פרויקט סיום בקורס מבוא לבינה מלאכותית

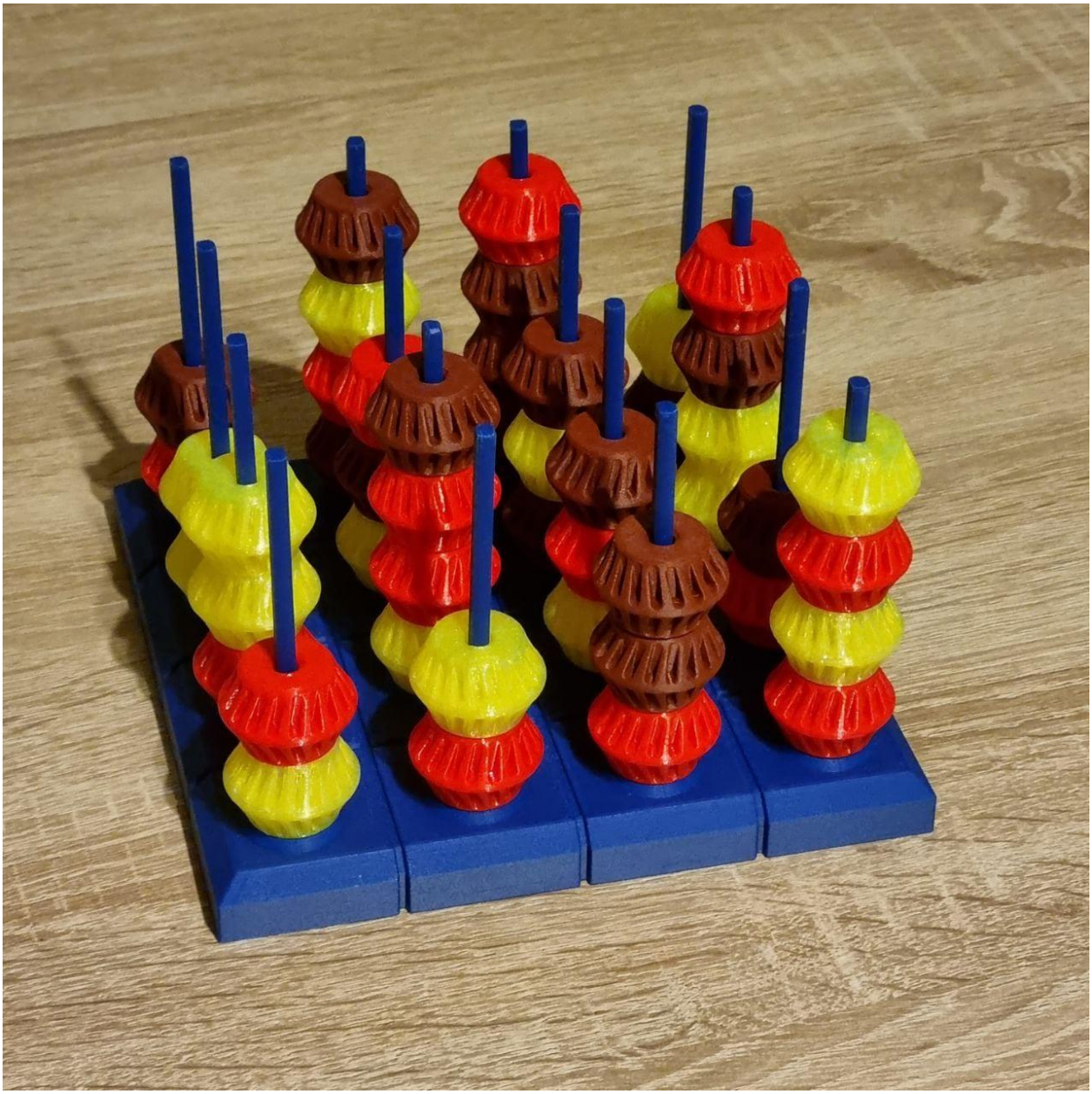
3D multi-player connect 4

מגישים:

איתמר שרם, 206762551

שלום בלוי, 319144762

עבד נירוך, 213668700



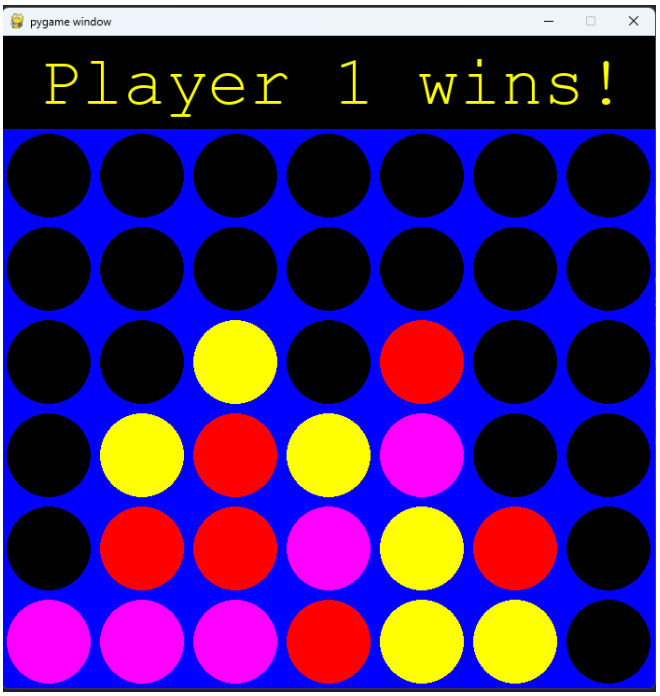
תוכן עניינים

קישור לקוד הפרוייקט

קישור לעמוד הפרוייקט ב-github:

https://github.com/itamarshrem/AI_project

מבוא:



המשחק שלנו מבוסס על המשחק 4 בשורה. המשחק המקורי מתקיים בלוח דו ממדי בגודל קבוע, לשני שחקנים. כל שחקן בתורו משחיל דיסקית בצבע שלו לאחת העמודות בלוח. מטרתו של כל שחקן להגיע ראשון לרצף של 4 דיסקיות בצבע שלו. הרצף יכול להיות בשורה, בעמודה או באלכסון.

בפרויקט שלנו הרחבנו את המשחק, על מנת לאתגר את הסוכנים שלנו ולבחון אותם בסביבות מורכבות יותר. השדרוג חל בכמה אופנים:

- לוח המשחק מוגדר כעת ע"י הרביעייה הבאה: (R,C,D,W), כאשר R הוא מספר השורות, C הוא מספר העמודות, D הוא העומק וW הוא רצף הנצחון
- ניתן להריץ את המשחק עם יותר משני שחקנים, כל אחד מתחרה בשאר השחקנים על מנת לנצח.

עץ המצבים של המשחק גדל בצורה אקספוננציאלית- $O((C*D)^t)$ כאשר t הוא מספר התורות עד לניצחון. כלומר סוכנים שיממשו אלגוריתמי חיפוש פשוטים ירוצו למשך זמן רב! לפיכך צריך לחשוב על אלגוריתמים חכמים, וזו הסיבה שבחרנו במשחק זה.

לאחר עבודה על האלגוריתמים במשחק, מצאנו באינטרנט UI מעוצב למשחק דו ממדי, אותו חיברנו למשחק שלנו. בשל אילוצים ויזואליים, נאלצנו להשאיר במשחק תלת ממדי את הממשק הטקסטואלי.

מימשנו סוכני AI שונים, לפי רעיונות שונים שלמדנו בכיתה, שמטרתם היא לנצח במשחק את השחקנים שיתמודדו מולם. הסוכנים שמימשנו הם:

1. minmax_agent
2. alpha_beta_agent
3. Q-learning_agent

נשים לב שהמשחק המקורי הוא משחק סכום אפס- נצחונו של שחקן אחד הוא הפסדו של האחר. לפיכך חשבנו כי האלגוריתם הטבעי ביותר להתחיל איתו הוא אלגוריתם המינמקס, שמותאם לנצח בסוג זה של משחקים. אלגוריתם מינמקס במימוש נאיבי, יורד עד לעומק עץ המצבים, ומהעלים מתחיל לשערך את הציון למצבים. במשחק 4 בשורה, שבו עץ המצבים עמוק מאוד, דבר זה אינו ישים. לכן נצטרך להגביל

את עומק החיפוש שלנו ולתת ציון למצב לוח למרות שהמשחק עדיין לא הסתיים. לשם כך השתמשנו בהיוריסטיקה שהגדרנו, אותה נתאר בחלק המתודולוגיה.

סוכן האלפא-בטא הוא הסוכן ההגיוני הבא- הוא פועל כמו סוכן המינמקס מבחינת ההחלטות שבוחר במהלך המשחק, אך מדלג על ענפים מסוימים ובכך מקצר את זמן הריצה.

לסיום, רצינו לבחון את ביצועיו של סוכן reinforcement learning . דרך אפשרית להפעיל את האלגוריתם במקרה שלנו, היא לתת לשחקן למידת החיזוק לשחק מול שחקן מינימקס/רנדומי ודרכו ללמוד איך לשחק את המשחק בצורה הטובה ביותר. סוכן זה ילמד באמצעות משחקים רבים את הפעולות הכדאיות בהינתן מצבי הלוח, כאשר במהלך הלמידה הסוכן מעדכן את הציון על מצב לוח מסוים, באמצעות הרצת סימולציות עד לנצחון/הפסד(monte carlo).

עבודות קודמות:

מינמקס ואלפא בטא

מצאנו שתי עבודות קודמות שמימשו סטודנטים בקורס זה.

בשתיהן, האלגוריתמים בהם השתמשו הם minmax ו-alpha beta pruning . כפי שאנחנו הסקנו, עץ המצבים במשחק גדול מאוד, ולכן החוקרים בחרו להגביל את עומק החיפוש ולחשב ציון ללוח של משחק שטרם הסתיים. ההבדלים בין האלגוריתמים שהם בחרו הוא בפונקציית האבליואציה שמחשבת ציון ללוח. נציג את ההיוריסטיקות המוצלחות ביותר בכל אחת מהעבודות.

נסמן ב-N את אורך הרצף המוגדר עבור נצחון במשחק(ברירת המחדל היא 4)

פונקציה א (שהוגדרה בעבודה זו):

במשחק 4 בשורה עם לוח דיפולטיבי(6 שורות על 7 עמודות) ישנן 69 פוזיציות בהן אפשר להגיע לנצחון(24 אופקיות, 21 אנכיות ו24 אלכסוניות). אם ברצף באורך N יש לשני השחקנים דיסקיות, נתעלם מרצף זה(שכן אף אחד מהם לא יכול לנצח שם). אחרת, לכל רצף בלוח ניתן ערך לפי כמות הדיסקיות שיש בו לשחקן מסוים. נסכום את כל הערכים הללו עבור השחקן הראשי, ונחסר מהם את סכום הערכים של היריב.

בהיוריסטיקה זו יש כמה חסרונות:

- הערכים שנבחרו מותאמים עבור לוח בגודל הדיפולטיבי בלבד, וקשה להכליל אותה למשחק עם לוח בגודל שונה, עם רצף נצחון שונה.
- ההיוריסטיקה יכולה לתת ערך זהה ללוח בו יש לנו הרבה רצפים קצרים, לעומת מעט רצפים ארוכים. אם נחשוב על הרלקסציה שבה ככל שיש לשחקן יותר דיסקיות ברצף, הוא יותר קרוב לנצחון, היוריסטיקה זו מתעלמת מעיקרון זה.

פונקציה ב(נקראה IBEF2, בעבודה זו)

נספור רצפים לא חסומים(שלא מכילים דיסקיות של היריב).ונחזיר 2^k , עבור k מספר הדיסקיות ברצף זה. הפונקציה תסכום את הערכים הללו ותחסיר מערך זה את סכום הערכים של היריב.

חסרונות של היוריסטיקה זו:

בדומה להיוריסטיקה הקודמת, יכולה לתת ערך זהה ללוח בו יש לנו הרבה רצפים קצרים, לעומת מעט רצפים ארוכים.

Reinforcement learning

בחיפושינו אחר עבודות קודמות, נתקלנו במחקר זה, בו אימנו החוקרים שחקן q-learning שבהתאם למקדם הדעיכה מתחיל לפעול באופן רנדומלי, ולאט לאט מתחיל לפעול על דעת עצמו.

חסרונות: במשחק כזה יש מספר רב של מצבי נצחון והפסד, ניתן להגיע לנצחונות במסלולים שאינם כדאיים, ולא ישקפו מסלול הגיוני במשחק מול שחקן מתוחכם. לכאורה עדיף היה לשקול מורה דרך במקום משחק רנדומלי.

מתודולוגיה:

מידול הבעיה:

החלטנו למדל את המשחק באופן שיקל על הסוכנים שלנו.

נגדיר כל לוח בו מונחות דיסקיות בצבעים שונים ובהתאם לחוקים(דיסקיות לא יכולות לרחף), כמצב במרחב המצבים. מצב התחלתי הוא לוח ריק, מצב נצחון הוא לוח שבו אחד השחקנים הגיע לרצף נצחון.

נגדיר פעולה בתור הנחת דיסקית בצבע מסוים, בעמודה ועומק מסוימים.

עבור אלגוריתם ה-rl agent , המצבים ייוצגו באופן שונה, על מנת להקטין את מרחב המצבים, כפי שנתאר בהמשך.

מינמקס, אלפאבטא והיוריסטיקות

הרחבת אלגוריתמי minmax,alphabeta עבור משחק של יותר משני שחקנים:

במקרה זה, השחקן הראשי יהיה שחקן המקסימום, ומבחינתו כל שאר השחקנים יהיו שחקני המינימום. בשערוך הערך עבור כל קודקוד בעץ ששחקן המקסימום מגדיר(עד עומק מסוים), שחקן המקסימום יבחר את הערך המקסימלי מבין ערכי כל ילדיו. כל אחד משחקני המינימום יבחר את המינימום מבין ערכי כל ילדיו.

נשים לב כי בגישה זו, מטרתו של כל שחקן מינימום היא לוודא ששחקן המקסימום לא מנצח, גם אם זה גורר ששחקן מינימום אחר ינצח- מכיוון שההיוריסטיקות מחושבות במינוס על כלל היריבים, אזי מטרת כל אחד משחקני המינימום היא שכל השחקנים היריבים(כולל הם עצמם) יקבלו ניקוד גבוה, וכך יורידו מערכו של שחקן המקסימום. בעיני שחקן המקסימום, כלל יריביו עשו יד אחת נגדו.

נשים לב כי גישה זו נכונה גם עבור ה-pruning באלגוריתם האלפא-בטא:

במשחק עם שני שחקנים, הערך אלפא הוא הערך שמייצג ערך מקסימלי שהגיע משחקן מקסימום קדמון לשחקן מינימום שרואה האם אפשר להפסיק את החיפושים. הערך בטא פועל באופן מנוגד לכך. נשים לב כי הערכים האלה לא בהכרח מייצגים ערכים של אב ישיר, ולכן גם במקרה של משחק רב משתתפים, בו שחקן המקסימום הוא סב של שחקן מינימום, הערך אלפא שעודכן על ידו עדיין רלוונטי לצורך גזימה.

כעת נציג את ההיוריסטיקה המרכזית ששימשה את אלגוריתם המינמקס:

היוריסטיקת complex

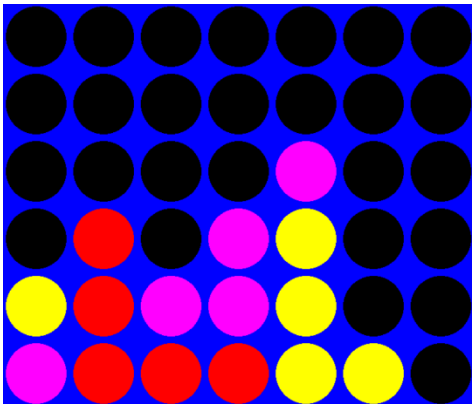
ההיוריסטיקה פועלת באופן הבא:

בהינתן לוח מסוים, ההיוריסטיקה תחזיר שני ערכים, שמייצגים את הרצף הארוך ביותר שאינו חסום שהושג בלוח, וכמות המופעים שלו. מאלו מחסרים את שקלול הערכים המתאימים ליריבים באופן הבא:

- מהרצף הארוך ביותר של שחקן המקסימום, נחסיר את ממוצע הרצפים המקסימליים של היריבים

- מכמות הרצפים הארוכים ביותר של השחקן המקסימלי, נחסיר רק את כמויות הרצפים המקסימליים של יריבים שהשיגו אותו רצף מקסימלי זהה לשל שחקן המקסימום- אנחנו לא רוצים לתת משקל ליריבים "חלשים" יותר בלוח נתון.

לדוגמא:



רצף הנצחון מוגדר להיות 4, שחקן המינמקס הוא השחקן הורוד. הרצף הכי ארוך שלו הוא 3, וכמות הרצפים באורך זה היא 1. לאדום יש רצף אחד באורך 3 (הרצף השני שלו באורך 3 חסום). לצהוב חמישה רצפים באורך 1 (ישנן דסקיות שנספרות יותר מפעם אחת) ולכן פונקציית ההיוריסטיקה תחזיר:

$$2^3 - \frac{1}{2}(2^3 + 2^1), 1 - 1 = 3{,}0$$

נשים לב כי בדרך זו, אנו נותנים עדיפות לאורך הרצף, ורק כאשר הרצפים זהים אנחנו מסתכלים על מספר המופעים. בכך אנו מתגברים על הבעיה שהצגנו בהיוריסטיקות מהעבודות הקודמות.

כאשר הרחבנו את המשחק ללוח תלת ממדי, נתקלנו בבעיה- על מנת לבדוק האם הנחה של דיסקית הובילה לנצחון, עלינו לבדוק רצפים רבים שיכולים להביא לנצחון. דבר זה פגע בזמני הריצה. לכן על מנת לשפר את זמני הריצה, אנחנו מחזיקים עבור כל לוח את מפת כל הרצפים המופיעים בו, וכאשר מוסיפים דיסקית, מעדכנים את מפות הרצפים רק באותו אזור מצומצם. הדבר שיפר משמעותית את זמני הריצה.

סוכן Q-Learning

המשחק שלנו מכיל יותר משחקן אחד, ולכן אלגוריתם ה-Q-learning שונה מהאלגוריתם שראינו בכיתה: הסוכן שלנו מבצע פעולה, וכאשר באים לבחון את השלכות הפעולה שלו, כלומר לאיזה מצב הגענו, יש להסתכל לאיזה מצב הגענו אחרי הפעולה שלנו, אך גם של היריבים שלו.

החלטנו לממש את אלגוריתם ה-Q-learning בעצמנו, כך שנוכל להתאים את האלגוריתם לצרכנו: ריבוי משתתפים, שינוי שיטת האימון והתאמה ל-API שלנו כך שיוכל להתאמן ולשחק עם שאר השחקנים שיצרנו.

שיטת אימון

בחירת יריב

מאחר ובמשחק שלנו קיימים מצבי לוח רבים, ועל מנת ללמוד השחקן צריך לראות כמה שיותר מצבי לוח, כדאי לבחור ביריב שיגרום לנו לטייל במרחב המצבים.

כמו כן, אם היינו בוחרים לאמן נגד יריב חזק, הסוכן שלנו כמעט ולא היה מנצח. שמנו לב שהאופן שבו האלגוריתם q-learning בנוי, הוא שכאשר מתעדכן ציון שלילי, לוקח זמן עד שהציון הזה יחלחל למצבים שקודמים לו. לכן בשיטה זו הלמידה של הסוכן מועטה, אם בכלל.

אולם, אם נבחר ביריב רנדומלי לחלוטין, ייתכן מצב בו ניצחנו אותו, למרות שמדובר בסדרת מהלכים שמול שחקן רגיל הייתה מובילה להפסד ודאי. כך יפעפע ציון חיובי עבור סדרת מהלכים זו.

לכן החלטנו לבחור ביריב שלעיתים מבצע החלטות מושכלות, ובשאר הזמן יבצע מהלכים הסתברותיים על מנת לאפשר לסוכן שלנו להיתקל בכמה שיותר מצבים.

לאחר ניסויים רבים, בחרנו את היריב להיות שחקן אלפא בטא עם ההיוריסטיקה complex, עם עומק 2, שיוחלש ע"י כך שבסיכוי 0.8 הוא מגריל צעד.

צעדי הסוכן בשלב בלמידה

על מנת לאפשר לסוכן לנצח את היריב, הוא התאמן בליווי "מאסטר": המאסטר הוא שחקן שמבצע החלטות מושכלות, וגם חזק יותר מהיריב. בתחילת דרכו שחקן הRL ייתן למאסטר לשחק בשבילו, ובאמצעות הקטנה של ה-exploration rate, המאסטר לאט לאט ישחרר את גלגלי העזר וייתן לסוכן לבצע יותר החלטות בעצמו.

בחרנו את המאסטר להיות שחקן אלפא בטא עם ההיוריסטיקה complex, עם עומק 2.

<u>פונקציית רווח</u>

בחרנו לתת ציון של 500 עבור לוח שבו הסוכן ניצח. עבור לוח שבו הפסיד, בחרנו לתת 500- ועוד מספר הצעדים ששוחקו עד להפסד- כלומר ככל שהסוכן "שרד" יותר זמן, אנו מחשיבים את ההפסד שלו לקטן יותר, מכיוון שיכול להיות שרק המהלכים האחרונים שלו הביאו להפסד, אך הוא שיחק טוב בתחילה.

עבור תיקו הבאנו ציון של 200- , שכן אנחנו רוצים לגרום לסוכן לנצח, אך אם הגיע לתיקו זה הרבה יותר טוב מאשר להפסיד.

לסיום, עבור כל צעד שלא הביא לאיזושהי הכרעה, נתנו לו ציון של 1- , על מנת שיעדיף בנצחון המהיר ביותר.

<u>ייצוג הלוח</u>

כדי שסוכן q-learning יציג תוצאות טובות, עליו להכיר הרבה מצבים במשחק, על מנת שידע לפעול בכל סיטואציה שייתקל בה. יתרה מזאת, אם יפגוש מצב שלא ראה קודם לכן, אפילו אם ראה מצב דומה(למשל תבניות דומות, אבל בהזזה מהמצב שכן ראה) הוא לא ידע כיצד לפעול.

היינו רוצים מרחב מצבים קטן ככל האפשר. מצד שני, ככל שייצוג המצב פשוט יותר, יותר מידע הולך לאיבוד.

נבחין כי בבואנו לבחור את הפעולה, דיסקיות שאינן חלק מרצף שיכול להביא לנצחון/הפסד, ולא משנה לנו של מי הדיסקיות.

לפיכך בחרנו בייצוג הבא: תחילה מקודדים את מספר הדיסקיות בכל עמודה ולאחר מכן את אורכי הרצפים שאינם חסומים, ואת מיקומם בלוח, הן שלנו והן של היריב.

תוצאות

על מנת להעריך את ביצועיהם של הסוכנים שלנו, החלטנו לממש מספר סוכני baseline, שיהוו מדד להצלחה. הסוכנים שמימשנו לצורך בדיקות הם:

שחקן offensive – מטרתו היחידה היא להשלים רצפים שלו לרצפי נצחון

שחקן defensive- מטרתו היא לחסום את שאר השחקנים

על מנת להימנע ממשחקים דטרמיניסטיים, שחקנים אלה יבחרו בפעולה שמניבה להם ציון גבוה ביותר בהסתברות גבוהה, אך בהסתברות נמוכה יכולים לבחור גם בפעולות אחרות. ההסתברות ממושקלת לפי כמה הפעולה טובה עבור השחקן ומטרתו.

שחקן IBEF2 – שחקן מינמקס עם ההיוריסטיקה החזקה ביותר שהצלחנו למצוא בעבודות קודמות

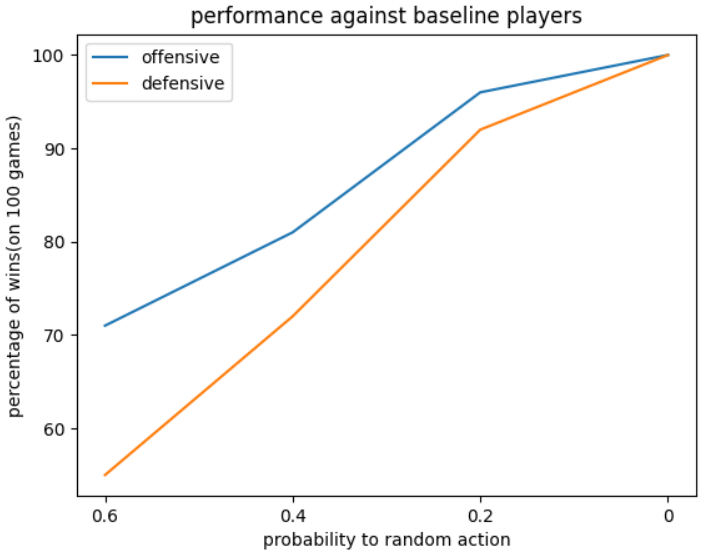
הערה: במקרים מסוימים ובאחוזים מועטים הסתיימו המשחקים בתיקו. לצורך הצגה ויזואלית נוחה יותר (אחוזי ניצחון בלבד) חילקנו את אחוזי התיקו בין השחקנים השונים.

סוכני מינמקס, ואלפא בטא

תוצאות ההיוריסטיקה מול שחקני baseline

ראשית, בדקנו סוכנים אלה במשחק עם שני שחקנים בלבד, בלוח תלת ממדי בגודל (6,7,5).

בבדיקות ראשוניות שביצענו, ראינו כי הסוכן שלנו מנצח את סוכני ה-baseline במאה אחוז מהמשחקים. מכיוון שרצינו לבדוק את הסוכן יותר לעומק, בחרנו להחליש את הסוכן שלנו ע"י כך שיבצע מהלכים אקראיים בהסתברות מסוימת. הגרף הבא מציג את ביצועי השחקן שלנו אל מול שחקני ה-offensive,defensive כתלות בסיכוי שלו לבצע מהלך אקראי:



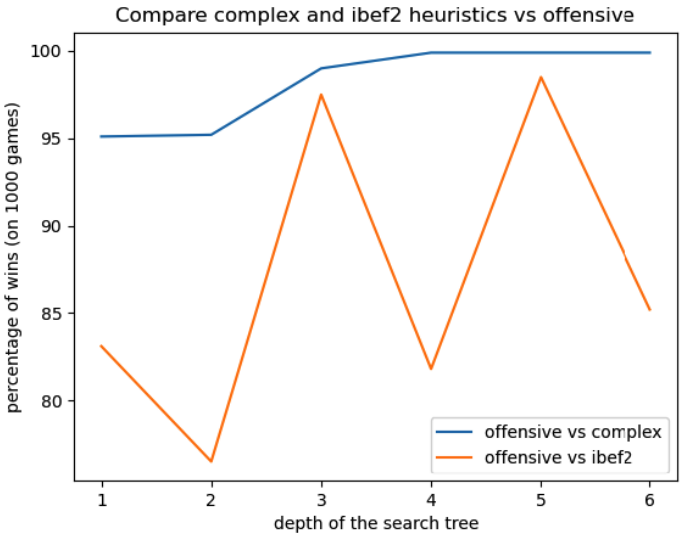
ניתן לראות כי כאשר החלשנו את הסוכן שלנו עם הסתברות של 0.6 לפעולה רנדומית, עדיין הוא מנצח את היריבים שלו בהסתברות לא מבוטלת. כמובן שככל שמקטינים ערך זה, הוא מנצח אותם אף יותר.

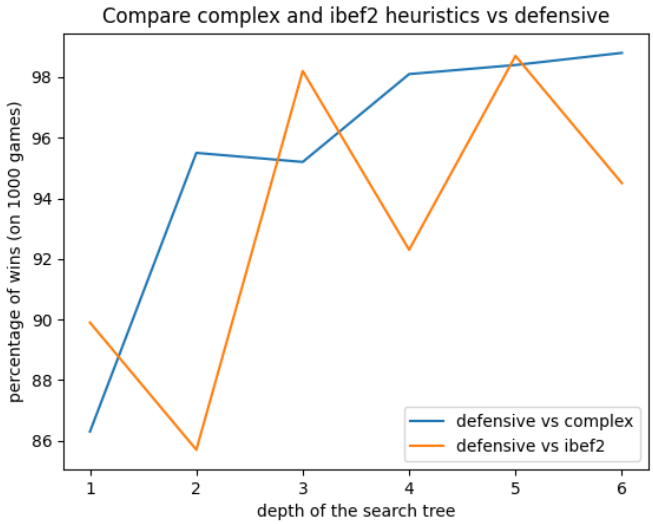
מעניין לראות כי ביצועיו נגד שחקן ה-defensive נמוכים יותר מאשר נגד שחקן ה-offensive: נשים לב כי כל אחד מהיריבים מממש אחת מהמטרות שמממש שחקן המינמקס. בעוד בניית רצף נצחון היא טקטיקה שמצריכה לפחות 4 מהלכים מחושבים, חסימת רצף היא פעולה קלה בהרבה. לכן

ניצח הרבה יותר את שחקן ה-offensive.

<u>ההיוריסטיקה שלנו לעומת היוריסטיקה מעבודות קודמות</u>

כעת, השווינו את ההיוריסטיקה שלנו אל מול שחקן IBEF2. שחקן זה לא מותאם לשחק בלוח תלת ממדי ולכן עברנו לשחק בלוח דו ממדי בגודל (6,7). בנוסף, השווינו כל אחת מההיוריסטיקות עבור עומקים שונים. מכיוון שהמהלכים שההיוריסטיקות משרות יוצאים יחסית דטרמיניסטיים(נבחר בפעולה שתקבל את הציון הגבוה היותר, ופעמים רבות במשחק יש רק מהלך אחד כזה), בחרנו להשוות כל אחת מההיוריסטיקות אל מול שחקני ה-baseline. להלן התוצאות:





ניתן לראות כי ביצועיו של הסוכן שלנו עקביים יותר, ואחוזי ההצלחה שלו עולים ככל שעומק עץ המינמקס גדל.

לעומתו סוכן ה-IBEF2 מציג תוצאות פחות טובות, ובנוסף פחות עקביות.

לצורך קיצור זמני הריצה, ומעתה ועד סוף הדוח, ההשוואות שנערוך יהיו על לוחות דו ממדיים- עדכון מפות האקטיבציה עבור הרצפים בכל הכיוונים לוקחת זמן רב יותר בלוח תלת ממדי מפני שיש יותר רצפים להתחשב בהם. לאחר שבדקנו את נכונותן, היה לנו חשוב יותר להעריך את ביצועיהם של אלגוריתמי הסוכנים שלנו במספר גדול יותר של משחקים.

<u>תוצאות ההיוריסטיקה במשחק רב משתתפים</u>

עתה, עברנו לבחון את ההיוריסטיקה שלנו במשחק עם 3 משתתפים:

על מנת לבחון את היוריסטיקת ה-complex במשחק של 3 משתתפים, הגדרנו סוכן חדש שנקרא only_best_opponent . סוכן זה הוא סוכן אלפא בטא, שמשתמש בהיוריסטיקה זהה כמעט לחלוטין להיוריסטיקת ה-complex, למעט הפרט הבא:

בהיוריסטיקת ה-complex, אנו מחסירים מהרצף המקסימלי שהשחקן שלנו השיג, את ממוצע הרצפים המקסימליים שהשיג כל אחד מהשחקנים האחרים.

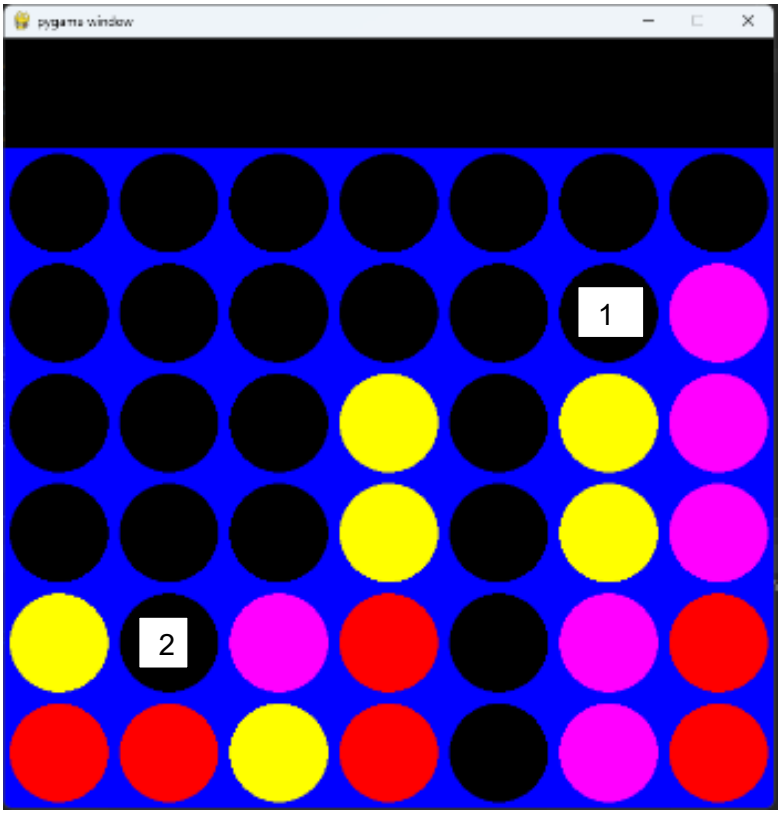
בהיוריסטיקת ה-only_best_opponent , נחסיר מהרצף המקסימלי של השחקן שלנו, את הרצף המקסימלי שהושג ע"י היריב שהשיג רצף הכי ארוך.

כפי שהסברנו במתודולוגיה, הגישה של סוכן המינמקס במשחק רב משתתפים, הוא שכולם פועלים נגדו(ולא אחד נגד השני). אך ישנן רמות שונות בתוך גישה זו. הגדרנו את ההיוריסטיקה של only_best_opponen כדי להדגים זאת: בהיוריסטיקה זו, הסוכן מניח כי ברגע שאחד השחקנים קרוב יותר לניצחון, שאר השחקנים מוותרים על הניצחון שלהם ומתמקדים בחסימת הסוכן.

לעומת זאת ההיוריסטיקה שלנו מעדנת במעט גישה זו, כאשר איתה הסוכן מניח כי השחקנים יעדיפו שלא לחסום אחד השני, אך כן ינסו לנצח בעצמם. בכך היוריסטיקה זו משקפת את המציאות מעט יותר טוב.

נמחיש זאת בדוגמה הבאה:

סוכן המקסימום הוא השחקן האדום, רצף ניצחון הוא באורך 4.



כאשר הסוכן מנחש את הצעד שיעשה הצהוב, הוא יבחן את הפעולות שמסומנות בספרות 1,2.

בהיוריסטיקה שלנו, הסוכן ייתן ללוח עם הפעולה 1, את הערך $2^1 - \frac{1}{2}(2^3 + 2^3) = -6$

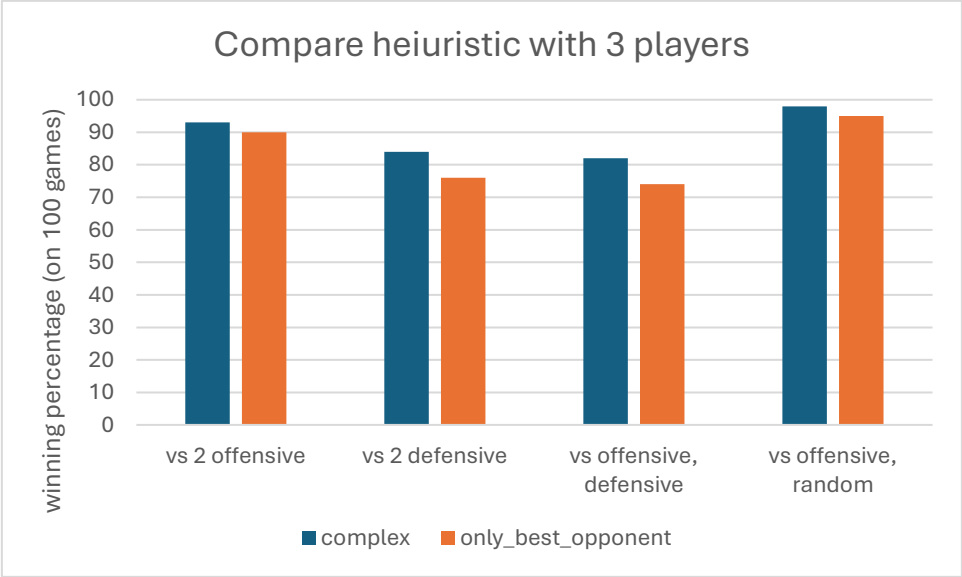
בכניסה הראשונה בציון הלוח

וללוח עם הפעולה 2 ייתן:

$2^1 - \frac{1}{2}(2^2 + 2^3) = -4$ בכניסה הראשונה בציון.

לכן הסוכן יניח שהצהוב יבחר בפעולה 1 שמקרבת אותו לניצחון.

לעומת זאת, בהיוריסטיקה only_best_opponent, הסוכן ייתן גם ללוח עם הפעולה 1 וגם ללוח עם הפעולה 2 את הערך 6- , ולכן יניח כי הצהוב יבחר בפעולה 2 שחוסמת כמות גדולה יותר של רצפים שלו.



נתנו לכל אחד מהסוכנים complex, only_best_opponent לשחק נגד מספר יריבים, וקיבלנו את התוצאות הבאות:

ניתן לראות שהיוריסטיקת ה-complex השיגה תוצאות טובות יותר, ביחס להיוריסטיקת ה-only_best_opponent.

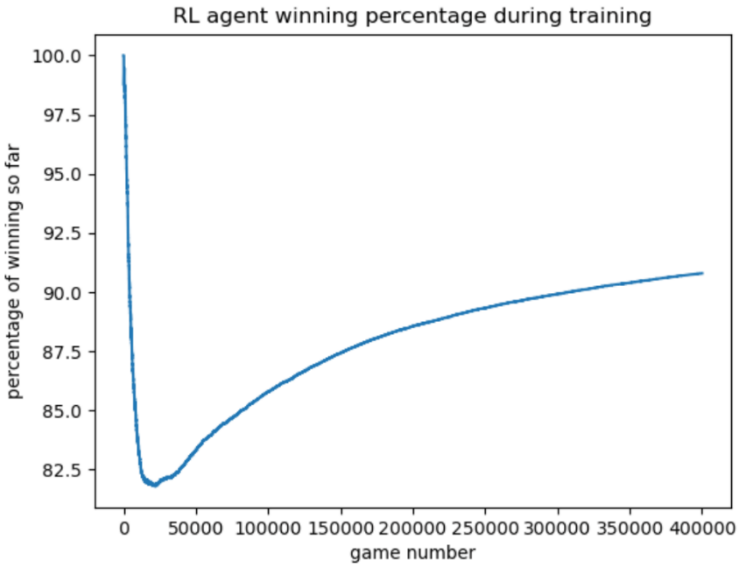
זאת מאחר שהיוריסטיקת ה-complex נותנת לסוכן הנחה על היריבים שקרובה יותר למציאות.

סוכן ה-Q-learning

אימון הסוכן

סוכן זה, בשונה מהסוכנים הקודמים, נדרש לשלב של למידה, בו ה-Q-table שלו תתעדכן. בחרנו לאמן את הסוכן שלנו לאורך 400000 משחקים, כאשר ה-exploration rate דועך למספר קטן ביותר לאורך האיטרציות- כלומר אנו רוצים שבתחילת האימון הסוכן יסתמך אך ורק על המאסטר שלו, כך הטבלה תתמלא בערכים שמייצגים בחירות מושכלות עבור מצבי לוח שונים. אך לאט לאט המאסטר ישחרר את המושכות וייתן לסוכן לבצע החלטות, ומכיוון שאנו מצפים שבטבלה הצטבר מספיק מידע משמעותי, הסוכן יבצע בהתחלה החלטות טובות יותר ויותר.

הגרף הבא מציג את אחוז הנצחונות כתלות בשלב האימון:



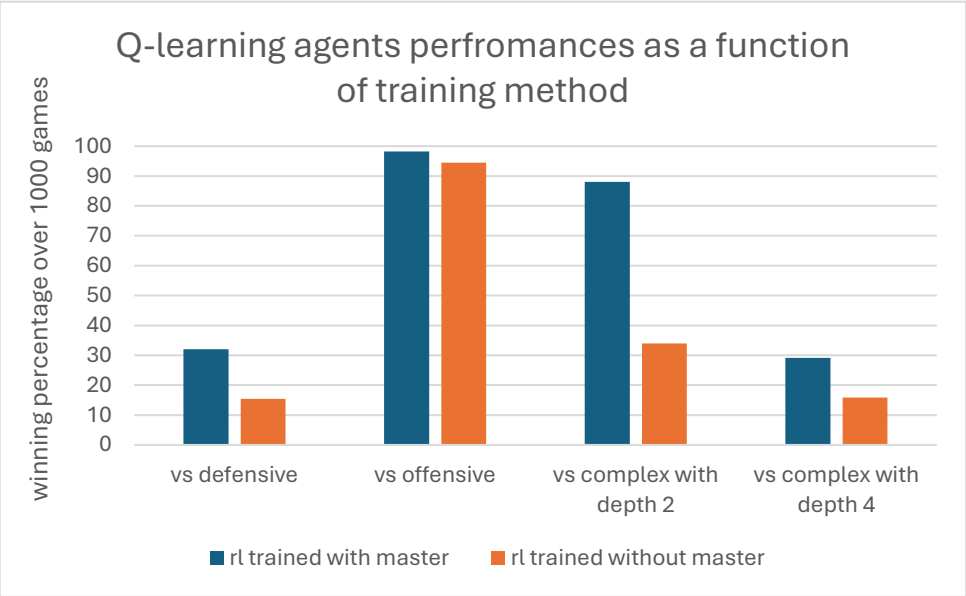
ניתן לראות כי בתחילת האימון הסוכן מנצח בהסתברות גבוהה. זאת מכיוון שהסוכן משתמש במאסטר שלו על מנת לבצע מהלכים. מכיוון שבחרנו במאסטר שיהיה טוב יותר מהיריב של הסוכן, אכן אנו מנצחים בהסתברות גבוהה במשחקים.

לאורך 30000 המשחקים הראשונים רואים דעיכה באחוז הנצחונות המצטבר- מכיוון שה-exploration rate הולך וקטן, המאסטר עוזר לסוכן פחות ופחות, והסוכן מקבל החלטות בעצמו אבל הטבלה שלו עדיין לא מכילה מספיק ערכים לגבי לוחות מסוימים, וגם הערכים עצמם עדיין לא מספיק מייצגים נכונה את המציאות.

לאחר מכן, ניתן לראות עלייה באחוז הנצחונות המצטבר- הטבלה של הסוכן כבר מכילה ערכים טובים יותר, ולכן הוא מצליח לנצח ביותר משחקים, ובנוסף ממשיך לעדכן את הטבלה שלו עם ערכים יותר ויותר מדויקים למדיניות האופטימלית, ובכך מסיים את האימון עם 90% הצלחה.

בחינת אסטרטגיית המאסטר

כפי שהסברנו במתודולוגיה, הנחנו כי סוכן שהתאמן עם מאסטר ייתן ביצועים טובים יותר מאשר סוכן שהתאמן בלי מאסטר, מכיוון שהוא בונה את הטבלה שלו על סמך נתונים שמשקפים נכון יותר את המציאות, ואיזה לוחות טובים ואיזה לא. על מנת לבדוק את ההנחות שלנו בצורה מעשית, אימנו שני סוכנים עם פרמטרים זהים כמעט לחלוטין, למעט העובדה שאחד התאמן עם מאסטר ואחד ביצע במהלך האימון פעולות רנדומיות. נתנו להם לשחק נגד מספר יריבים שונים, להלן התוצאות:



ניתן לראות כי הסוכן שהתאמן עם מאסטר הצליח להשיג תוצאות גבוהות יותר, מול כל היריבים נגדם שיחק, בהשוואה לסוכן שהתאמן בלי מאסטר.

נקודה מעניינת נוספת היא שמול סוכן אלפא בטא עם היוריסטיקת complex בעומק 2 , הסוכן שהתאמן עם מאסטר ניצח בהסתברות גבוהה, בעוד הסוכן שהתאמן בלי מאסטר הפסיד ביותר מחצי מהמשחקים!

נגד סוכן אלפא בטא עם היוריסטיקת complex בעומק 4, ונגד שחקן ה-defensive , שני סוכני ה-Q-learning לא היוו יריב ראוי. לדעתנו הסיבה לכך טמונה בעובדה שהטבלה מתעדכנת בקלות כאשר הסוכן מנצח והמידע מפועפע מטה, אך יותר קשה לעדכן את הטבלה ולהבין שישנם שחקנים שינסו גם לחסום את הסוכן מלנצח.

מכך ניתן להבין שאימון סוכן Q-learning היא בעיה מורכבת, וכאשר משחקים מול שחקנים מורכבים יותר, הטבלה של הסוכן צריכה להכיר מספר גדול ביותר של מצבים, כדי לדעת איך להתמודד בכל תרחיש.

הערה: ברגע האחרון, שמנו לב שעל מחשבי האקוריום טעינת ה-RL agent ארוכה באופן משמעותי, ולכן מטעמי נוחות החלטנו להעלות לgit סוכן שאומן על 100000 משחקים, שביצועיו היו די קרובים לסוכן המקורי.

זמני ריצה

במשחק שלנו, הלוח יכול להיות גדול מאוד, ואף להיות תלת ממדי. בנוסף לכך, כאשר מגדילים את כמות השחקנים, על מנת להגיע לתוצאות טובות, על שחקני החיפוש להגדיל את עומק עץ החיפוש.

לפיכך, על מנת להריץ משחקים בזמן פיזבילי, היה לנו לאפטם את זמני הריצה, ע"י עדכון דינמי של מפות הרצפים בלוח, כאשר העדכון מבוצע אך ורק באזור קטן סביב הדיסקית שהונחה אחרונה, שאינו תלוי בגודל הלוח.

את זמני הריצה המוצגים בטבלה, בדקנו על לוח בגודל (6,7,1). את הבדיקות ביצענו על מחשבי האקוריום.

שחקן	זמן ריצה ממוצע למהלך(ב-100 משחקים)
Minmax depth 1	0.00115
Minmax depth 2	0.00801
Minmax depth 3	0.05212
Minmax depth 4	0.33927
Minmax depth 5	2.30415
Alpha beta depth 1	0.00115
Alpha beta depth 2	0.00469
Alpha beta depth 3	0.01773
Alpha beta depth 4	0.05953
Alpha beta depth 5	0.24708
Rl agent	0.00012

נשים לב כי בדומה להנחות התיאורטיות שלנו, שחקן האלפא בטא משיג זמני ריצה טובים יותר משחקן המינמקס כאשר הם מסתכלים באותם עומקים בעץ החיפוש.

בנוסף, שחקן ה-Q-learning השיג את זמני הריצה הטובים ביותר, שכן על מנת לבצע החלטה הוא מסתכל בטבלה שלו ושולף את הפעולה בעלת הערך מקסימלי, בלי לבצע חישובים נוספים.

סיכום:

משחק ה-4 בשורה אמנם נראה משחק פשוט לכאורה, אך בעל מרחב מצבים גדול ביותר. בפרט המשחק שאנו יצרנו- 3D multiplayer connect 4 , הוא משחק מורכב אף יותר, מכיוון שהלוח יכול לגדול משמעותית, ובהתאם גם מרחב המצבים. בנוסף במשחק מרובה משתתפים כל שחקן צריך להתחשב בפעולותיהם של יותר מיריב אחד, מה שמעלה את רמת הקושי של קבלת החלטות נכונות.

בחנו מספר אלגוריתמים:

התחלנו עם אלגוריתם המינמקס שמותאם למשחק סכום אפס עם שני שחקנים. אותו הרחבנו להתמודד עם מספר משתנה של שחקנים. הגדרנו את היוריסטיקת ה-complex שבה ישתמש האלגוריתם. לאחר מכן התאמנו גם את אלגוריתם האלפא בטא לשחק במשחק מרובה משתתפים.

על מנת לבדוק את ביצועיהם של האלגוריתמים, הבנו כי אין טעם לשחק נגד שחקן רנדומי, ורצינו לאתגר אותם נגד שחקנים מורכבים יותר. לכן הגדרנו מספר שחקני baseline מולם יתמודדו.

ראינו כי ההיוריסטיקה שלנו בשילוב עם אלגוריתם המינמקס חזקה מאוד, ואנו מצליחים לנצח את שחקני ה-baseline בהסתברות גבוהה. את שחקן ה-offensive קל יותר לנצח מאשר את שחקן ה-defensive , והדבר הגיוני כי בעוד בניית רצף נצחון היא טקטיקה שמצריכה לפחות 4 מהלכים מחושבים ,חסימת רצף היא פעולה קלה בהרבה.

על מנת להשוות את עצמנו אל מול היוריסטיקות מעבודות קודמות, הבנו שאין טעם לשחק נגדן באופן ישיר, מפני שמהלכים של אלגוריתמי מינמקס דטרמיניסטיים כמעט לגמרי, ולכן נתנו לכל אחד מהסוכנים לשחק מול שחקני ה-baseline . ראינו שברוב המקרים הצלחנו להשיג ביצועים טובים יותר, ולפיכך יכולנו להסיק שההיוריסטיקה שלנו משקפת לאלגוריתם המינמקס מידע נכון יותר לגבי מצב הלוח.

ניתחנו את התנהגות המינמקס במשחק מרובה משתתפים ואת הנחות שחקן המקסימום לגבי שאר היריבים. הראינו כי כדאי לשקף לו הנחות כמה שיותר קרובות למציאות כדי להשיג תוצאות טובות במשחקים. עם זאת, גם במימוש שלנו, הנחתו על המציאות היא שהיריבים יעדיפו שלא לחסום את היריבים האחרים.

ראינו כי ככל שאנו מגדילים את עומק החיפוש שלנו, אנו משיגים תוצאות טובות יותר. עם זאת זמן הריצה גדל, ולכן ישנה חשיבות רבה להשתמש באלגוריתם האלפא בטא. במיוחד כאשר הלוחות גדלים (יחד עם מימד השלישי בלוח), אלפא בטא ביצע יותר גיזומים ולכן זמן הריצה קטן משמעותית ביחס לזמן הריצה של מינמקס. ראינו כי הגיזומים מתבצעים גם במשחק מרובה משתתפים.

לאחר מכן עברנו לממש את סוכן ה-Q-learning . סוכן זה היה מורכב הרבה יותר למימוש, שכן היינו צריכים לחשוב על גורמים רבים שיביאו לאימון מיטבי ביותר- סמלוץ משחק מרובה משתתפים, היריב שמולו יתאמן, פונקציית רווח שתגרום לו ללמוד נכון מצבים שונים. אפילו בחירת ייצוג הלוח, מכיוון שרצינו שהסוכן ייחשף לכמה שיותר מצבים ולכן עדיף לייצג את הלוח במימד קטן יותר, אך לא קטן מדי באופן שיוריד מידע חשוב מהלוח.

כל אלה היוו היפר פרמטרים שונים, שכל אחד השפיע על ביצועיו של הסוכן.

חידוש אחד שלנו במימוש סוכן זה, הוא בחירת המאסטר- ראינו כי סוכן שהתאמן עם סוכן חכם ששיחק עבורו, השיג ביצועים טובים יותר לעומת סוכן RL שבשלב האימון למד על בסיס החלטות רנדומיות בלבד.

מכיוון שהאימון עצמו היה מורכב ביותר, בחרנו להתמקד בתוצאות עבור סוכן שהתאמן בקונפיגורציה הקלאסית של המשחק, והוא השחקן שמשחק ראשון. אם נרצה לשחק בקונפיגורציה אחרת ו/או בתור אחר, נצטרך לאמן את הסוכן שוב בהתאם למשחק אותו נרצה שהוא ישחק. אנו מניחים כי ניתן היה להכליל את האימון של הסוכן ליותר ממצב משחק אחד, ע"י התאמת ייצוג מצבי המשחק. אולם התאמה שכזו הייתה עולה לנו בביצועי הסוכן.

בדקנו את ביצועיו של הסוכן אל מול שחקני ה-baseline, וראינו כי מול שחקני ה-defensive והאלפא בטא עם היוריסטיקת ה-complex בעומק 4, לא הצליח להשיג תוצאות טובות כל כך. ההערכה שלנו היא, שלמידע על לוחות שליליים לוקח הרבה יותר זמן לחלחל מטה מאשר לוחות בהם הסוכן קרוב לנצחון, ולכן הצליח להתמודד פחות טוב נגד יריבים שטובים בלחסום את המתמודדים שלהם.

לסיכום, נראה כי אלגוריתם מינמקס, על אף עקרון הפעולה הפשוט שלו, הצליח להשיג תוצאות טובות לעומת סוכן ה-RL , זאת בזכות פונקציית ההיוריסטיקה המורכבת שיצרנו עבורו.

ביבליוגרפיה:

- עבודה קודמת של סטודנטים על 4 בשורה, מספר 1 :
https://www.cs.huji.ac.il/course/2021/ai/projects/old/4InRow_1.pdf
- עבודה קודמת של סטודנטים על 4 בשורה, מספר 2 :
https://www.cs.huji.ac.il/course/2021/ai/projects/old/4InRow_2.pdf
- הסבר על המשחק 4 בשורה: https://en.wikipedia.org/wiki/Connect_Four
- 4 בשורה עם reinforcement learning
https://web.stanford.edu/class/aa228/reports/2019/final106.pdf