Pac-Man Agent

IZRADA INTELIGENTOG AGENTA ZA REŠAVANJE NIVOA PAC-MAN IGRE KORIŠĆENjEM NEAT METODE

Mentori:
PhD Đorđe Obradović
Mihailo Isakov

Autori:
RA 63/2012 – David Vuletić
RA 75/2012 – Nikola Todorović

SADRŽAJ

UVOD

Ovaj rad se bavi detaljnom analizom problema izrade inteligentnog agenta za prelaženje nivoa arkadne igrice *Pac-Man*.

Sastoji se od pet poglavlja kojima se analiziraju domen problema, već postojeća rešenja, njihove prednosti i mane. Nakon toga se ulazi u nešto detaljniju analizu NEAT metode, koja je odabrana kao odgovarajuća za izradu inteligentnog agenta. Potom se predstavljaju koraci implementacije, da bi se na kraju prikazali i analizirali dobijeni rezultati.

POTREBAN SOFTVER

Za pokretanje projekta potreban je sledeći softver:

- Emulator za NES igre BizHawk, koji je moguće skinuti sa sledećeg sajta - BizHawk Emulator
- Pac-Man (USA) (Namco), koji je moguće skinuti sa sledeceg sajta - Pac-Man
- LUA skripta koja se pokreće preko BizHawk emulatora, koju je moguće skinuti sa sledećeg sajta - PacManAgent.lua.

UPUTSTVO ZA POKRETANJE

Kako bi se pokrenulo obučavanje evolutivne neuronske mreže, potrebno je učiniti sledeće:

- Pokrenuti Pac-Man ROM preko FILE -> OPEN ROM.
- Pokrenuti nivo Pac-Man-a, i u trentku dok se on nije još nije pokrenuo sačuvati stanje pod nazivom „pacman.state“ u direktorijum u kom se nalazi „PacManAgent.lua“ skripta. Čuvanje stanja se vrši preko FILE->SAVE STATE->SAVE NAMED STATE.
- Pokrenuti Lua skriptu koristeći Lua konzolu, na sledeći način:
    - Preko TOOLS->LUA CONSOLE otvoriti Lua konzolu.
    - U prozoru Lua konzole otići na SCRIPT->OPEN SCRIPT, i odabrati već pomenutu „PacManAgent.lua“ skriptu.

Nakon ovoga, obučavanje mreže je započeto.

MOTIVACIJA

UKRATKO O PAC-MAN IGRI

Pac-Man je jedna od najprepoznatljivijih igara širom sveta. Izrađena je od strane Namco korporacije i puštena u promet 1980. godine, i napravila je pravu revoluciju u industriji video igara.

Ideja igre jeste da se Pac-Man uz pomoć komandi vodi kroz lavirint i da pojede što veći broj žutih bobica. Međutim, to mora da se uradi tako da se izbegnu 4 duha koja, svaki svojom strategijom, pokušavaju da spreče Pac-Man-a u jedenju bobica. Za lakšu borbu protiv duhova, Pac-Man-u je omogućeno da i on njih pojede, ali samo u kratkom vremenskom periodu nakon jedenja velikih bobica – Powepill-ova. Nakon što ga Pac-Man pojede, duh se vraća na sredinu lavirinta. Nivo je gotov kada Pacman pojede sve bele bobice u lavirintu.

Slika 1. – prikaz scene iz igre

CILj PROJEKTA

Cilj projekta je bio stvoriti inteligentnog agenta koji će samostalno naučiti da analizira svoju okolinu i sa određenom uspešnošću rešava jedan nivo Pac-Man igre. U tu svrhu, Pac-Man kreće bez ikakvog znanja o svom svetu – šta je po njega dobro, a šta loše, i vremenom uči na osnovu toga što biva nagrađen za određene akcije.

Kao uzor i inspiracija poslužio je MarI/O - Machine Learning for Video Games, inteligentni agent koji uspešno rešava nivo Super Mario Bros igre.

Takođe, cilj je bio naučiti i primeniti novu metodu sa kojom nismo do sada imali prilike da se upoznamo.

SLIČNA REŠENjA U DOMENU PROBLEMA

U okviru domena izrade AI agenta za prelaženje Pac-Man igre, naišli smo na različite metode izrade, koje su se međusobno razlikovale po vrsti veštačke inteligencije koju su koristile, po ulaznim parametrima i po načinu vrednovanja uspešnosti.

AI METODE

Jedna od metoda koje su korišćene je A*. Da bi se ona implementirala potrebno je definisati heuristiku koja će govoriti koliko je neka putanja dobra u datoj situaciji. S obzirom da za heuristiku mora eksplicitno da se naglasi šta je to dobro, a šta loše, ova metoda se nije poklapala sa našom željom da Pac-Man samostalno nauči da rešava nivo i procenjuje okolinu.

Druga metoda koja je često korišćena je Q-Learning. Ova metoda traži optimalan sled akcija za konačni skup stanja. Međutim, postavlja se pitanje da li je ovde skup stanja konačan. Takođe, problem predstavlja definisanje funkcije koja bi opisala korisnost akcije u svakom stanju. Ni ovom metodom Pac-Man neće sam naučiti da rešava problem. Na kraju, dokazano je da je Q-Learning sporiji u nekim situacijama od NEAT metode[1].

Posle analize mogućih metoda, odlučeno je da se radi NEAT metoda, koja je detaljno opisana u istoimenom poglavlju.

ULAZI

Kao i kod AI metoda, i ovde smo naišli na različita rešenja. Dva moguća pristupa su:

- Unutrašnji podaci iz igre
- Mapa okoline

Za unutrašnje podatke iz igre se mogu uzeti informacije kao što su PacMan-ova pozicija na mapi, lokacije ili udaljenosti od duhova, lokacija najbliže bobice i slično.

Mapa okoline predstavlja ono što i čovek može da vidi igrajući igru. Različiti elementi na mapi se predstavljaju različitim simbolima, na primer, različitim numeričkim vrednostima za različite elemente.

Koji od ova dva načina predstavljanja okoline Pac-Man-u će biti odabran prevashodno zavisi od odabrane AI metode. S obzirom da je odlučeno da se radi NEAT metodom, za ulaz je odabrana mapa okoline, pri čemu se ona mora ograničiti radi ubrzanja procesa učenja.

Mapa okoline se može generisati na više načina – OCR-om, ručnim kodiranjem mape i direktnim pristupom RAM memoriji. Problem kod OCR-a je dugotrajnost zbog česte obrade slike, koja nije poželjna za on-line obučavanje agenta. Ručno kodiranje mape je dugotrajno i primenjivo samo na jedan nivo (hard coding), što ga čini veoma ograničenim. Poslednja opcija, koja je ujedno i naš izbor, jeste direktan pristup RAM memoriji, koji je brz i može se primeniti na različitim mapama.

Način na koji su predstavljeni elementi mape, kao i okolina koja je odabrana, detaljno su opisani u poglavlju *Koraci implementacije – ulazi u neuronsku mrežu*.

NEAT METODA

UVOD U NEAT

Neuroevolucija je veštačka evolucija neuronskih mreža koja koristi genetske algoritme. Ona pretražuje prostor ponašanja za mrežu koja dobro izvršava zadati zadatak. Zbog toga što neuronska mreža traži ponašanje umesto vrednosne funkcije, efikasna je u rešavanju problema sa kontinualnim i višedimenzionalnim prostorima stanja.

Kod standardnih evolutivnih neuronskih mreža, struktura se određuje pre početka eksperimenta i to je obično mreža sa jednim skrivenim slojem. Evolucijom se traže najpogodnije težine veza između neurona, koje određuju funkcionalnost mreže.

Međutim, osim težina veza, i struktura neuronske mreže je ta koja određuje njenu funkcionalnost.

Neuroevolucija rastućih topologija (NEAT) je dizajnirana tako da iskoristi strukturu na takav način da se minimizira dimenzionalnost prostora pretraživanja težina veza. Ako struktura evoluira na takav način da se struktura minimizira i inkrementalno povećava, značajno se uvećava brzina učenja [1].

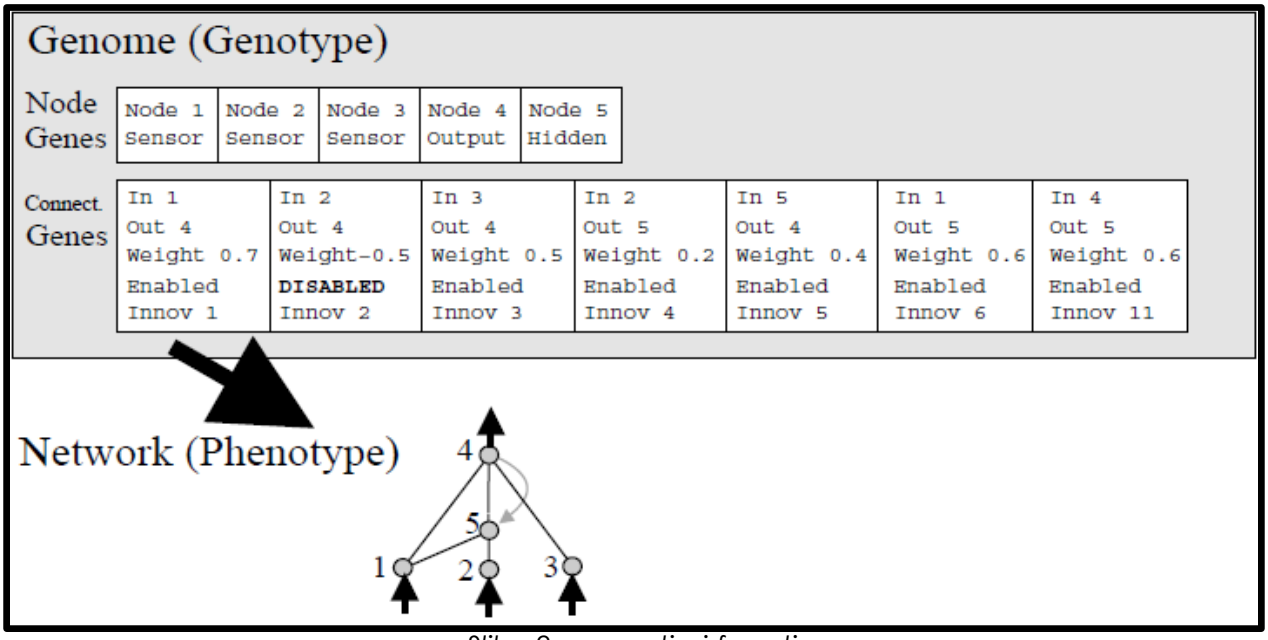
NEAT minimalnu dimenzionalnost obezbeđuje tako što počinje sa uniformnom populacijom mreža bez skrivenog sloja. Strukture rastu inkrementalno tako što dolazi do mutacija, pri čemu preživljavaju samo one strukture koje se pokažu korisnim kroz evaluaciju fitnes funkcijom.

GENETSKO ENKODIRANJE

NEAT metoda koristi direktno enkodiranje, što znači da šema enkodiranja u genomu specificira svaku vezu i svaki čvor koji će se pojaviti u fenotipu. Genom, odnosno genotip, u NEAT metodi predstavlja skup svih gena – gena čvorova i gena veza. Fenotip je sama neuronska mreža (Slika 2)

Gen čvora poseduje informacije o tome koji čvor predstavlja, i koja je to vrsta čvora – ulazni, izlazni i čvor skrivenog sloja.

Gen veze poseduje sledeće informacije o vezi između dva čvora. Te informacije su: koji ulazni čvor, koji je izlazni čvor, težina veze, da li je veza uključena, i poreklo gena – *History marking,* o kome će biti reči u istoimenom poglavlju.

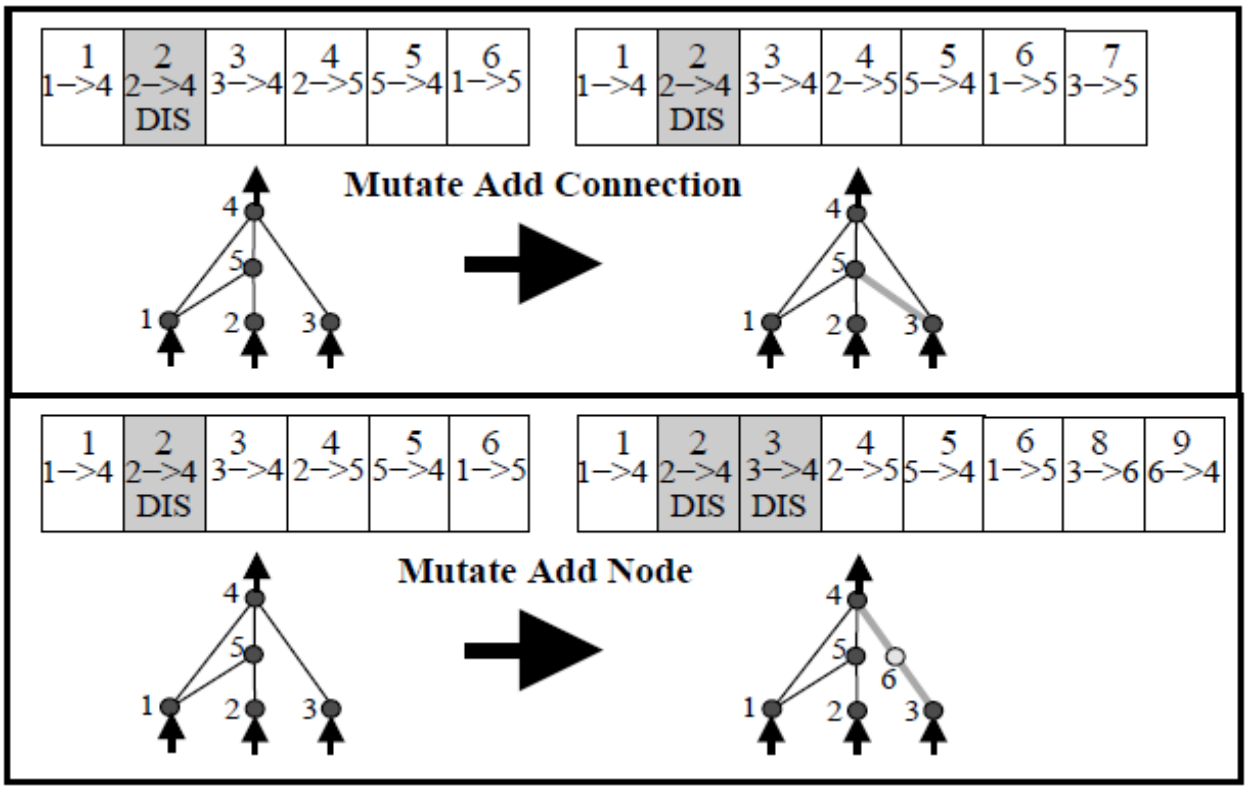


Slika 2 – genotip i fenotip

MUTACIJE

Mutacije u NEAT metodi mogu da promene i težine veza i strukture mreža. Mutacije mreža se dešavaju tako što se svaka veza pobudi (promeni težinu) ili ne, u svakoj generaciji. Strukturne mutacije se dešavaju na dva načina – dodavanjem veze ili dodavanjem čvora (Slika 2).

Pri dodavanju nove veze, u genom se dodaje novi gen veze, koji povezuje dva prethodno nepovezana gena čvora, sa sa slučajnom težinom veze. Pri dodavanju novog čvora, postojeća veza se deli na dva, a novi čvor se dodaje na mesto stare veze. Stara veza se isključuje, i dodaju se dva nova gena veze u genom. Nova veza koja ulazi u novi čvor dobija težinu 1, dok nova veza koja izlazi iz novog čvora dobija težinu stare veze. Ovakvo ubacivanje za cilj ima smanjenje prvobitnog efekta mutacije.



Slika 3 - mutacije

HISTORY MARKINGS

History markings, odnosno oznake porekla se koriste kako bi se u strukturno raznovrsnoj populaciji utvrdilo poklapanje gena. Poklapanje gena je bitna karakteristika, neophodna kako bi se uspešno moglo realizovati *ukrštanje jedinki*. Dva gena sa istom oznakom porekla moraju predstavljati istu strukturu (gen veze), mada sa mogućom razlikom u težini. To je posledica toga što su izvedene iz istog gena pretka u nekom trenutku u prošlosti.

Oznaka porekla je predstavljena inovacionim brojem, koji predstavlja redni broj pojavljivanja gena u sistemu – svaki put kada se pojavi novi gen, globalni inovacioni broj se povećava, i dodeljuje se novom genu.

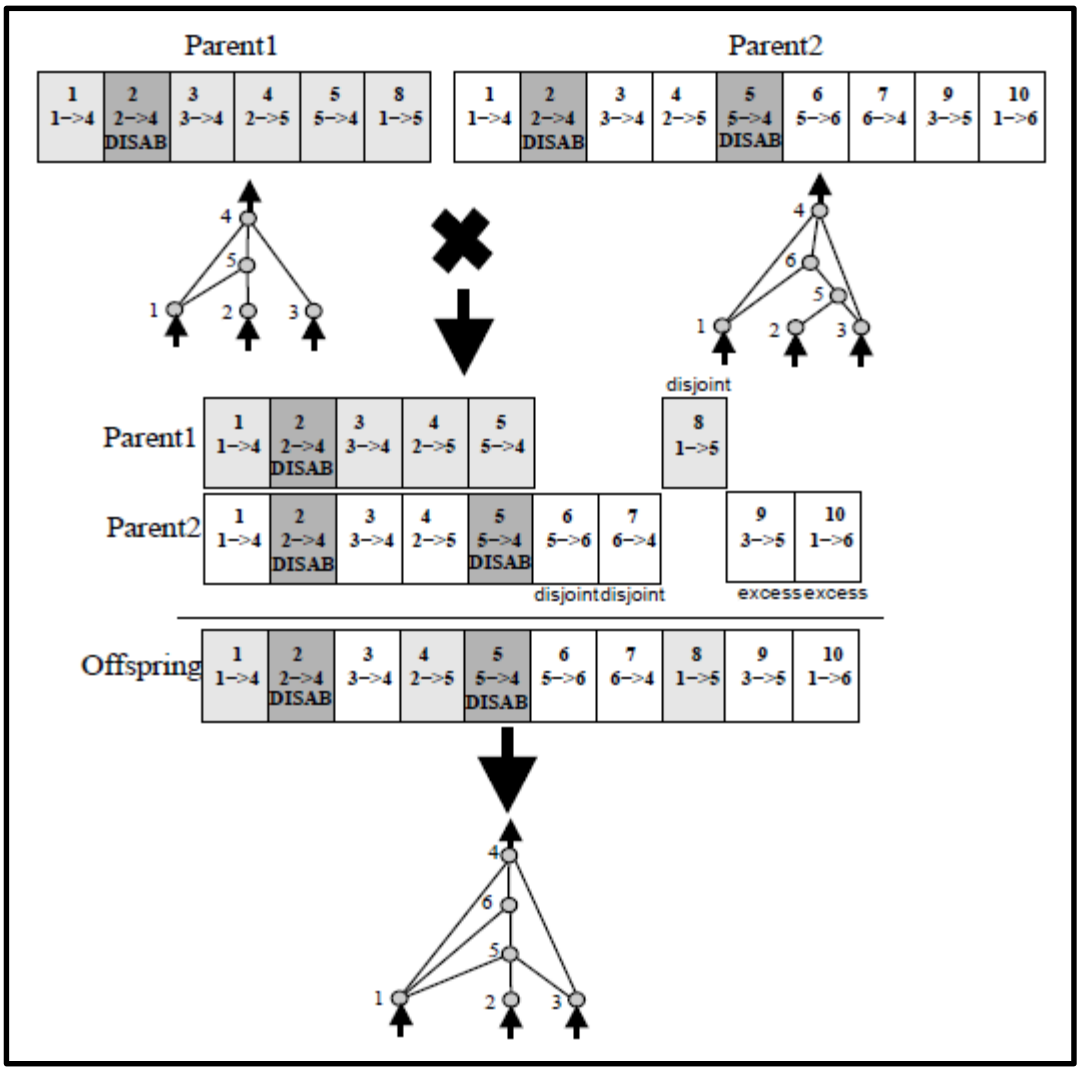
Problem koji se pri ovakvom označavanju može pojaviti jeste da se identične strukturne inovacije dobiju različit inovacioni broj u istoj generaciji. Međutim, čuvanjem liste inovacija koje su se dogodile u trenutnoj generaciji rešava se i ovaj problem.

Ovime NEAT metoda dobija snažnu osobinu – sistem sada zna tačno koji gen da upari sa kojim.

UKRŠTANjE JEDINKI

Ukrštanje se u NEAT metodi odvija tako što se geni veza roditeljskih jedinki postave tako da im se poklapaju inovacioni brojevi. Ovakvi geni geni se nazivaju *poklapajući (matching) geni*. Geni koji nemaju svog para mogu biti *neupareni (disjoint) geni* ili *geni viška (excess)*, u zavisnosti od svoje pozicije u nizu. Geni koji nemaju svog para, a nalaze se u sredini niza su neupareni geni, dok se geni bez para na krajevima niza nazivaju geni viška.

Pri ukrštanju jedinki, potomak će poklapajuće gene od roditelja naslediti slučajnom metodom. Sa druge strane, neupareni geni i geni viška se nasleđuju od roditelja sa većim fitnesom. U slučaju jednakog fitnesa oba roditelja, ovi geni se takođe nasleđuju nasumično.



Slika 4 – ukrštanje jedinki sa istom vrednošću fitnes funkcije

Dodajući nove gene u populaciju i smisleno ukrštajući genome koji predstavljaju različite strukture, sistem može da formira populaciju vrlo raznolikih struktura.

SPECIJACIJA

Iako se opisanim metodama mutacije i ukrštanja dobija raznolika populacija, ispostavlja se da takva populacija ne može sama po sebi održavati topološke inovacije. Razlog tome je to što se manje strukture brže optimizuju od većih struktura, kao i to što dodavanje novih čvorova u strukturu inicijalno smanjuje fitnes jedinke. To znači da te jedinke imaju male šanse za preživljavanje duže od jedne generacije, iako inovacije koje one nose sa sobom na duže staze mogu biti ključne za rešavanje zadatog problema. Rešenje za ovaj poprilično veliki problem jeste *specijacija*.

Podela populacije na vrste dozvoljava jedinkama da se takmiče sa jedinkama svoje vrste, a ne sa celom populacijom. Na ovaj način, strukturne inovacije su zaštićene, tako da imaju vremena da optimiziju svoje strukture. Populaciju je neophodno podeliti u vrste tako da se slične strukture nalaze u istoj vrsti. Oznake porekla igraju bitnu ulogu u razvrstavanju jedinki.

Broj neuparenih gena i gena viška između dva genoma je prirodna mera njihove različitosti. Što manje imaju parova, to im je manja zajednička prošlost, i manje su kompatibilni. Mera različitosti dve jedinke se računa sledećom formulom:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \overline{W}$$

δ je mera različitosti dve jedinke, E broj gena viška, D broj neuparenih gena, W prosečna razlika težina uparenih gena, uključujući i isključene gene, N broj gena u većem genomu, a c1, c2 i c3 koeficijenti koji nam dozvoljavaju da podesimo značaj tri faktora.

Mera različitosti jedinki δ nam omogućava specijaciju korišćenjem praga podudarnosti δ_t. Održava se uređena lista vrsta. U svakoj generaciji, genome se sekvencijalno razvrstavaju. Svaka postojeća vrsta je predstavljena nasumičnim genomom iz vrste prethodne generacije. Genom *g* iz aktuelne generacije stavlja se u prvu vrstu sa čijim je predstavnikom kompatibilan. Na ovaj način, vrste se ne preklapaju. Ukoliko *g* nije kompatibilan ni sa jednim predstavnikom vrste, stvara se nova vrsta sa njim kao predstavnikom.

Kao mehanizam reprodukcije kod NEAT metode koristi se eksplicitno deljenje fitnesa (*explicit fitness sharing*), što znači da organizmi iste vrste moraju da dele fitnes cele vrste. Na taj način, vrste su sprečene da postanu prevelike i preplave populaciju, čak iako se većina jedinki vrste pokaže dobrim.

Prepravljeni fitnes jedinke *i,* u oznaci f_i', računa se na osnovu mere različitosti δ u odnosu na svaku drugu jedinku *j* u populaciji. Funkcija deljenja *sh* je postavljena na 0 kada je mera različitosti između jednki *i* i *j* iznad praga podudarnosti δ_t. U suprotnom dobija vrednost 1. Suma rezultata funkcija deljenja svih drugih jedinki *j* smanjuje broj jedinki u istoj vrsti u kojoj je organizam *i*.

$$f_i' = \frac{f_i}{\sum_{j=1}^{n} \mathrm{sh}(\delta(i,j))}$$

Svakoj vrsti se dodeljuje potencijalno drugačiji broj potomaka u proporciji sa sumom prepravljanih fitnesa njenih jedinki članica. Vrste se dalje ukrštaju najpre eliminišući najslabije članove populacije. Cela populacija se zatim zamenjuje potomcima preostalih organizama u vrsti.

KORACI IMPLEMENTACIJE

NEAT metoda je implementirana skriptom napisanom u Lua programskom jeziku i namenjeno je njeno pokretanje u BizHawk emulatoru u skladu sa uputstvom navedenim u uvodnom poglavlju.

ULAZI U NEURONSKU MREŽU

Za ulaz u neuronsku mrežu odabrana je Pac-Man-ova okolina, odnosno generiše se matrica veličine 11x11 u čijem je centru u svakom trenutku Pac-Man, što znači da se ulazni sloj neuronske mreže sastoji od 121 čvora.

Elementi Pac-Man-ove okoline mapirani su korišćenjem podataka iz RAM memorije igre. Oni su predstavljeni na sledeći način:

- Barijera – „-1“
- Pac-Man – „0“
- Prazno polje – „1“
- Bobica i powerpill – „2“
- Duhovi – „3“

U slučaju konzumiranja powerpill-a, u periodu dok Pac-Man može da jede duhove, oni se uklanjaju sa ulaza kako se ne bi poremetilo njihovo značenje neuronskoj mreži.

IZLAZI IZ NEURONSKE MREŽE

Izlaze neuronske mreže čine četiri moguća smera Pac-Man-ovog kretanja koji se preslikavaju na kontrole upravljača, što znači da se izlazni sloj neuronske mreže sastoji od četiri čvora.

Pac-Man na svakih 50 frejmova prikaza analizira okolinu i reaguje na nju. Ova brojka je odabrana eksperimentalnom metodom.

Radi ubrzanja obučavanja, Pac-man umire svaki put kada se kroz 150 frejmova zadrži na istoj poziciji.

FITNES FUNKCIJA

S obzirom da je fitnes funkcija ta koja treba da odredi valjanost jedinke, postavlja se pitanje od kojih parametara ona treba da zavisi i u kojoj meri. Mogući parametri u Pac-Man-u su broj pojedenih bobica, sakupljeni poeni, dužina trajanja partije i slično.

U rešenju predstavljenom ovim projektom, odlučeno je da se u obzir uzme samo broj pojedenih bobica. Ova odluka proizilazi iz želje da se Pac-Man fokusira na prelaženje nivoa radije nego na skupljanje poena. Vremenska komponenta nije uzeta u obzir da bi se naglasilo skupljanje bobica kao važnije od izbegavanja duhova.

REZULTATI I ANALIZA

Uspešnost obučavanja se analizira na osnovu sledećih stavki:

- Uspešnost najbolje jedinke obučavanja
- Prosečne uspešnosti svake generacije, kako bi se sagledao generacijski napredak populacije
- Brzina rasta uspešnosti populacije po generacijama

PARAMETRI OBUČAVANJA

Na osnovu testnih obučavanja, uočeno je da se nakon određenog broja generacija jedinke iz iste vrste počinju jednolično ponašati, odnosno da je neophodna veća raznovrsnost kako bi se uvele nove inovacije, koje bi potencijalno mogle doprineti rešavanju problema.

Parametri za koje je pretpostavljeno da bi mogli da utiču na rešavanje ovog problema su sledeći: veličina populacije, šansa za mutaciju težine veze, šansa za ukrštanje, kao i šansa za mutaciju dodavanjem novog gena čvora. Na osnovu ovoga osmišljene su 2 različita obučavanja, opisana u narednim poglavljima.

Testno obučavanje je predstavljeno u poglavlju *OBUČAVANJE I,* i korišćeno je kao uporedno u odnosu na dva obučavanja sa izmenjenim parametrima.

Sva obučavanja su trajala više 30 sati, pri čemu je neophodno napomenuti da bi se uspešnosti najboljih jedinki možda razlikovale da je vreme obučavanja duže, ali da za takvo obučavanje nije bilo vremena.

OBUČAVANJE I

Parametri:

- Veličina populacije – 300 jedinki
- Šansa za mutaciju težine veze – 25%
- Šansa za ukrštanje dve jedinke – 75%
- Šansa za dodavanje novog gena čvora – 50%

Rezultati obučavanja:

- Broj generacija obučavanja – 125
- Uspešnost najbolje jedinke – 107/192 bobica (55.7%)
- Generacija pojavljivanja najbolje jedinke – 110
- Prosečna uspešnost po generaciji – Tabela 1.

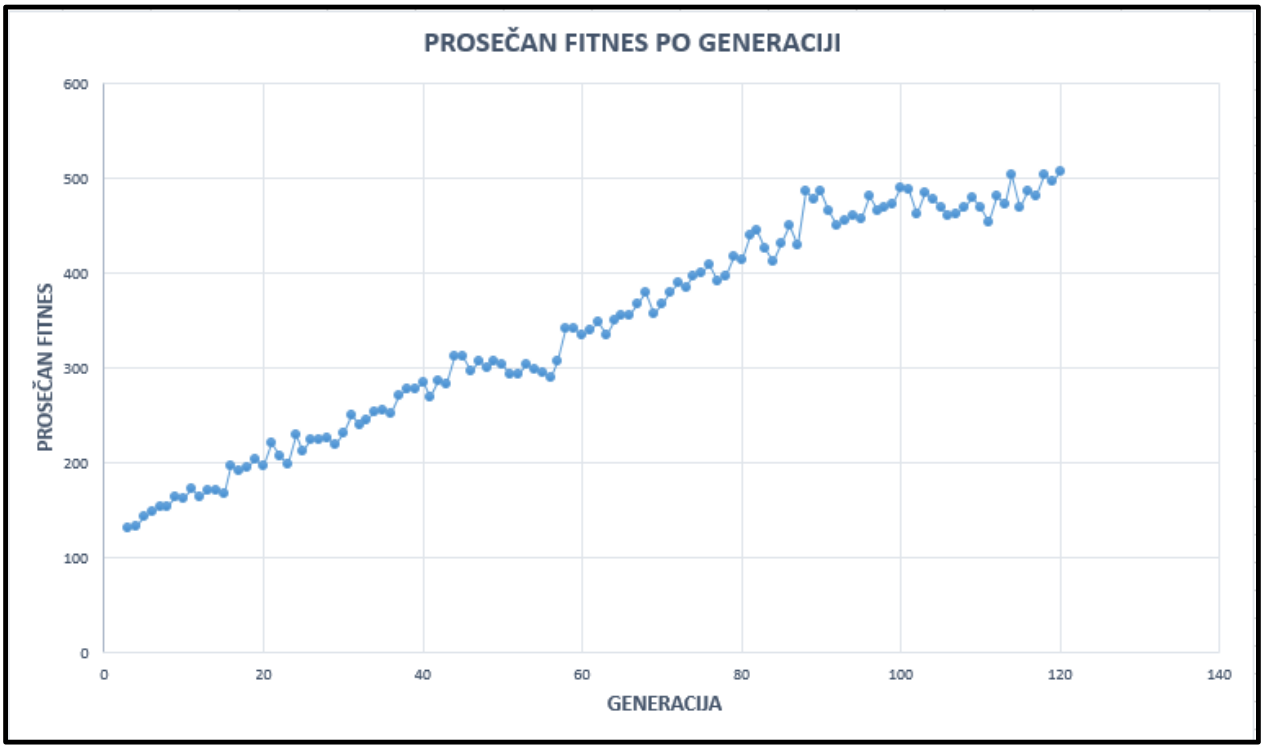


Tabela 1 – prosečan fitnes po generaciji od 0. do 120. generacije

OBUČAVANJE II

Parametri:

- Veličina populacije – 500 jedinki
- Šansa za mutaciju težine veze – 35%
- Šansa za ukrštanje dve jedinke – 70%
- Šansa za dodavanje novog gena čvora – 50%

Rezultati obučavanja:

- Broj generacija obučavanja – 145
- Uspešnost najbolje jedinke – 116/192 bobica (60.04%)
- Generacija pojavljivanja najbolje jedinke – 142
- Prosečna uspešnost po generaciji – Tabela 2.

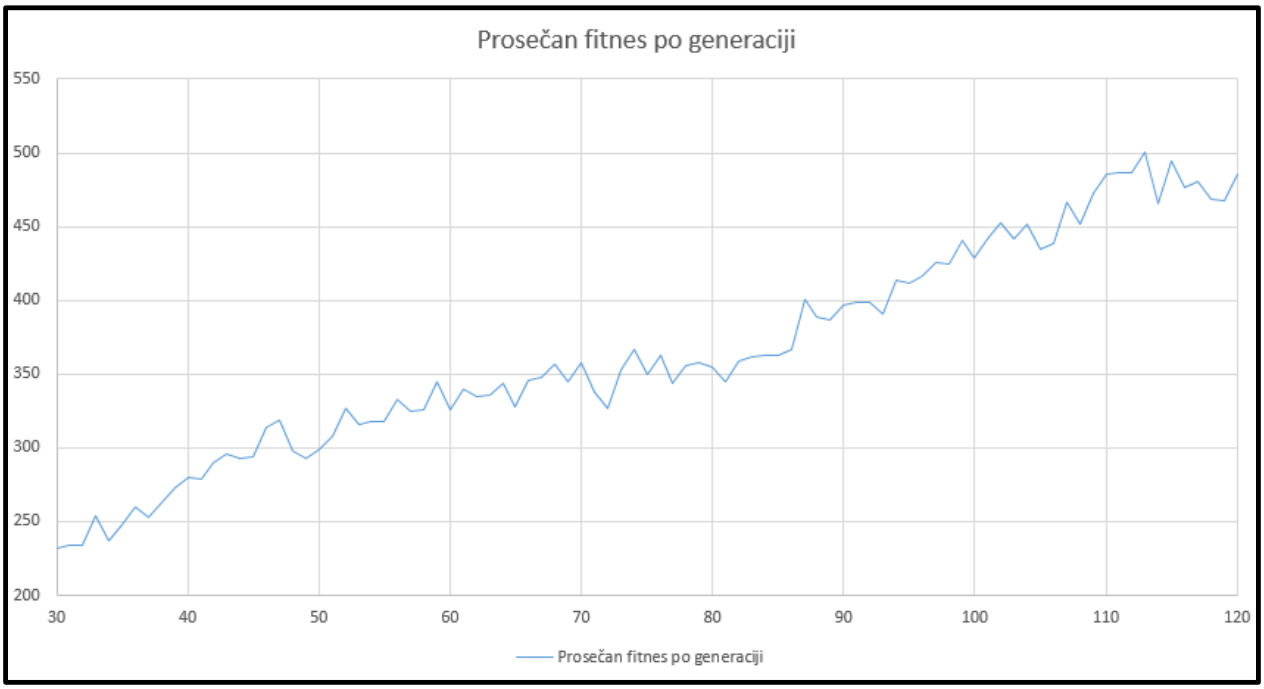


Tabela 2 – prosečan fitnes po generaciji od 30. do 120. generacije

OBUČAVANJE III

Parametri:

- Veličina populacije – 400 jedinki
- Šansa za mutaciju težine veze – 50%
- Šansa za ukrštanje dve jedinke – 75%
- Šansa za dodavanje novog gena čvora – 75%

Rezultati obučavanja:

- Broj generacija obučavanja – 125
- Uspešnost najbolje jedinke – 110/192 bobica (57.3%)
- Generacija pojavljivanja najbolje jedinke – 118
- Prosečna uspešnost po generaciji – Tabela 3.

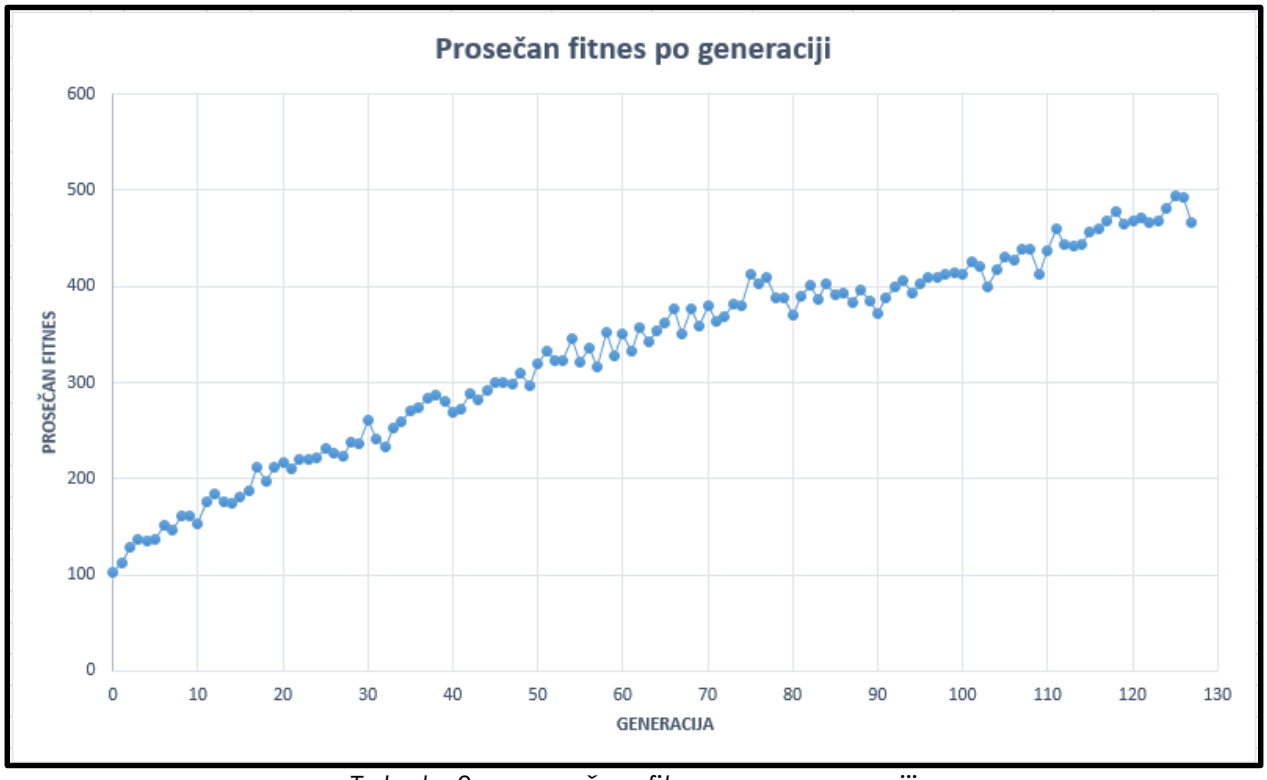


Tabela 3 – prosečan fitnes po generaciji od 0. do 125. generacije

ANALIZA DOBIJENIH REZULTATA

Prilikom analize rezultata obučavanja Pac-Man agenta, uočene su pravilnosti koje se ponavljaju pri svakom obučavanju, nezavisno od vrednosti već pomenutih parametara. Prosečni fitnesi su u konstantnom, ali sve sporijem rastu (poput logaritamske ili korenske funkcije), i kod svakog obučavanja je prag od prosečnog fitnesa 500 po generaciji dostignut između 110. i 130. generacije. Takođe, uočeno je i da se maksimalni dostignuti fitnes pri obučavanju uvek kreće između 55-60% od najvećeg mogućeg fitnesa. Nažalost, jedinka koja dostigne taj maksimum nakon toga iz generacije u generaciju ponavlja svoje ponašanje, ne uspevajući da se poboljša, a na kraju i potpuno promeni svoje ponašanje zbog mutacije koja se loše odrazi na njeno ponašanje.

Najbolji rezultat je dostignut u Obučavanju II, i iznosi 116 pojedenih bobica od 192, odnosno 60% od najvećeg mogućeg fitnesa. Analizom putanja agenta, procenjeno je da bi, u perspektivi, najviši dostignuti rezultat mogao da dosegne oko 125 bobica, odnosno 65.1%.

LITERATURA

- [1] – Evolving Neural Networks through Augmenting Topologies, Kenneth O. S, Risto M.
- [2] – Pac-Man USA Namco, 02. FEBRUAR 2016.
- [3] – Infinite Mario AI using A* Search – Baumgarten R.