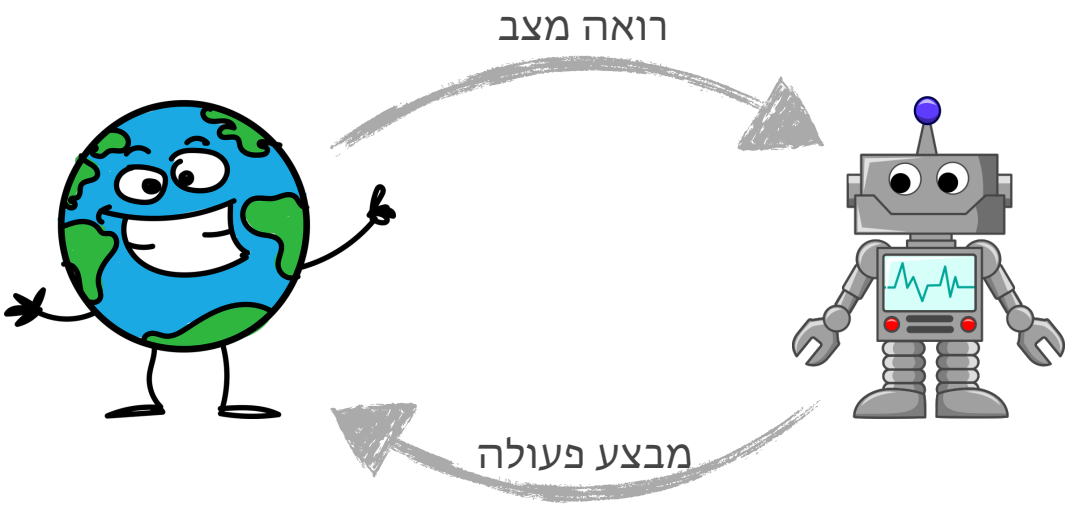
אסטרטגיה אבולוציונית
(Evolution Strategies)

הקדמה

אבולוציה ביולוגית היא תהליך כ״כ מוצלח עד שגרם להתפתחות של יצורים חד תאיים לכדי יצורים מורכבים כמו בני אדם. מה יקרה אם נשתמש באבולוציה על מנת לבנות בינה מלאכותית? זאת השאלה שהובילה להתפתחות התחום במדעי המחשב שנקרא אלגוריתמים אבולוציוניים, זוהי משפחה של אלגוריתמי בינה מלאכותית המבוססים על רעיונות השאובים מאבולוציה בטבע על מנת לפתור בעיות.

מטרה

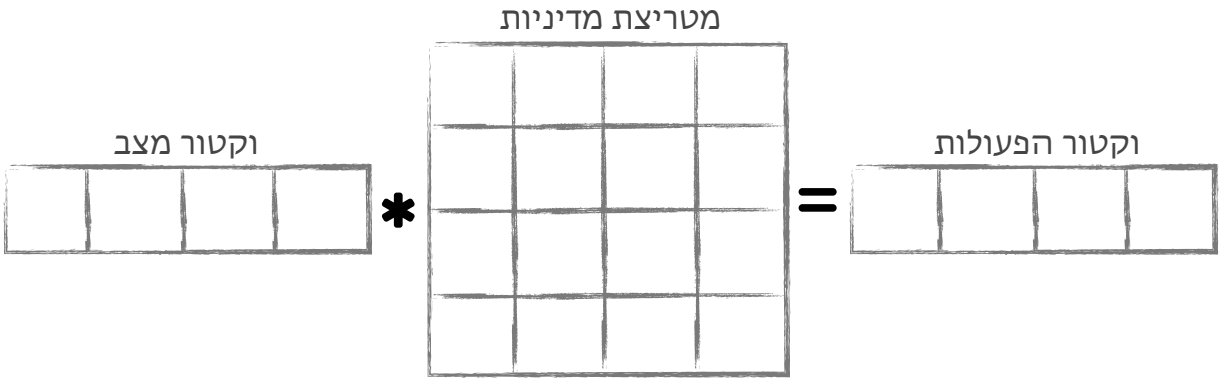
המטרה של אסטרטגיה אבולוציונית דומה לזו של למידה חיזוקית (Reinforcement leraning) ובאופן מדויק יותר לזו של Q-Learning. המטרה היא, בהינתן עולם וסוכן למצוא את המדיניות (Policy) שעל פיה הסוכן צריך לפעול כדי להצליח בצורה הטובה ביותר לעמוד ביעדים שניתנו לו. הסוכן יכול להיות למשל שחקן במשחק מחשב, רובוט שמנקה את הבית או טייס של מטוס קרב והיעדים יכולים להיות לנצח במשחק בציון גבוהה ככל הניתן, לנקות את הבית כמה שיותר מהר בלי לפספס אף חלק או להפיל כמה שיותר מטוסי אוייב. כדי להבין את הכוונה ב״מדיניות״ אפשר לחשוב על מודל כזה:



המדיניות היא בעצם הדרך שבה הסוכן יקבל החלטה לבצע פעולה כלשהי בהינתן מצב מסויים.

אז איך זה עובד?

מכיוון שאנחנו מדברים על מדעי המחשב אי אפשר בלי קצת וקטורים ומטריצות. אם נתאר את המצב של העולם שלנו כוקטור ואת המדיניות בתור מטריצה אז תוצאת המכפלה בניהם תהיה וקטור של הסתברויות עבור כל אחת מהפעולות האפשריות.



כמובן שזוהי הפשטה של הרעיון אבל נוכל להשתמש בה כדי להבין את דרך הפעולה של האסטרטגיה האבולוציונית.

האלגוריתם מבוסס ביסודו על הרעיון של מוטציה וברירה טבעית, כך שבתהליך איטרטיבי (בלולאה) ניצור מוטציות ונבחר את הטובות ביותר. בכל שלב נייצר אוכלוסיה, האוכלוסיה תהיה מורכבת מקבוצה גדולה של עותקים של המדיניות כאשר לכל אחד מהם נבצע מוטציה אקראית. ניתן לכל צאצא באוכלוסיה לפעול בעולם ולאחר מכן נתבונן בביצועים שלו. נאסוף את הצאצאים שמביאים את התוצאות הטובות ביותר במה שנקרא מטא-אוכלוסיה ונבצע איחוד שלהם כך שנקבל נקודת מוצא חדשה או "אב" עבור האוכלוסיה שבשלב הבא בלולאה, כך נחזור שוב ושוב עד שנקבל מדיניות טובה.

קצת יותר פרטים

עכשיו כשהבנו באופן בסיסי איך עובד האלגוריתם נתאר אותו בצורה קצת יותר מתמטית (אם לא בא לכם אפשר לקפוץ ישר לחלק הבא). נסמן ב-θ את המדיניות, F תהיה הפונקציה שמעריכה את ביצועי המדיניות על העולם, λ מייצגת את גודל האוכלוסיה ו-μ מייצגת את גודל המטא-אוכלוסיה. נאתחל את המדיניות בערכים רנדומליים ובכל שלב המוטציה

תהיה הוספת דגימה אקראית מהסתברות יוניפורמית כפול גודל צעד הלימוד שנסמנו σ ניתן לחשוב על σ בתור ה-learn rate. על מנת לחשב את מדיניות הבסיס החדשה עבור האיטרציה הבאה נבצע ממוצע משוכלל של μ הצאצאים הטובים ביותר, נכפיל אותו בצעד הלימוד σ ונחבר זאת למדיניות הבסיס של הצעד הנוכחי.

כעת יש בידינו את כל החלקים הדרושים ע"מ להבין את האלגוריתם המלא:

Algorithm 1 CanonicalES Algorithm

Input:
σ - Mutation step size
θ_0 - Initial Policy parameters
F - Policy evaluation function
λ - Offspring population size
μ - Parent population size
Initialize:

$$w_i = \frac{\log(\mu + 0.5) - \log(i)}{\sum_{j=1}^{\mu} \log(\mu + 0.5) - \log(j)}$$

for $t \in \{0, 1, \dots\}$ **do**
  for $i \in \{1, \dots, \lambda\}$ **do**
    Sample noise: $\epsilon_i \sim \mathcal{N}(0, I)$
    Evaluate score in the game: $s_i \leftarrow F(\theta_t + \sigma * \epsilon_i)$
  Sort$(\epsilon_1, \dots, \epsilon_\lambda)$ according to s (ϵ_i with best s_i first)
  Update policy: $\theta_{t+1} \leftarrow \theta_t + \sigma * \sum_{j=1}^{\mu} w_j * \epsilon_j$
  Optionally update step size σ

למה זה טוב?

לאסטרטגיה אבולוציונית יש 2 יתרונות בולטים על פני שיטות קלאסיות של למידה חיזוקית. 1- האלגוריתם אינו מוטה כמעט בכלל להשפעות המתכנת כך שיש לו חופש גדול יותר להתפתח. 2- האלגוריתם ניתן להרצה בצורה מקבילה באופן פשוט מאוד ולכן חוסך זמן רב בלימוד בהינתן כוח חישוב רב מספיק. למען האמת בכל שלב בלולאה אין קשר בין הצאצאים בכלל ולכן כל אחד מהם יכול לפעול במקביל בצורה שאינה קשורה לאחרים ורק בסיום הזמן המוקצה לשלב בלולאה יאוחדו התוצאות שלהם ויווצרו צאצאים חדשים. לכן הגורם היחד שמשפיע על גודל האוכלוסיה הוא כמות המשאבים.

דוגמא לביצועי האלגוריתם ניתן לראות בוידאו הבא:

https://youtu.be/0wDzPBiURSI?t=372

בוידאו רואים שחקן של משחק המחשב Qbert שאומן למשך 5 שעות בלבד בשימוש באוכלוסיה בגודל 400 ופיתח שיטת משחק מעניינת מאוד. ראשית ניתן לראות ב 6:24 שהשחקן קופץ ממקום למקום ומחכה שהיריב יזוז לכיוונו כדי שיוכל לנצח במשחק כלומר

השחקן למד "להטעות" את היריב. ולאחר מכן ב 6:27 ניתן לראות באג שהשחקן גילה במשחק שלא היה ידוע עד כה שמאפשר לשחקן לקבל אינסוף נקודות במשחק.

מידע נוסף

חלק מהכתבה מבוססת על המאמר שניתן למצוא בלינק כמו כן ניתן למצוא מימוש שלי הכולל הרחבה לאלגוריתם זה בפרויקט שלי בגיטהאב.