ביולוגיה חישובית תרגיל 3

מגישים:
דניאל יאנובסקי 314826975
גל גלעדי 312491616
סרטון הסברה : https://www.youtube.com/watch?v=m7t8PkeTKEc
גיט : https://github.com/dani020799/Targil3_biology/tree/main

Neural Network - instructions to use:

In order to run the splitter, so you get the 'training.txt' file and the 'testing.txt' file, you should use the 'spliter.py' program, and give it 2 args:
First arg is the file you wish to split, for example 'nn0.txt' file.
Second arg is the percentage of the lines you wish to have in the 'training.txt' file.
For example, 0.8 is 80%, so 80% of the lines of the 'nn0.txt' will be splitted into the 'training.txt' file, and 20% of the lines will be splitted into the 'testing.txt' file.
Then we will have output of 2 new files, 'training.txt' and 'testing.txt' according to the split we wished to have:

```
C:\Users\Gal Giladi\Downloads\Targil3_biology-main\Targil3_biology-main>python spliter.py nn0.txt 0.8
```

In order to run the buildnet, you should use the 'buildnet0.exe' or 'buildnet1.exe'.
We should give it 2 args:
First arg is the 'training.txt' file, and second arg is the 'testing.txt' file.
The output will be the best fit calculated, and in the end, it will make a 'wnet.txt' file:

```
C:\Users\Gal Giladi\Downloads\Targil3_biology-main\Targil3_biology-main>buildnet0.exe training.txt testing.txt

best fit: 14082
```

Notice that we have got 'training.txt' file, and 'testing.txt' file, using the 'spliter.py' program above, which splits the 'nn' file using percentage.
In our testing, and training files, we decided to split by 80%, so basically the 'training.txt' file contains 16,000 lines, while the 'testing.txt' file contains 4,000 lines.

In order to convert ‘nn0’ and ‘nn1’ files to be without classifications, you should use the ‘extract_bits.py’ program, and give it as an arg the file you want to change and eliminate the classification:

```
C:\Users\Gal Giladi\Downloads\Targil3_biology-main\Targil3_biology-main>python extract_bits.py nn0.txt
Extraction complete. The 16-bit strings without the classification have been saved to output_nn0.txt.txt

C:\Users\Gal Giladi\Downloads\Targil3_biology-main\Targil3_biology-main>python extract_bits.py nn1.txt
Extraction complete. The 16-bit strings without the classification have been saved to output_nn1.txt.txt
```

In order to run the runnet, you should use the runnet0.exe and runnet1.exe.
For example, if we use it on ‘nn0’ file, we should give it 2 args:
First arg is the ‘wnet0.txt’, and second arg is the ‘nn0’ file without the classification (in order to have the nn file without the classification, use the program above).
Then we will get a file as an output which is the output we wished to have.. Namely an output which is similar to the ‘nn0.txt’ file in our case, with all the classifications (or at least with 90% success):

```
C:\Users\Gal Giladi\Downloads\Targil3_biology-main\Targil3_biology-main>runnet0.exe wnet0.txt output_nn0.txt.txt
Weights list:
(-61.98233332950807, 61.98233332950807)
(-54.82781540434033, 54.82781540434033)
(-56.34180033955468, 56.34180033955468)
(-50.58677504806931, 50.58677504806931)
(-66.92621074637158, 66.92621074637158)
(-51.47085746094315, 51.47085746094315)
(-50.94787974599488, 50.94787974599488)
(-51.199420198100825, 51.199420198100825)
(-53.104467425038905, 53.104467425038905)
(-56.16928221766889, 56.16928221766889)
(-48.970343849974356, 48.970343849974356)
(-54.03738165758614, 54.03738165758614)
(-61.46988778126618, 61.46988778126618)
(-49.441509942615546, 49.441509942615546)
(-45.02591873862913, 45.02591873862913)
(-56.43421001561905, 56.43421001561905)
Output file 'output_with_classification_nn0.txt' created successfully.
```

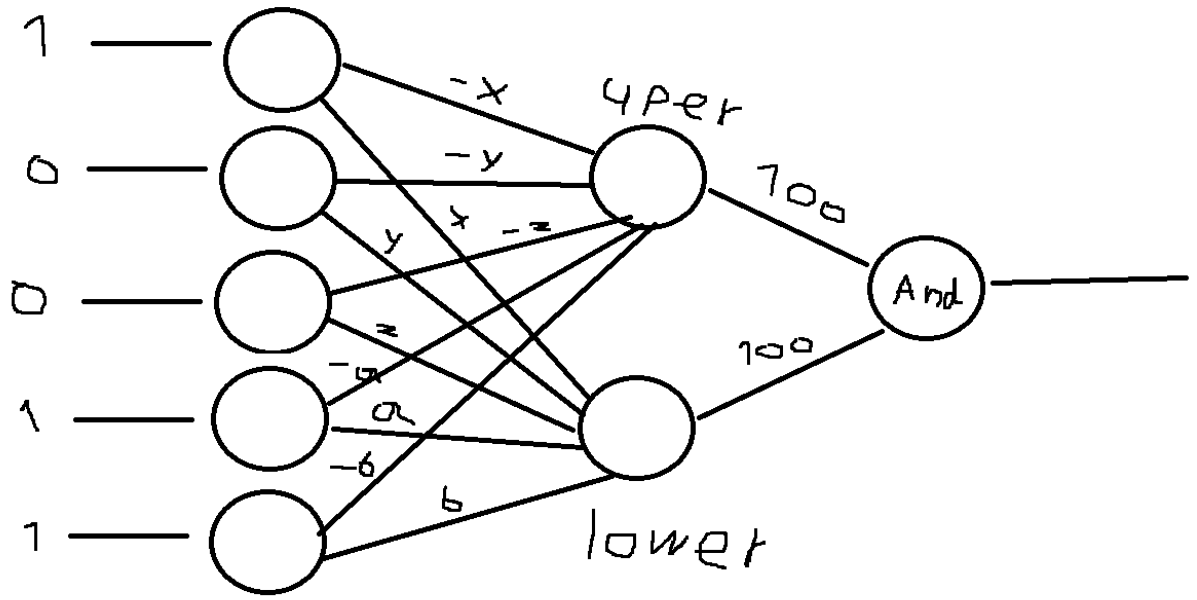
In order to check if the output with the classifications we got is similar/identical, you should use the compare_files.py program, and give it the 2 files you want to compare.
For example, I give as a first arg the ‘nn0.txt’ file we got at the assignment, and the second arg is the ‘output_with_classification_nn0.txt’ file we have got from the ‘runnet0.exe’ program above.
The output will be the different lines between them:

```
C:\Users\Gal Giladi\Downloads\Targil3_biology-main\Targil3_biology-main>python compare_files.py nn0.txt output_with_clas
sification_nn0.txt
Number of different lines: 209
```

חוקיות של הקבצים

בשני הקבצים כל המשקלים זהים.
החוקיות בקובץ הראשון היא כמות הפעמים שהספרה '1' מופיעה, שהיא גדולה שווה ל 8 וקטנה שווה ל 12.
החוקיות בקובץ השני היא כמות הפעמים שהספרה '1' מופיעה, שהיא קטנה שווה ל 8.

הרשת ניורונים שיצרנו



יצרנו רשת שהשכבה הראשונה שלה היא 16 נוירונים שיקבלו את הביטים. שכבה שנייה של שני נוירונים, כאשר הנוירון הראשון יהיה החסם העליון של הסכום שיקבל מספרים שליליים. לחסם העליון יהיה בייס חיובי ויקבל מספרים שליליים.
כדי לדעת אם עברנו את החסם העליון, נחשב את הסכום של המשקלים

כפול המערך ביטים ועוד הבייס שלילי, אם הסכום הוא שלילי - זה אומר שעברנו את החסם העליון.
נגיד בקובץ השני החסם עליון זה 8 כולל אם הבייס 850 וכל המשקלים הם מינוס 100 מ 9 ביטים ומעלה הסכום יהיה שלילי זה אומר שהניורון יחזיר 0.
הנוירון השני בשכבה השנייה הוא החסם התחתון ויקבל משקלים חיוביים. המשקלים בנוירון הזה לעומת הראשון שווים בערך מוחלט.
הבייס של הנוירון השני יהיה שלילי ויקבל מספר חיוביים ככה שהסכום של המשקלים כפול הקלט ועוד הבייס יראה אם עברנו את החסם התחתון או לא.
נגיד אם החסם התחתון הוא 8 וכל המשקלים הם 100 אז הבייס יהיה נגיד מינוס 750 אז 8 ביטים ומעלה יוחזר מספר 1 מהנויורון כי זה אומר שעברנו את החסם תחתון.

השכבה השלישית היא נוירון אחד שיהיה בעצם AND GATE .
הבייס שלו הוא מינוס 150 ויקבל ביט כפול המשקל 100 מכל נוירון.
אם שני הנוירונים יחזירו אחד זה אומר שאנחנו בין החסם העליון לחסם התחתון ולכן זה תקין. והנוירון של השער 'וגם' יחזיר אחד.
אם אנחנו עברנו את החסם עליון או לא עברנו את התחתון אז אחד הניורונים יחזיר 0 ולכן השער וגם יחזיר 0.

האלגוריתם הגנטי

קודם כל יש אובייקט של רשת נוירונים שהיא מורכבת מ Tupples של משקלים: אחד לכל ניורון ושני נוירונים שאחד מהם זה חסם תחתון והשני עליון, נוירון בשכבה השלישית שהוא שער 'וגם'.
האוכלוסייה היא 150.
בהתחלה יוצרים 150 רשתות רנדומליות, המשקלים הם float בין 0 ל 100, והבייסים של השכבה השנייה הם רנדומליים בין 0 לסכום של כל המשקלים החיוביים.
הפיטנס זה כמה תוצאות יצאו כמו מה שמצופה.
עשינו שני פונקציות cross, אחד עושה ממוצע בין המשקלים, השני בוחר נקודה שהחלק ראשון של האובייקט במשקל הראשון, והחלק השני מהשני.
בשתי הפונקציות עושים ממוצע של הבייסים.

יצירת הדור הבא

קודם כל עשינו מיון לפי הפיטנס ככה שהגבוה ביותר יהיה הראשון, ואותו נכניס כמו שהוא לדור הבא.
לאחר מכן, שמינית מהדור הבא עושים אליטיזם בוחרים מהטופ 15 ועושים קרוסים ביניהם .
ואז 90 אחוז בוחרים בחירה בררנית ככה שמי שיש לו פיטנס גבוה יותר, הסיכוי שלו להיבחר כמובן גבוה יותר, והשאר יוצרים רשתות רנדומליות חדשות.

קובץ wnet

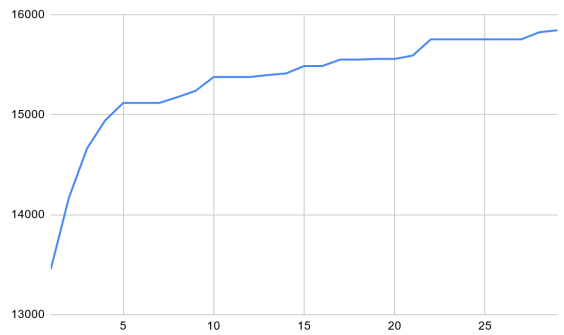
בקובץ יש 18 שורות.
16 שורות שהם המשקלים החיוביים כדי לבדוק את החסם התחתון בתוכנית runnet אנחנו עושים אותם גם שלילים לחסם העליון.
שורה 17 - הבייס של החסם העליון המקבל את הנוירונים עם המשקלים השליליים.
שורה 18 - הבייס של החסם התחתון המקבל את הנוירונים עם המשקלים החיוביים.
בחרנו לבצע את הקובץ כך לשם הנוחות, שכן אנו לא צריכים מידע נוסף, וקל להוציא מהקובץ את הנתונים שצריכים.

הרצות

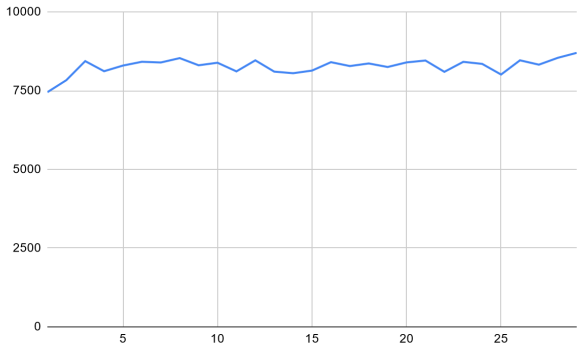
buildnet0

הרצה מספר 1

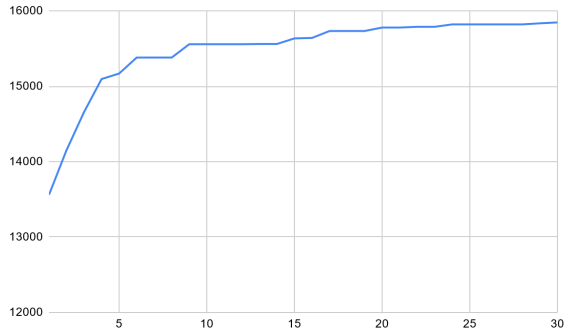
best

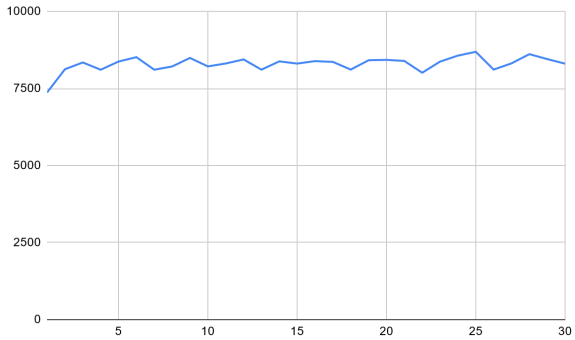


Avrage

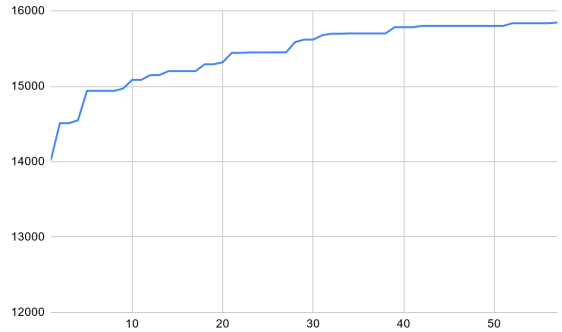


הרצה מספר 2:



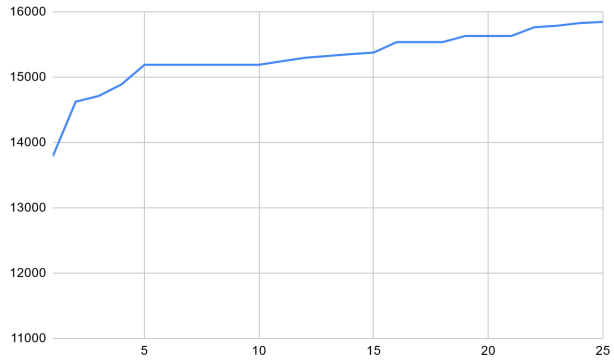


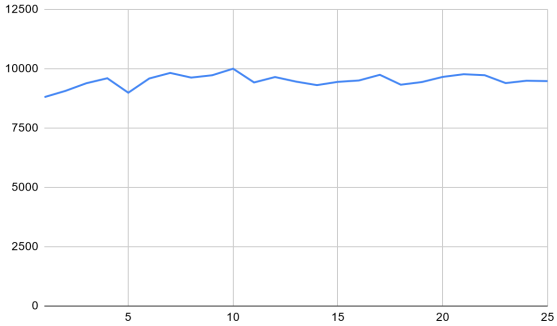
הרצה מספר 3:



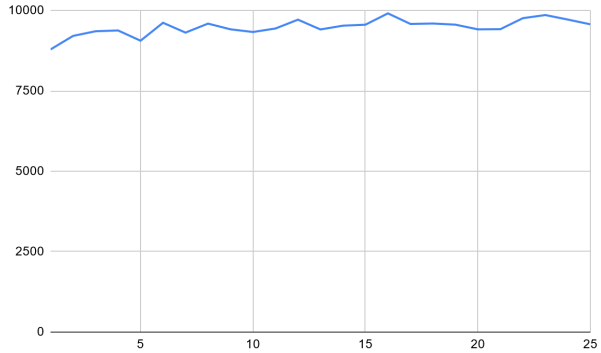
Buildnet1

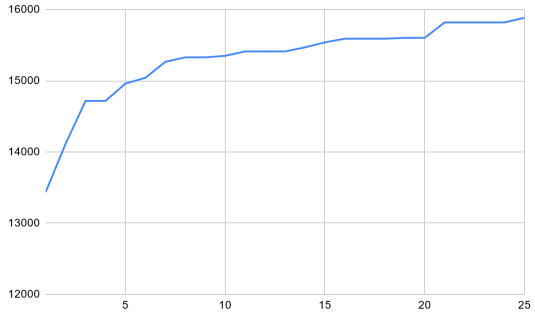
הרצה מספר 1



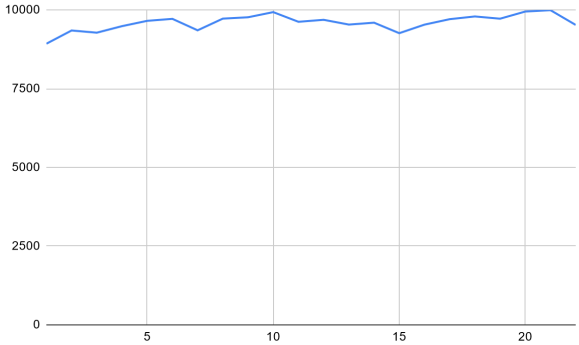


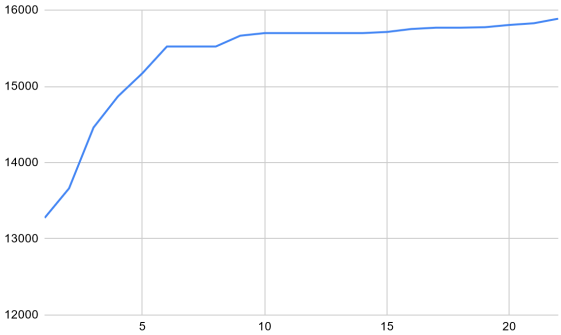
הרצה מספר 2





הרצה מספר 3





ניתוח תוצאות

אם מסתכלים על הפיטנס הטוב ביותר, רואים שמלכתחילה, יוצא פיטנס טוב של 75 אחוז לפחות.
ניתן לראות התקדמות די גדולה בהתחלה ואז רואים תקיעה הרבה דורות.
הפיטנס לא משתנה ומדי פעם יש קפיצה קטנה, וכן הלאה.
קל לראות שהגרף דומה בהרצה על שני הקבצים, למעט ההגעה לפתרון בקובץ השני בצורה מהירה יותר.
מבחינת הממוצע רואים שהוא ב'זיגזג', ועולה בצורה איטית לעומת ההתקדמות ההתחלתית, אבל עדיין הממוצע בסוף לא שונה בהרבה מהממוצע בהתחלה.
כך זה מתקיים עבור שני הקבצים.