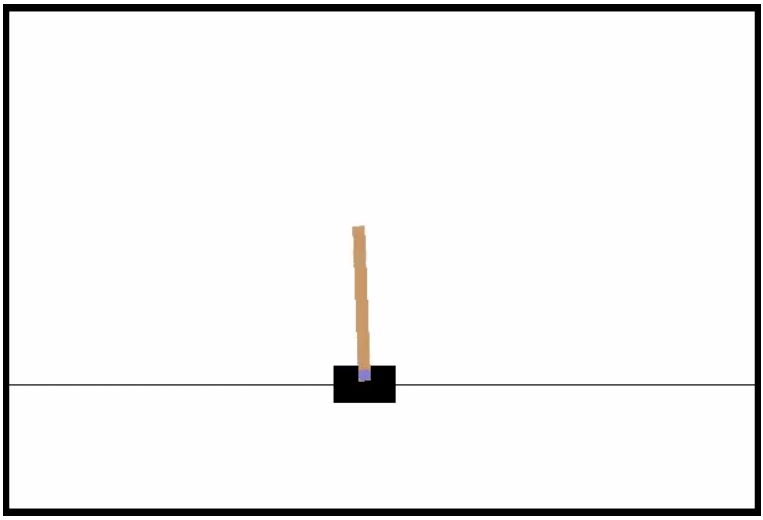
Okruženje 1



Cartpole-v0

https://github.com/openai/gym/blob/master/gym/envs/classic_control/cartpole.py
https://gym.openai.com/envs/CartPole-v1/

Opis

Cilj okruženja jeste balansirati štap koji je postavljen uspravno na kutiju.
Observation (stanje) se sastoji iz četiri vrednosti koje se odnose na poziciju kutije, brzinu kutije, ugla štapa i brzine vrha štapa.
Postoje dve moguće akcije a to su guranje kutije u levo (-1) i u desno (1).
Okruženje je ograničeno na 500 iteracija po epizodi. Okruženje vraća nagradu od 1 za svaku iteraciju u kojoj štap nije pao ispod linije. Ako štap padne ispod linije onda se dobija nagrada od -1 i epizoda se završava.

Rešenja

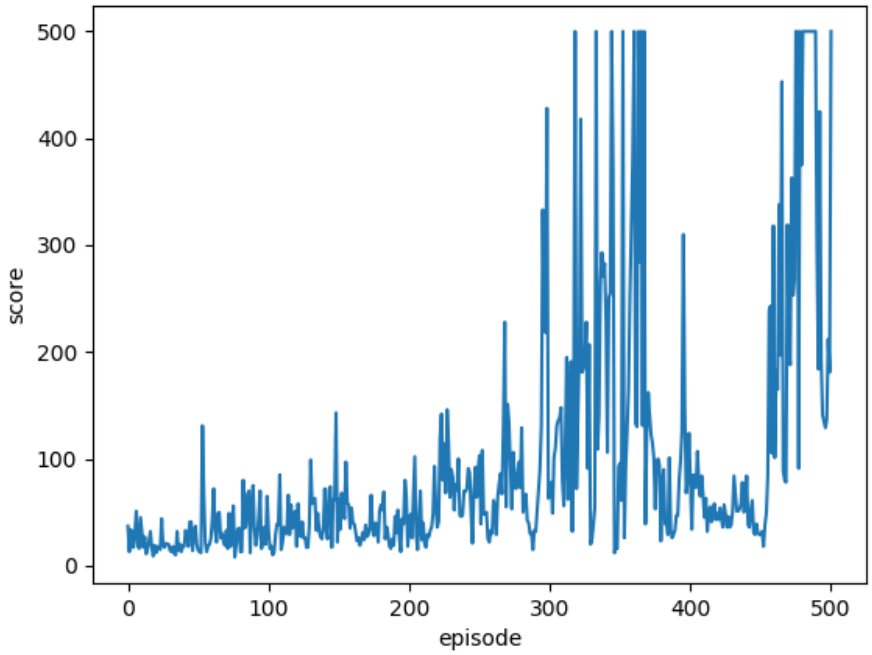
- Prvi pokušaj je bio pomoću Deep Q Learning (uz experience replay) sa neuronskom mrežom koja se sastoji od dva skrivena sloja sa po 24 neurona. U prvih nekoliko pokretanja sam dobio model koji je naizgled uspešno konvergirao, ali taj rezultat nisam uspeo da reprodukujem. Generalno mi se čini da je ovakva implementacija overkill za ovaj zadatak, i verovatno bi bolje radio sa jačim hardware-om. (za ovu impl. sam najviše podešavao parametre za batch size i exploration decay, ali nisam nešto značajno uočio)

- Za drugu implementaciju sam uklonio experience replay i smanjio model na NM sa jednim skrivenim slojem. S ovom konfiguracijom su dobijeni dosta bolji rezultati. Model konvergira u prvih 200-300 iteracija, i često osvaja i maksimalnu nagradu od 500. Prednost ove implementacije jeste da proces treniranja ide znatno brže, i samim tim se istraži mnogo više mogućih stanja. Mana koju sam primetio jeste da se svaki put, posle još dužeg treniranja, dobijaju znatno lošije performanse (prosečan skoro pada na ispod 20).

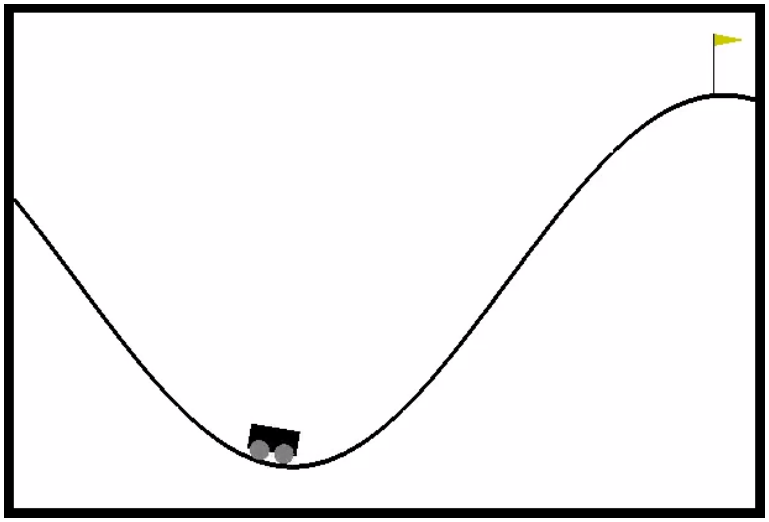
- Na kraju sam pokušao da primenim *discounting.* U slučaju cartpole-a sam dodao negativnu nagradu u slučaju da se epizoda završi pre 500 iteracija (što bi značilo da je došlo do neuspeha). Parametre vezane za discounting sam nazvao DISCOUNTING_MAX, DISCOUNTING_RANGE i DISCOUNTING_DECAY, i njih sam dalje podešavao. I u ovom slučaju se dešava da score modela naglo opadne, ali se ovaj put oporavi kasnije.

Rezultati

Na grafu ispod su prikazani rezultati kroz epizode treniranja gde su dobijeni najbolji rezultati (model je često dostizao score od 500). Parametri za discounting su DISCOUNTING_MAX=-400, DISCOUNTING_RANGE=-40 i DISCOUNTING_DECAY=0.8.



Okruženje 2



MountainCar-v0

https://github.com/openai/gym/blob/master/gym/envs/classic_control/mountain_car.py
https://gym.openai.com/envs/MountainCar-v0/

Opis

U ovom okruženju agent kontroliše auto koji se nalazi između dve planine. Cilj jeste da auto dospe na vrh desne planine, ali njegov motor nije dovoljno snažan da se popne uz planinu iz prve.
Observation (stanje) se sastoji iz dve vrednost, gde prva označava poziciju auta a druga brzinu kojom se auto trenutno kreće.
Postoje tri moguće akcije a to su kretanje auto u levo (-1), u desno (1) I mirovanje (0).
Okruženje je ograničeno na 200 iteracija po epizodi. Okruženje vraća nagradu od -1 u svakoj iteraciji gde auto nije dospeo do cilja. Kada auto dostigne cilj, vraća se nagrada 1.

Rešenja

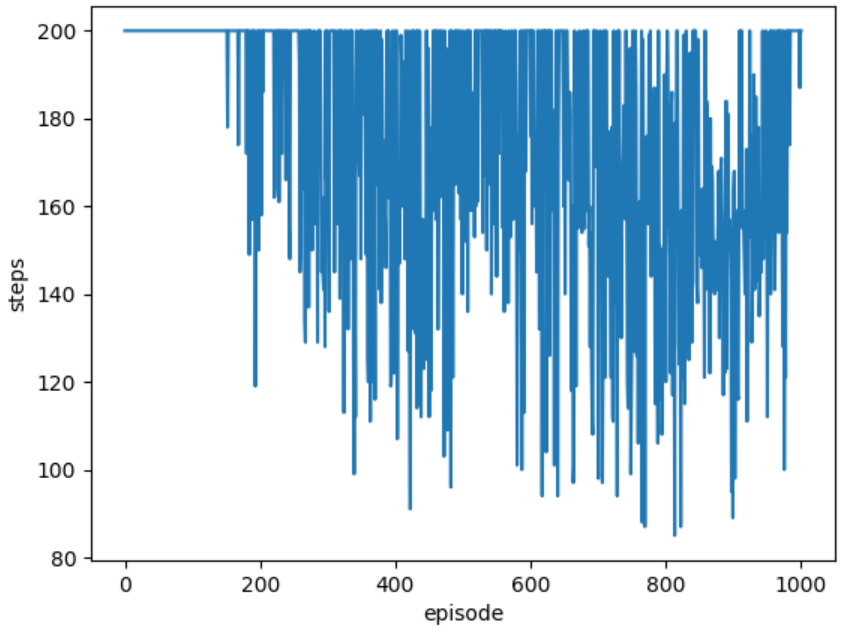
- I sa ovim okruženjem sam prvo pokušao Deep Q Learning (uz experience replay) sa neuronskom mrežom koja se sastoji od dva skrivena sloja sa po 24 neurona. Ponovo nisam dobijao rezultate s ovim, ali sam ovaj put primetio da agent uopšte ne dostiže cilj, čak ni posle 1000 epizoda. Zbog toga sam pokušao da podešavam parametre vezane za *exploration rate* i *exploration decay*, ali bez mnogo uspeha.

- Uočio sam da je najveći problem kod ovog okruženja sistem nagrade. Agent će veoma retko, ili skoro nikada, dostići cilj slučajnim putem. A čak i kada ga dostigne nagrada je previše niska. Zbog toga sam izmenio sistem nagrada, tako što se agent nagrađuje na nekoliko milestoneova a ne samo na krajnjem cilju. Milestoneovi je lakše dostići od krajnjeg cilja i služe da navode agenta u dobar smer. U ovom slučaju je bilo dovoljno davati manje nagrada kada se autić samo približava desnoj strani (ali sam pokušao i nagrađivati agenta kada dostigne veće brzine auta, što nije tolko dobro radilo). Implementacija sa korigovanim sistemom nagrada mi je pokazala znatno bolje rezultate, i model bio počeo da kovergira posle oko 300 epizoda.

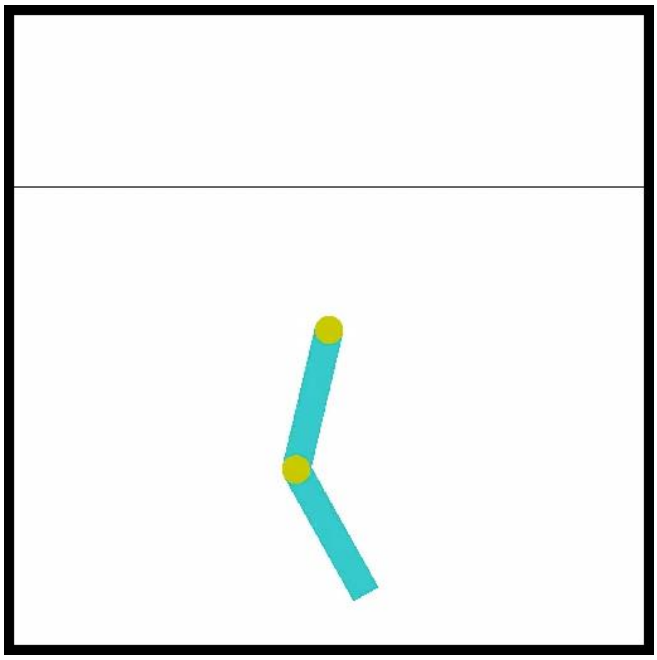
- Pokušao sam da primenim i discounting umesto da ručno postavljam nagrade na milestonove. Tu je opet problem bio da agent nikada ne uspe da dostigne cilj iz prve, tako da se discounting nije ni primenjivao.

Rezultati

Na grafu ispod su prikazani rezultati kroz epizode treniranja gde su dobijeni najbolji rezultati. Steps predstavlja dužinu neke epizode, i 200 znači da je ta epizoda neuspešna (agent nije dostigao cilj na vreme). Korištena je konfiguracija sa experience replay i modifikovanim sistemom nagrada. Takođe sam korigovao parametre za exploration rate tako da agent češće istražuje nova stanja. Parametri su:
EXPLORATION_MAX = 2
EXPLORATION_MIN = 0.1
EXPLORATION_DECAY = 0.996



Okruženje 3



Acrobot-v1

https://github.com/openai/gym/blob/master/gym/envs/classic_control/acrobot.py
https://gym.openai.com/envs/Acrobot-v1/

Opis
Okruženje se sastoji iz dve tačke (*joints*/zglobovi) i dve linije (*links*). Gornja tačka je zakucana u mesto i cilj jeste da se, primenom rotacionih sila na zglobove, donja linija prebaci preko sive linije.
Observation (stanje) se sastoji iz sinusne I kosinusne vrednosti za oba zglova i njihove ugaone brzine.
Moguće akcije su primena -1, 0 ili 1 obrtne sile na donji zglob.
Za svaku akciju koja ne dovodi do terminalnog stanja (prebacivanje preko sive linije) okruženje vraća nagradu od -1. Izvrašvanje je ograničeno na 500 iteracija po epizodi.

Rešenja
- Kao i u predhodnim okruženjima, počeo sam sa Deep Q Learning (uz experience replay) sa neuronskom mrežom koja se sastoji od dva skrivena sloja sa po 24 neurona. Isti problem, treniranje traje predugo, čak i kada agent slučajno dođe do cilja nagrada nije dovoljno velika da od toga nauči nešto.

- Ovo okruženje sam rešio na sličan način kao cartpole. Jedina razlika jeste da sam primenio pozitivno nagradu umesto negativne kada se radi discounting (jer želimo da završimo što ranije a ne da preživimo što duže kao kod cartpole). Nova stvar koju sam ovde isprobavao je podešavanje batch size-a tako da ne bude konstantan, već da jedna epizoda predstavlja jedan batch. Ovim se optimizuje discounting, pošto se pre nisu primenile nagrade na sva stanja (ako nisu u istom batchu). Isto tako sam korigovoa exploration rate, pošto su sad u proseku batchevi dosta veći (ponaročito u početku gde svaka epizoda traje 500 iteracija). Sa svime ovim sam dobijao dosta dobre rezultate, i discounting je bio ponaročito koristan prilikom rešavanja ovog okruženja.

Rezultati

Na grafu ispod su prikazani rezultati kroz epizode treniranja gde su dobijeni najbolji rezultati. Steps predstavlja dužinu neke epizode, i 500 znači da je ta epizoda neuspešna (slično kao kod mountcar). Korišćena je konfiguracija sa sledećim parametrima:

EXPLORATION_MAX = 1.0
EXPLORATION_MIN = 0.01
EXPLORATION_DECAY = 0.199

DISCOUNTING_MAX = 500
DISCOUNTING_RANGE = 100
DISCOUNTING_DECAY = 0.8