

Deep Q-Learning Agent za igru Tennis

Vladimir Popov SV29/2021

May 10, 2024

1 Definicija problema

Problem se odnosi na jednu od igara za *Atari 2600* konzolu pod nazivom *Tennis*. Naime, igra predstavlja simulaciju tenisa, gde igrač kontroliše narandžastog lika, protiv kompjuterski upravljanoj plavog lika. Igra prati sva pravila tenisa i igra se jedan set od šest gemova.

Igra se igra pomoću svih 18 mogućih kontrola na džojstiku za *Atari 2600* (gore, dole, levo, desno, gore-desno, gore-levo, dole-desno, dole-levo, udarac, gore-udarac, dole-udarac, i ostale kombinacije...). Opis i dokumentaciju same igre moguće je pročitati na sledećem linku).

2 Motivacija

Usled sve veće profitabilnosti i popularnosti zabavne industrije, pogotovo njene grane koja se bavi računarskim igrama, problemi u kojima se razvija agent za neku od igara postaju sve privlačniji. Njihovim rešavanjem i ugrađivanjem takvih agenata u samu igru ona postaje teža i samim tim zabavnija za krajnjeg korisnika, koji će posledično sve više vremena provoditi za ekranom igrajući tu igru.

Pored toga, *Reinforcement Learning* predstavlja značajnu granu veštačke inteligencije, jer uz pomoć nje možemo da generišemo nova znanja i do sada neotkrivene pristupe za već poznate probleme, za razliku od ostalih grana koje koriste gotove, ljudski proizvedene podatke da bi simulirali nešto što čovek već zna. Samim tim svako saznanje iz ove oblasti, dobijeno eksperimentisanjem sa različitim projektima, je veoma značajno za njen dalji razvoj.

3 Skup podataka

S obzirom da se radi o *Reinforcement Learning*-u skup podataka kao takav ne postoji u svom izvornom obliku. Podaci koji će se koristiti kao ulaz neuronske mreže su frejmovi same igre. Dok će se za evaluaciju akcije koristiti reakcija (nagrada ili kazna) samog okruženja.

4 Način pretprocesiranja podataka

Sami frejmovi igre biće pretvoreni od strane okruženja u crno-bele slike formata 250x160.

5 Metodologija

Kao prvo potrebno je postaviti i instancirati okruženje. Kako bi testirali okruženje napravićemo agenta koji pravi nasumične poteze i s njim ćemo kasnije upoređivati dobijene agente. Nakon toga, potrebno je napraviti model neuronske mreže, u nekoj od biblioteka, koja će nam davati Q-vrednosti za naše akcije. Sledeći korak je treniranje prethodno postavljene mreže, upotrebom nagrade i kazne koju dobijamo od strane okruženja. Na kraju procesa učenja potrebno je agenta evaluirati i videti da li smo zadovoljni njegovim sposobnostima, na taj način dobijamo sveobuhvatnu sliku na osnovu koje možemo proceniti naše zadovoljstvo agentom i ponoviti proces treniranja uz promenu hiperparametara, ukoliko uvidimo potrebu za tim.

6 Način evaluacije

Za evaluaciju akcija koje agent sprovodi koristiće se reakcija okruženja. Pošto igra prati pravila tenisa za svaki postignut poen agent će dobiti nagradu, dok će za svaki protivnikov poen biti kažnjen. Kako bi evaluirali samog agenta, posmatraćemo njegov učinak protiv ugrađenog protivnika.

7 Tehnologije

Plan za tehnologije koje će se koristiti (skup nije striktno ograničen, i tokom razvoja može doći do njegove izmene ukoliko za to bude potrebe):

- Python - odabrani programski jezik

- OpenAI Gym - okruženje za Q-učenje iz kojeg možemo da ekstrahujemo frejmove i nagrade
- Arcade Learning Environment - okruženje izgrađeno na emulatoru za *Atari 2600* igre, koje nam pomaže da integrišemo igru u *OpenAI Gym*
- TensorFlow (po potrebi Keras) - bilblioteka za mašinsko učenje

8 Literatura

- Deep Q Learning Tutorial: minDQN - Mike Wang [link]
- World Models - David Ha, Jürgen Schmidhuber [link]
- Cooperative multi-agent game based on reinforcement learning - Hongbo Liu [link]
- Hallucinogenic Deep Reinforcement Learning Using Python and Keras - David Foster [link]
- Using Deep Reinforcement Learning to Play Sonic the Hedgehog - Daniel Bourke [link]
- Deep Q-Network (DQN) - Shruti Dhumne [link]
- Rainbow: Combining Improvements in Deep Reinforcement Learning - DeepMind [link]