דו"ח פרויקט – Reinforcement learning of mini-poker agents:

מגישים:

איתי גיא, 305104184

דין שרעבי, 311138747

6. השוואה סטטיסטית בין ביצועי הסוכן החכם אל מול הסוכן האקראי (עבור total money=10):

a. כפי שניתן לראות בהדפסה המצורפת הסוכן החכם מצליח לבלוף את יריבו הסוכן האקראי ~1.108% מסך הסיבובים הכולל
כאשר הגדרנו "בלוף" בתור:
- האחוזון ה-K מלמטה של תוחלת כל המצבים ששונים רק ב-feature ה-pot שלהם
  - K זה פרמטר בקוד שנקבע לעת עתה להיות 0.10

פונקציה זו מתבצעת בכל סיום של סיבוב:

```
def updateIfBluff(self,state):
    if self._states.getFinalState() == States.WIN and self.action == Poker.ALLIN:
        idx = self._states.getLinearIndex(state)
        qtable = self._states.getQTable()
        stateRank = self._states.getExpectedAction(state)
        stateRank = stateRank[1]
        if stateRank < States.POOR_PERCENTILE*self._states.getTotalPercentile():
            self.bluffs += 1
```

b. כמות המשחקונים שהסוכן החכם מנצח לאורך כמות ה-epochs הינה ~57% [ישנם למידות שלאחריהם סיים את משחק ה-test עם 61% הצלחה] כאשר ניתן להבחין שכמות הסיבובים הכוללת שבה הסוכן החכם מנצח היא ~75%
כמות המשחקונים הממוצעת שבה הסוכן החכם מנצח היא ~1.327, כלומר שזה נע בין משחק ל-2 משחקים עד להבסת היריב

c. ממוצע העונשים שהסוכן החכם קיבל לאורך תקופת הלימוד הוא ~-0.369 , כאשר העונשים הם לא אחידים

```
^^^^^^^^^^^^^^^^^^^^^^^^ game epoch 2000 ^^^^^^^^^^^^^^^^^^^^^^^^
----------- round 1 -----------
player 1:  [J♣, J♦]
player 2:  [10♣, A♠]
player 2 makes allin
flop: [8♣, 2♣, A♥, Q♦, 2♦]
player 1 makes allin
QAgent (id:1) wins
QAgent(id:1) current round wins: 75.14999999999999 %
QAgent(id:1) current game wins: 56.599999999999994 %
RandomAgent(id:2) current round wins: 24.85 %
RandomAgent(id:2) current game wins: 43.4 %
gameover: [[1]]
mini AI poker game is finished.

Summery Results:
----------------
QAgent(id:1) total round wins: 75.14999999999999 %
QAgent(id:1) total game wins: 56.599999999999994 %
QAgent(id:1) average bluffs: 1.108153078202995
QAgent(id:1) average game wins: 1.3277385159010602
RandomAgent(id:2) total round wins: 24.85 %
RandomAgent(id:2) total game wins: 43.4 %
```

d. <u>מידול המרחב:</u>

1. <u>כל מצב הוא וקטור של חמישייה שמורכב מהצורה:</u>

- $[rank\_card1(0-12), rank\_card2(0-12), is\_same\_suit(0-1), pot\_amount(0-40), is\_big\_blind(0-1)]$
- סך הכל מידלנו **13*13*2*41*2 = 27716 מצבים**.
- מספר המצבים גדל בעיקר בגלל כמות הכסף שרצינו לקוונטז – ניתן להוריד לפחות מצבים ע"י קוונטיזציה חזקה יותר אבל זה לא הוביל לשינוי דרמטי בביצועים.

<u>כל וקטור כמו הנ"ל ממופה באופן יעיל לזוג מהצורה:</u>

- $[reward/penalty(allin), reward/penalty(fold)]$

כאשר כל איבר בוקטור זו מכיל את ערך ה-reward/penalty שהצטבר עד לצעד t במשחק

2. כל reward/penalty שניתן לסוכן הוא כפונקציה של זוג הקלפים שבהם אחז, הפעולה שביצע ותוצאת הסיבוב.

למשל, אם אחז [A,A] ועשה ALLIN וזכה יקבל 12.0 ואם הפסיד יקבל 3-

טבלת מידול ה-reward/penalty של הסוכן:

```
self._R_simple = {States.WIN : 1.0,States.LOSE : -1.0}
self._R_high = {States.WIN : 2.0,States.LOSE : -1.0}
self._R_twoHighs = {States.WIN : 3.0,States.LOSE : -1.0}
self._R_pair = {States.WIN : 6.0,States.LOSE : -2.0}
self._R_color = {States.WIN : 6.0,States.LOSE : -2.0}
self._R_highPair = {States.WIN : 12.0,States.LOSE : -3.0}
self._R_highColor = {States.WIN : 12.0,States.LOSE : -3.0}
```

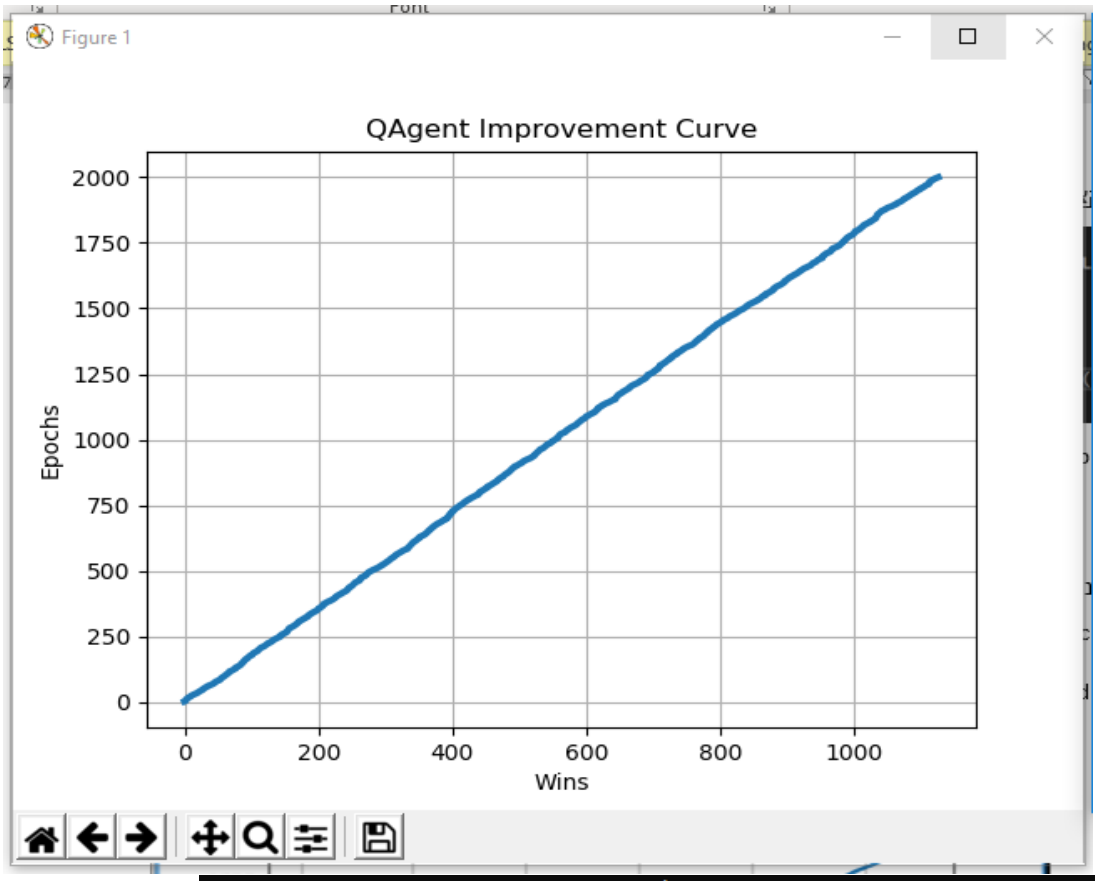
- בנוסף לכך, אם הסוכן זכה בכל המשחקון יקבל reward/penalty נוסף שמזכה אותו בהטבה עבור משחקון טוב בצורה של backtrack ל-rounds שהיו במשחק שהסתיים מתוך מחשבה שכנראה המהלך באופן גלובלי היה טוב:

```
weight = -20
if game.isGameover():
    if agent.getStatus() == States.WIN:
        agent.updateGameWins()
        agent.backtrackUpdateStates(True)
        weight = 100
    else:
        agent.updateGameLose()
        agent.backtrackUpdateStates(False)
```

3. הקלפים שמוגרלים לסוכן לאחר הלמידה – בזמן משחק ה-test – הינם מוגרלים מתוך המצבים שנתקל בהם בזמן הלמידה בכדי לייצר test אמין שמשקף את למידת הסוכן עבור מצבים וקלפים שכבר נתקל בהם (ללא טריק זה אחוז הביצועים של הסוכן יפגע וכתוצאה מכך אחוז ההצלחה שלו ירד ולא יהיה יציב כלל בין משחק למשחק)
4. הפעולה שהסוכן החכם יבחר לבצע תלויה בממוצע הגבוה ביותר בין הפעולה שיש לה ערך מצטבר גבוה יותר של המצב בטבלה לבין ממוצע המצבים האחרים ששונים מהמצב הזה בערך ה-pot בלבד (זה מבוצע מהסיבה שיכול להיות שהסוכן ראה כבר את המצב הזה אבל עם סכום כסף אחר ששם הוא לקח החלטה שתלוי גם בסכום הכסף וכעת יצרה לבצע פעולה אבל הוא לא נתקל במצב עם סכום הכסף הזה ולכן נראה להתנות את ההחלטה גם במצבים אחרים שכבר נלמדו ושונים רק בסכום הכסף הנוכחי)
5. המצב הבא שהסוכן משכלל בנוסחת העדכון הוא ה-blind ההפוך מה-blind שהוא שיחק בו כרגע ולזה מחושבת התוחלת שמתווספת ל-reward/penalty

e. <u>גרף שיפור ביצועים של הסוכן החכם לאורך נסיונות הלמידה:</u>

ניתן להבחין שישנה עליה לינארית יחסית חלקה בין תקופת הלימוד לבין כמות הנצחונות במשחקונים, כלומר ישנה הטיה של הסוכן החכם כלפי המצבים שבאמת כדאי לשחק בהם:

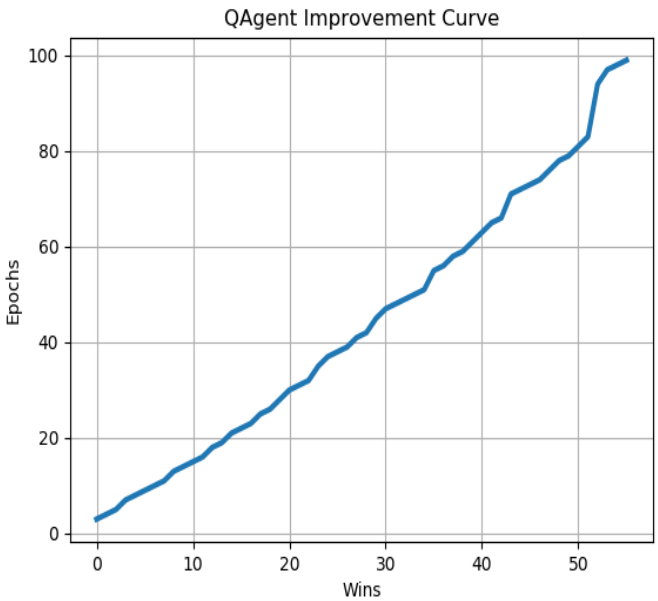



```
^^^^^^^^^^^^^^^^^^^^^^ game epoch 2000 ^^^^^^^^^^^^^^^^^^^^^^
----------- round 1 -----------
player 1:  [2♠, 6♥]
player 2:  [5♠, 10♣]
player 2 makes allin
flop: [8♠, 4♠, 6♥, 6♠, 5♠]
player 1 makes allin
QAgent (id:1) wins: reward/penalty(allin) = 11.8321516152333
QAgent(id:1) current round wins: 74.3 %
QAgent(id:1) current game wins: 56.3 %
RandomAgent(id:2) current round wins: 25.7 %
RandomAgent(id:2) current game wins: 43.7 %
gameover: [[0]]
model learning is finished.

Summery Results:
----------------
QAgent (id:1) average penalties: -0.3699861687413555
QAgent(id:1) total round wins: 74.3 %
QAgent (id:1) total game wins: 56.3 %
QAgent (id:0) saved all data.
QAgent plotting learning process graph..
RandomAgent(id:2) total round wins: 25.7 %
RandomAgent (id:2) total game wins: 43.7 %
```

f. <u>קבענו את הפרמטרים להיות:</u>

GAMA=0.01, EPSILON=0.1, ALPHA=0.2

```
^^^^^^^^^^^^^^^^^^^^^^ game epoch 100 ^^^^^^^^^^^^^^^^^^^^^^
----------- round 1 -----------
player 1:  [10♥, K♠]
player 2:  [2♦, 4♣]
player 2 makes allin
player 1 makes fold
RandomAgent (id:2) wins
----------- round 2 -----------
player 1:  [8♦, Q♦]
player 2:  [7♦, A♥]
player 1 makes allin
flop: [4♥, J♦, 2♥, 8♦, A♥]
player 2 makes allin
RandomAgent (id:2) wins
QAgent(id:1) current round wins: 78.0 %
QAgent(id:1) current game wins: 56.00000000000001 %
RandomAgent(id:2) current round wins: 22.0 %
RandomAgent(id:2) current game wins: 44.0 %
gameover: [[1], [1]]
model learning is finished.

Summery Results:
----------------
QAgent (id:1) average penalties: -0.34306569343065696
QAgent(id:1) total round wins: 78.0 %
QAgent (id:1) total game wins: 56.00000000000001 %
QAgent (id:0) saved all data.
QAgent plotting learning process graph..
```



ניתן להבחין בגרף הנ"ל שהוא שונה בצורתו מהגרף שתואר מקודם מהסיבה ששינינו את פרמטר קצב הלמידה ALPHA מ-0.01 ל-0.2 וכך תוך 100 epochs הצליח ללמוד את מה שהפרמטר הקודם למד ב-2000 epochs, בנוסף לכך שינינו את פרמטר הרנדומיזציה מ-0.01 ל-0.1 וזה מה שגרם לגרף להראות לא חלק כמו הקודם אבל כפי שכבר הזכרנו בזכות זה ראינו ולמדנו מצבים יותר מהר ובאופן יותר רחב – בנוסף ממוצע ה-penalty ירד מעט.

בנוסף לכך, ניתן להבחין כי הלמידה היתה אפקטיבית יותר כי בזמן המשחק קיבלנו את התוצאות הבאות על epochs 1000:

```
^^^^^^^^^^^^^^^^^^^^^^^^ game epoch 1000 ^^^^^^^^^^^^^^^^^^^^^^
----------- round 1 -----------
player 1:  [5♠, 7♥]
player 2:  [8♥, 9♣]
player 2 makes fold
RandomAgent (id:2) wins
----------- round 2 -----------
player 1:  [7♣, 8♦]
player 2:  [4♦, 10♣]
player 1 makes allin
flop: [K♥, J♥, 6♥, 6♣, Q♠]
player 2 makes allin
RandomAgent (id:2) wins
----------- round 3 -----------
player 1:  [3♣, K♠]
player 2:  [6♠, Q♣]
player 2 makes allin
flop: [6♦, 4♦, 4♠, Q♥, 8♠]
player 1 makes allin
QAgent (id:1) wins
----------- round 4 -----------
player 1:  [9♠, A♠]
player 2:  [9♦, A♥]
player 1 makes allin
flop: [3♦, Q♦, A♥, 6♦, 6♠]
player 2 makes allin
RandomAgent (id:2) wins
QAgent(id:1) current round wins: 76.1 %
QAgent(id:1) current game wins: 57.3 %
RandomAgent(id:2) current round wins: 23.9 %
RandomAgent(id:2) current game wins: 42.699999999999996 %
gameover: [[0], [1], [1, 0], [1]]
mini AI poker game is finished.

Summery Results:
----------------
QAgent(id:1) total round wins: 76.1 %
QAgent(id:1) total game wins: 57.3 %
QAgent(id:1) average bluffs: 1.1147540983606556
QAgent(id:1) average game wins: 1.3280977312390925
RandomAgent(id:2) total round wins: 23.9 %
RandomAgent(id:2) total game wins: 42.699999999999996 %
```

- שיפור משמעותי רק באמצעות הפרמטרים

רגישות של התוצאות:

- ניתן להבחין שכאשר GAMA גדול, למשל 0.2 הוא תורם ללמידת הסוכן רק לאורך זמן וכמות משחקים גדולה ובכמות משחקים קטנה הוא רק יגרע כי עדיין לא יתגבש על ערכים מספיק מייצגים של המצבים
  - למשל, עבור למידה של 100 מצבים הוא מנצח 49% עם GAMA=0.2 ואילו 62% עם GAMA=0.01

- עומת זאת ניתן להבחין שעבור 1000 מצבים עם ALPHA=0.7 נקבל הפחתה בביצועים לאורך זמן
  - למשל, עבור 100 מצבים הוא מנצח 61% ואילו ב-1000 מצבים הוא מנצח 55%

```
QAgent(id:1) total round wins: 68.0 %
QAgent (id:1) total game wins: 60.0 %
```

- לעומת זאת ניתן להבחין שעבור 1000 מצבים עם EPSILON=0.3 נקבל שהשינוי הוא לא רגיש כל כך לאורך זמן
  - למשל, עבור 100 מצבים הוא מנצח 56% ואילו ב-1000 מצבים הוא מנצח 53%

סיכום:

האלגוריתמיקה והתהליך כולה של למידת הסוכן היא מאוד משמעותית אבל כאשר יש לנו כבר מודל שניתן לאמן אותו באופן אידיאלי עובר חוט מאוד דק שצריך לחקור אותו בין סוכן טוב לסוכן לא טוב שתלוי בעיקר בפרמטרים ובכמות המשחקים שניתן לאמן אותו עליהם. אם סכום הכסף משתנה הסטטיסטיקה תראה טיפה אחרת אבל אחוזי ההצלחה עדיין שומרים על יציבותם כטובים עם בחירת פרמטרים בצורה נכונה. למשל עבור total money=5 נקבל ב-100 משחקים כ-65% הצלחה בעוד שגם כמות ה-penalty תרד משמעותית (בערך בחצי).