تمرین هفتم - یادگیری ماشین - بخش عملی

دانیال ملک محمد-۹۴۱۰۰۰۹۲

سوال۲)

من چند فرض را قبل از سوال در نظر گرفتم:

۱) وقتی از یک خانه ی Freeze به خانه ی Goal می رویم ، در این فضا reward=20 در این حین داده می شود اما من این را پیش فرض گرفتم که این reward=0 باشد. به جای آن وقتی در Goal هستیم هر حرکتی انجام دهیم ، ۲۰ reward گرفته می شود و بازی خاتمه می یابد. این تغییر کوچک باعث می شود وقتی value های خانه ها بعد از اجرای الگوریتم هایی مثل policy iteration و value iteration ،به دست می آیند، خانه ی Goal بیشترین مقدار را داشته باشد و هنگام انتخاب بهترین حرکت بر اساس value ها، agent سعی کند به خانه ی Goal برود. این در حالی است که در حالت قبلی ، مقدار value خانه ی goal صفر می ماند اما خانه های مجاور آن مقدار مثبت زیادی خواهند گرفت و لذا وقتی agent به این خانه های مجاور goal می رسد، به جای اینکه به goal برود، سعی می کند در آن خانه های مجاور رفت و آمد کند چون Value زیادی دارند.
من اینکار را برای خانه های Hole هم انجام دادم اگرچه در این جا فقط یک reward برای رسیدن به goal وجود دارد.

```
to_freeze_reward = 0
hole_to_end_reward = 0
goal_to_end_reward = 20
```

۲) این سه متغییر در کد اثر می گذارند و در ابتدای فایل تعریف شده اند تا بتوانید تغییرشان دهید.

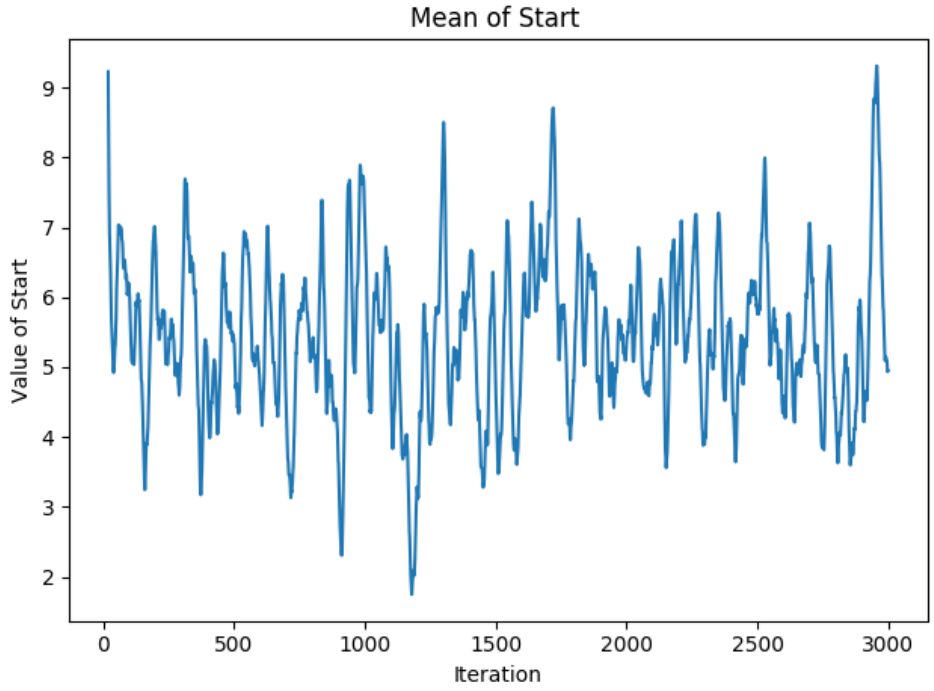
۳) از اپیزود ۱۹ به بعد، هر ۲۰ اپیزود قبلی میانگین و واریانس گرفته می شدند و در زیر نمودارشان آورده شده است.

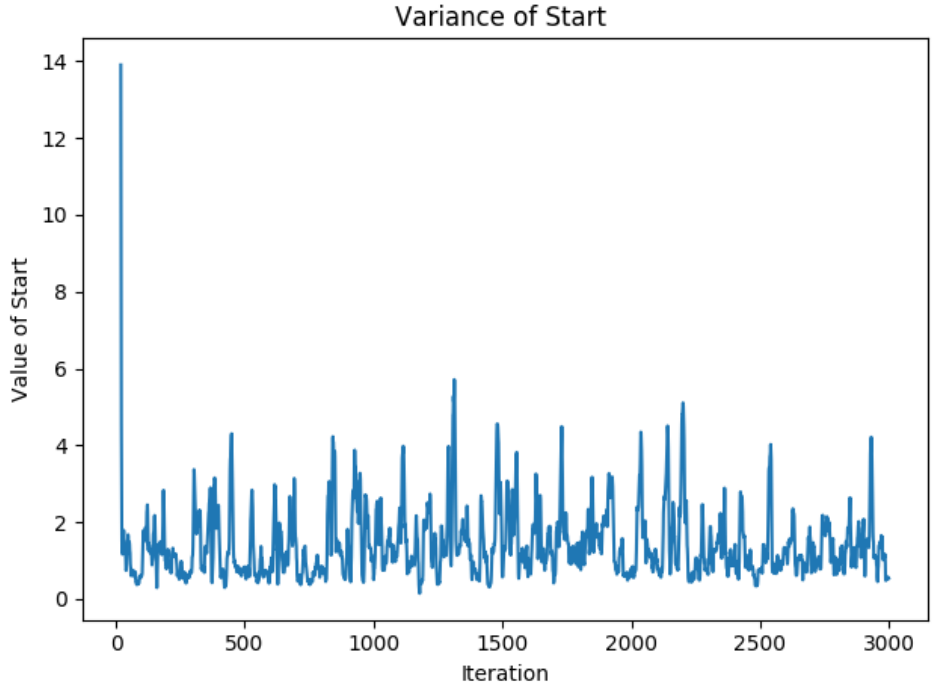
۴) درپایان کد، در یک حلقه ی Whileبی نهایت وجود دارد که بازی را طبق پالیسی انجام می دهد. در ابتدای این قسمت،متغییرPolicy_type وجود دارد که مطابق کامنتش،اگر می خواهید بازی طبق پالیسی Policy Iteration اجرا شود،learner1 ، مطابقalpha_MC، Learner2 و اگر می خواید مطابق TD اجرا شود، به Learner3مقدار دهی شود.

5)

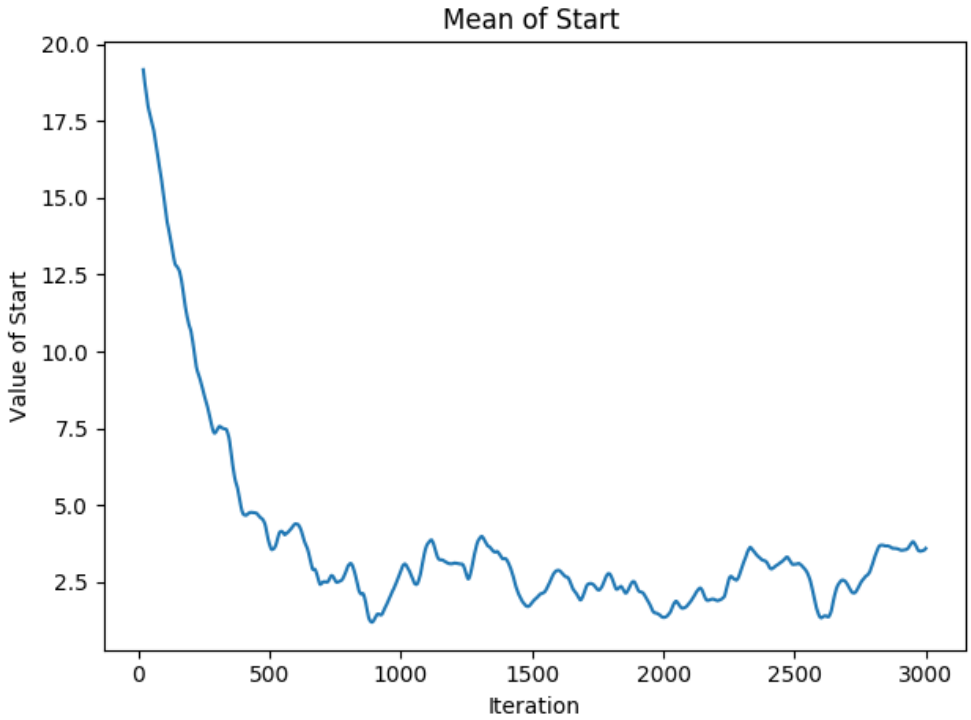
طبق نمودار های زیر واضح است که همگرایی واریانس و میانگین در طول Iteration ها در روش TD بهتراست.همچنین بازه ی تغییرات واریانس که در آن بالا پایین می رود را هم در نظر بگیرید.

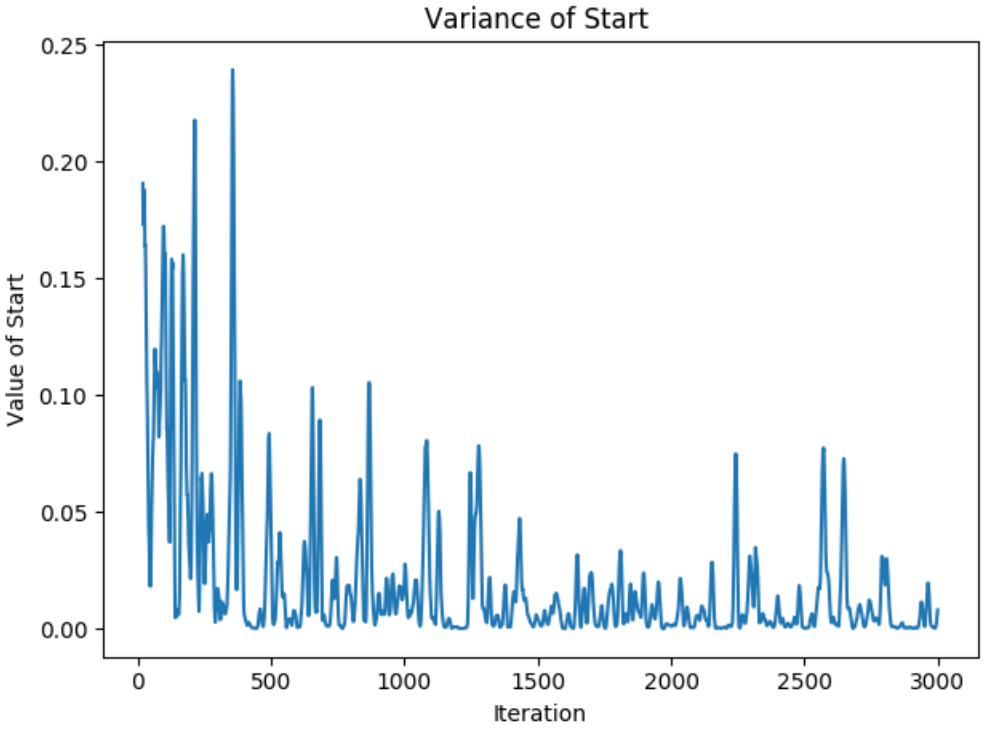
نمودار مربوط به روشAlpha Monte Carlo:





نمودار مربوط به روش Temporal Difference:





سوال سوم)

تابع statTOstate خروجی step را که چهار عدد x,xdot,theta,thetadot است را به یک عدد که شماره ی state گسسته است، تبدیل می کند.تعداد این state های گسسته ی را ۹۶ تا گرفتم. تبدیل حالات پیوسته به گسسته توسط تابع discritizer(x, portion)
انجام می گیرد که portion یک لیست از مقادیری مرتب است که خروجی این تابع تعیین می کند که مقدار x بین کدام دو مقدار از لیست portion است. فرض کنید portion ، k عضو دارد، خروجی عددی از صفر تا k+1 است.
مطابق داکیومنت، نتوانستم حداکثر تعداد step های episode را کنترل کنم، لذا به صورت دستی این مقدار را کنترل کردم و در ابتدای کد مقدارش قابل تغییر است.
max_episode_steps = 500

دو تابع Train و Play مطابق اسمشان تعریف و پیاده سازی شده است .