לימוד מדיניות משחק לרובוט במטלב

שם הסטודנטים: אושר אזולאי 203099247

ליהי קלקודה 305277071

1. תקציר

במסגרת הקורס "מערכות אוטומציה נבונות", למדנו על שיטות שונות ללימוד רובוט באמצעות חיזוקים (Reinforcement learning). בתחום זה, הבסיס הינו ללמוד מתוך ניסיון והמטרה הינה למפות סיטואציות בהתאם למצב הסביבה כאשר נותנים למערכת ללמוד באמצעות פרסים קטנים. הפרסים הינם ביחס להחלטות שנקבעות מראש.

כחלק מהמשימה השנייה בפרויקט, נדרשנו ליישם אלגוריתם לימוד לרובוט הנדרש לפגוע בכדור במגרש. קיימות מספר שיטות עבור לימוד הרובוט ובחרנו להשתמש בשיטת Q-learning שהינה שיטה המתעסקת בשערוך בשילוב אפסילון גרידי. הלמידה הינה מתוך ניסוי וטעיה בין סוכן (הרובוט הלומד) לבין סביבתו. מתבצעות מספר אינטרקציות שבהן הסוכן מקבל מידע מסביבתו לגבי המצב במגרש ובוחר פעולה אפשרית. בתגובה לפעולה מתקבל ערך מספרי (reward) המבטא את הניקוד לפעולה שבוצעה וכך התוכנה לומדת מהי הפעולות המועדפת למצב. כחלק מהיישום, בנינו גרפיקה למגרש המשחק, הכנסנו פיזיקה ויושם האלגוריתם. הסיבה שבחרנו באלגוריתם Q-learning הינה שהוא מאפשר לקבל את הפעולה האופטימלית עבור המשחק. גם בעבודה זו בחרנו להשתמש בתוכנת המטלב מאחר וקיימות עבורה פונקציות נוחות המאפשרות את יישום האלגוריתם. ביצענו ניסויים והרצות רבות לבדיקת האלגוריתם על מנת לבדוק את אחוזי ההצלחה לפגיעת הרובוט בכדור.

2. רקע ספרותי

RL כפי שציינו הינה קטגוריה של שיטות לימוד באמצעות חיזוקים. בשיטה זו, נותנים למערכת פרסים קטנים על מנת ללמד אותה מהי הדרך פעולה הרצויה בהתאם למצב הסביבה. הלימוד מתבצע באמצעות אינטרקציה בין סוכן לסביבה כאשר הסוכן הוא החלק שעליו יש שליטה מוחלטת. מתבצע מיפוי לסיטואציות עבור הפעולות לפי ה-Reward שיתקבלו עבורם, בדרך זו התוכנה "לומדת" איזו פעולה מפיקה את הפרס הגבוה ביותר.

קיימות מספר שיטות לפתרון RL ובהן Temporal difference. שיטה זו מתעסקת בשערוך ומאפשרת לחזות מטריצה Q (המכילה מצבים ופעולות) שתלויה בערכים עתידיים של הפעולה הנוכחית. קיימים מספר אלגוריתמים למימוש ובהם SARSA ו-Q-learning. ההבדל המרכזי ביניהם הינו ש-SARSA הוא אלגוריתם on-policy, כלומר, מתבצע עדכון למדיניות ממנה

מגיע הערך לפי הפעולה הנוכחית ו- Q-learning הוא off-policy, כלומר, עדכון המדיניות מתבצע מפעולות רנדומליות שונות. למעשה, עבור SARSA יתקבל הדרך לפתרון הקצר ביותר בעוד עבור Q-learning יתקבל הדרך לפתרון היעיל ביותר. בתרגיל שלנו, החלטנו לנסות לקבל עבור הרובוט את הפתרון האפקטיבי ביותר ולכן בחרנו ליישם את אלגוריתם Q-learning במשחק שבנינו. הסיבה לכך הינה שרצינו לבחון את ההשפעות של מציאת מהלך משחק אופטימלי לעומת משחק בטוח וזאת באמצעות מספר מקסימלי של איטרציות ללא כישלון. שילבנו את האלגוריתם עם אפסילון גרידי שהינה מדיניות משחק לבחירת פעולה רנדומלית. ככל שהאפסילון הנבחר נמוך יותר, כך ההסתברות שהרובוט יפעל באופן רנדומלי קטנה יותר ולהפך. לפיכך, במהלך הלמידה נשתמש באפסילון גבוה כדי שהרובוט יבצע הרבה פעולות אקראיות וכך ימפה את המספר הרב ביותר של המצבים והפעולות.

3. שיטה

נרחיב כעת על הפונקציה אותה בנינו לפי שלבים. נציין כי לצורך הנוחות בנינו קובץ הרצה אחד ובתוכו חילקנו את הפעולות מספר פונקציות:

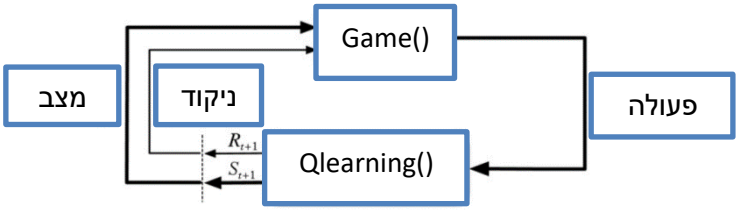
1. Pingpong – הפונקציה הראשית. מגדירה את כל הפרמטרים הגלובליים, מאתחלת מסך, תוחמת גבולות מגרש ומציירת ישויות של רובוט וכדור במשחק. מפעילה את פונקצית qlearning.

2. Qlearning – מכילה כמה פונקציות ומפעילה את האלגוריתם המרכזי לשיטה:
   - אתחול מטריצה Q(state,action)
   - עבור כל אפיזודה:
     - אתחל s (באמצעות פונקציות getBallPos, getRobotPos, getVelocity)
     - עבור כל צעד באפיזודה:
       - בחר action לפי Q,π
       - בצע action וקבל reward (פונקציות DoAction, Game)
       - $Q(s_t,a_t)=Q(s_t,a_t)+\alpha[r+\gamma r_{max_a}Q(s',a')- Q(s_t,a_t)]$
       - s=s'

3. getBallPos – מעדכנת את מיקום הכדור בחלוקה במגרש (איור 3).
4. getRobotPos – מעדכנת את מיקום הרובוט במגרש, 0 למעלה,1 באמצע, 2 למטה.
5. getVelocity – מעדכנת את גודל מהירות הכדור ואת מצב זווית ההתקדמות.

6. DoAction – מקבלת את ה- action שנדרש לעשות ומבצעת אותו ואת Game.
7. Game – מעדכנת ה-Reward (score) בהתאם לתנאי המשחק ואת התנהגות הכדור (שינוי גודל וכיוון למהירות) ומדפיסה למסך. במידה והמשחק נכשל מפעילה Restart.
8. Restart – מבצעת איתחול מחדש למשחק כדי להתחיל אפיזודה חדשה.

ביישום שביצענו, פונקציית Game מייצגת את הסוכן ו-Qlearning את הסביבה.



איור 1: תרשים סוכן-סביבה ב-RL עבור היישום במערכת

על מנת להקנות מספר מצבים סופי שיאפשר להגדיר את המצב, ביצענו תיאור למצבים הקיימים ומספרם:

Position Robot	Velocity angle	Velocity abs	Ball position Y	Ball position X
0-2	0-7	0-3	0-8	0-5

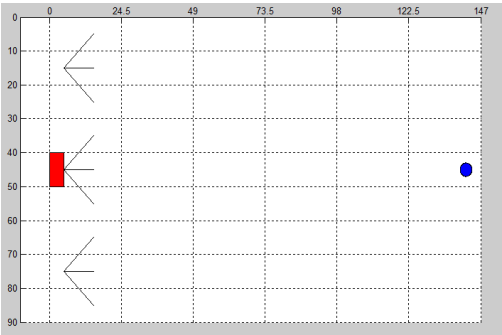
איור 2: כלל המצבים האפשריים עבור המערכת

4. ניסוי

על מנת לבדוק את האלגוריתם שכתבנו, ביצענו מספר בדיקות במהלך פיתוחו:

בדיקת הניקוד עבור האלגוריתם – נעשו מספר ניסויים לבחינת הניקוד הנדרש על מנת לאפשר לימוד מיטבי לרובוט. בשלב הראשון, ניסינו לתת reward זהה לפגיעה בכדור (הצלחה) ולפספוס (כישלון). ראינו לאחר הרצות רבות שהתנאים אינם מספקים כי אחוזי ההצלחה היו נמוכים מאוד (באזור ה-15% גם לאחר הרצות רבות, לפחות 10,000 משחקונים ובדיקת גרידיות עבור אפסילון של 0). בעקבות כך, החלטנו לתת ניקוד חמור יותר עבור מצב של כישלון. לאחר הרצות רבות, צפינו שיפור כבר מההרצות הראשונות עבור אחוזי ההצלחה. עבור הבדיקה הגדרנו לכדור מהירות ומיקום התחלתי זהים כדי לאפשר פעולה אחידה ולקצר את הבדיקה הראשונית. חילצנו פרמטר הסוכם את מספר הפגיעות והצגנו אותו בטבלה.

מספר הרצה	מספר הפגיעות מתוך 10000 צעדים:
הרצה 1	2489
הרצה 2	3143
הרצה 3	2876
הרצה 4	3255



איור 3: מימין טבלת הניסוי ומשמאל מצב המגרש בתחילת כל משחקון עבור הניסוי

נראה שיש שיפור בסטטיסטיקת הפגיעות מהניקוד הקודם לנוכחי (עלייה מכ-15% לכ-30% הצלחות עבור אותם התנאים). ניתן לראות שהלמידה בטבלה אינה אחידה, אך מאחר ואלה האפיזודות הראשונות לבניית מטריצת הלמידה, הלימוד איננו מיטבי בשלב זה. המצב ההתחלתי ללימוד עבור בדיקה זו נתון באיור 3. בנוסף, ביצענו הרצות רבות למתן הניקוד אך אין אפשרות להציג את כל תוכנם בדו"ח זה.

לאחר הרצות רבות כ-100,000 משחקונים, על מנת לבנות טבלה רחבה שתאפשר כיסוי למספר מצבים ופעולות גדול, ביצענו הרצה למשחק עם מדיניות גרידית (אפסילון = 0) כדי להקטין את הרנדומליות בבחירת המצבים ולאפשר משחק טוב המתבסס על הלמידה. בנוסף, הוכנסו מהירויות שונות לכדור, רעש ומיקומי התחלה שונים. אחוזי הפגיעה לרובוט עלו לכ-60%.

5. דיון ומסקנות

כפי שהצגנו בפרק הניסוי, בוצעו הרצות רבות עבור מקרים שונים ולכן עבור כל שינוי ובדיקה נדרש זמן ריצה רב של התוכנה על מנת לתחקר את המימוש. ראינו כי ניתן להמשיך תמיד ללמד את הרובוט על מנת לשפר את מטריצת הניקוד וכי יש אופציות רבות למתן הניקוד. בנוסף, לאחר מחקר ספרותי למדנו כי אין דרך ספציפית לתת ניקוד טוב או רע ללמידה ושניתן לקבוע את השיטה מניסויים בלבד. ישנה חשיבות רבה לאופן מתן הניקוד ואופן למידת התוכנה מאחר ואין אפשרות להקביל את החלטות התוכנה להחלטות האדם. קיימים מצבים בהם ניקוד על פעולה שנראה לנו הגיוני עלול להיות לא מתאים ללימדת התנהגות הרובוט.

כחלק מבדיקות האלגוריתם, גילינו תקלות כאשר הלמידה לא השתפרה מעבר לאזור ה-60%. לצורך בחינת השיפור ניתן להשתמש בעתיד באלגוריתם למידה מתקדם יותר כגון אלגוריתם המתבסס על מספר רב של צעדי זמן קדימה ולא רק על הצעד האחד הבא ומתן משקלים שונים לניקוד מכל צעד זמן.

בנוסף, במהלך כתיבת הדו"ח, עלו לנו תהיות להבדלים שהיינו מקבלים אילו יישמנו את אלגוריתם SARSA מאחר ודרך פעולתו שונה והוא מתבסס על הפעולה היותר בטוחה. בדיקה נוספת לשיפור הלימוד הינה בחינת האלגוריתם מול האחד שמומש במטרה לבדוק אם ניתן לקבל אחוז פגיעות גבוה יותר. נוסף על כך, ניתן היה לבצע בדיקה מעמיקה יותר לטיב אפקטיביות האלגוריתם אם היינו מגדילים את הרעש הפיזיקלי לפעולת הכדור במגרש, התייחסות לתוספת האנרגיה בעקבות המכה של הרובוט ומודל חיכוך לכדור. יש לציין כי במודל הנבחר, לא בוצע שיערוך למיקום הכדור בצעד זמן הבא ולכן אם ניתקל בהפרעה בזמן אמת עקב תלות במצלמה אז ניתן יהיה להוסיף את השערוך באמצעות אלגוריתמים שונים.