Car Racing GYM environment

2020.1.28

Mahsa Ghaderan & Hossein Alipour
Iran University of Science and Technology
Artificial Intelligence project
DR. Pillevar

Overview

پروژه ی CarRacing از محیط GYM را با استفاده از الگوریتم های OpenCV , Reinforcement Learning , شبکه عصبی پیاده سازی کردیم. اگر در این مسئله در ۷۳۲ فریم از همه trackها عبور کرده باشیم ۱۰۰۰ امتیاز کسب میکنیم که ماکسیمم جواب مسئله میباشد. به طور کلی مسئله دارای سه نوع اکشن گاز ترمز و فرمان است که بازی هرکدام به ترتیب [0,1] , [0,1] , [1,1-] می باشد. محاسبه ی تمام اکشن ها غیر ممکن است و بهترین راه تبدیل کردن استیت ها به اعداد ثابت می باشد.

Goals

هدف پیاده سازی الگوریتم Renforcement learning(RL می باشد به طوری که ماشین موجود در صفحه ی بازی بتواند مسیر مشخص شده را در سریع ترین زمان ممکن طی کند و از زمین خارج نشود.

Description

در ابتدا لازم است برای ارتباط با محیط بازی عکس های به فرمت RGB هستند که در هر فریم در اختیار برنامه قرار میگیرد. ما برای پیش پردازش عکس ها از OpenCV استفاده کردیم. هر عکس را در سه مرحله پردازش میکنیم.

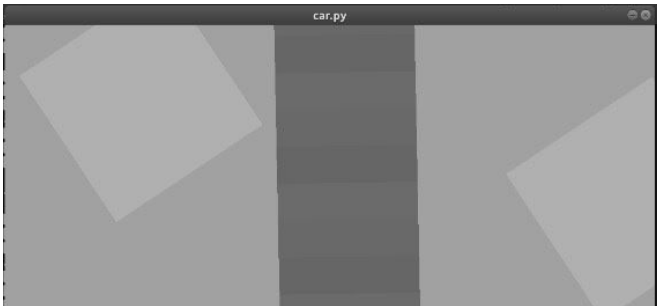
1. پیدا کردن و جدا کردن ناحیه ی مورد نظر از عکس
2. تبدیل عکس ها به فرمت gray scale
3. تبدیل عکس ها به فرم باینری برای تشخیص شی مشخص

هر عکس ورودی برنامه به سه قسمت تقسیم کردیم:

1. Game board 2. State fetch(control base) 3.car position(CarField)

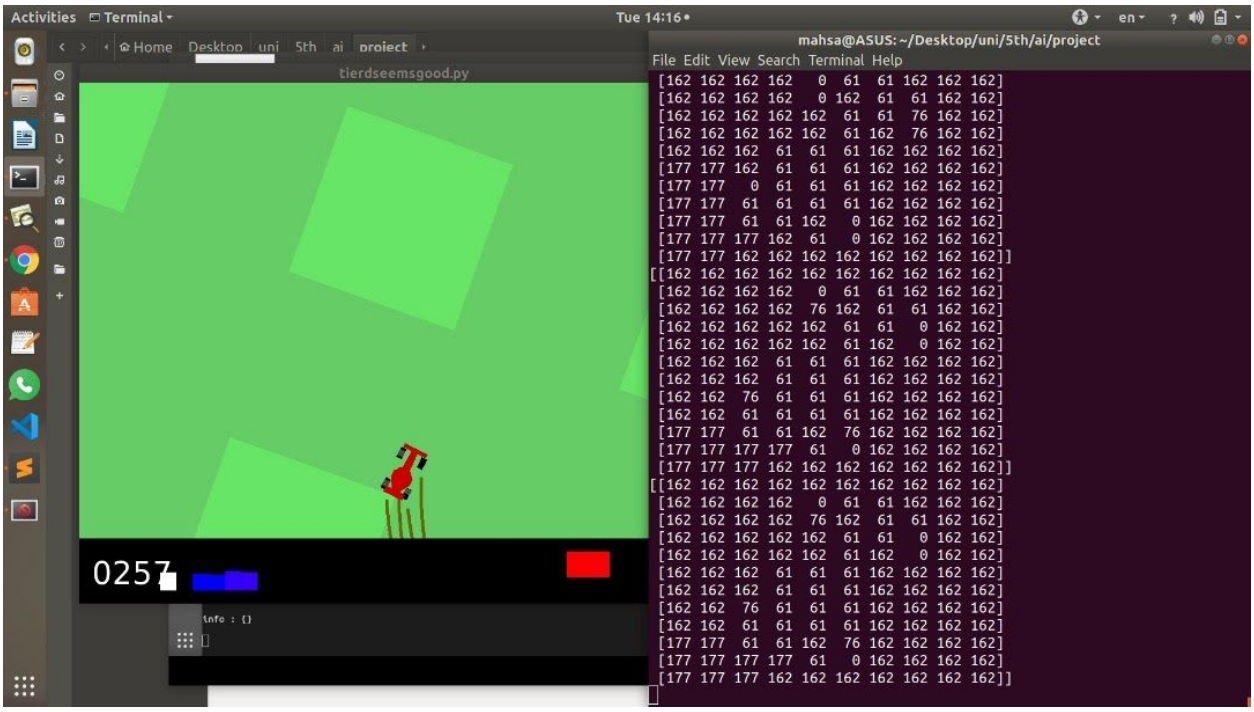
- قسمتی که شامل مسیر و چمن هاست را به صورت سیاه و سفید از نوع np یا مقدار های ۰ و ۱ تبدیل کردیم همان طور که در عکس قابل مشاهده است رنگ جاده تیره تر از چمن ها می باشد و برای تبدیل gray scale به ۰و ۱ باید عددی را به عنوان استانه در نظر میگرفتیم و طبق ازمون و خطا عدد ۱۲۰ مناسب بود.

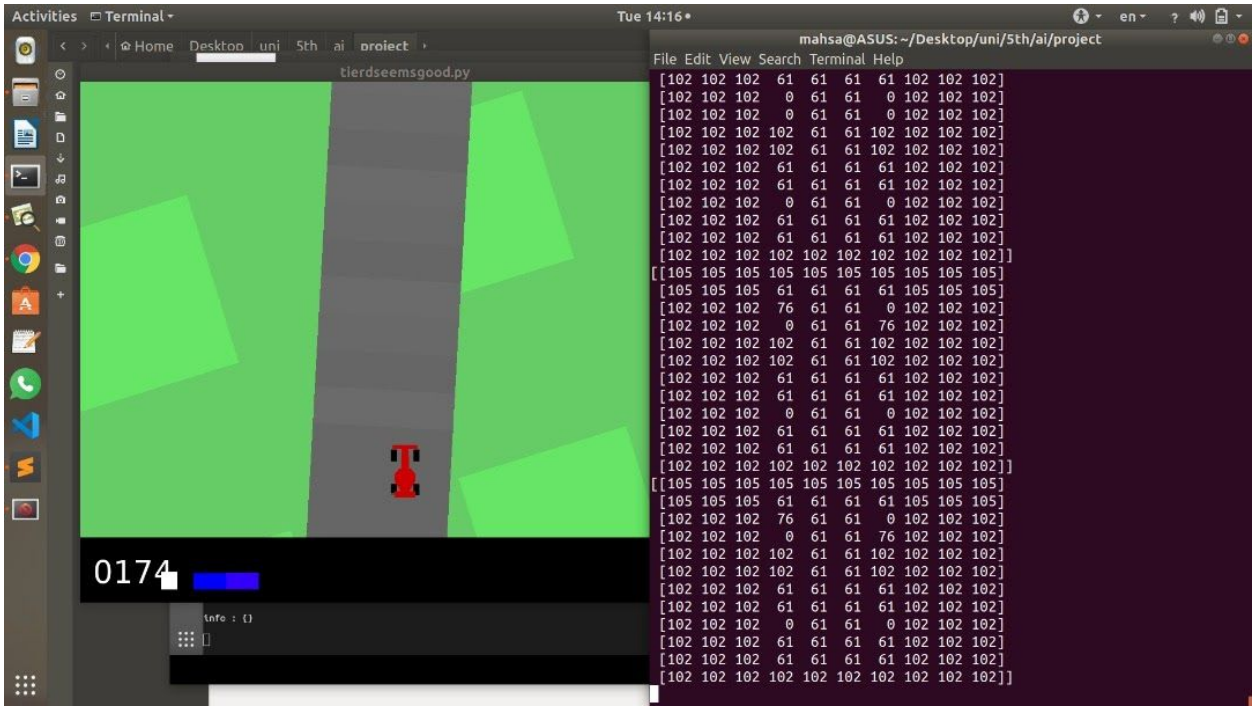
تبدیل عکس به gray scale-*

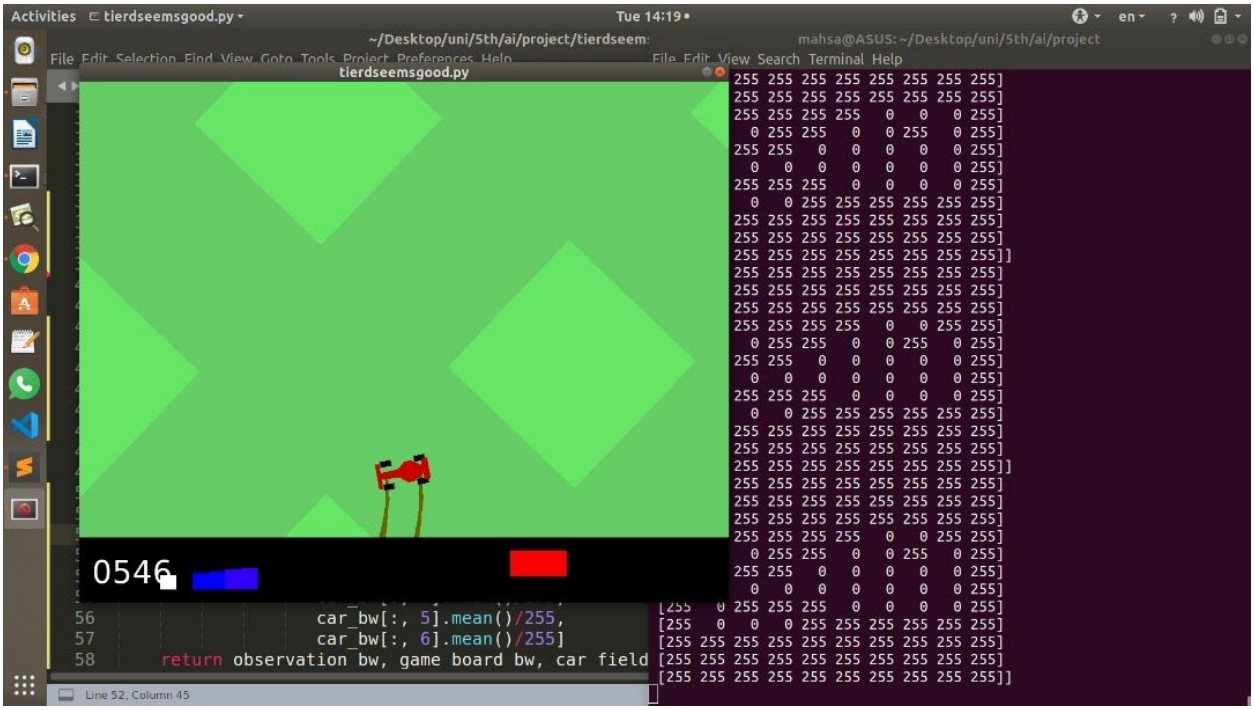


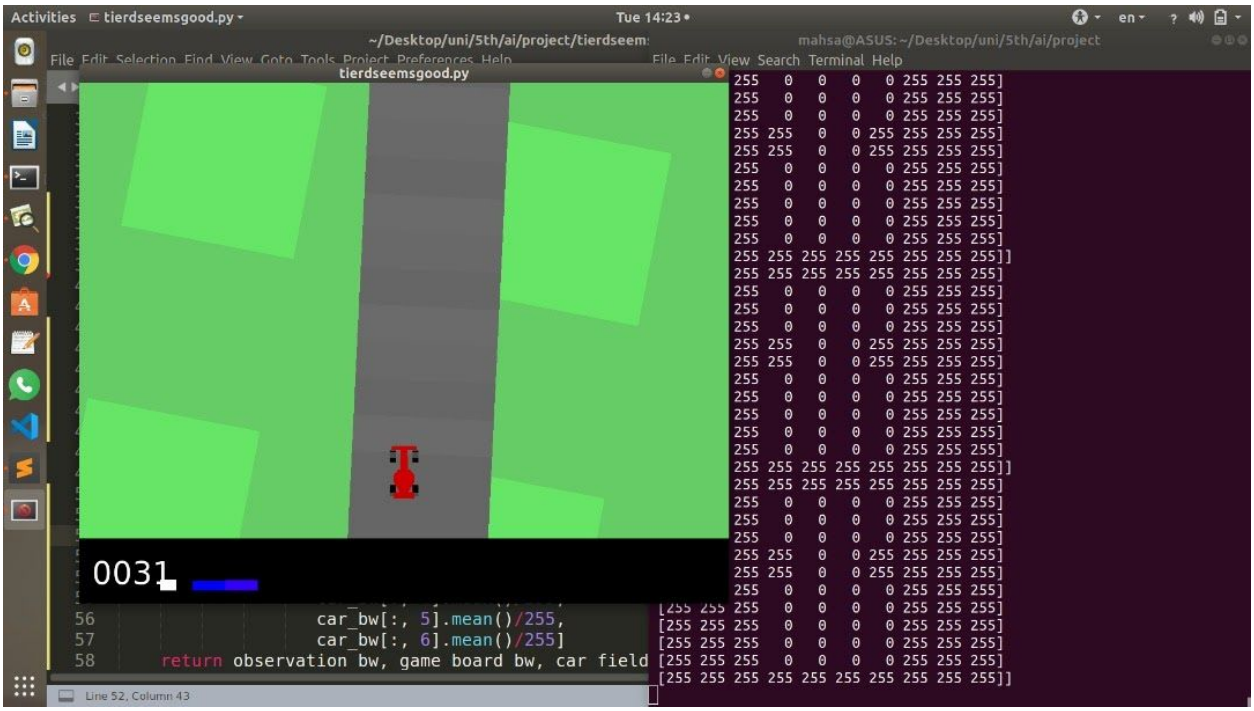
- در state fetch نوار پایین بازی را به صورت آرایه ی ۰ و ۱(سیاه و سفید) تبدیل کردیم. و چون مطمئن بودیم که پس زمینه ۰ است همه ی مقادیر به جز ۰ را به ۱ map کردیم. سپس با به دست آوردن مقادیر مختلف از جمله وضعیت فرمان و سرعت و مقدار لیز خوردن چرخ ها.
- بخش آخر که مربوط به خود ماشین است هم ابتدا به صورت gray scale درآوردیم و بعد آن را به صورت مطلق ۰ و ۱ تبدیل کردیم. این طوری می توان استیت ماشین را به راحتی تشخیص داد. عکس های زیر مربوط به فرایند تبدیل rbg به 0,1 است.

برای مثال در عکس قسمت ماشین مقدار gray scale آن به طور متوسط ۶۵ است و طبق مشاهده ی چند آزمایش عدد ۷۸ برای جداسازی ماشین از صفحه ی زیرش را انتخاب کردیم.




Activities
Terminal
Tue 14:16
mahsa@ASUS: ~/Desktop/uni/5th/ai/project
File Edit View Search Terminal Help
tierdseemsgood.py
0174


Activities
tierdseemsgood.py
Tue 14:19
mahsa@ASUS: ~/Desktop/uni/5th/ai/project
File Edit View Search Terminal Help
tierdseemsgood.py
0546
car_bw[:, 5].mean()/255,
car_bw[:, 6].mean()/255]
return observation_bw, game_board_bw, car_field
Line 52, Column 45



تبدیل عکس ها به آرایه های ۰ و ۱ (مربوط به خود carField)*-

در مرحله ی بعد اکشن ها را به صورت گسسته تعریف کردیم. در ابتدا با ۴ اکشن شروع به کار کردیم. اکشن ها شامل :

۱. فقط به جلو (گاز) ۲.فقط ترمز ۳.فقط به چپ ۴. فقط به راست

تعداد کمتر اکشن ها باعث شد تا عامل ما سریع تر train کند و همین باعث شد تا زودتر به امتیاز های بالا و نتیجه ی قابل مشاهده برسد(کمتر از ۶۰۰ اپیزود). هرچند عامل ما در بازی کردن به علت کم بودن اکشن ها مشکلاتی داشت. برای مثال چون همزمان نمی تواند گاز بدهد و به راست یا چپ بپیچد نمیتواند از این روش امتیاز بیشتری را کسب کند. زیرا هر چه عامل سرعت بیشتری داشته باشد امتیاز بیشتری را میتواند کسب کند.

به همین منظور متد دیگری را برای پیاده سازی در نظر گرفتیم . به این صورت که با یازده اکشن پیاده سازی کردیم. اکشن ها مانند حالت اول فقط جلو و فقط ترمز را داشتند. ولی برای فرمون ماشین مقادیر مختلفی در نظر گرفتیم. این کار باعث شد تا زمان train عامل بیشتر شود و بعد از گذراندن حدود ۲هزار اپیزود ...

روش سومی که برای بدست آوردن اکشن های بازی در نظر داشتیم تا پیاده سازی کنیم به این صورت است که برای یک سری اکشن خاص مقدار های خاصی نیز در نظر بگیریم.

۱. فقط به جلو (گاز با سرعت ۰.۵) ۲.فقط ترمز ۳.فقط به چپ ۴. فقط به راست ۵.گاز بدهد و ترمز کند
۶.گاز بدهد به راست برود (با سرعت ۰.۵)

۷.گاز بدهد و به چپ برود(با سرعت ۰.۵) ۸.گاز بدهد و به راست برود (با سرعت ۰.۳) ۹.گاز بدهد و به چپ برود (با سرعت ۰.۳) ۱۰. ترمز راست ۱۱.ترمز چپ

متاسفانه به علت اینکه نتوانستیم پروژه را روی Google Colab بالا بیاوریم نتوانستیم از این روش خروجی ای داشته باشیم.

مشکل ما در بالا آوردن پروژه در Google Colab نصب نشدن wheel2 در هنگام نصب کردن ماژول box-2D بود. :(