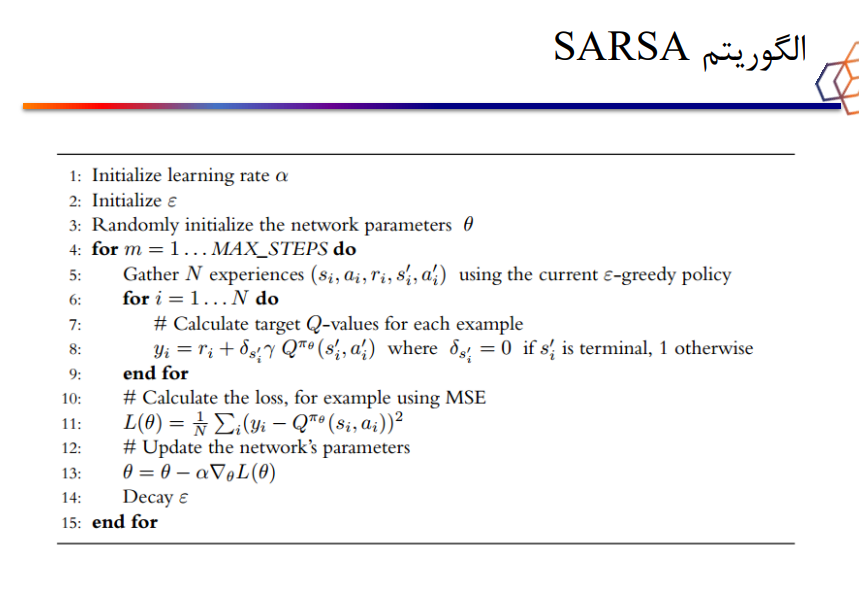
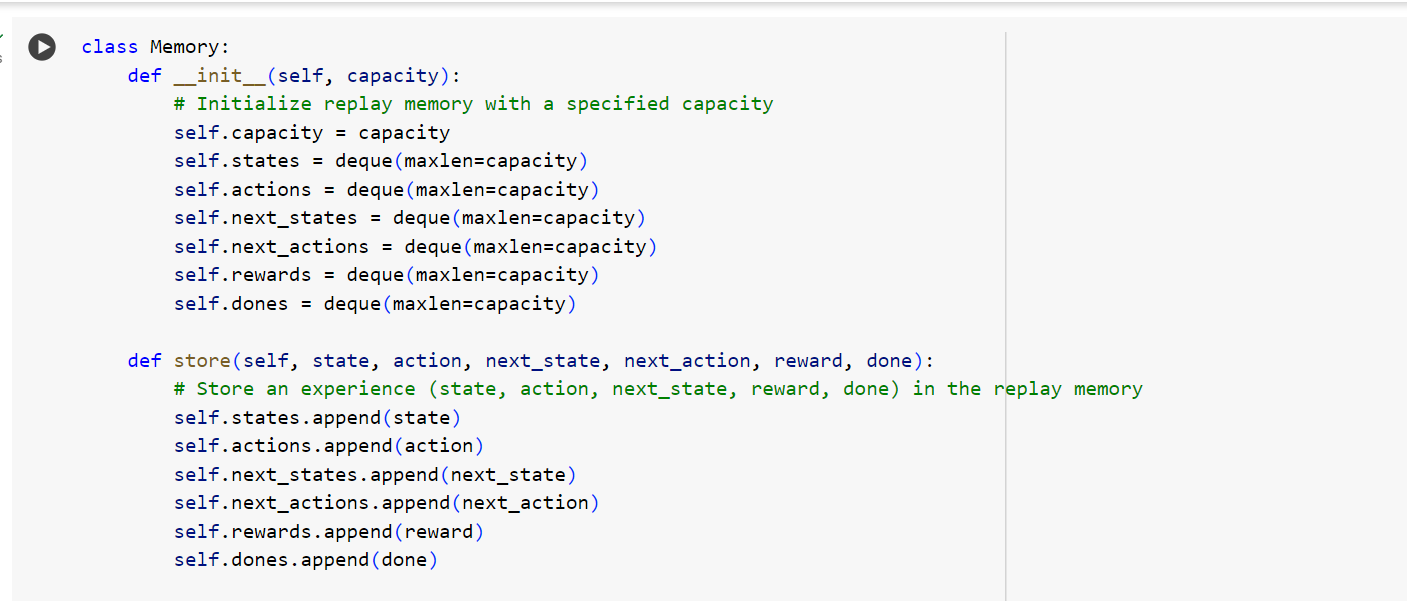
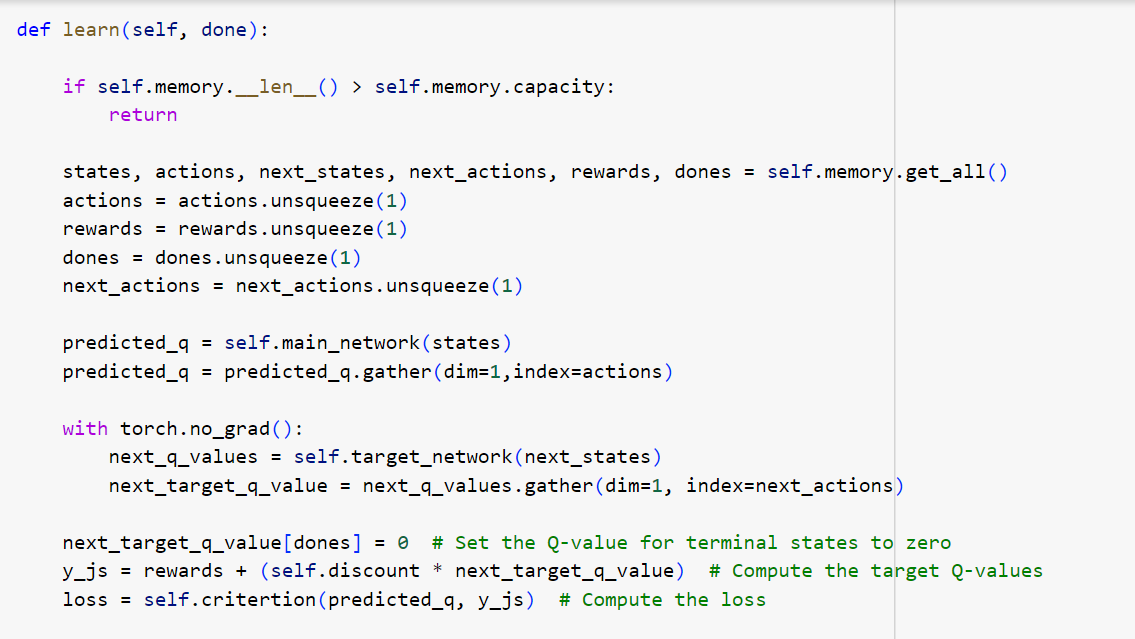
ابتدا SARSA را هم توضیح میدهیم سپس نمودار ها را بررسی میکنیم:  
**SARSA:**همانند انچه در این الگوریتم دیده میشود، دیگر از replay\_momory نباید استفاده کرد. صرفا تعدادی state, action, reward, next state, next action را جمع آوری میکنیم، و آن ها را با شبکه اصلی و تارگت خود بررسی میکنیم و پس از backpropagation، دیگر برخلاف DQN، از آنها استفاده مجدد نمیکنیم.

پس از پیاده سازی SARSA بدلیل به نتیجه نرسیدن، تحیلیل های بسیاری انجام شد که پس از توضیح کد ها به سراغ آنها میرویم. همانند الگوریتم ذکر شده:  
  
ابتدا یک حافظه میسازیم که صرفا این بخش از الگوریتم را پیاده سازی میکند:  
  
  
A screenshot of a computer

Description automatically generated  
همانطور که مشاهده میکنیم همیشه پس از دریافت محتوای حافظه، همه آن را با دستور clear پاک میکنیم.

A computer screen shot of a program

Description automatically generatedاین بخش ساخت مدل، همانند DQN است.  
   
در بخش SARSA\_agent، به سراغ توضیح بخش های تغییرداده شده نسبت به DQN میرویم:  
  
تفاوت این بخش با DQN، استفاده از q در s’ با استفاده از a’ یا همان next\_state است. در حالی که در بخش قبلی DQN از ماکسیمم استفاده میکردیم:  
A close-up of a computer screen

Description automatically generated  
در بخش train\_test:  
A computer screen shot of a program

Description automatically generated  
برای انتخاب action بعدی از سیاست فعلی استفاده میکنیم، و هربار این وقایع را در مموری ذخیره میکنیم. اگر حافظه پر شده باشد، شروع به یادگیری این رخداد ها میکنیم، A computer screen with text

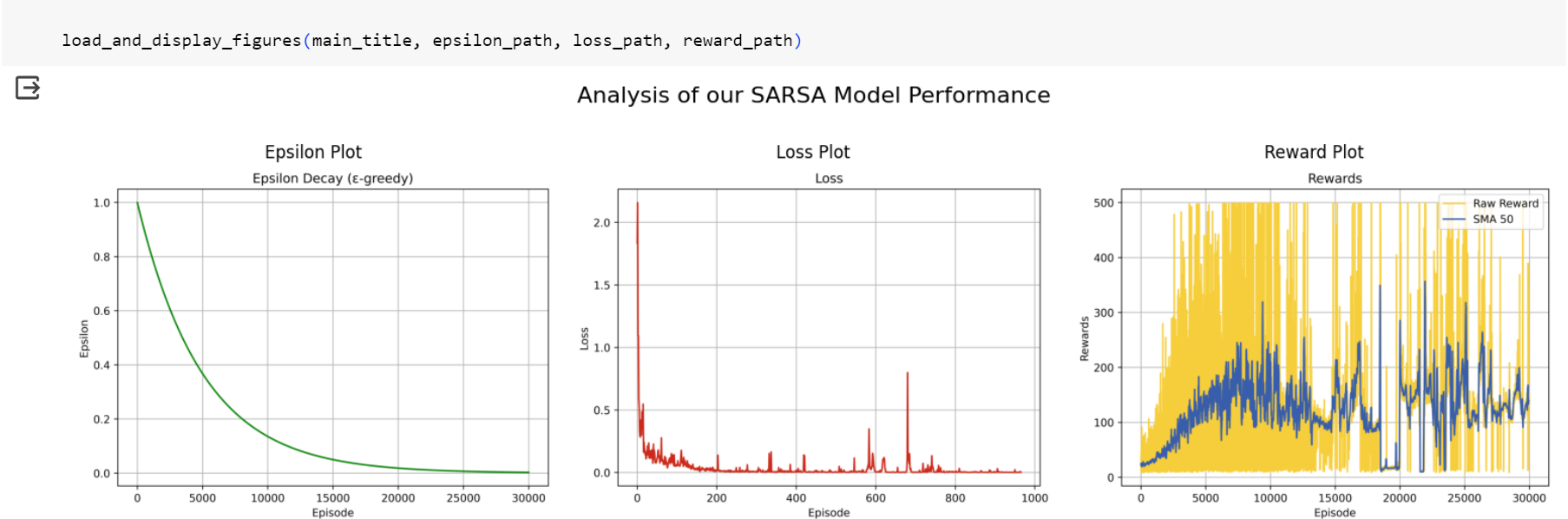
Description automatically generated  
و پس از یادگیری، دوباره حافظه را خالی میکنیم.  
پس از بررسی همه این موارد، شروع به یادگیری با استفاده از SARSA کردیم، اما نتایج پس از تعداد اپیزود های برابر با DQN، بسیار ضعیف بود.  
البته که DQN، چون از حافظه استفاده میکند، هربار پس از رخ دادن یک SARSA، یکبار دیگر دچار اپدیت میشود. درحالی که چون در SARSA باید منتظر پر شدن حافظه باشیم، پر کردن ان طولانی میشود، پس حتما به تعداد اپیزود های بسیار بیشتری از DQN نیاز دارد. در واقع تعداد دفعاتی که DQN دچار اپدیت میشود برابر است با:   
max\_episode – batch\_size (صرفا در مرتبه اول منتظر میشود که حافظه اش به اندازه یک batch پر شود، پس از آن دائما دچار اپدیت میشود)  
= 968   
که این تعداد اپدیت ها، تعداد زیادی رویداد تکراری دارد (به اندازه حافظه مموری ما که برابر با 4000 است – بسیار بیشتر از تعداد تجربه ها – و هر بار به اندازه یک batch از آن را لرن میکنیم).  
حال اگر با 1000 اپیزود در SARSA منتظر به جواب رسیدن بودیم، درواقع باید انتظار داشتیم که   
A white sky with clouds

Description automatically generated  
پس از 31 بار اپدیت، به نتیجه برسد، که کاملا ناعالانه بود.  
رویکرد دیگر، استفاده مجدد از داده ها و درواقع replay\_memory مانند DQN است که پس از بررسی ها، متوجه میشدیم تا حد خوبی به جواب میرسد، اما این کار خودش را از ذات الگوریتم SARSA کاملا جدا میکند.  
پس تنها موجود ما، افزایش تعداد max\_episode به همین روش، افزایش نرخ کاهش، به طریقی بتواند در میان یادگیری، از سیاست های exploration استفاده کند، بود.  
A white sky with clouds

Description automatically generated  
پس max\_episode را برابر 30 000 قرار داده، و برای epsilon\_decay داریم:  
A close-up of numbers

Description automatically generated  
پس 0.9998 میتواند مقدار مناسبی باشد.

حال با همه این بررسی ها:  
A screenshot of a computer program

Description automatically generated  
هایپر پارامتر هارا انتخاب میکنیم که یادگیری طولانی داشته باشیم!:  


پس از یادگیری، متوجه میشویم که گرچه مدل بتواند به نتیجه برسد، اما میانگین عملکرد آن، باز هم ضعیف تر از مدل DQN ما میباشد. DQN از حافظه خود استفاده میکند و چون هرکدام را صرفا سمپل کرده و دوباره برمیگرداند، احتمال انتخاب دوباره رویدار ها وجود دارد و این گونه قبلا آن ها را دیده است، با رخ دادن چندباره این اتفاق، سعی در کاهش لاس خود با داده های از پیش دیده شده دارد.  
اما SARSA داده ها را همیشه دور میاندازد و تجربه هایش همیشه برایش تازه است اما همین تازگی میتواند مشکلاتی را برایش ایجاد کند. از طرفی چون learning\_rate ها معمولا زیاد نیست، از همان تجربه ها هم، نمیتواند به طور کامل یادگیری مناسبی داشته باشد. شاید اگر این مسئله نیز کنترل میشد (که همان داده هایی که یکبار میبند را خوب اموزش ببیند – یا چندبار روی آن یادگیری انجام شود و سپس دور انداخته شود) آنگاه SARSA در برابر حالت تست هایی که قبلا تجربه نشده اند، بهتر از DQN عمل میکرد.  
  
توجه داریم که اگر در SARSA نیز از حافظه استفاده میشد که داده های قبلی را ببیند، بدون شک به جواب مناسبی میرسید اما از این هدف الگوریتم دور است.

ویدیو این بخش نیز در تایتل SARSA و سپس سابتایتل visualizing وجود دارد و لینک کولب قابل مشاهده است.  
  
A screenshot of a computer

Description automatically generated  
  
A video game with a stick

Description automatically generated with medium confidence