בסייעתא דשמיא

דו"ח ביולוגיה חישובית- תרגיל 3

רשתות נוירונים- מימוש SOM

מגישות: שיר בן אהרון, תמר סעד

הוראות הרצה לקוד:

כדי להריץ את התוכנית יש צורך להריץ את קובץ ה- python הנתון באמצעות **דאבל קליק** על הקובץ מהתיקייה בה הוא נמצא, או באמצעות **windows cmd** ע"י שימוש בפקודה- python ex3.py. (אנחנו מניחות שקיים python על המחשב והוא מקושר ל-cmd).

<u>התוכנית מורידה בעצמה חבילות נדרשות להרצתה, ולא דורשת פרמטרים חיצוניים לפקודת ההרצה.</u>

(בשל הנאמר מעלה והורדת החבילות, יתכן כי ההרצה הראשונה תיכשל ולכן יש לנסות להריץ פעם נוספת).

- קלט מהמשתמש- בתחילת ההרצה המשתמש מתבקש להכניס את שם קובץ ה-csv שמכיל את נתוני ההצבעה. <u>יש לתת את השם המלא עם הסיומת המתאימה לקובץ באופן הבא:</u>

```
Please Enter A File Name For The Program:
Elec_24.csv
```

- אם קובץ הקלט לא נמצא בתיקייה שבה ממוקם קובץ הריצה, יש להכניס בשם הקובץ את הנתיב המלא אליו.

תיאור הפלט:

- מספר ה-epochs שנדרשו להרצת התוכנית. הגבלנו את התוכנית ל-100 epochs, ואפשרנו לה לעצור בשלב מוקדם יותר במקרה של התכנסות.
- <u>חלוקת היישובים לפי נוירונים:</u>
  התוכנית מדפיסה את מיקום הנוירון על הגרף ולאחר מכן את כל הישובים שמופו כקלאסטר לנוירון זה יחד עם המצב הכלכלי שלהם.
- <u>תמונת התוצאה הסופית של רשת הנוירונים:</u>
  בתמונה מופיעים הנוירונים בצורת משושים, כאשר הצבע של כל נוירון נקבע לפי ממוצע המצב הכלכלי של כל היישובים שמופו אליו.
  נוירון בצבע בז' הוא נוירון שלא מופה אליו אף ישוב.
- <u>ציון ההרצה:</u>
  בסוף ריצת האלגוריתם מחושב ממוצע המרחקים בין הישובים לבין הנוירונים אליהם הם מופו. ככל שהמרחק הממוצע קטן יותר, התוצאה שהתקבלה טובה יותר.

מימוש האלגוריתם:

לאחר טעינת הדאטה, הפרדנו את עמודת המצב הכלכלי מכל יישוב כך שנשארנו רק עם נתוני ההצבעות של כל יישוב.
נרמלנו את ערכי ההצבעה של כל ישוב לפי התפלגות ההצבעות בו, כך שהמרנו את הערכים המספריים של התפלגות ההצבעות לאחוזים (מספרים בין 0 ל-1).
בנוסף, אתחלנו 61 נוירונים באורך זהה לאורך הווקטור של כל ישוב-14 features, עם ערכים אקראיים בין 0 ל-1.

<u>בכל epoch במהלך ההרצה (100 epochs, או עד הגעה להתכנסות):</u>

- **חישוב המרחק בין יישוב בקלט לנוירון:** חיפשנו עבור כל ישוב את הנוירון שהערך הווקטורי שלו הוא הקרוב אליו ביותר (השתמשנו ב-RMS distance).
- **קירוב התא ושכניו לקלט שמופה אליו:**
  - עדכנו את הערכים של אותו נוירון "להתקרב" לערכים של הישוב באמצעות הנוסחה:
    $$new\ neuron = neuron + 0.3 * (town - neuron)$$
  - עדכנו את הערכים של 6 השכנים הקרובים לנוירון (מעגל ראשון) "להתקרב" לערכים של הישוב באמצעות הנוסחה:

בסייעתא דשמיא

$$new\ neuron = neuron + 0.2 * (town - neuron)$$

בצורה כזו גרמנו לנוירונים השכנים להתקרב לישוב שהתמיין לנוירון המרכזי, אבל בקנה מידה קטן יותר.

- **בדיקת התכנסות:** בדקנו האם כל הישובים באיטרציה הנוכחית מופו לאותם הנוירונים כמו באיטרציה הקודמת. במקרה שכך היה, עצרנו את ההרצה.

תוצאות:

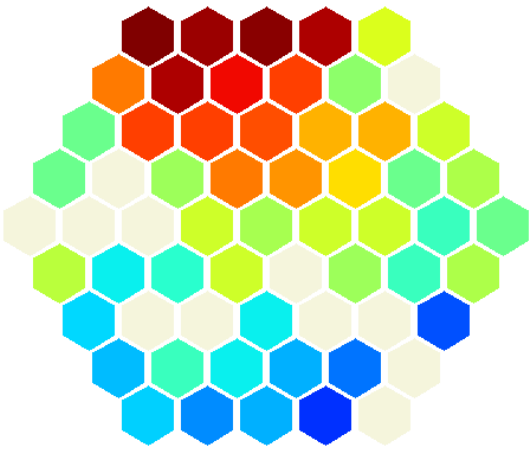
על מנת לקבל תוצאות טובות ככל האפשר, בדקנו את האלגוריתם בשינוי מספר פרמטרים:

- עדכון ערכי נוירונים של שכנים ברמה ראשונה בלבד (ששת הנוירונים הסמוכים לנוירון הנבחר)
- עדכון ערכי נוירונים של שכנים ברמה ראשונה בלבד, וכן ערבוב הישובים בתחילת כל epoch
- עדכון ערכי נוירונים של שכנים ברמה ראשונה ושניה (גם שנים עשר הנוירונים שמקיפים את השכנים בדרגה הראשונה)
- עדכון ערכי נוירונים של שכנים ברמה ראשונה ושניה, וכן ערבוב הישובים בתחילת כל epoch

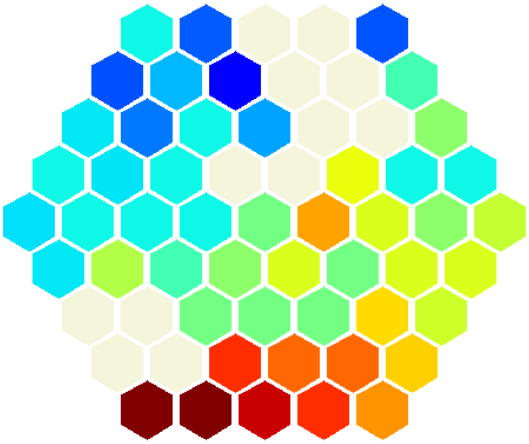
כל אחת מגרסאות ההרצות הנ"ל נבדקה ל-10 הרצות שונות של האלגוריתם, על מנת לבחון מי מהן המוצלחת ביותר.
*יש לציין כי בשתי הגרסאות בהן לא היה ערבוב בכל epoch, נעשה ערבוב יחיד לפני תחילת ההרצה.
*בגרסאות בהן עדכנו רמה נוספת של שכנים, העדכון נעשה בצורה דומה למתואר בחלק "מימוש האלגוריתם". מקדם המכפלה היה 0.1.
בסיום כל ריצה חישבנו על הלוח הסופי שהתקבל את המרחק הממוצע של כל ישוב מהנוירון שאליו הוא התמיין (מתוך הנחה שככל שהמרחק הממוצע נמוך יותר, כך הלוח מייצג פתרון טוב יותר).
לבסוף, השווינו בין התוצאות של הגרסאות השונות:

הלוחות שהתקבלו:

שכנים מדרגה ראשונה:



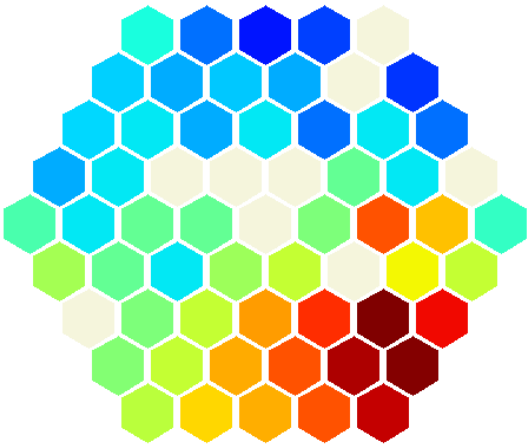
שכנים מדרגה ראשונה + ערבוב:

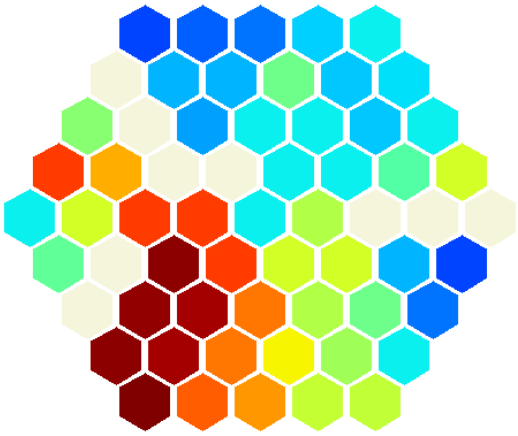


שכנים מדרגה שניה:

שכנים מדרגה שניה + ערבוב:

בסייעתא דשמיא





צבע כל נוירון נקבע לפי ממוצע המצבים הכלכליים של הישובים שמופו כקלאסטר לנוירון.
ניתן לראות כי בכל התמונות קיימים מספר נוירונים בצבע בז', משמע שום נוירון לא שמופה אליהם.
ניתן לראות בכל התמונות כי נוצרים אזורים בהם הנוירונים הם בעלי צבעים זהים ואף קשת הצבעים דומה, כלומר באמצעות המיפוי ניתן לומר כי ישנה הלימה מסוימת בין מצב כלכלי של ישוב לבין התפלגות ההצבעות שלו בבחירות, ואכן בהדפסת הישובים שמופו לכל נוירון ניתן לראות כי לרוב המצב הכלכלי של הישובים באותו נוירון זהה או קרוב מאוד.
על מנת לקבל מושג טוב יותר על טיב כל פתרון שהתקבל ע"י אלגוריתם, נתבונן בטבלת המרחקים הממוצעים של כל ישוב מהנוירון שלו:

1st neighbors	1st neighbors with shuffle	2nd neighbors	2nd neighbors with shuffle
0.025945943	0.02828929	0.032476634	0.036283775
0.026733835	0.026325813	0.033846932	0.03197768
0.028268987	0.027637295	0.03014556	0.031590351
0.026162919	0.028358751	0.034526839	0.033117525
0.027472845	0.027993568	0.031562287	0.034200334
0.026038356	0.028446304	0.032364505	0.035339525
0.025042852	0.027819586	0.032251958	0.033551756
0.027238378	0.026310319	0.032840992	0.03244749
0.028277284	0.026511812	0.032949797	0.032676144
0.028034803	0.027262442	0.034427252	0.032663563

	1st neighbors	1st neighbors with shuffle	2nd neighbors	2nd neighbors with shuffle
min	0.025042852	0.026310319	0.03014556	0.031590351
max	0.028277284	0.028446304	0.034526839	0.036283775
avg	0.02692162	0.027495518	0.032739276	0.033384814

בטבלה העליונה ניתן לראות את ממוצע המרחקים עבור כל איטרציה של כל גרסת הרצה.
בטבלה שמתחתיה ניתן לראות נתונים מסכמים: מינימום, מקסימום וממוצע.
ניתן לראות כי הגרסה שנתנה לנו את התוצאות הטובות ביותר היא זאת שבה מתבצע עדכון של רמה אחת של שכנים, וכן שאין בה ערבוב בכל epoch, אף על פי שהתוצאות של הגרסה הזו שכן מבצעת ערבובים טובות כמעט באותה המידה.
עם זאת, הגרסה ללא הערבוב נתנה לנו התכנסות מהירה בחלק גדול מהמקרים, בעוד הגרסה עם הערבוב לא התכנסה גם לאחר 100 epochs.

לכן, זו הסיבה שבחרנו להגיש את האלגוריתם כמו שהוא, עם עדכון של רמת שכנים אחת בלבד וללא ביצוע ערבוב בכל epoch.