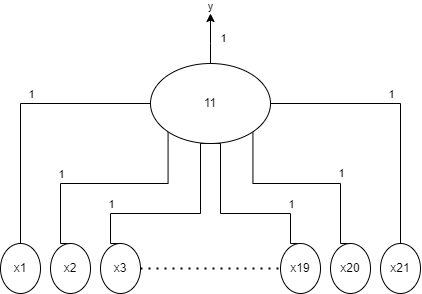
**ממ"ן 13**

**שאלה 1**

א. ניתן לבצע משימה זו עם פרספטרון חד שכבתי, שכן מדובר בבעיית סיווג לקבוצות על פי סף (מס' אחדות גדול שווה/קטן מ-11). להלן ציור של פרספטרון מתאים:



ב. שאלה 12:

a. להלן המטריצה:

b. נראה שהמטרציה יציבה. לכל אחד מהוקטורים בשאלה, נחשב האם הוא צריך להשתנות על פי שכניו, ונקבל כי הוא כבר יציב:

initial vector: (1, 1, 1, 1, 1)

\* v1 is selected

1 \* (-1) + 1 \* 1 + 1 \* 1 + 1 \* (-1) = 0

v1 stays 1

\* v2 is selected

1 \* (-1) + 1 \* 1 + 1 \* 1 + 1 \* 3 = 4

v2 stays 1

\* v3 is selected

1 \* 1 + 1 \* 1 + 1 \* (-1) + 1 \* 1 = 2

v3 stays 1

\* v4 is selected

1 \* 1 + 1 \* 1 + 1 \* (-1) + 1 \* 1 = 2

v4 stays 1

\* v5 is selected

1 \* (-1) + 1 \* 3 + 1 \* 1 + 1 \* 1 = 4

v5 stays 1

initial vector: (1, -1, -1, 1, -1)

\* v1 is selected

(-1) \* (-1) + (-1) \* 1 + 1 \* 1 + (-1) \* (-1) = 2

v1 stays 1

\* v2 is selected

1 \* (-1) + (-1) \* 1 + 1 \* 1 + (-1) \* 3 = (-4)

v2 stays (-1)

\* v3 is selected

1 \* 1 + (-1) \* 1 + 1 \* (-1) + (-1) \* 1 = (-2)

v3 stays (-1)

\* v4 is selected

1 \* 1 + (-1) \* 1 + (-1) \* (-1) + (-1) \* 1 = 0

v4 stays 1

\* v5 is selected

1 \* (-1) + (-1) \* 3 + (-1) \* 1 + 1 \* 1 = (-4)

v5 stays (-1)

initial vector: (-1, 1, -1, 1, 1)

\* v1 is selected

1 \* (-1) + (-1) \* 1 + 1 \* 1 + 1 \* (-1) = (-2)

v1 stays (-1)

\* v2 is selected

(-1) \* (-1) + (-1) \* 1 + 1 \* 1 + 1 \* 3 = 4

v2 stays 1

\* v3 is selected

(-1) \* 1 + 1 \* 1 + 1 \* (-1) + 1 \* 1 = 0

v3 stays (-1)

\* v4 is selected

(-1) \* 1 + 1 \* 1 + (-1) \* (-1) + 1 \* 1 = 2

v4 stays 1

\* v5 is selected

(-1) \* (-1) + 1 \* 3 + (-1) \* 1 + 1 \* 1 = 4

v5 stays 1

**matrix is stable for all vectors!**

c. נתבונן בווקטור ונחשב את ערכו המתקבל על פי הרשת באיטרציות, עד שנקבל וקטור יציב:

initial vector: [1, -1, 1, 1, 1]

\* v1 is selected

(-1) \* (-1) + 1 \* 1 + 1 \* 1 + 1 \* (-1) = 2

v1 stays 1

\* v2 is selected

1 \* (-1) + 1 \* 1 + 1 \* 1 + 1 \* 3 = 4

v2 changes from (-1) to 1

\* v3 is selected

1 \* 1 + 1 \* 1 + 1 \* (-1) + 1 \* 1 = 2

v3 stays 1

\* v4 is selected

1 \* 1 + 1 \* 1 + 1 \* (-1) + 1 \* 1 = 2

v4 stays 1

\* v5 is selected

1 \* (-1) + 1 \* 3 + 1 \* 1 + 1 \* 1 = 4

v5 stays 1

[1, 1, 1, 1, 1]

initial vector: [1, 1, 1, 1, 1]

\* v1 is selected

1 \* (-1) + 1 \* 1 + 1 \* 1 + 1 \* (-1) = 0

v1 stays 1

**[1, 1, 1, 1, 1]**

קיבלנו את הווקטור המקורי.

d. נתבונן בחישוב עבור קואורדינטה יחידה i של הווקטור U על מנת להחליט האם לשנות את סימנו. נסמן את מצב הרשת S:

*כאשר הסימן לא משתנה אם*

*כעת נתבונן בחישוב זה עבור ההופכי ל-U לבין מצב הרשת S.*

*אם ניקח אחד מווקטורי U ממנו נבנתה הרשת, נקבל תמיד .*

*לכן אם ניקח את הווקטור ההופכי לו V, נקבל בהכרח , לפי (\*).*

*כלומר שהסימן של לא ישתנה לשום i כבר מהסיבוב הראשון, ולכן לא ישתנה.*

*מכיוון שלכל הווקטורים בסעיף מתקיים , אזי שלושתם לא ישתנו כתוצאה מפעולת הרשת.*

שאלה 2

*א. וידאו*

*ב. הוראות קימפול והרצה:*

1. *התקן פייתון 3.9+.*
2. *חלץ את קובץ ה-zip לתיקייה לבחירתך*
3. *התקן את הספריות matplotlib ו-NumPy – לשם כך ניתן להריץ את הפקודה:*
   1. *pip install -r requirements.txt*
4. *הרץ את הקוד בעזרת הפקודה:*
   1. *python main.py*

ג. 0. **מציאת ועיבוד הדאטה:** קובץ הדאטה: <https://www.conwaylife.com/patterns/all.zip>.

\* הקובץ מכיל מעל 2000 קונפיגורציות של משחק החיים בקבצי RLE. בעזרת rle\_decoder שכתבתי, המרתי כל קובץ למטריצה. לאחר מכן "גזרתי" את האפסים במסגרת המטריצה. סיננתי כל מטריצה שגודלה מעל 32\*32 שכן היא גדולה מדי עבור הרשת (היה עליי לבחור סף כלשהו). נותרתי עם 160 קבצי oscillators ו-80 non\_oscillators.

בעת טעינת הסט עבור אימון ובדיקה, טענתי כל אחד מהקבצים המסוננים בצורה דומה. את המטריצה שהתקבלה לכל קובץ ריפדתי ב-0 חזרה לגודל של 32\*32 ואז שיטחתי אותה לווקטור בגודל 1 על 1024, שיהווה קלט לרשת, לכן רוחב השכבה הראשונה, של הקלט, הוא 1024 נוירונים.

בנוסף , לכל מטריצה יצרתי 4 אוגמנטציות של סיבוב ב-90 מעלות, שכן זהו גם כן oscillator, וסיננתי אוגמנטציות שחזרו על עצמן. נותרתי עם 552 oscillators ועוד 318 non-oscillators.

את אוסף המידע חילקתי ל-train set, test set באופן אקראי, כאשר 70% מהדוגמאות הופנו ל-train set ו-30% ל-test set.

1. הפרמטרים בהם בחרתי לבסוף:

**ארכיטקטורת הרשת:**\* עומק – 4 שכבות:  
 - שכבת קלט ברוחב 32\*32=1024 נוירונים.  
 - שכבה חבויה ברוחב 512 נוירונים.  
 - שכבה חבויה ברוחב 128 נוירונים.   
 - שכבת פלט ברוחב 1 נוירון.  
\* קישוריות: רשת CNN  
\* פונקציית אקטיבציה: סיגמויד

2. **פרמטרי אופטימיזציה**:

\* פונקציית השגיאה: L2-norm  
\* גודל batch: שליש מהדאטה   
\* קצב הלמידה: 0.0001

3. גרף השגיאה על סט האימון ועל סט הבדיקה:

**גרף סופי:**

Chart, line chart

Description automatically generated

**הפרמטרים הנ"ל נבחרו בעקבות ניסוי ותהייה ובעזרת שיטוט באינטרנט ואינטואיציה מתמטית. כל ניסוי היה בלתי תלוי ונוסה עם אפוקים של full batch. הניסויים השונים כללו בין היתר:**

**\*** אוגמנטציות – ביצעתי ניסויים שונים ללא ביצוע אוגמנטציות ועם אוגמנטציות (של סיבוב הקונפיגורציה ההתחלתית ב-90, 180, 270 מעלות). בהרצה עם אוגמנטציות, זמן ההתייצבות היה ארוך משמעותית (סביב 9000 אפוקים לעומת קרוב ל-3000) וכל אפוק דרש זמן רב יותר, שכן כמות הדגימות הוכפלה כמעט פי 4. ללא אוגמנטציות הגעתי ל-overfitting ברור יחסית, בעוד שעם אוגמנטציות הגעתי לאחר סדר גודל של פי 3 אפוקים למצב יציב יחסית אבל עדיין לא מובהק כ-overfitting – השגיאה על סט האימון התייצבה, והשגיאה על סט הבדיקה עלתה מעט מאוד לאורך האפוקים הבאים. בשל מגבלות חומרה, לא היה באפשרותי להריץ עוד אפוקים.

בגרף הסופי לעיל נראה אימון ובדיקת הרשת עם אוגמנטציות. למטה בסעיף על batches, ניתן לראות את שינון הרשת על הדוגמאות בהרצה על full batch. להלן גרף האימון והבדיקה ללא אוגמנטציות לשם השוואה:

גרף 2:

\* חלוקה בין סט האימון והבדיקה – תחילה החלוקה הייתה 40%-60% לטובת סט האימון, אך לאחר ניסויים הגעתי לתוצאות טובות יותר עם חלוקה של 30%-70%, כנראה בשל כמות הדאטה.

\* שימוש בפונקציית השגיאה L1-norm – ציפיתי ששימוש ב-L1 לחישוב השגיאה יביא לתוצאות טובות יותר, שכן הוא נותן יותר דגש לערכים קטנים של שינוי מאשר L2, כלומר מאפשר מרחב פתרונות יותר רחב. להפתעתי התוצאות היו כמעט זהות, בעוד שתהליך האימון ב-L2 היה מעט מהיר יותר, שכן הוא מבצע נרמול מהיר של ערכים גדולים, וההבדל בין המטריצות מכל סוג לא גדול מספיק.

\* שימוש בפונקציית השגיאה L1-norm – ציפיתי ששימוש ב-L1 לחישוב השגיאה יביא לתוצאות טובות יותר, שכן הוא נותן יותר דגש לערכים קטנים של שינוי מאשר L2, כלומר מאפשר מרחב פתרונות יותר רחב. להפתעתי התוצאות היו כמעט זהות, בעוד שתהליך האימון ב-L2 היה מעט מהיר יותר, שכן הוא מבצע נרמול מהיר של ערכים גדולים, וההבדל בין המטריצות מכל סוג לא גדול מספיק.

\* גודל batch – ניסיתי לאמן את הרשת עבור בצ'ים של 10%, 30%, 50% מהדאטה. אימון על בצ'ים של 33% מהדאטה הביא לתוצאות הטובות ביותר, אם כי בהבדל מזערי. גם מספר האפוקים הנדרש היה דומה. להערכתי זוהי תוצאה של כמות הדאטה הקטנה עמה עבדתי.

גרף 3 – משמאל Full batch, 50%, 0%.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedFull batch


\* פונקצית אקטיביציה נוספת – tanh -

Chart, histogram

Description automatically generatedChart, line chart

Description automatically generated Chart

Description automatically generated

4. **Overfitting –** ללא אוגמנטציות, הרשת הגיעה למצב של overfitting לאחר כ-6000 אפוקים, כפי שניתן לראות בגרף 2. גם עם אוגמנטציות, נראה כי לאחר כ-4000 אפוקים האימון "זניח", שכן השגיאה על סט האימון יציבה – לא עולה ולא יורדת כמעט, שכן הרשת כבר שיננה את הסט. עם זאת, השגיאה על סט הבדיקה לא ירדה עוד, כנראה בשל הגיוון של הדאטה – שמנע הגעה מהירה ל-overfitting.

גרף 4:

Chart, line chart

Description automatically generated

5. **מקרים חריגים** – דוגמאות למקרים חריגים

דוגמאות לקונפיגורציות התחלתיות בעייתיות (או גיפים):

6. **תוצאות הרצה עם פונקציות אקטיבציה בינאריות** –

7. **הסבר ל-6:**