מערכות אוטומציה נבונות, תרגיל 2 – לימוד מדיניות משחק

מגישים: יובל גבאי ומרטין עבאדי

תקציר

כחלק ממטרה כללית ליצור אלגוריתם משחק בין רובוט לאדם ולאחר שביצענו עקיבה על הכדור, נרצה שהרובוט שלנו ילמד לשחק את המשחק. למידת המשחק נעשתה באמצעות סימולציה שמדמה את תנועת הכדור כפי שהמצלמה תופסת אותה, ובעזרת אלגוריתם למידה המבוסס על תגמולים שמטרתם לכוון את הרובוט להצלחה במשחק. פגיעה תוגדר כהצלחה ופספוס ככישלון והרובוט יתוגמל בהתאם. בכל מצב של הסימולציה הרובוט מעדכן את פונקציית הערך של הפעולות בהינתן מצב מגרש מסוים. השיטה שהשתמשנו לחישוב ערכי פונקציית הערך של מדיניות היא שיטת Sarsa ממשפחת ה-Temporal Differencing. לבסוף, הפלט בצורת טבלה שבה כל שורה היא מצב וכל טור הוא פעולה אפשרית, והיא תכיל את אומד לערך התועלת מביצוע הפעולה. בכך, יוכל הרובוט להחליט בהמשך איזה פעולה לבצע בזמן המשחק.

רקע ספרותי

<u>למידה בשיטת החיזוקים</u>

בלמידה בשיטת החיזוקים, אנו מקבלים תגמול מהמערכת בכל מעבר ממצב לאחר. המטרה של המערכת היא למצוא שיטה בו אנו נמקסם את ערך ההחזרה שלנו מהסביבה.

הערך מהפונקציה מתקבל מחיזוי של התשואה הזמינה מכל מצב [Rummery & Niranjan, 1994].

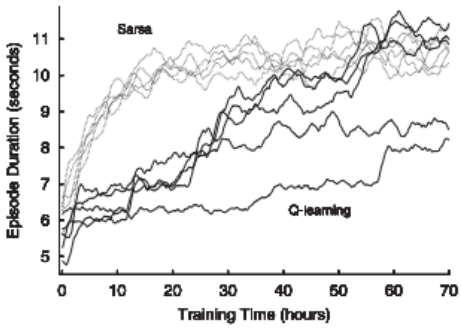


Figure 15 Learning curve comparison for Q-learning and Sarsa(λ).

Exploration – הוא מרכיב חשוב באלגוריתם. בעת בחירת הפעולה בעלת התשואה הגבוהה ביותר, האלגוריתם מאפשר מדי פעם, ובהתאם למדיניות שנקבעה, בחירת פעולה שתשואתה אינה הגבוהה ביותר. אם לא כך היה המצב, האלגוריתם תמיד היה ממשיך לבחור בפעולה שבחר עד כה ולא חוקר פתרונות אפשריים אחרים [Rummery & Niranjan, 1994].

במאמר נעשתה השוואה ללמידת תפקיד השוער בין שיטת sarsa לשיטת Q-learning. מהתרשים ניתן לראות כי לשיטת ה- Q-learning לוקח יותר זמן להתכנס מאשר ל-Sarsa, הדורש 40-50 שעות של זמן למידה לעומת 15-20 שעות. כמו כן, למדיניות הנלמדת מ- Q-learning יש שונות גבוהה יותר בביצועים לאורך כל הריצות. זה יכול לבטא את חוסר היציבות של Q-Learning בעת שימוש קירוב הפונקציה [Stone, Sutton, Kuhlmann (2005)].

הסברים על בחירת אלגוריתמים והרכיבים של הקוד

מבין האלגוריתמים שנלמדו בכיתה אנו בחרנו את אלגוריתם Sarsa . Sarsa הוא מסוג האלגוריתמים Temporal Differencing, אשר הוא שילוב של סימולציות מונטה קרלו ושל תכנות דינמי ומבצע בכל צעד עדכון של האמדים.

המדיניות שנבחרה היא מדיניות אפסילון גרידי, שבו הרובוט יבחר בצורה גרידית את הפעולה בעלת השווי הגבוה ביותר בהינתן למצב מסוים ב- (1-epsilon) מהמצבים. שיטה זו נבחרה מכיוון שהיא מביאה איזון בין exploitation ל-exploration (מה שמאפשר לתור אחרי מרחבים חדשים).

הקוד מכיל שלושה חלקים עיקריים שהן: 1. סימולציה 2. Sarsa 3. רובוט

1. <u>סימולציה</u>

הסימולציה מייצרת את הסביבה, המגרש בה המערכת מריצה את המשחק ואת הפרמטרים הדרושים לרובוט כדי שיוכל ללמוד לשחק. הסימולציה מאתחלת את כל הערכים והפרמטרים הנדרשים על מנת ליצור מבחר רב של אפיזודות עבור הרובוט. בכל סימולציה נמדל את תנועת הכדור על המגרש, ובדומה לקלט שאמור להיקלט מהמצלמה, נספק ערכי מהירות, אמפליטודה, מיקום X ו-Y על המגרש, והמרת המידע הרציף הזה לדיסקרטי. כל אפיזודה מתחילה בתנועת הכדור אל עבר הרובוט, ומסתיימת באחת מארבעת הסיבות הבאות: הכדור עבר את מיקום הרובוט, מהירות הכדור הגיעה ל-0, הכדור שינה כיוון והוא לא פונה לרובוט או שהרובוט הצליח לבעוט בכדור.

בתוך אפיזודה, מידלנו כך שהמידע שמגיע לרובוט הוא בהתאם למהירות הפריימים של המצלמה (25 פריימים בשניה), וכך dt הינו 0.04 שניות. בכל פעם שהרובוט יבחר פעולה שהיא אינה הישארות במקום, dt יגדל להיות 1. האפיזודה תריץ מצבים של התקדמות הכדור והשינויים שנעשו במהירויות ובזוויות של הכדור (מפורט בהמשך), לאחריו ייתן לרובוט לבחור את הפעולה שיבצע בהתאם ללמידה של הרובוט על גבי כל האפיזודות עד כה.

2. <u>Sarsa</u>

Sarsa היא מחלקה בה מתבצעת הלמידה של הרובוט. המחלקה מקבלת את הפרמטרים ללמידה, החלקה והיוון של פעולות (זאת על מנת שלא תהיה תנודתיות רבה בלמידה), וכמו כן יקבל את הפעולות השונות שהרובוט יכול לבצע. בעת הפעלת האלגוריתם, ה-Sarsa יעדכן את התועלת של בחירת הפעולה בהינתן מצב הסימולציה והתגמול שנבחר עבורו בעקבות תוצאות הפעולה שביצע. לאחריו, ייבחר האלגוריתם עבור הרובוט את הפעולה הבאה שמשתלם לו לבצע בהתאם לפרוטוקול בחירת פעולה שנקבע.

3. <u>הרובוט</u>

הרובוט מאתחל לעצמו מופע של Sarsa שיחשב עבורו את החישובים הנ"ל ויבצע עבורו את הלמידה. בנוסף, הרובוט מכיל את המיקום שלו, מחשב עבורו את התגמולים במידה והוא פגע או לא פגע בכדור והפעולה אותה ביצע, ומעדכן את ה-Sarsa שלו בהתאם.

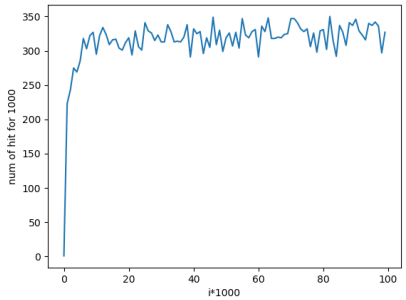
בדיקות הקוד

הבדיקות שנעשו על הקוד היה מבחינת תקינות הסימולציה, תקינות ביצוע הפעולות והתגמולים עבורם ותקינות תהליך הלמידה. תחילה הרצנו אפיזודה בודדת בכל פעם. בדקנו שהמהירות יורדת בהתאם לפונקציה הפיזיקלית של תאוצה, וכמו כן שהמיקום של הכדור בצירים מתעדכנת בהתאם למיקום נוכחי בתוספת המהירות כפונקציה של הזמן. ווידאנו שהכדור משתנה בהתאם ברגע שהוא פוגע בדפנות ומשנה את הזווית שלו.

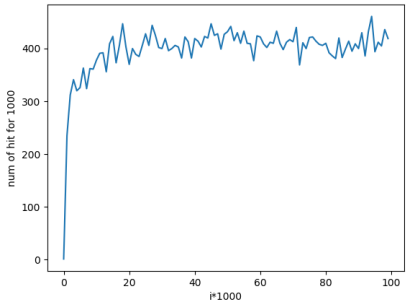
```
------------------------------------------------------------------------------ 0
Real X:  79.7394742925     Real Y:  72.6593804936    Cell X:  3  Cell Y:  7
 action= 3 Angle:  108.744893466  State Angel:  2  Velocity:  37.62   Velocity State:  1
 hit?:  0

------------------------------------------------------------------------------ 1
Real X:  67.6500966309     Real Y:  37.0347914046    Cell X:  2  Cell Y:  3
 action= 3 Angle:  108.353746718  State Angel:  2  Velocity:  37.2438   Velocity State:  1
 hit?:  0
```

הבדיקה הבאה הייתה בדיקה של המרה נכונה של נתוני הסימולציה להמרה למרחב הדיסקרטי שנקבע לרובוט. לבסוף בדקנו שהאפיזודה נגמרת כאשר התנאים שקבענו לסיומה אכן קורים. לאחריו בדקנו את התגמול בהתאם לפעולה שנבחרה והתוצאה שלה. לבסוף, הרצנו הרבה אפיזודות ובדקנו את שיפור הפגיעות של רובוט כל 1000 אפיזודות.

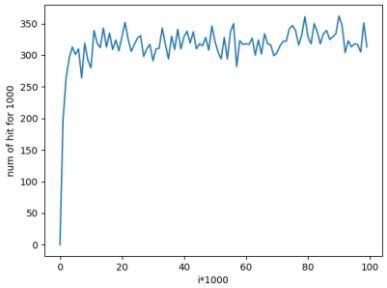
לאחר מכן בדקנו את תוצאות הלמידה בשינוי ערכי פרמטרים שונים של המודל (גאמה, אפסילון ואלפא).



איור 1 אפסילון 0.5



איור 2 גאמה 0.3



איור 3 אלפא 0.5

כאשר שינינו את אפסילון ואפלא ניתן לראות בבירור שהם פחות טובים ממה שבחרנו (ראו תוצאות). שינוי הגאמה גם הוא פחות טוב אם כי לא באופן משמעותי.

תיאור מבנה הקוד, הפלט והקלט

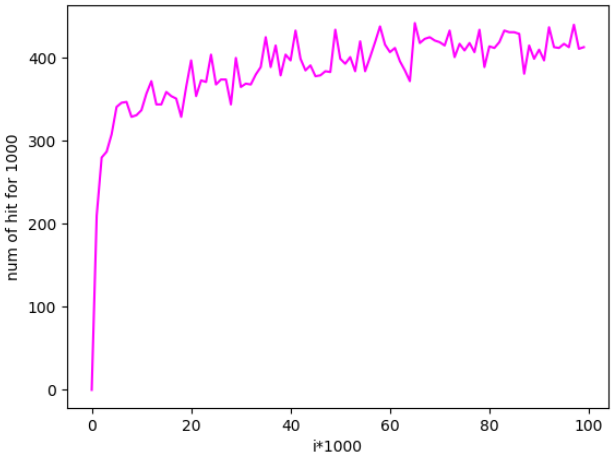
<u>תיאור הפסאודו קוד</u>:

1. אתחול משתנים ורובוט
2. לולאה של הרצת אפיזודות
   - 2.1. אתחול תנועה של כדור
   - 2.2. לולאה עד סוף האפיזודה: כל עוד לא הייתה בעיטה או הכדור עבר את הרובוט או שהכדור נעצר:
     - 2.2.1. החלטה על פעולה
     - 2.2.2. ביצוע הפעולה
     - 2.2.3. בדיקת מצב סביבה
     - 2.2.4. עדכון ערכים

3. שמירת מדיניות
4. העברת המידע לאקסל

הפרמטרים שבחרנו: $\alpha = 0.1;\ \gamma = 0.9;\ \varepsilon = 0.1$ בנוסף, בחרנו 3 ערכי תגמול: פגיעה בכדור 100, פספוס 10- , אחרת 0 . תחילה נתנו תגמול על פגיעה 10 בלבד והתוצאות פחות טובות.

דיון במשמעות ובתוצאות



ניתן ללמוד מהגרף שהרובוט אכן לומד לפגוע בכדור בעזרת התגמולים. כמות הפגיעות שלו עולה ככל שהאפיזודות מתקדמות. מידלנו אך ורק תנועת כדור לכיוון הרובוט ולא כשהכדור מתרחק ממנו, מכיוון שזה לא מצב שנותן מידע רלוונטי לרובוט ואנו מצפים שהוא יישאר במקום ולא יבחר לבצע פעולה. ניתן לראות כי עבור הפרמטרים שנבחרו נוצר תהליך למידה מגמתי שלאחר 100,000 הרצות התכנס ל42.5% הצלחות.

אנו נציע בהמשך להשוות את תהליך הלמידה לQ-learning, ולחפש פרמטרים אופטימליים (גאמה, אפסילון אלפא ותגמולים).

ביבליוגרפיה:

1. Rummery, G. A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical report CUED/FINFENG/TR 166, Cambridge University Engineering Department.
2. Peter Stone, Richard S. Sutton, Gregory Kuhlmann (2005). Reinforcement Learning for RoboCup Soccer Keepaway.

נספח

הוראות הפעלת קוד:

1. יש לפתוח את הפרויקט בפייטון 2.7
2. יש לייבא את החבילות הרשומות מעלה בקובץ Simulation.
3. יש להריץ את קובץ Simulation, שם נמצאת הפונקציה הראשית.

* אם רוצים לשנות את הפרמטרים של המודל, ניתן לשנותם מהקובץ Robot.