פרויקט Reinforcement Learning - סדנה מתקדמת בלמידה חישובית מעשית 67652

אוריין חרמוני 302170204

הערות:

- **יצוג הלוח** - עבור שני הסוכנים, בחרתי לייצג את הלוח באמצעות וקטור באורך 3 המייצג את הפרס המיידי בכל אחד המהלכים האפשריים , למשל המצב $(5, 0, -10)$ מייצג מצב בו אם הנחש יפנה שמאלה הוא ירוויח 5 נקודות, ימשיך ישר ירוויח 0 נקודות, ויפנה ימינה הוא יפסיד 10 נקודות. בחרתי לייצג כך את מצב המשחק כיוון שזהו היצוג המינימלי ביותר שעבד תחת אילוצי הזמנים של הקלאסטר, כל יצוג אחר (וכל חישוב נוסף שהייתי צריך לבצע כדי לעבור למרחב מצבים קטן יותר) עלה הרבה מבחינת זמני ריצה. ניסיתי יצוג של 4 וקטור שמייצג את הערך הממוצע של תזוזה ימינה/שמאלה/למעלה/למטה , ניסיתי לשטח את לוח המשחק , עבור הסוכן המיוחד ניסיתי גם לשרשר את הלוח הקודם ואת הלוח הנוכחי כדי לעבוד עם שכבות קונבולוציה שיוצרות field of view גדול יותר אך זה דרש יותר מדי זמן לעבור דרך רשת הנוירונים. לכן בחרתי את היצוג המינימלי הנ"ל, וראיתי כי אכן ניתן ללמוד איתו אך לא ביעילות. הביצועים על הקלאסטר היו כל כך גרועים (גם בבחירה חכמה של הקלאסטר לפי המייל שעמרי שלח לפני מספר ימים) שהתחברות בssh למכונה רגילה והרצה עליה הייתה עם פחות timeouts.
  - על מצב ופעולה אפשרית הוספתי גם את המצב הסימטרי - כלומר אם במצב $(5, 0, -10)$ המהלך האידאלי הוא תזוזה שמאלה , אזי המצב הסימטרי הוא $(-10, 0, 5)$ והמהלך האידאלי הוא תזוזה ימינה . הפרס של הסוכן זהה בשני המקרים.
- כמו שציינתי הקלאסטר היה בעייתי, עד מצב שהרבה פעמים הוא זרק אותי בזמן שחיכיתי למשאבים ואמר שאין מכונות פנויות (הוא הפנה אותי למכונות שונות מההגדרות ששמתי בגלל שהמכונות שהקציתם לנו היו תפוסות, אבל גם המכונות האחרות היו תפוסות והוא פשוט זרק אותי כי אין עליהן טנזורפלואו שמקומפל עם SSE שלא תואם את המכונה - למשל חיבר אותי הרבה פעמים לsulfur). לכן הרבה פעמים הייתי צריך להריץ על מכונה רגילה , מה שעלה לי בהרבה זמן יקר שיכולתי להשקיע בדיבוג. בנוסף, גם עם בחירת מכונות ספציפיות מתוך סדרת sm אני מוקצה למכונות אחרות, איטיות יותר (למשל sm-2 שלוקח לה 17 דקות להריץ סוכן לינארי) .

1 סוכן לינארי

עבור סוכן זה בחרתי לממש רשת נוירונים עם שכבת Dense בודדת , ללא אקטיבציה בסוף כיוון שפלט הרשת מייצג את ערך טבלת Q ואין אנו מחשבים את ההסתברות לבחור פעולה אלא את הערך שלה. עדכון הטבלה התבצע לפי הנוסחא הבאה (שנלקחה מההרצאה של שש"ש):

$$Q_{t+1}(s, a) = Q(s_t, a_t) - \alpha\left(r_t + \gamma \max_{a' \epsilon A} Q(s_{t+1}, a')\right)$$

כאשר α הוא קצב הלמידה ו γ הוא ה$discount\,factor$.

ניסיתי לבנות מודלים בכמה שיטות שונות , פעם באמצעות בניית מודל Sequential של קראס (בשיטה זו בחרתי לבסוף), פעם באמצעות יצירת מחלקה היורשת מKeras.Model וכתיבת פונ' אימון מיוחדת , ופעם נוספת על ידי עדכון ידני של טבלת numpy המייצגת את טבלת הQ . שתי השיטות האחרונות הניבו ביצועים ירודים למדי (במיוחד לאחרונה היו ביצועים די מחרידים, למרות שמבחינת משאבים היא דרשה הכי פחות מהמערכת). פונ' ההפסד בה השתמשתי כדי לקרב אותה היא MSE.

ב50 הסיבובים הראשונים בחרתי לעשות exploration של הלוח כדי להכיר אותו כמה שיותר . גיליתי שיותר סיבובים מכך הסוכן יכול באותה מידה תמיד לבחור פעולה באקראי , ופחות מכך לא משפיע במיוחד על פעולת הסוכן כיוון שלקלאסטרים לוקח המון זמן לטעון את המודל וכל פעולת פרדיקציה לא תסתיים בזמן.

עשיתי חיפוש פרמטרים יסודי , וגיליתי ש $batches$ בגודל 8 הם הגודל המקסימלי שמאפשר למודל ללמוד ללא הפרת פרמטרים של זמנים , קצב למידה של 0.01 (אופטימייזר אדאם), גודל $replay\,buffer$ של 128 מצבים,$discount\,factor$ של 0.01 , ואפסילון (אחוז המהלכים האקראיים) 0.05.

סוכן זה

דוגמא להרצה שבה הסוכן הפסיד:

```
game_id,player_id,policy,score
140173570789560,0,Linear,-0.0100
140173570789560,1,Avoid,0.0820
140173570789560,2,Avoid,0.1120
140173570789560,3,Avoid,0.0540
140173570789560,4,Avoid,0.0000
```

במקרה זה סוכן 1 הוא הסוכן שלי , סוכנים 5-2 הם סוכני Avoid עם אפסילונים $P2 = 0, P3 = 0.05, P4 = 0.1, P5 = 0.2$.

דוגמא נוספת להרצה שבה הסוכן ניצח מול אותם סוכנים:

```
game_id,player_id,policy,score
139902649245256,0,Linear,0.0880
139902649245256,1,Avoid,0.0960
139902649245256,2,Avoid,0.0920
139902649245256,3,Avoid,0.0290
139902649245256,4,Avoid,0.0110
```

בריצה של 5000 סיבובים הסוכן נמצא רוב הזמן עם פרס שלילי אך עולה מונוטונית , כאשר בכמה מאות הסיבובים האחרונים הפרס נהיה חיובי. אף פעם לא הצלחתי לנצח את P2/P3 אך התקרבתי מאוד לניקוד שלהם רוב הזמן, ובערך חצי מהזמן אני מנצח את P4 ואת P5.

2 סוכן מיוחד

עבור סוכן זה בחרתי גם לממש רשת נוירונים , ללא אקטיבציה בסוף כיוון שפלט הרשת מייצג את ערך טבלת Q ואין אנו מחשבים את ההסתברות לבחור פעולה אלא את הערך שלה. עדכון הטבלה התבצע לפי אותה נוסחא כמו בסוכן הלינארי ואותה פונ' הפסד. ניסיתי כאן כמה ארכיטקטורות שונות, שכל אחת מהן הניבה תוצאות פחות מזהירות (כמעט תמיד נגמר בפרס ממוצע של כ-1.3 , כל כך נמוך שזה נראה כאילו הרשתות לומדות רק את המהלכים ה**לא נכונים**). הארכיטקטורה אותה בחרתי לבסוף היא:

$$
\begin{aligned}
& Dense256ReLU \\
& \rightarrow Conv2D16Ch(stride1, kernel3x3) \\
& \rightarrow Conv2D32Ch(stride1, kernel5x5) \\
& \rightarrow Dense256ReLU \\
& \rightarrow Dense\,(NUM\_ACTIONS)\,Softmax
\end{aligned}
$$

ניסיתי כאן כמה ארכיטקטורות שונות, שכל אחת מהן הניבה תוצאות פחות מזהירות (כמעט תמיד נגמר בפרס ממוצע של כ-1.3 , כל כך נמוך שזה נראה כאילו הרשתות לומדות רק את המהלכים ה**לא נכונים**).

ארכיטקטורה נוספת שניסיתי (אותה מצאתי באינטרנט אחרי המון נסיונות נואשים לבנות ארכיטקטורה משלי) :

$$
\begin{aligned}
& Dense120ReLU \rightarrow Dropout\,(0.15) \\
& \rightarrow Dense120ReLU \rightarrow Dropout\,(0.15) \\
& \rightarrow Dense120ReLU \rightarrow Dropout\,(0.15) \\
& \rightarrow Dense\,(NUM\_ACTIONS)(both\,with\,softmax\,activation\,and\,no\,activation)
\end{aligned}
$$

הבחירה לממש באמצעות רשת נוירונים את סעיף זה נבעה בעיקר מקוצר בזמנים , התחלתי מאוחר יחסית את הפרויקט ולא נשאר לי מספיק זמן כדי לנסות פתרונות ושיטות אחרות . ניצלתי את העובדה שכתבתי תשתית שעובדת במקרה הלינארי והרחבתי את הרשת לארכיטקטורה עמוקה יותר ו (בתקווה) יותר חכמה. הבחירה החכמה ליצוג מצב הלוח הייתה להשתמש בלוח השלם ולהכניסו ישר לרצף של שכבות קונבולוציה (כדי לקבל field of view גדול ע"מ לקבל החלטות חכמות יותר) אך זה גרר קשיים מבחינת ביצועים על הקלאסטר.

לצערי, לא הגעתי למצב שבו הסוכן שלי מצליח ללמוד כמו שצריך עם אף ארכיטקטורה , כנראה בגלל שילוב של מרחב המצבים דל , ארכיטקטורה והיפר פרמטרים לא מתאימים, וזמני ריצה חריגים. כאשר הרצתי את הסוכנים שלי מול עצמם הם תמיד סיימו עם ערכים חיוביים , כלומר כולם הצליחו ללמוד בהצלחה רוב הזמן (זה נכון עם כמעט כל היפר פרמטר שניסיתי) , אך הם לא הצליחו להתמודד עם סוכני Avoid משום מה (אפילו עם סוכנים בעלי אפסילון גבוה מ0.2 היו מנצחים אותם הרבה). לכן אני מכתיר את הסוכן המיוחד שלי בתור כישלון חרוץ , כיוון שלא עמדתי בתנאים של ההגשה :(אציין לעומת זאת כי הסוכנים רובם ככולם כן מצליחים ללמוד באופן כלשהו, וכי ניתן לראות בפלט שלהם כי הפרס הממוצע במגמת עליה לאורך כל הריצה אך העליה כל כך מינורית שהפרס שלהם תמיד שלילי .

ב50 הסיבובים הראשונים בחרתי לעשות exploration של הלוח כדי להכיר אותו כמה שיותר . גיליתי שיותר סיבובים מכך הסוכן יכול באותה מידה תמיד לבחור פעולה באקראי , ופחות מכך לא משפיע במיוחד על פעולת הסוכן כיוון שלקלאסטרים לוקח המון זמן לטעון את המודל וכל פעולת פרדיקציה לא תסתיים בזמן.

עשיתי חיפוש פרמטרים , הבחירה הכי מוצלחת שמצאתי היא של $batches$ בגודל 16 הם הגודל המקסימלי שמאפשר למודל ללמוד ללא הפרת פרמטרים של זמנים , קצב למידה של 0.0005 (אופטימייזר אדאם), גודל $replay\,buffer$ של 128 מצבים,$discount\,factor$ של 0.9 , ואפסילון (אחוז המהלכים האקראיים) 0.1.

ריצה של הסוכן שלי מול סוכני Avoid עם אפסילונים $P2 = 0, P3 = 0.05, P4 = 0.1, P5 = 0.2$ (למשך 50000 סיבובים):

```
game_id,player_id,policy,score
140145080156344,0,Custom,-0.0470
140145080156344,1,Avoid,0.1760
140145080156344,2,Avoid,0.0910
140145080156344,3,Avoid,0.0490
140145080156344,4,Avoid,0.0390
```

ריצה זו יחסית מוצלחת באופן מפתיע, אך זהו outlier ובדרך כלל המודל שלי מקבל ערכים בין -1.3 ל-1.4 . לא צירפתי ריצות נוספות כיוון שזה היה חסר פואנטה , כולן תמיד מפסידות ... *הכנס סמיילי עצוב נוסף*