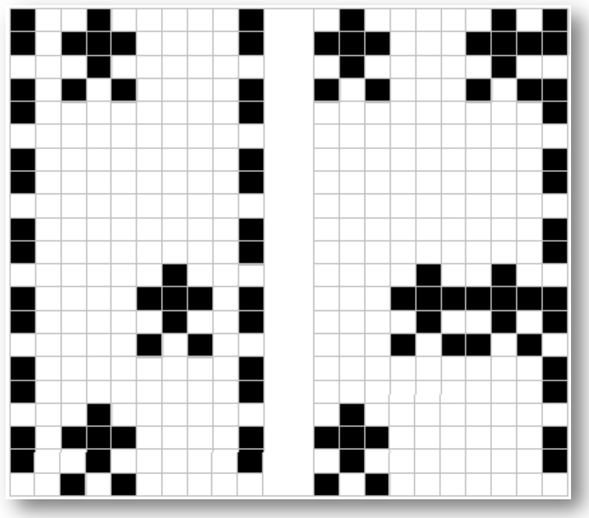
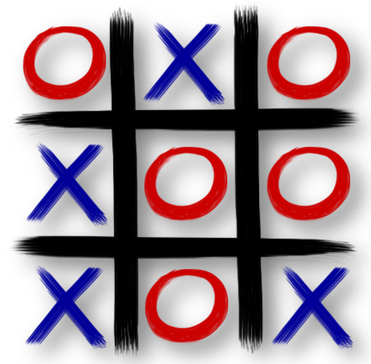
Autonomno računarsko igranje igara inspirisano „Deep Learning“ principima

Nikola Jovanović (1996), seminar računarstva
IV razred, Deveta gimnazija „Mihailo Petrović – Alas“, Beograd
mentor: Dragan Toroman





Uvod

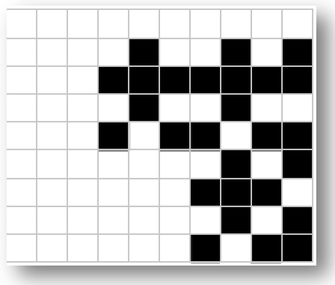
- Veštačka inteligencija je jedna od oblasti računarstva koje se trenutno najviše razvijaju. Takođe je i sve češća tema u naučno-popularnim člancima. Ideja ovog rada je da čitaocima približi koncept mašinskog učenja, kao jednog od oblika veštačke inteligencije.
- Osnovni principi mašinskog učenja prikazani su pomoću dva slična algoritma koji uče da igraju igrice. Algoritmi su posebno dizajnirani u ovom radu, a testirani su na igrama Iks-oks i dve verzije igre Trkanje sa konzole Brock Game (popularno Tetris). Ove igre su odabrane kao primeri logičke i arkadne igre.
- Inspiracija za ovaj projekat je rad „Playing Atari with Deep Reinforcement Learning“, u kome grupa autora objašnjava svoj univerzalni algoritam za igranje igrica. Njihov algoritam je naučio da irgra pet igrica za računar Atari, a to su: *Pong*, *Breakout*, *Space Invaders*, *Seaquest* i *Beam Rider*. Na nekima je čak nadmašio ljudskog eksperta ili naučio da koristi bagove u igrici u svoju korist.

Osnovne ideje korišćenih algoritama

- Ključna osobina algoritama kojima se bavi ovaj rad je **univerzalnost**. Algoritmi su dizajnirani tako da uče da igraju igru bez poznavanja pravila i ciljeva igre.
- Informacije kojima algoritmi raspolažu su iste kao i one kojima bi raspolagao čovek koji prvi put vidi igru:
  - Trenutno stanje ekrana (trenutna slika na ekranu)
  - Broj osvojenih poena (trenutni skor)
- Algoritmi su dizajnirani tako da će program vremenom igrati sve bolje i bolje, učeći na svojim greškama.

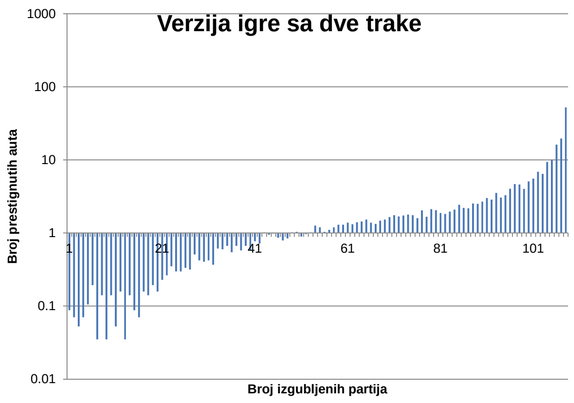
Tetris trkanje

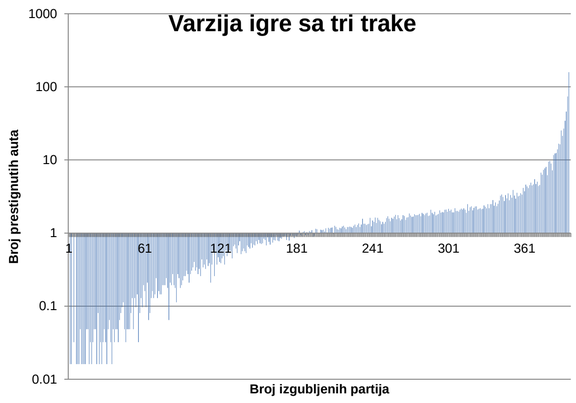
- Dve verzije igre trkanja su preuzete sa konzole *Brick Game*. U prvoj verziji put kojim se automobili kreću ima dve trake, a u drugoj tri trake. Igrač kontroliše auto sa dva tastera koji označavaju komande levo i desno.
- Protivnički automobili se pojavljuju na vrhu ekrana u nasumično odabranim trakama i kreću se na dole, cilj igre je ne sudariti se sa protivničkim automobilima.
- Ekran koji se prosleđuje algoritmu je predstavljen matricom sa 20 redova i 10 kolona, sa mogućim stanjima 0 i 1, koja odgovaraju crnim i belim pikselima.
- Igrica je dizanjnirana tako da algoritam dobija negativne poene kada se njegov automobil sudari sa drugim i izgubi život. Tada algoritam zabranjuje akciju koja je dovela do gubitka života.
- Ovakav pritup ne može da se primeni na drugu verziju igre zato što u slučaju prikazanom na slici ne postoji akcija kojom može da se izbegne gubljenje života. Ovaj probelm je rešen tako što algoritam takva stanja obeleži kao stanja na kojima je već izgubio život, a zatim uči kako njih da izbegne.



Rezultati učenja igara trkanja

- Na sledeća dva grafika prikazana je uspešnost algoritma koji je učio prvu i drugu verziju igre trkanja. Na logaritamskoj skali je prikazan odnos prosečnog broja osvojenih poena u odnosu na broj izgubljenih partija od početka učenja, nakon velikog broja ponavljanja.





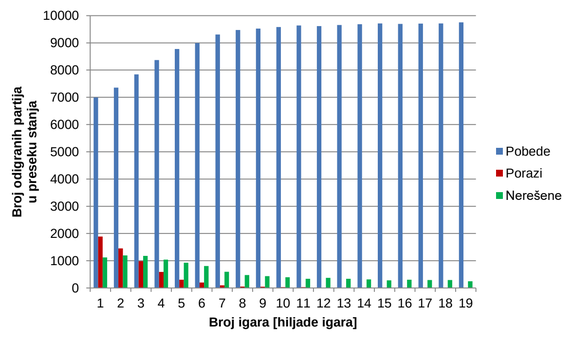
- Sa prethodnih grafika se može zaključiti da:
  - Algoritam ima veoma loše rezultate sve do trenutka kada u potpunosti ne nauči igru.
  - Oba grafika prikazuju isti oblik zavisnosti uspešnosti od broja izgubljenih života.
  - Druga verzija igre zahteva oko četiri puta više odigranih partija da bi bila savladana.

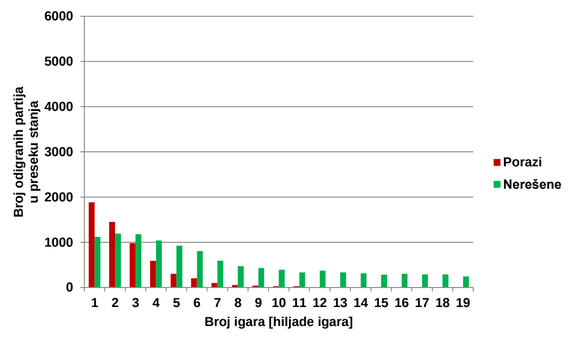
Iks-oks:

- Ekran koji se prosleđuje algoritmu je predstavljen matricom sa tri reda i tri kolone, sa mogućim stanjima 0, 1 i 2. Ova stanja predstavljaju prazno polje, polje obeleženo sa X i polje obeleženo sa O.
- U implementaciju igre je uključen sistem bodovanja table. To znači da je za svako stanje na tabli moguće izračunati koji potez je optimalan.
- Algoritam uči da igra, igrajući protiv drugog algoritma koji uvek odabira potpuno nasumične poteze, kako bi se osiguralo da će biti istestirane sve kombinacije.
- Kada algoritam izabere akciju i potez se odigra, informacija koja se vraća algoritmu predstavlja vrednost efikasnosti poteza koji je odigran. Ona ima vrednost 1 ukoliko je izabran optimalan potez, u suprotnom ima vrednost 0.

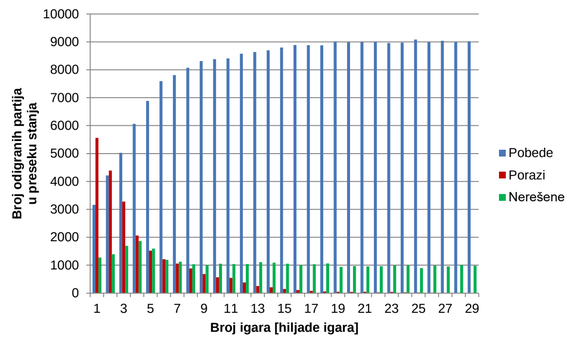
Rezultati učenja igre Iks-oks

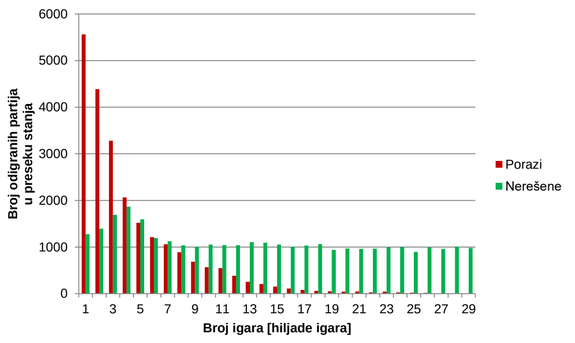
- Sledećia dva grafika prikazuju uspešnost algoritma koji je učio da igra poziciju X u igri Iks-oks. Na graficima su prikazane pobede, porazi i nerešene partije, čiji broj je resetovan na svakih 10000 partija. Na drugom grafiku je izbačen broj pobeda, kako bi se bolje video odnos broja izgubljenih i nerešenih paritja.





- Sledeći grafici prikazuju uspešnost algoritma koji je učio da igra poziciju O.





- Sa prethodnih grafika se vidi da:
  - Iako procenat pobeda brzo raste, potrebno je mnogo više treniranja da bi broj izgubljenih partija pao na nulu.
  - Drugi par grafika se slaže sa činjenicom da je O podređena pozicija u odnosu na X, zbog sporijeg učenja i većeg broja nerešenih partija.

Zaključak

- Korišćenjem prikazanih algoritama postižu se dobri rezultati jer oni nakon određenog boja iteracija nauče da pobeđuju.
  - Ograničenje ovih algoritama je to što ima je za složenije igre potrebno mnogo više vremena i memorijskog prostora.
  - Takođe oni mogu da prepoznaju samo identična stanja ekrana i ne bi bili u mogućnosti da iskoriste prethodno znanje u sličnim situacijama, koje bi čovek sa lakoćom prepoznao.
- Algoritmi se mogu unaprediti primenom naprednijih metoda za mašinsko učenje, kao što su analiza signala i neuralne mreže.

LITERATURA:

- Mnih V; Kavukcuoglu K; Silver D; Graves A; Antonoglou I; Wierstra D, Riedmiller M (2013). "***Playing Atari with Deep Reinforcement Learning***".

XIV godišnja konferencija polaznika Istraživačke stanice Petnica "Korak u nauku"
novembar 2015