגיעה לדיוק של 99% הוא מדווח לנו באיזה דור זה קרה.

לבסוף הוא רושם את כל נתוני הרשת לתוך קובץ בשם wnet0.txt או wnet1.txt בהתאם לקובץ שהורץ בהתחלה (nn0.txt וnn1.txt בהתאמה)
פורמט שמירת הנתונים הוא:
שורה ראשונה היא מספר השכבות שיש לנו ללא שכבת הפלט

לאחר מכן מספר הנוירונים בשכבה הראשונה.
לאחר מכן ערכי המשקולות שמחברות בין השכבה הראשונה לשנייה

לאחר מכן כמות הנוירונים בשכבה השנייה
לאחר מכן ערכי המשקולות שמחברות בין השכבה השנייה לשלישית

וכך הלאה...

אופן טיפול בהתכנסות מוקדמת:

בכל ריצה עשירית (כלומר ריצה 10,20,30...)
הוא בודק האם הfitness הטוב ביותר בריצה הנוכחית השתפר בלפחות 0.001 מהfitness הכי טוב שהיו ב10 ריצות הקודמות.
במידה ולא:
מוסיפים לmutation_rate 0.25. כלומר מעלים את גודל האוכלוסייה שמבוצעת עליה mutate ב25% (במידה ולהוסיף 0.25 גדול מ-1 הערך יהיה 1, כלומר על כל האוכלוסייה מבצעים mutate)
באופן דומה, לselection_percentage (אחוז הרשתות שממשיכות כמו שהן לדור הבא) מורידים 0.1,כלומר גודל האוכלוסייה של הדור הבא יורכב מ10% פחות רשתות שממשיכות כמו שהן ובמקומן יבוצע CROSSOVER, אם הערך קטן מ0.2 לאחר ההפחתה של 0.1 - משאירים את זה 0.2.

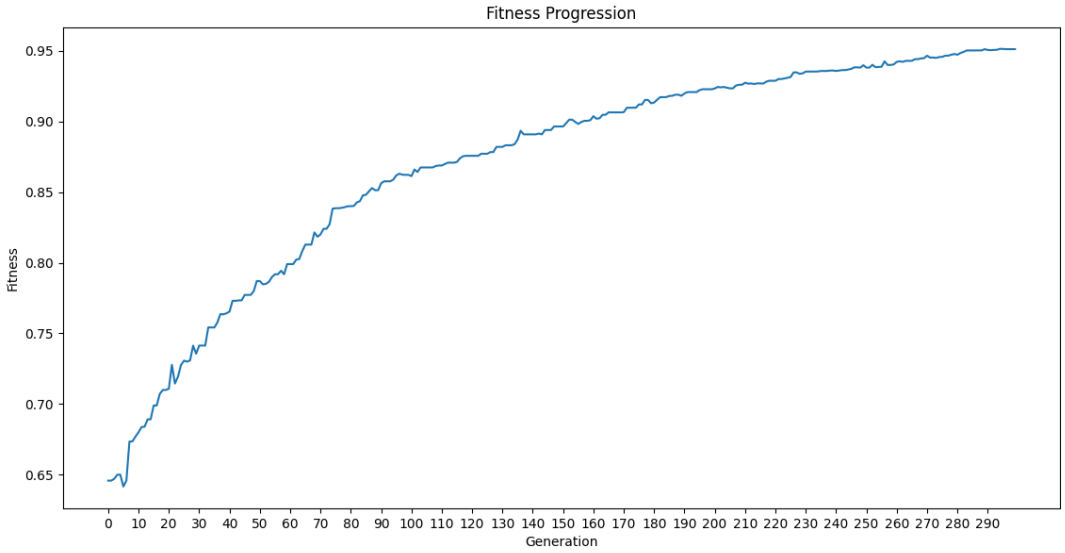
ביצועי התוכנית

nn0:

```
[0.9514375]
Test accuracy is 93.825
Model was saved in 'wnet0.txt' file.
```

דיוק בtrain 95.14%
דיוק בtest 93.825%

גרף fitness הכי גבוה – דור עבור הtrain:

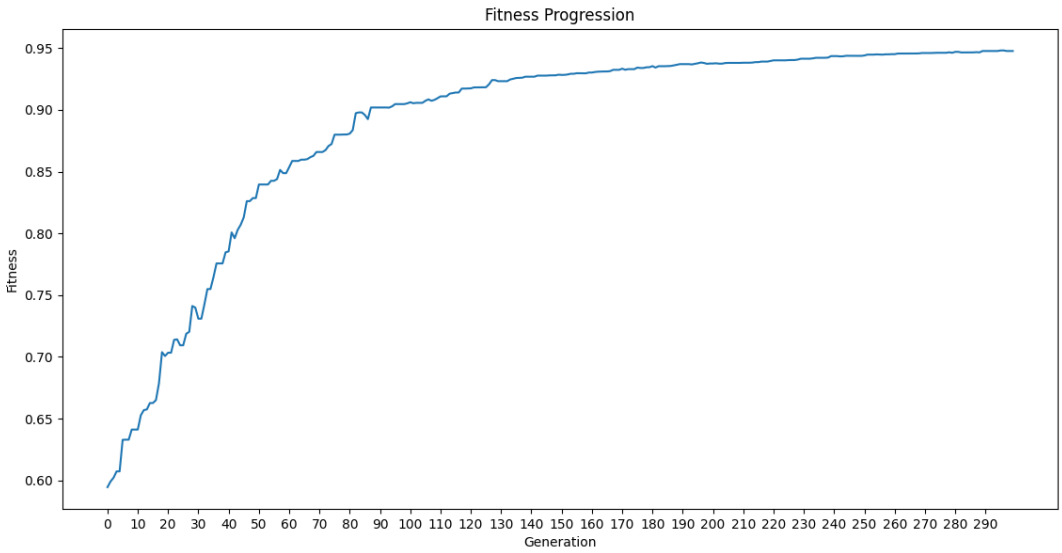


:nn1

```
[0.9479375]
Test accuracy is 94.75
Model was saved in 'wnet1.txt' file.
```

דיוק בtrain 94.79%
דיוק בtest 94.75%

גרף fitness הכי גבוה – דור עבור הtrain:



חוקיות

nn0 – לפחות 8 אחדות בסטרינג (כולל 8) מקבל label 1 אחרת label 0

nn1 – לפחות 8 אחדות בסטרינג (כולל 8) מקבל label 0 אחרת label 1