**מבוא לבינה מלאכותית – תרגיל בית 2 – פאקמן:**

1. **הכרות עם הקוד והמשחק:**
   1. שחקן בסיסי ReflexPlayer בוחר בכל צומת החלטה את הפעולה הבאה לביצוע ע"י בחינת המצבים הבאים אליהם כל פעולה אפשרית מובילה, ובחירת הפעולה המובילה למצב בעל הניקוד הגבוה ביותר. פונ' היוריסטיקה היא, אם כן, הניקוד המקס' של המצב המבין סט המצבים הבאים
2. **בניית סוכן משופר:**
   1. היוריסטיקה שלנו תהא מורכבת מקומבינציה לינארית של המאפיינים הבאים (הפרמטרים ניתנים לכוונון):
      * ניצחון/הפסד
      * מרחק מינימלי לאוכל
      * ניקוד
      * כמות האוכל במגרש
      * כמות הקפסולות במגרש
      * האם קפסולה
      * כמות הרוחות הרעות במגרש
   2. החישוב שלנו עבור היוריסטיקה מובא להלן: 
      * יש לשים לב למקדמים השליליים עבור חלק מהפרמטרים. הם קיימים על מנת לאפשר לוגיקה הפוכה; המצב נבחר כמקסימלי, לכן נרצה לתת ערך גבוה ככל האפשר עבור:
        1. כמות אוכל קטנה יותר – נבחר במצב בו אוכלים על פני מצב בו לא אוכלים.
        2. מרחק מינימלי לאוכל – נבחר להתקדם לכיוון האוכל הקרוב ביותר על פני כיוון האוכל הרחוק.
        3. במקרה בו המשבצת הסמוכה אלינו מכילה קפסולה, נבחר להתקדם אליה על פני משבצת בה אין קפסולה.
        4. נבחר להתקדם למשבצת בה מס' הרוחות הפעילות נמוך יותר, כלומר למשבצת בה ביצענו אכילה של רוח רעה, על פני משבצת בה לא ביצענו זאת.
      * ניתן היה לממש יוריסטיקה עוד יותר טובה באמצעות monkey hacking קטן לסוכן עצמו, אבל בחרנו שלא לעשות זאת. העובדה הזו נובעת מהאופן בו מומש הסימולטור, בייחוד במשבצות בהן גם אוכלים קפסולה, ומיד באותו צעד הרוחות משתנות לרוחות טובות.

הדבר גרם לנו להתנגשות של 2 יוריסטיקות מתחרות – השאיפה לאכול קפסולה, והשאיפה לנוע לכיוון רוחות טובות.

ניתן היה למנוע זאת באמצעות שמירת אינדיקטור על אכילת הקפסולה, כך שיופעל סעיף היוריסטיקה של תנועה לכיוון רוחות טובות, רק לאחר אכילת הקפסולה.

1. **בניית סוכן Min-Max**
   1. על ידי שימוש בעץ אסטרטגיה, אנו למעשה מניחים כי:
      * קבלת החלטות במשחק מתקבלת בתורות. פעם תור הסוכן ובפעם תור כל אחת מהרוחות. כלומר, אנו מניחים שרוח לא תקבל החלטה פעמיים בלי שהסוכן קיבל החלטה פעם אחת באמצע. הנחה זו לא בהכרח נכונה שכן קבלת ההחלטות מתקבלת אצל הסוכנים השונים במקביל.
      * השחקן היריב גם משחק בצורה אופטימלית.
2. דרך נוספת לחישוב ערך minimax היא להתייחס לכל הרוחות יחד כסוכן אחד. בדרך זו נשארת שכבת min אחת בעץ האסטרטגיה.
   * + חסרון שיטה זו בא לידי ביטוי במקדם הסיעוף. בעץ בעלת שכבתmin אחת לכל רוח, מרחב המצבים השכנים יכלול את המצבים הבאים לאחר פעולת הרוח הספציפית. בעץ בעל שכבת min אחת בלבד, מרחב המצבים השכנים יכלול את המצבים הבאים לאחר פעולה של כל אחת מהרוחות ולכן זהו מרחב מצבים גדול בצורה ניכרת, כלומר מקדם הסיעוף B גדול במצב זה משמעותית.

לפי משפט מהתרגול, סיבוכיות הזיכרון היא . אפילו שעומק העץ הצטמצם, כיוון שמקדם הסיעוף גדל כל כך, חיסרון השיטה היא בכמות הזיכרון הגדולה הנדרשת לי*ישומה לעומת השיטה של עץ בעל כמה שכבות min.*

* + - *יתרון שיטה זו בא לידי ביטוי בעומק העץ. לפי משפט מהתרגול סיבוכיות הזמן אקספוננציאלית בעומק - . אפילו שמקדם הסיעוף גדל, עצם העובדה שהצלחנו להקטין את עומק העץ תביא לשיפור בזמן ריצת התוכנית לעומת שיטה המשתמשת בעץ אסטרטגיה בעל כמה שכבות min.*

1. **בניית סוכן Alpha-Beta**
   1. מבנה העץ החדש ישפיע על אלגוריתם alpha-beta באותו אופן שבו השפיע על אלגוריתם minimax. באלגוריתם זה יש שימוש בשני חסמים, (חסם תחתון מקסימלי) ו- (חסם עליון מינימלי) שיאפשרו צמצום של הצמתים שנפתח בעץ. נפתח רק צמתים שיקיימו בכל זמן נתון *.*

*גיזום : פיתוח צומת min. אם* החסם של גבוה מהערך המתקבל באחד השכנים נבצע גזימה *של הענף כיוון שמובטח לנו שערכו של צומת ה-max העליונה יהיה בתחום ולכן אין צורך לפתח ענף המביא לערכים שלא בטווח. אמנם בעץ זה מעל צומת min אין בהכרח צומת max כמו בעץ אסטרטגיה של שני משתתפים, אך לפי הגדרת העומק אנו יודעים כי בסוף כל שכבות ה-min של הרוחות, תמצא שכבת max של פקמן. לכן גם בעץ זה אין טעם לפעפע מעלה מסלולים שבמילא לא יתחשבו בהם בשכבת ה-max הבאה.*

*גיזום : פיתוח צומת max. אם החסם של נמוך יותר מהערך המתקבל באחד השכנים, נבצע גזימה של הענף. אין צורך לפתח ענף שבודאות לא יבחר בשכבת ה-min הבאה.*

*מאותם שיקולים כמו בגיזום , למרות שמבנה העץ שונה, מובטח לנו שאחרי כל שכבת max, תהיה שכבת min ולכן נכונות הגיזום בפרט והאלגוריתם בכלל, נשמרת.*

1. *סוכן ה-alpha-beta יתנהג שונה מסוכן minimax בשני האופנים:*
   * + *זמן הריצה יתקצר – זמן חישוב הצעד הבא יתקצר שכן עבור אלגוריתם זה אין טעם לחשב את ערך minimax עבור כל המסלולים, אלא מסלולים שיודעים מראש שלא יבחרו לא נלקחים בחשבון. מספר החישובים קטן לעומת אלגוריתם minimax וכתוצאה מכך גם זמן הריצה.*
     + *ערך ה-minimax הסופי יישאר זהה לזה של סוכן ה-minimax, אך תתכן בחירת מהלכים שונה וזאת בעקבות מצבים של שוויון בין ערכי minimax. עבור אלגוריתם זה אם יש שוויון וערך minimax של צומת שווה ל- או ל- , הענף נגזם והצומת בוודאות לא תיבחר לפעפע הלאה, זאת בניגוד לאלגוריתם minimax שיכל היה לבחור אחרת.*
2. **בניית סוכן Expectimax לרוח רנדומלית**

כעת יש בידנו מידע נוסף, תנועת הרוחות נבחרת על ידי התפלגות אחידה על כל האפשרויות. Minimax ו-alpha-beta שניהם מניחים כי היריב יעשה את הצעד הכי טוב בשבילו וכאן אנחנו יודעים שלא כך המצב. אין סיבה לתת לצומת את ה-value הכי נמוך, אלא ניתן להעריך אותה כתוחלת הערכים האפשריים.

* + - נצפה שבממוצע מספר הפעמים ש- סוכן Expectimax יפסיד יהיה גדול יותר ממספר הפעמים שסוכני *minimax* או *alpha-beta* יפסידו.
    - נצפה שבמצב שבו פקמן לכוד בין רוחות נראה התנהגות שונה בין אלגוריתם זה לאלגוריתמים הקודמים. עבור האלגוריתמים הקודמים, ההנחה היא תמיד שהרוחות יעשו את הצעד הכי גרוע לפקמן, פקמן יחשוב שאין לו מצב שלא מוביל להפסד ולכן יפסיד בודאות. מה שקורה בפועל הוא שיש סבירות מסוימת שהרוחות ילכו לכיוון שלא מוביל לפקמן, כלומר לא הכי גרוע בשבילו. סוכן Expectimax ידע להתחשב באפשרות זו ולכן חלק מהפעמים יחלץ מהמלכוד וינצח.

1. **בניית סוכן Expectimax לרוח לא רנדומלית**
   1. הרוח הכיוונית מקיימת מדיניות סטוכסטית של תנועה בהסתברות גבוהה בכיוון המקיים מרחק הקצר ביותר מפאקמן אם היא במצב פעיל, ותנועה בכיוון המרחק הארוך ביותר מפאקמן בהסתברות גבוהה אם היא במצב מבוהל.

ספציפית ההסתברות נבנית כך שאם קיימים מספר כיוונים בעלי מרחק הקצר ביותר/ארוך ביותר, ההסתברות עליהם אחידה, וכן אחידה עברו המצבים המשלימים להם.

למשל אם הרוח במצב פעיל, ו – 3 כיוונים מקיימים מרחק מנהטן מינימלי לפאקמן, נניח בה"כ למעלה, ימינה, למטה. אז ההתפלגות תהא: 

נשים לב גם שככל שיש פחות כיוונים אפטימליים במידה שווה, ההסתברות עליהם גדלה ואילו ההסתברות על כל אחד מהמצבים המשלימים קטנה.

* 1. בקוד.
  2. ההבדל הוא באופן חישוב התוחלת. בעוד randomExpectimax הניח התפלגות אחידה של הבחירות של הרוח, וביצע חישוב של התוחלת באופן הנ"ל, directionalExpectimax מבצע משקול שונה של הבחירות של הרוח לכיוון התקדמות, לפי ההסתברות הידועה לנו שהבחירה אכן תילקח.
  3. רעיונות לשיפור הרוח:
     + ניתן להציע פונ' התפלגות טובה יותר עבור הבחירות של הרוחות, כך שתתקיים הבדלה בין מצבים לא אופטימליים יותר ופחות. לדוג', נציע בחירה של הכיוונים האופטימליים בהסתברות אחידה כפי שמתקיים עכשיו, ופילוג גאוסי על היתר. זה יבטיח בחירה בכיוון הכי פחות אופטימלי בהסתברות נמוכה יותר מאשר הכיוון הלא אופטימלי הבא אחריו.
     + בניה ועדכון של מודל הבחירה של פאקמן בהתאם לצעדים שהוא לוקח. תחת ההנחה שמודל הבחירה של פאקמן לא ידוע לרוח (הנחה חזקה שכן אנחנו הנחנו בdirectional שמודל הבחירה של הרוח כן ידעו לנו), נוכל לאמן רשת נוירונים שמטרתה לשערך את פונ' הevaluation של פאקמן, ואז תמיד לבחור פעולה אופטימלית מולו. נבצע את הצעדים הבאים:
       1. נאתחל את כל הבחירות של פאקמן להסתברות שווה
       2. נמדוד פרמטרים שונים של המשחק (מיקום פאקמן, מיקום רוחות, כמות אוכל, כמות רוחות פעילות וכו')
       3. נבצע feed forward של הפרמטרים הנמדדים דרך רשת נוירונים המבצעת classification, נקבל פילוג על הכיוונים, ונבחר במקס'.
       4. נבצע צעד ונמדוד את הבחירה שפאקמן עשה.
       5. נעדכן את פרמטרי הרשת על פי פונ' מחיר הלוקחת בחשבון את הבחירה של פאקמן ואת השערוך שביצענו.
       6. לאחר מספר איטרציות כאלו, במידה והרשת התכנסה נגיע לשערוך אופטימלי של הבחירה של פאקמן בכל צעד.

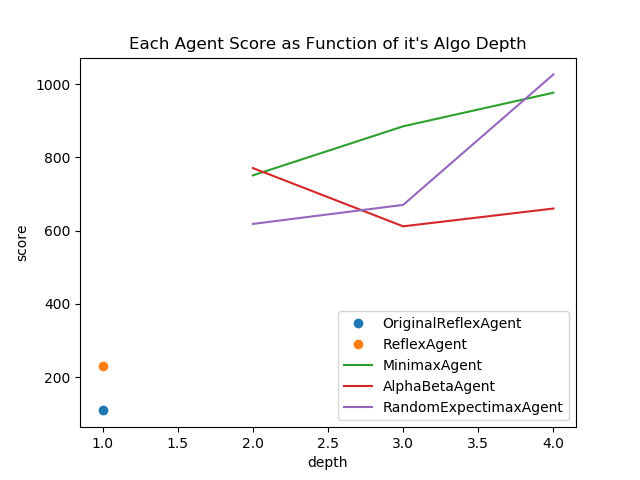
1. **ניסוח השערות במשחק פאקמן**
   1. נגדיר משתנה אקראי ברנולי עבור משחק בודד של הסוכן שלנו מול המחשב: 
   2. נניח השערת האפס כי הסוכן לא טוב מהמחשב, בממוצע. כלומר, ההסתברות לנצח זהה להסתברות להפסיד למחשב: 
   3. נגדיר את ההשערה הנגדית כהנחה כי קיימת הסתברות גבוהה יותר לניצחון הסוכן על המחשב
   4. אם כן:  (מבחן חד צדדי)
   5. נגדיר t סטטיסטי של הנתונים עבור המבחן: 
   6. נגדיר את חוק המבחן: נדחה את השערת האפס אם  עבור c כלשהו.
   7. נשים לב שעבור הסטטיסטי t הנ"ל מתקיים: 

לכן עבור המשערך של p מתקיים: 

ואז עבור הסטטיסטי t כולו, נשים לב שמתקיים נרמול למשתנה נורמלי ממורכז: תחת חוק המספרים הגדולים המקיים: 

* 1. לסיכום, נדחה את השערת האפס בהסתברות , ונקבע את הקבוע לפי רמת המובהקות הקריטית הדרושה.

1. **ניסויים תוצאות ומסקנות**
   1. מצורף קובץ experiments.csv.

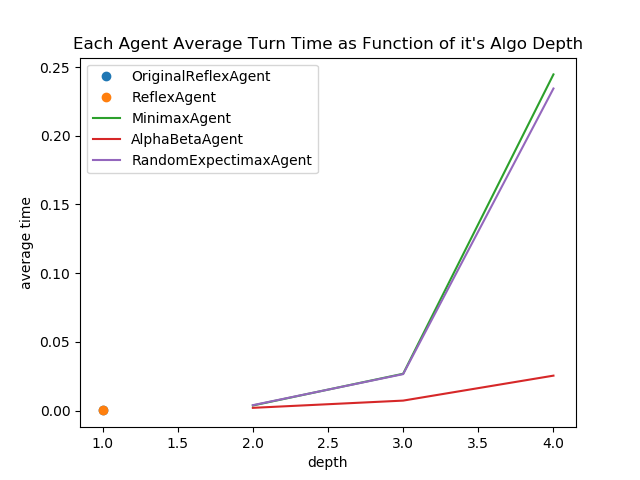


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **1** | **2** | **3** | **4** |
| OriginalReflexAgent | 109.314 |  |  |  |
| ReflexAgent | 230.586 |  |  |  |
| MinimaxAgent |  | 751 | 884.956 | 976.758 |
| AlphaBetaAgent |  | 771.043 | 611.914 | 660.485 |
| RandomExpectimaxAgent |  | 618.416 | 670.443 | 1026.728 |

* 1. ראשית יש לשים לב כי האלגוריתמים הרפלקסיביים (לא מנסים לתכנן קדימה), השיגו ביצועים נחותים באופן משמעותי מכן היתר.   
     שנית, נשים לב כי אלגו' ה – Expectimax הינו בעל תוצאות נחותות בממוצע ברוב העומקים לעומת שאר האלגוריתמים המתכננים. זה גם הגיוני משום ששאר האלגוריתמים מניחים את המצב הגרוע ביותר מבחינתם (הבחירה הטובה ביותר של היריב), ולכן נוקטים פעולה אופטימלית תמיד. סוכן Expectimax לקוח בחשבון את התגובה הממוצעת של היריב, ולכן לפעמים מבצע פעולות המביאות אותו להפסד שהיה יכול להימנע.

אין לי הסבר טוב לנחיתות המשמעותית של אלגו' AlphaBeta לעומת Minimax, מלבד טעות במימוש. **יש לבדוק זאת!**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **1** | **2** | **3** | **4** |
| **OriginalReflexAgent** | 7.89E-05 |  |  |  |
| **ReflexAgent** | 0.000232 |  |  |  |
| **MinimaxAgent** |  | 0.003645 | 0.026743 | 0.244631 |
| **AlphaBetaAgent** |  | 0.001944 | 0.007208 | 0.025376 |
| **RandomExpectimaxAgent** |  | 0.003885 | 0.026493 | 0.23436 |

* 1. התוצאות תואמות היטב את ציפיותינו. קל להבחין בשיפור המשמעותי שמשיג אלגו' AlphaBeta על פני שאר האלגו' המתכננים. חשוב להדגיש גם כי ככל שהעומק גדל, כן היתרון של האלגו' גדל. נק' חשובה היא העומק 2 – ניתן להבחין בכך שלכל האלגו' אותו זמן בקירוב, וזה עקב אופן הפעולה של אלגו' AlphaBeta אשר צריך לפתח עוד שלב למטה על מנת לבצע גיזום (מסתכל על 'הנכד' של השכן), ולכן עבור עומק 2 לא מבצע גיזום.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RandomExpectimaxAgent | 4 | trickyClassic | 1858 | 7.48E-01 |  | RandomGhost |
| RandomExpectimaxAgent | 4 | trickyClassic | 1678.8 | 6.01E-01 |  | DirectionalGhost |
| DirectionalExpectimaxAgent | 4 | trickyClassic | 1316.8 | 5.35E-01 |  | RandomGhost |
| DirectionalExpectimaxAgent | 4 | trickyClassic | 1315.8 | 1.13E+00 |  | DirectionalGhost |

כפי שניתן לראות, עבור סוכן המניח התפלגות אחידה ורוח שאכן מקיימת התפלגות אחידה, הניקוד גבוה יותר מאשר עבור רוח המקיימת התפלגות לא אחידה. עבור סוכן המניח התפלגות מכוונת התקבל ניקוד זהה עבור שני סוגי הרוחות. אין לי במה לתלות את זה מלבד בעובדה שמדובר על מדגם קטן מידי של 5 הרצות בלבד. כן חשוב לציין בהקשר הזה שזמן חישוב התור הממוצע ארוך הרבה יותר עבור הרוח הכיוונית וזאת עקב החישוב העודף המשמעותית הדרוש לשם חישוב ההתפלגות.

1. **תחרות בקורס**