گزارش کار

یادگیری ماشین

تمرین اول

آرمین خیاطی

9931153

پویان انصاری راد

9960565

فهرست

مقدمه

بازی دوز یک بازی قدیمی دو نفره است که در یک صفحه 3 در 3 انجام میشود و هر بازیکن نماد خود (دایره/ضربدر) خود را دارد و باید بتواند نماد های خود را در یک سطر یا ستون یا بصورت قطری در این صفحه بازی قرار دهد تا برنده بازی شود. اگر هیچ یک موفق به اینکار نشوند بازی مساوی میشود.

در این تمرین ما با پیاده سازی الگوریتم تخمین تابع برای این بازی سعی میکنیم یک Agent برای این بازی تربیت کنیم. همانطور که در کتاب تام میشل نیز ذکر شده است، بازی دوز یک مسئله یادگیری well-posed هست که بصورت کامل تر در بخش بعدی توضیح خواهیم داد.

سیستم یادگیری بازی دوز

ایده اصلی پشت این سیستم یادگیری این است که سیستم باید بتواند عملکرد (Performance) (P) خود را بر اساس مجموعه ای از Task ها (T) به وسیله یادگیری از Training Experience (E) ها ارتقا دهد. این تجربیات یادگیری (E) میتوانند بصورت مستقیم یعنی با استفاده از یک سری مجموعه داده برچسب گذاری شده از قبل تهیه شده و یا بصورت غیر مستقیم یعنی نمونه های آموزشی بدون برچسب باشند. در مورد مسئله ما خواهیم داشت:

- Task (T) : بازی کردن دوز
- Performance (P) : درصد بازی های برنده شده ی Agent در مقابل انسان

- Experience (E) : بازخورد غیر مستقیم از طریق تاریخچه بازی که هنگام بازی کردن کامپیوتر با خودش ایجاد میشود.

یادگیری از روی تجربه ها (Experience)

بصورت کلی یک تابع هدفی را ما باید یاد بگیریم که در هر حالت از صفحه بازی، بهترین حرکت بعدی را پیشنهاد دهد. در اینجا تابع ما یک تابع خطی هست که یک وضعیت بازی را دریافت و یک مقدار حقیقی یعنی امتیاز را خروجی میدهد.

- V(boardState) → R، که R امتیاز آن حالت از بازی است.
- V_hat(boardState) ←(W.T) * X، که W وزن های تابع هدف و X ویژگی های استخراج شده از صفحه بازی هستند.
- برای آپدیت وزن ها نیز از فرمول زیر استفاده میکنیم.

$$W_{i_{new}} = W_{i_{old}} + lr * (V_{hat}(state) - V_{hat}\big(Successor(state)\big) * X$$

پیاده سازی

بر اساس کتاب تام میشل، طراحی نهایی به چهار ماژول تقسیم میشود.

1. Experiment Generator : وظیفه این ماژول تولید یک مسئله جدید در ابتدای هر Epoch میباشد که در این مسئله یک صفحه بازی خالی است.
2. Performance System : وظیفه این ماژول، دریافت یک صفحه بازی تولید شده توسط Experiment Generator و استفاده از یک الگوریتم یادگیری برای تولید یک دنباله جواب برای مسئله در هر

Epoch می باشد. برای مسئله ما این ماژول کارش را با شبیه سازی یک بازی بین دو بازیکن انجام میدهد. این بازیکن ها تصمیمات خود برای انتخاب حرکت بعدی را با استفاده از تابع هدف کنونی میگیرند.

3. Critic : این ماژول یک دنباله جواب را از ماژول قبلی میگیرد و به عنوان نمونه های آموزشی به ورودی ماژول Generalizer میدهد.

Training Examples ←[<Features(boardState) , R >,<>….]

4. Generalizer : این ماژول با استفاده از نمونه های آموزشی دریافت شده در مرحله قبل، تابع هدف را آپدیت میکند.

برای تابع هدف این مسئله ما دوازده ویژگی و یک بایاس در نظر میگیریم به ترتیب زیر:

1. Bias : X0
2. X1 : وجود تنها یک ضربدر در یک ردیف یا یک ستون خالی
3. X2 : وجود تنها یک دایره در یک ردیف یا یک ستون خالی
4. X3 : وجود دو ضربدر پشت هم در یک ردیف
5. X4 : وجود دو دایره پشت هم در یک ردیف
6. X5 : وجود سه ضربدر در یک ردیف
7. X6 : وجود سه دایره در یک ردیف
8. X7 : وجود یک ضربدر در یکی از دو قطر اصلی یا فرعی که خالی است.
9. X8 : وجود یک دایره در یکی از دو قطر اصلی یا فرعی که خالی است.
10. X9 : وجود دو ضربدر در یکی از دو قطر اصلی و فرعی
11. X10 : وجود دو دایره در یکی از دو قطر اصلی و فرعی
12. X11 : وجود سه ضربدر در یکی از دو قطر اصلی و فرعی

13. X12 : وجود سه دایره در یکی از دو قطر اصلی و فرعی

پیاده سازی ما به صورتی هست که علاوه بر یادگیری در هنگام بازی کامپیوتر با خودش، هنگام بازی کامپیوتر با انسان نیز مدل ارتقا پیدا کرده و قوی تر میشود.

شرح کد

بعضی از توابع استفاده در بعضی از کلاس ها تکراری هستند و به همین دلیل فقط به شرح توابع غیر تکراری میپردازیم.

کلاس ExperimentGenerator

تابع generateNewProblem

یک صفحه بازی خالی تولید میکند.

کلاس Player

کلاس بازیکن شامل تمام توابع مورد نیاز برای بازی می باشد.

تابع isGameOver

اتمام بازی را کنترل میکند.

تابع lookForLegalMoves

لیستی از حرکت های مجاز در هر وضعیت تخته بازی برمیگرداند.

تابع extractFeatures

فیچر ها در هر وضعیت تخته بازی را محاسبه میکند.

تابع boardPrint

صفحه بازی را چاپ میکند.

تابع calculateNonFinalBoardScore

امتیاز هر وضعیت از صفحه بازی را محاسبه میکند.

تابع chooseMove

بهترین حرکت از بین حرکت های مجاز در هر وضعیت از صفحه بازی را میابد.

تابع chooseRandomMove

یک حرکت از بین حرکت های مجاز هر وضعیت از صفحه بازی بصورت رندم انتخاب میکند.

کلاس PerformanceSystem

مسیر انتخاب تمام وضعیت های صفحه بازی داده شده تا اتمام بازی را می یابد.

تابع generateGameHistory

مسیر انتخاب تمام وضعیت های صفحه بازی تا اتمام بازی را تولید میکند.

کلاس Critic

کلاس تولید نمونه های آموزشی براساس سابقه بازی می باشد.

تابع calculateFinalBoardScore

امتیاز وضعیت پایانی صفحه بازی را محاسبه میکند.

تابع generateTrainingSamples

نمونه های آموزشی را بر اساس سابقه بازی تولید میکند.

کلاس Generalizer

کلاس بهبود وزن های تابع هدف می باشد.

تابع lmsWeightUpdate

تابع محاسبه وزن های جدید به روش Least Mean Squares می باشد.

نتیجه

بخش اول : کامپیوتر با کامپیوتر

نتایجی که بعد از آموزش مدل در 1000 تکرار برای دو بازیکن به دست می آید به صورت زیر می باشد:

First Player Training Results: (Player Wins = 765, Player Lose = 148, Game Draws = 87)

Second Player Training Results: (Player Wins = 727, Player Lose = 164, Game Draws = 109)

وزن های به دست آمده برای 13 ویژگی به همراه بایاس متعلق به هر بازیکن نیز به شرح زیر می باشد:

First Player Final Learned Weight Vector:

[71.44560870863126- ,245.17811003388516 ,111.95797585816914,

10.734171255002673 ,0.6862067591637899 ,42.530228819088805 -,

132.06480838648295- ,127.9042291708675- ,15.22818111721062,

13.747848875988776- ,57.90152128425417 ,54.08550436965859 -,

119.14923430487816]

Second Player Final Learned Weight Vector:

[6.9798029117994425- ,208.70100142657418 ,12.540660929730969,

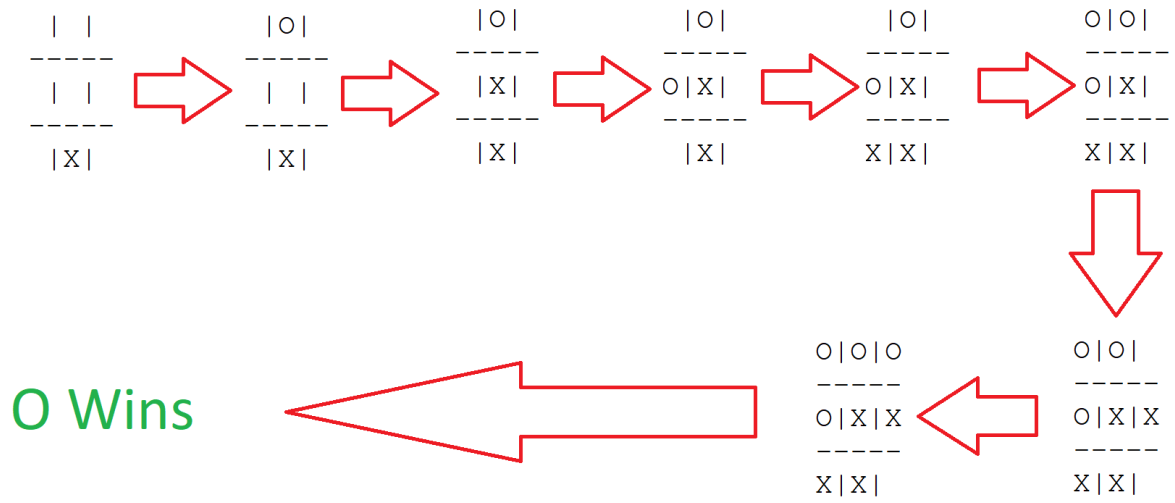
28.112940430297865- ,15.059685629438865 ,97.78743124495205 -,

98.20712597701547 ,7.9649989252153475- ,75.19655283619173,

41.55345685130794- ,22.905654125448994 ,90.86416135654544 -,

73.84098057221422]

نمونه ای از بازی کامپیوتر با خودش را در زیر مشاهده میکنید.



بخش دوم : کامپیوتر با انسان

همانطور که گفتیم مدل در هنگام بازی با انسان نیز میتواند فرایند یادگیری را طی کرده و وزن ها را آپدیت کند. در زیر فرایندی که بعد از انجام یک بازی با کامپیوتر طی شده و خود کامپیوتر نیز برا نماد ضربدر برنده شده است را میتوانید مشاهده کنید.