**פרויקט SOM**

**דו"ח סיכום מאת שלומי אוחנה ונעה בייר**

**מבוא**

הפרויקט אשר בחרנו לבצע הינו אלגוריתם לבניית מפות SOM – שהוא בעצם רשת נוירונים לעיבוד נתונים. בקצרה, מדובר באלגוריתם המבצע סיווג/חלוקה של כמות נתונים גדולה, ממימד כלשהו, למספר מצומצם יותר של קבוצות (תאים במפה).

הנתונים אותם אנו מקבלים מן המשתמש ייוצגו ע"י וקטורים. האלגוריתם מחלק את הנתונים למספר תאים במפה, וזאת כאשר כל תא מיוצג באמצעות וקטור, ומתעדכן בשלב הלמידה בהתאם לנתונים המשויכים אליו. אל התא משויכים מספר לא קבוע של וקטורים.

השכנים של כל תא מעודכנים גם-כן בכל הכנסה של מידע חדש לאחד מהתאים הסמוכים לתא הנתון.

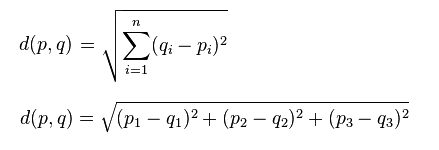
אלגוריתם זה מספק לנו תובנות על מידע גדול כאשר אין ביכולתנו לסווג את המידע.

**האלגוריתם**

שלב 1:

ניתן לכל Node (תא) במפת ה SOM שלנו וקטור משקל אקראי, אשר יוגרל בעת יצירת מבנה הנתונים. הוקטור האקראי אותו נגריל צריך להיות מאותו המימד כמו בנתונים אותם נרצה להכניס אל המפה.

לאחר מכן כל נתון ישתמש בפונקציית מרחק על מנת לזהות את התא הכי קרוב אליו (כלומר, ה-BMU שלו, שמהווה argmin על המרחקים); בפועל, אנו עושים שימוש בפונקציית המרחק האוקלידית המוכרת לנו:



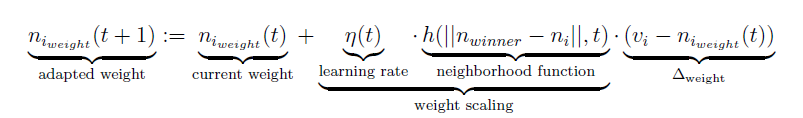
שלב 2:

נשייך את הוקטור אל התא הנבחר ונתחיל במלאכת העדכונים: נעדכן את התא הנבחר ע"י פונקציית עדכון משקל. כעת, נעשה זאת גם לתאים השכנים של התא הנבחר, בהתחשב במרחק שלהם מהתא הנבחר.

לאחר שסיימנו להכניס את כל הוקטורים אל תוך המפה, נמחק אותם מן התאים, אך הפעם משקלם של התאים ישמר מהאיטרציה הראשונה ולא יוגרל אקראית.

כעת סיימנו cycle (איטרציה) אחד ונמשיך לעשות זאת כמספר ה cycles אותו נבחר.

עדכון המשקלים:



החלק הגדול של הפרויקט נשען על עדכון משקלם של התאים הנבחרים או שכניהם ומתבצע ע"י פונקציית עדכון משקל המורכבת ממספר פרמטרים:

* המשקל הנוכחי של התא
* קצב למידה – נוסחה התלויה במספר האיטרציות אותן נעשה.
* t הינו מספר האיטרציה הנוכחית.
* n0 הינו בעל ערך התחלתי קבוע של 0.1.
* tmax הינו מספר האיטרציות הכולל אותה התוכנית תסמלץ.



* פונקציית השכנים – זהו נרמול המרחק בין השכן לבין התא המנצח (מקבל 1 כאשר נעדכן את התא המנצח בעל ה-BMU הקטן ביותר מהוקטור אשר אותו נכניס).



* המרחק בין הוקטור אותו אנו מכניסים לבין התא הנוכחי.

לאחר שחישבנו את הפרמטרים עליהם הרחבנו למעלה, נחברם ונקבל וקטור משקל מעודכן לתא אותו עדכנו.

**רשימת מחלקות**

**NODE**– מחלקה שמייצגת תא במפה. שדות: אינדקסים שמציינים את מיקום התא, משקל, ומערך של שכני התא.

**DataVector** - מחלקה המייצגת נתון אחד. מיוצג באמצעות וקטור משקל.

**SOM**- מחלקה המייצגת את המפה עצמה. האלגוריתם של מפת SOM ממומשת באחת מהפונקציות של המחלקה. שדות המחלקה: מימידי המפה, מימדי וקטור הנתונים, מפת סיווג שמסווגת כל נתון לאינדקס במפה, מערך דו מימדי של NODE כלומר מבנה הנתונים שמחזיק את מפת SOM. מתודה **setSomMap(int max cycle, List<DataVector> data)** שמריצה את אלגוריתם הקלסיפיקציה. הפרמטרים הם הנתונים ומספר האיטרציות. מתודה **updateWeightNeighbers** – מעדכנת את המשקל של התא שמעובר בפרמטר על פי הפרמטרים המעוברים. מתודה זה בעצם היא המימוש לפונקציית עידכון השכנים שהוסברה בדף הקודם. מתודה **printMap()** שמדפיסה את תוצאות המפה לקובץ פלט.

**ResultColoursSom** – מחלקה זאת אחראית על ייצוג בגוי של תוצאות מפת SOM. יש לה גישה לתוצאה הסופית של המפה, ולקלסיפקציה של הנתונים לכל תא.

**ResultWords** – מחלקה זו אחראית על ייצוג גוי של נתונים שאינם לקוחים מתחום הצבעים RGB. הם מייצגים רק את האינדקסים של כל תא ובעזרת מתן צבע לכל קשת המחברת בהם, ניתן להבחין בדימיון של התאים. (צהוב- אין קרבה גדולה, אדום כהה יש קרבה גדולה \*קיים גם בייצוג גוי של צבעים).

**OpeningPageController** – אחראי על פתיחת המסך הראשי בגוי ועל קליטת קלט מהמשתמש ובדיקת תקינותו. לאחר לקיחת הקלט מהמשתמש, חלון זה אחראי לפתוח את חלון גוי שמייצג את תוצאות המפה לפי קלט המשתמש.

**Main** – מפעיל את חלון OpeningPageController על ידי JAVAFX . פותח את קובץ FXML שOpeningPageController אחראי על השליטה בקלט שמגיע מחלון זה.

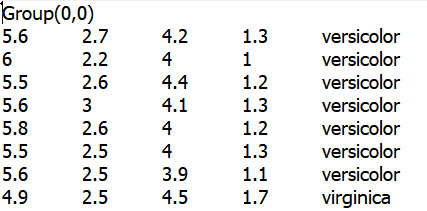
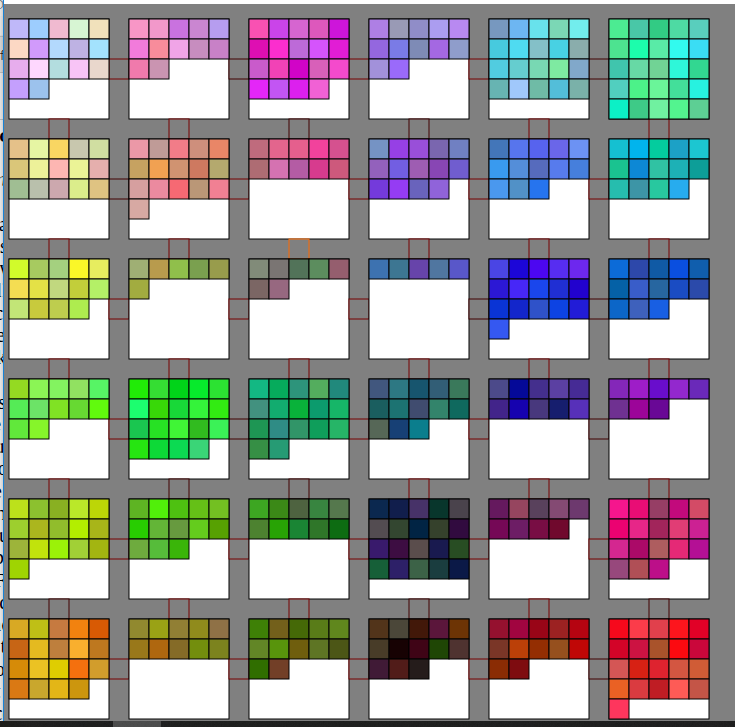
**הפעלת התוכנית**

בעת הפעלת התוכנית יוצג למשתמש מסך עם מספר פרמטרים אותם יצטרך להכניס על מנת להמשיך:

* גודל המפה אליה ירצה המשתמש למפות את נתוניו (אופציות אפשריות הינן מפות בגדלים 3X3 ו-6X6).
* טווח מספרים מקסימלי ומינימלי של הוקטור.
* מספר הוקטורים (נתונים) אותם נרצה למפות אל מפת ה-SOM.
* מימדם של הוקטורים.
* אופציה להצגה של צבעים (לצורך הצגת הפרויקט), בה נשתמש במידה ונרצה להציג נתונים העומדים בקריטריונים של RGB (וקטורים ממימד 3 בעל טווח מספרים של 0-255).
* שם הקובץ כולל סיומת txt אשר ממנו נקרא את הוקטורים ונכניסם היישר אל המפה אותה אנו יוצרים.

פלט:

פלט התוכנית יודפס אל קובץ אחד ולפני כל מספר וקטורים של מידע יופיע מספר התא אליו הוא השתייך בסוף הריצה. בנוסף, נציג בגוי את התאים ובין התאים יהיו קווים בצבע אשר מסמלים את הדמיון בין התאים. צהוב – התאים אינם דומים מרחקם גדול מ-200, אדום כהה- התאים מאוד דומים ומרחקם קטן מ-50. לפלט של וקטורי RGB הוספנו פלט ויזואלי עם הצבעים אותו נראה בעת סיום הרצת התוכנית זאת במידה וסימנו את הריבוע "colors".

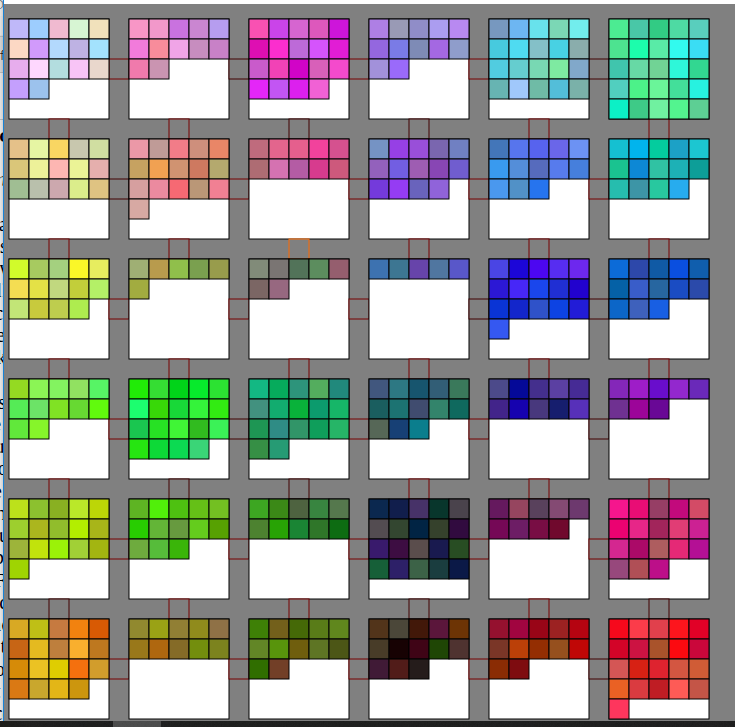


**הדגמות ותוצאות**

הדוגמא הראשונה אותה הרצנו היו וקטורי RGB. הנתונים אותם הזנו אל מסך ההפעלה של התוכנית היו:

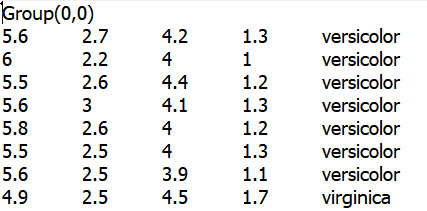
* גודל מפה – 6X6
* טווח מספרים בוקטור – 0 עד 255
* מספר הנתונים אותם נכניס אל המפה – 500
* מימד הוקטורים – 3
* שם קובץ – input.txt

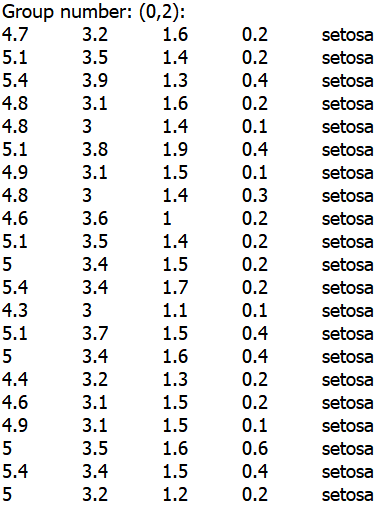
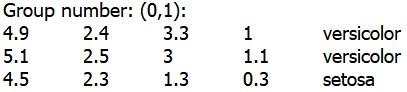
פלט התוצאות הוצג ע"י פלט ויזואלי:

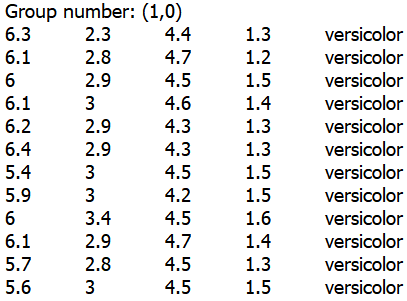


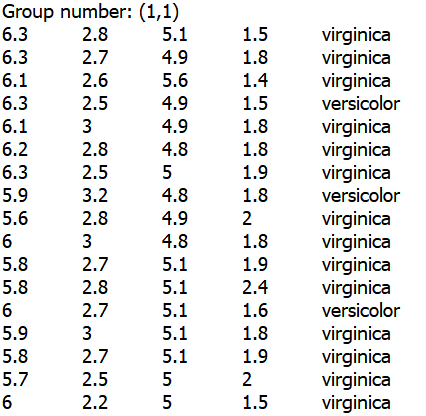
הדוגמא השנייה אותה הרצנו הייתה על מאגר נתונים בשם iris; מדובר בפרמטרים גיאומטריים של עלי הכותרת והגבעול. הנתונים אותם הזנו אל מסך ההפעלה של התוכנית היו:

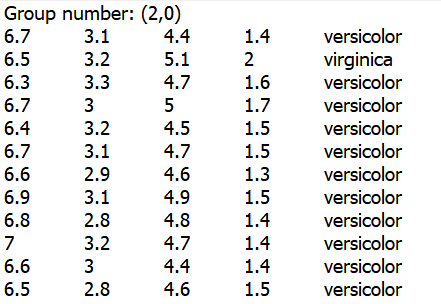
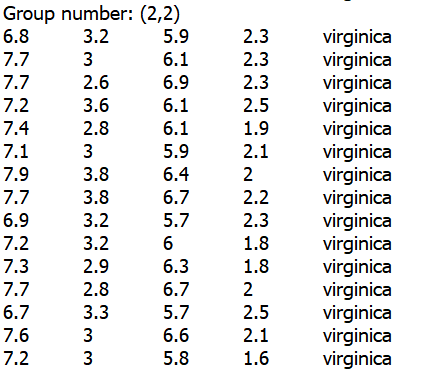
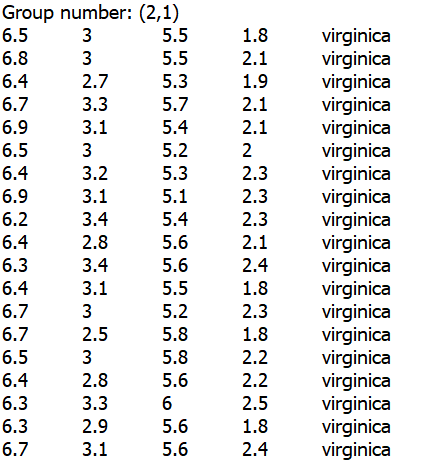
* גודל מפה – 3X3
* טווח מספרים בוקטור – 0 עד 255
* מספר הנתונים אותם נכניס אל המפה – 150
* מימד הוקטורים – 4
* שם קובץ – input.txt

פלט התוצאות הוצג ע"י פלט אל תוך קובץ טקסט:









**בעיות במהלך הדרך**

במהלך הכנת הפרויקט נתקלנו בלא מעט קשיים אשר הפכו את העבודה לקשה יותר, אך למרות הכל צלחנו אותן בצורה טובה. הנה חלק מן הבעיות העיקריות איתן התמודדנו:

**בדיקת תוצאות** - הבעיה העיקרית בה נתקלנו היא בדיקת התוצאות אותן קיבלנו. המפה אמורה לסייע לנו לסווג מספר רב של נתונים, דבר שהקשה עלינו להמשיך הלאה. כאשר נתקלנו בבעיות שונות בתוצאות אותן קיבלנו היה קושי רב לדבג.

**הבנת האלגוריתם** – האלגוריתם של המפה נראה לנו פשוט במבט ראשוני, אולם עם הזמן הבנו כי יש לנו הרבה דברים שלא היו כל כך ברורים. למזלנו, עופר היה שם.

**בעיית הדיבאג -** הייתה בעיה לדבג את האלגוריתם במקרה של תוצאות לא הגיוניות. קשה לדבג עם מספר נתונים כל כך גדול, אך עם הרבה סבלנות התגברנו גם על זה.

**GUI** – במהלך התואר לא עבדנו עם כזו רמה של GUI, אולם לאחר מאמצים רבים עמדנו גם בזה.

ביבליוגרפיה:

Brauer, C .(2012) .Proseminar Artificial Intelligence, Proseminar Paper.