Trening i Implementacija AI za Air Hockey

Teodor Vidaković SV33/2021

Danilo Cvijetić SV25/2021

July 3, 2024

Uvod

- **Opis Projekta**:
  - Prvo smo razvili igricu Air Hockey koristeći Pygame kako bismo simulirali okruženje igre. Ova simulacija omogućava realistično prikazivanje dinamike, uključujući fizičke interakcije između paka i štapova, kao i zidova.
  - Zatim smo radili na razvoju i obuci agenata veštačke inteligencije za simulaciju igre koristeći tehnike dubokog učenja. Korišćenjem neuronskih mreža, agenti su obučeni da prepoznaju obrasce i optimizuju svoje ponašanje tokom igre.
  - Koristi se metoda Proširenog DDPG (MADDPG) za koordinaciju više agenata u okruženju igre. Ovaj algoritam omogućava agentima da deluju u zajedničkom prostoru i da komuniciraju i koordinišu svoje akcije kako bi postigli bolje rezultate.
- **Ciljevi Projekta**:
  - Razviti efikasne strategije za kontrolu dva agenta u simulaciji. Cilj je da agenti budu sposobni da donose brze i precizne odluke tokom igre, optimizujući svoje pokrete i strategije za postizanje golova i odbranu svog polja.
  - Implementirati nagrade i kazne kako bi se optimizovalo ponašanje agenata. Različiti scenariji u igri, poput postizanja gola, držanja paka, i sudara sa pakom, su povezani sa specifičnim nagradama i kaznama koje pomažu agentima da uče efikasnije taktike i ponašanja.
- **Motivacija**:
  - Razvoj inteligentnih agenata za igre pruža uvid u primenu veštačke inteligencije u dinamičnim i nepredvidivim okruženjima. Ova istraživanja mogu doprineti napretku u različitim oblastima, uključujući robotiku, autonomna vozila, i druge sisteme gde je potrebno brzo donošenje odluka.
  - Igre kao što je Air Hockey predstavljaju idealno okruženje za testiranje i razvoj algoritama za multi-agentne sisteme, jer uključuju kompleksne interakcije i strategije koje se mogu primeniti u širem spektru realnih aplikacija.
- **Izazovi**:

- Jedan od glavnih izazova u ovom projektu je balansiranje između nagrada i kazni kako bi se postiglo optimalno ponašanje agenata. Preterane kazne mogu demotivisati agenta, dok previše nagrada može dovesti do suboptimalnih strategija.
- Još jedan izazov je obezbeđivanje da agenti mogu brzo da prilagode svoje strategije u promenljivom okruženju igre. Ovo zahteva sofisticirane algoritme za učenje i adaptaciju.
- Resursi za treniranje su takođe značajan izazov. Trening agenata veštačke inteligencije zahteva značajne računske resurse, uključujući moćne grafičke procesore (GPU) i puno vremena za simulacije i učenje. Ograničenja u dostupnim resursima mogu usporiti proces treniranja i otežati postizanje optimalnih rezultata.

Tehnologije i Alati

- **Programski Jezik**:
  - Python
- **Biblioteke i Alati**:
  - PyTorch za duboko učenje
  - Pygame za grafički interfejs i simulaciju igre
  - Sklearn za obradu podataka

Video Demonstracija

- **Igra na Djelu**:

Pogledajte video demonstraciju ovde

Struktura Koda

- **Klasa GameCore**:
  - Upravljanje stanjem igre, uključujući položaj i brzinu paka, pozicije štapova, detekcija kolizija i računanje nagrada.
  - Ključne metode: `update_game_state`, `get_reward`, `move_paddle`, `check_paddle_collisio`
- **Klasa ActorModel i CriticModel**:
  - Definicija neuronskih mreža za agenta.
  - Ključne metode: `forward`, `reset_parameters`.
- **Klasa Agent_DDPG**:
  - Upravljanje učenjem agenta, uključujući radnje i ažuriranje modela.

  - Ključne metode: `act`, `learn`, `soft_update`.

- **Klasa Agent_MADDPG**:
  - Koordinacija više agenata i njihovo kolektivno učenje.
  - Ključne metode: `step`, `act`, `learn`.

- **Klasa GUICore**:
  - Grafički interfejs za igru, uključujući prikaz stanja igre, štapova, paka i rezultata.
  - Ključne metode: `update`, `close`, `draw_predicted_path`.

- **Klasa ReplayBuffer**:
  - Implementacija repozitorijuma za skladištenje iskustava agenata (stanja, akcije, nagrade, sledeća stanja, oznake završetka epizoda).
  - Ključne metode: `add`, `sample`.

- **Klasa OUNoise**:
  - Implementacija Ornstein-Uhlenbeck noise-a za istraživanje tokom treninga agenata.
  - Ključne metode: `reset`, `sample`.

Algoritam i Logika Učenja

DDPG (Deep Deterministic Policy Gradient)

- **Algoritam za kontinuirane akcije**:
  - DDPG je off-policy algoritam za učenje pojačanjem koji se koristi za probleme sa kontinuiranim akcijama.
  - Algoritam kombinuje ideje iz DQN (Deep Q-Network) i DPG (Deterministic Policy Gradient) algoritama kako bi omogućio efikasno učenje u okruženjima sa velikim prostorima akcija.

- **Korišćenje aktorske i kritičke mreže**:
  - **Aktor**: Neuronska mreža koja mapira stanja na akcije. Aktor generiše akcije koje agent treba da izvrši.
  - **Kritičar**: Neuronska mreža koja procenjuje vrednost Q-funkcije, tj. očekivanu sumu budućih nagrada za dati par stanja i akcije. Kritičar procenjuje koliko je dobra ili loša data akcija u datom stanju.
  - Algoritam koristi dva skupa aktorskih i kritičkih mreža: glavne mreže (koje se ažuriraju tokom treninga) i ciljane mreže (koje se ažuriraju sporije kako bi stabilizovale učenje).
  - Gubitak aktora se računa kao negativna procena Q-vrednosti kritičara, dok se gubitak kritičara računa pomoću TD (Temporal Difference) greške.

MADDPG (Multi-Agent DDPG)

- **Proširenje DDPG za okruženje sa više agenata**:
  - MADDPG je proširenje DDPG algoritma koje omogućava primenu u okruženjima sa više agenata.
  - Svaki agent ima svoju aktorsku mrežu, dok se kritičke mreže mogu deliti ili biti specifične za svakog agenta.
- **Koordinacija akcija i deljenje informacija**:
  - Agenti dele informacije o svojim stanjima i akcijama kako bi unapredili proces učenja. Ovo omogućava svakom agentu da uzme u obzir akcije drugih agenata pri donošenju sopstvenih odluka.
  - Kritičar koristi informacije o akcijama svih agenata kako bi procenio Q-vrednost, omogućavajući koordinaciju među agentima.
  - Deljenje informacija između agenata pomaže u smanjenju neizvesnosti i poboljšava kolektivne performanse sistema.
- **Ažuriranje mreža i stabilnost učenja**:
  - Kao i u DDPG, ciljane mreže se koriste za stabilizaciju učenja. One se ažuriraju pomoću soft update tehnike, gde se parametri ciljane mreže polako približavaju parametrima glavne mreže.
  - Koristi se iskustvo iz replay buffer-a, gde agenti čuvaju svoja prošla iskustva i koriste ih za treniranje mreža. Ovo omogućava efikasnije i stabilnije učenje, smanjujući korelaciju između uzastopnih iskustava.

Implementacija Nagrada i Kazni

- **Osnovne Nagrade**:
  - **Nagrada za gol**: Dodeljuje se agentu kada postigne gol. Ova nagrada podstiče agenta da postiže ciljeve i igra efikasno.
  - **Nagrada za razdaljinu od paka**: Pozitivna nagrada kada se agent približi paku. Ova nagrada pomaže agentima da aktivno učestvuju u igri i drže kontrolu nad pakom.
  - **Nagrada za sudar sa pakom**: Pozitivna nagrada kada agent sudari sa pakom, podstičući ga da aktivno prati i udara pak.
  - **Nagrada za direktno usmeravanje paka prema protivničkom golu**: Dodeljuje se kada agent uspešno usmeri pak prema protivničkom golu tokom simulirane putanje paka.
  - **Nagrada za ubrzanje paka**: Pozitivna nagrada kada agent povećava brzinu paka.
- **Kazne**:

- **Kazna za dribling u svojoj polovini**: Oduzima se bod za predugo zadržavanje paka u svojoj polovini. Ova kazna podstiče agenta da se kreće prema protivničkom golu.
- **Kazna za sporu aproksimaciju paka**: Dodeljuje se kazna ako agent prilazi paku presporo, podstičući ga da se kreće brže i efikasnije.
- **Kazna za stajanje paka**: Kazna kada pak stoji mirno, podstičući stalnu aktivnost i dinamiku igre.
- **Kazna za pogrešno usmeravanje paka**: Kazna za usmeravanje paka prema sopstvenom golu, čime se sprečava autogol.
- **Kazna za zadržavanje paka iza sebe**: Kazna za situaciju kada je pak iza agenta, podstičući agenta da se postavi ispred paka.
- **Kazna za usporavanje paka**: Dodeljuje se kada agent smanjuje brzinu paka.

Rezultati i Diskusija

- **Performanse Agenta**:
  - Tokom treninga, agenti su pokazali solidne performanse u učenju osnovnih pravila igre. Njihova sposobnost da kontrolišu štapove, sudaraju se sa pakom i postižu golove značajno je poboljšana.
  - Agenti su razvili efikasne strategije tokom treninga, uključujući bolju koordinaciju pokreta, brže reakcije na poziciju paka i optimizaciju udaraca ka golu protivnika.
  - Iako su performanse agenata zadovoljavajuće, dalji trening bi mogao dodatno poboljšati njihovu strategiju i efikasnost.
- **Izazovi i Problemi**:
  - **Podešavanje hiperparametara**: Jedan od ključnih izazova bio je podešavanje hiperparametara. Svaki od ovih parametara imao je značajan uticaj na stabilnost i brzinu konvergencije algoritma.
  - **Stabilnost učenja**: Algoritmi dubokog učenja, posebno oni koji se koriste u okruženjima sa višestrukim agentima, skloni su problemima sa stabilnošću. Često je dolazilo do oscilacija u ponašanju agenata, posebno u ranom stadijumu treninga. Stabilnost učenja je poboljšana kroz upotrebu tehnika kao što je replay buffer.
  - **Problemi sa istraživanjem**: Agenti su povremeno previše eksploatisali naučene strategije, što je dovodilo do suboptimalnih rezultata. Implementacija Ornstein-Uhlenbeck noise-a pomogla je da se ovo prevaziđe, omogućavajući agentima da istraže šire opsege mogućih akcija.
  - **Resursi**: Trening dubokih mreža je računarski intenzivan proces. Ograničeni resursi, uključujući hardverske kapacitete i vreme, predstavljali su značajan izazov. Koristili smo NVIDIA CUDA na RTX 3050 Ti grafičkoj kartici da bismo ubrzali trening i smanjili opterećenje na sistemu.

Zaključak

- **Postignuća**:
  - Uspešno smo implementirali agente veštačke inteligencije koji mogu efikasno igrati. Agenti su pokazali sposobnost da kontrolišu štapove, udaraju pak, postižu golove i razvijaju strategije za optimalno igranje.
  - Primena tehnika dubokog učenja, konkretno DDPG i MADDPG algoritama, omogućila je agentima da uče iz iskustava i kontinuirano poboljšavaju svoje performanse. Ovaj projekat je demonstrirao efikasnost ovih tehnika u složenim multi-agent okruženjima.
  - Agenti su uspešno integrisali različite nagrade i kazne kako bi optimizovali svoje ponašanje, pokazujući napredak u razvoju sofisticiranih strategija kroz trening.
  - Iskoristili smo hardverske resurse, uključujući NVIDIA CUDA na RTX 3050 Ti grafičkoj kartici, za ubrzanje treninga i postizanje boljih performansi, što je omogućilo bržu konvergenciju modela i efikasniju upotrebu resursa.
- **Budući Rad**:
  - Dalje unapređenje algoritama može dodatno poboljšati performanse agenata. Planiramo eksperimentisanje sa različitim hiperparametrima.
  - Planiramo integraciju više složenih scenarija i okruženja kako bi agenti mogli da uče u još izazovnijim uslovima, što bi ih pripremilo za potencijalne realne aplikacije u robotici i automatizaciji.
  - Dodatno ćemo istražiti mogućnosti korišćenja drugih biblioteka i alata za poboljšanje grafičkog interfejsa i simulacije, kako bismo poboljšali vizuelizaciju i interaktivnost tokom treninga.
  - Planiramo da istražimo mogućnosti korišćenja distributivnog treninga na više GPU-ova ili čak korišćenja cloud resursa kako bismo ubrzali proces treniranja i omogućili rad sa većim modelima i složenijim okruženjima.
  - Kreiranje botova sa različitim strategijama kako bi mogli trenirati jedni protiv drugih, čime bismo obezbedili raznovrsniji trening i razvoj naprednijih taktičkih sposobnosti agenata.