

Rešavanje igrice korišćenjem neuronskih mreža

Milan Ilić

February 2019

1 Opis problema

U ovom radu biće opisano par tehnika za rešavanje kompjuterske igre poput igre koja se nalazi na sajtu [1]. Pod *rešavanjem* smatra se pravljenje agenta koji je u mogućnosti da potpuno samostalno igra igru i dodje do njenog kraja. Cilj igre je postići što veći rezultat izbegavajući prepreke. U svakom trenutku moguće je naterati igrača da skoči, nastavi da trči ili čučne, što je dovoljno da se izbegne bilo koja data prepreka. Kako vreme prolazi, tako igra postaje teža i teža, jer se brzina nadolazećih prepreka ubrzava, pa je potrebno prilagoditi se novim uslovima. Implementacija igre i njenog rešenja sa svim pratećim materijalima, nalazi se na sledećem *github* linku [2].

2 Pregled srodnih tehnika za rešavanje problema

Za rešavanje problema ovog tipa koriste se razne metode računarske inteligencije, neke od njih su:

- Genetski algoritmi
- Nadgledano učenje
- Učenje potkrepljivanjem
- Hibrid genetskih algoritama i neuronskih mreža

Svaka od gore predloženih tehnika rešavaju problem na specifičan način i opisivanje svake prevazilazi razmere ovog rada. U referencama se mogu naći radovi koji opisuju korišćenje navedenih tehnika za rešavanje sličnih problema i čitaocu se toplo preporučuje čitanje istih za bolje shvatanje datog problema ([4], [3], [5]).

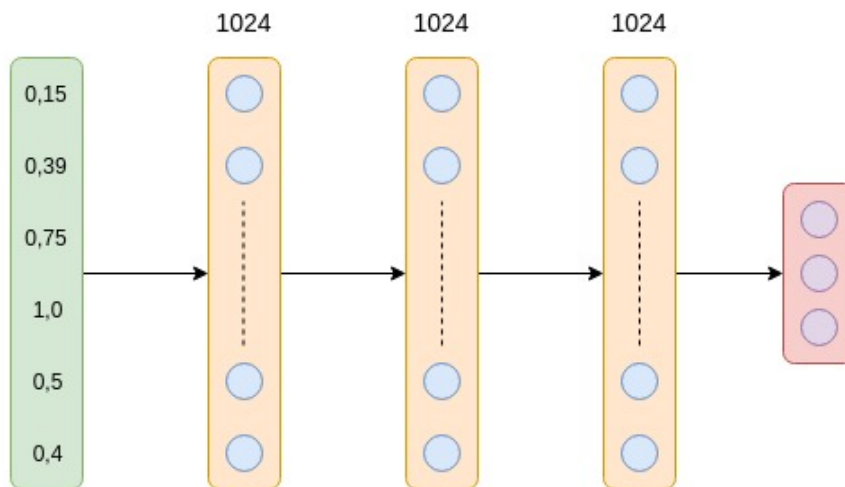
3 Predloženi pristupi

U ovoj sekciji biće ukratko predstavljena dva pristupa nadgledanog učenja za rešavanje datog problema. Prvi pristup koristi duboku neuronsku mrežu

sa propagacijom unapred, dok drugi koristi konvolutivnu neuronsku mrežu. U oba pristupa mreža predviđa jednu od tri moguće akcije (skakanje, trčanje i čučanje), koja se izvršava u sledećem frejmu igre. Parametri treniranja su menjani i izabrani su oni koji su dali najbolje rezultate. U nastavku sledi detaljan opis oba pristupa.

3.1 Rešavanje problema korišćenjem neuronske mreže sa propagacijom unapred

Za rešavanje ovim pristupom, potrebno je najpre napraviti prikladan skup podataka nad kojim je moguće treniranje. Za svaki frejm igre se izvlači vektor izabranih atributa koji imaju uticaj na sledeću akciju kao što su udaljenost od najbliže prepreke, visina date prepreke, brzina nadolazećih prepreka... Skup podataka se sastoji od vektora pomenutih atributa, koji se nakon učitavanja deli na trening skup i skup za validaciju. Inicijalni skup podataka se prikuplja igranjem igre od strane čoveka. Treniranje se izvršava inkrementalno, naime nakon treniranja na inicijalnom skupu, prikupljaju se atributi nastali igranjem igre od strane nastalog modela, sve dok model ne postane nepogrešiv i ne pobedi u svakom igranju.

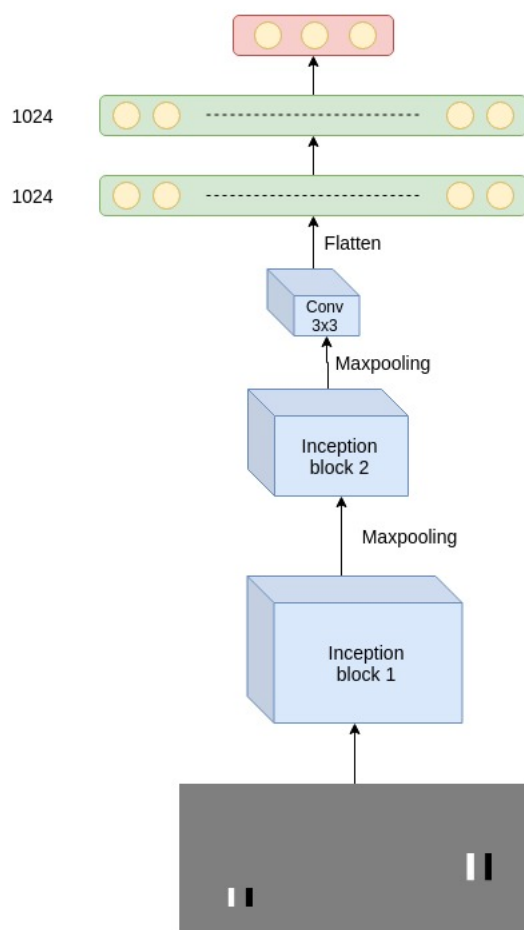


Slika 1. Arhitektura neuronske mreže korišćena za prvi pristup.

Nad izlazom iz poslednjeg sloja mreže primenjuje se softmax aktivaciona funkcija. U toku igranja igre nakon treniranja, sledeća akcija bira se iz raspodele dobijene primenom aktivacione funkcije, čime se postiže istraživanje i manje verovatnih stanja igre.

3.2 Rešavanje problema korišćenjem konvolutivne neuronske mreže

Kod pristupa koji koristi konvolutivnu mrežu potrebno je pripremiti skup podataka u vidu slika, međutim zbog povećanja brzine prepreka kroz vreme, samo jedna slika frejma nije dovoljna. Zbog toga se vrši preprocesiranje, koje podrazumeva spajanje dva susedna frejma u jednu sliku. Naime, prvo pretvaramo sliku u monohromatski format, nakon čega piksele koji se nisu promenili u dva susedna frejma bojimo u sivu boju, dok piksele koji su postali sivi u crnu. Time dobijamo informacije o kretanju i brzini prepreka i igrača koje će olakšati mreži da predvidi sledeću akciju.



Slika 1. Arhitektura neuronske mreže korišćena za drugi pristup.

U ovom slučaju istraživanje manje verovatnih stanja nije primenjeno, već se za sledeću akciju uzima najveća vrednost u izlaznom sloju.

4 Rezime

Pokazalo se da izabrani pristupi mogu biti korišćeni za rešavanje problema ovog tipa i uz dat inicijalni skup podataka, bez velikog vremenskog utroška daju dobre rezultate. Mana ovakvih pristupa jeste upravo prikupljanje inicijalnog skupa podataka koje nije moguće za svaki tip računarske igre. Medjutim, u slučaju ove igre, prikupljanje podataka i odabir odgovarajuće akcije je trivijalno, zbog čega su upravo i iskorišćeni ovi pristupi za rešavanje.

References

- [1] Google. <https://elgoog.im/t-rex/>.
- [2] Milan Ilic. <https://github.com/milanilic332/Simple-dino-game>.
- [3] Prafulla Dhariwal Alec Radford Oleg Klimov John Schulman, Filip Wolski. <https://arxiv.org/abs/1707.06347>.
- [4] Julian Togelius Sebastian Risi. <https://arxiv.org/abs/1410.7326>.
- [5] Mehdi Mirza Alex Graves Timothy P. Lillicrap Tim Harley David Silver Koray Kavukcuoglu Volodymyr Mnih, Adrià Puigdomènech Badia. <https://arxiv.org/abs/1602.01783>.