

## מבוא לבינה מלאכותית

# פרויקט Simulated Annealing Hundred-dollar, Hundred-digit Challenge

מגשים : תומר גרוסמן ואוראל סומך

בפרויקט התבקשנו להכין מערכת Simulated Annealing המוצאת את פתרון המינימלי האופטימלי הגלובלי לפנק' הבאה :

$$\exp(\sin(50x)) + \sin(60e^y) + \sin(70 \sin x) + \sin(\sin(80y)) - \sin(10(x + y)) + 1/4(x^2 + y^2)$$

פונקציה זו הינה אחת מעשר הבעיות שניתנו ב Hundred-dollar, Hundred-digit Challenge בשנת 2002 ע"י המתמטיקאי האמריקאי Nick Trefethen.

לשם הכנת מערכת ה-SA כתבנו קוד בשפת Python בגרסה 3.9 תוך שימוש בספריות וה-packages הבאים :

**Numpy** – ספרייה לביצוע חישובים מתמטיים.  
**Frigidum** – חבילה המאפשרת מימוש של אלגוריתם Simulated Annealing.  
**Random** – ספרייה לחישוב מספר רנדומלי.  
**Math** – ספרייה לפעולות אריתמטיות מתמטיות ופונקציות גיאומטריות כגון  $\sin$ ,  $\cos$ ,  $\exp$ ,  $\text{power}$ .

הפרויקט הוכן והורץ לבדיקה ועבד תקין ב-IDE הבאים :

- Visual Studio Code 2019

- REPL

---

**חשוב:** יש לוודא כי החבילות Frigidum ו-Numpy מותקנות לפני הרצה \*\*

במידה והחבילות לא מותקנות, יש להתקין אותם ידנית ע"י הרצת פקודה (לדוגמה) :

- pip install frigidum

- pip install numpy

או לחילופין להיכנס ל-package manager של ה-IDE ולהתקין ידנית.

---

## מבנה הקוד :

הקוד נכתב באופן הבא :

- (1) הגדרת פונקציות הנחוצות למערכת.
- (2) משתנה למתודת Simulated Annealing.

הקוד מתבסס בעיקרו על השימוש במתודה frigidum.sa מתוך החבילה frigidum.

המתודה מורכבת ממספר פרמטרים ופונקציות :

- ערך התחלתי רנדומלי בעל שיעורי  $X$  ו- $Y$ . טווח החיפוש ב- $X$  וב- $Y$  הינו בין 0 ל-1.
- פונקציה אובייקטיבית – הפונקציה המתמטית שבה אנו רוצים למצוא את הערך המינימלי הגלובלי.
- פונקציות שכנות – רשימה של פונקציות הנבחרות באופן רנדומלי כאשר כל 'שכן' מייצג פעולה אחרת. בפרויקט שלנו הפונקציות מוגדרות כסוג התנועה שתבוצע :
  - תנועה גדולה – הערך המתקבל לפונקציה ייבדק האם הוא קטן מ-0.5. אם כן, ערך ה- $X$  יישאר ללא שינוי וערך ה- $Y$  יגדל פי 0.5. אחרת, ערך ה- $X$  יגדל וערך ה- $Y$  יישאר ללא שינוי.
  - תנועה קטנה – פועל בדומה לאופן הפעולה של תנועה גדולה, רק עם שינוי גדילה פי 0.02.
- טמפרטורה התחלתית וטמפרטורת עצירה.
  - חישוב ועדכון הטמפרטורה כלולים בתוך המתודה ולכן אין צורך בהגדרתם.
- אלפא – מוגדרת כברירת מחדל וניתנת לשינוי. האלפא משפיעה על הנמכת הטמפרטורה.
- המתודה יודעת לחשב את מספר האיטרציות הדרושות לפעולת המערכת.
- המתודה מדפיסה גרף התקדמות ויזואלי למשתמש באמצעות תוסף tqdm 3.4.0.

## אופן פעולת המערכת:

המערכת בוחרת נקודה דו-ממדית רנדומלית בין 0-1 ומגדירה אותה כנקודת ההתחלה למצב הקיים. לאחר מכן נבחרת אחת התנועות שיבוצעו מהמצב הקיים באופן רנדומלי ותתבצע תנועה בציר ה- $X$  או בציר ה- $Y$ . המצב הקיים מתעדכן בהתאם לנקודה החדשה שאליה התבצעה התנועה האחרונה. במקביל הטמפרטורה מתעדכנת והנקודה הנמוכה ביותר נשמרת במערכת ומעדכנת בעת הגעה לנקודה חדשה נמוכה יותר. כמות האיטרציות מחושבת בעת הפעלת המערכת וכך המערכת חוזרת על הפעולות בהתאמה לכמות האיטרציות עד להגעה לטמפרטורת העצירה והפסקת פעילות המערכת.

ביצענו מספר הרצות על המערכת וראינו כי מערכת ה-SA בפרויקט זה מגיעה קרוב לפתרון הידוע הטוב ביותר : 3.306868647-.

להלן צילום מסך של הדפסת תוצאות מערכת ה-SA למשתמש בסיום ההרצה :

```
T: 0.096, M: 0.28, O_min: -3.3069, O_current: -3.2950: 100%| 66/66 [00:35<00:00, 1.86cooling/s]---
Neighbour Statistics:
(proportion of proposals which got accepted *and* changed the objective function)
  random_little_move      : 0.867231
  random_large_move       : 0.613549
---
(Local) Minimum Objective Value Found:
-3.30686531
Press any key to continue . . .
```

### סיכום:

במהלך העבודה על הפרויקט חקרנו את נושא האלגוריתם Simulated Annealing והעמקנו את ידיעותינו בנושא זה ע"י התמודדות עם בעיה מתמטית מורכבת ומאתגרת וכך ראינו כי באמצעות האלגוריתמים אותם אנו לומדים בקורס 'מבוא לבינה מלאכותית' ניתן לפתור בעיות מתמטיות מורכבות שייתכן ובלעדיהם ההגעה לפתרון הטוב ביותר הייתה קשה יותר או על סף הבלתי אפשרית ועצם הבנה זו מחדדת את חשיבות הלמידה והתרגול של אותם כלים היכולים לשמש אותנו במהלך הלימודים ואף מעבר לכך.

### קרדיטים:

Frigidum Author: Willem Hendrik

Numpy Author: Travis Oliphant

Challenge Author: Nick Trefethen