

# Hibrid neuronske mreže, genetskog algoritma i reinforcement learninga za igranje top-down shooter igre

Seminarski rad u okviru kursa  
Računarska Inteligencija  
Matematički fakultet

Daković Branko, Filip Kristić, Krčmarević Mladen  
brankodjakovic08@gmail.com, filip.kristic96@gmail.com  
mladenk@twodesperados.com

31. avgust 2019.

## Sažetak

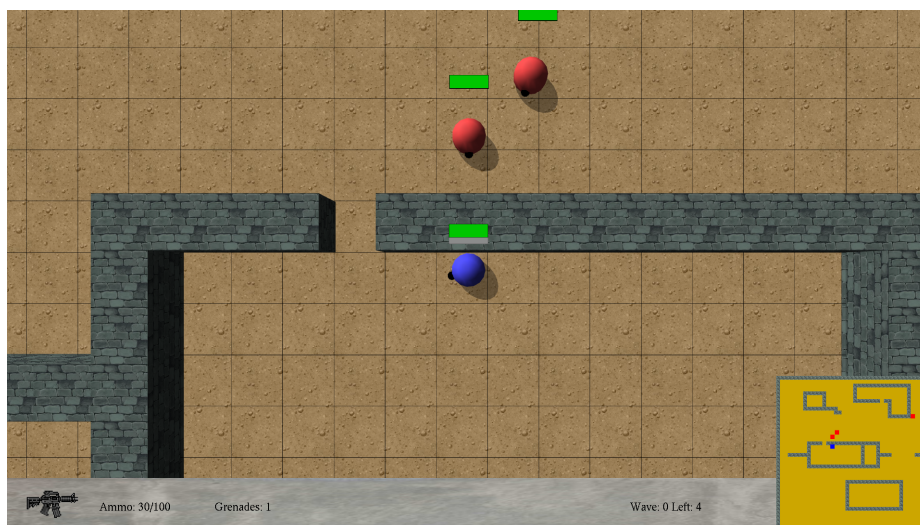
U radu će biti prikazana upotreba reinforcement learninga i genetskog algoritma za generisanje težina neuronske mreže koja igra top-down shooter igru "Shrodinger's shooter". Ovaj pristup spada u Neuro-evoluciju[2], sa tim što se menjaju samo težine neruonske mreže dok je struktura fiksna. Za pisanje kodova je korišćen programski jezik C++ i biblioteka FANN sa omotačem za C++. Biće dat kratak opis igre kao i dva rešenja koja se razlikuju po načinu reprezentacije ulaza za neuronsku mrežu.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Opis rešenja</b>	<b>2</b>
2.1	Selekcija, ukrštanje i mutacija . . . . .	4
<b>3</b>	<b>Uporedjivanje rešenja</b>	<b>5</b>
<b>4</b>	<b>Zaključak</b>	<b>6</b>
	<b>Literatura</b>	<b>6</b>

## 1 Uvod

“Shrodinger’s shooter” je shooter igra koju smo razvili za predmet Razvoj Softvera u kojoj je cilj igrača da što duže preživi nalete protivnika. Igrica koristi 2D fiziku uz 3D grafiku a mapa je kvadratnog oblika i sadrži zidove. Moguće akcije igrača su: idi gore, dole, levo, desno kao i dijagonalno kretanje njihovom kombinacijom, podešavanje pozicije nišana, repetiranje oružja i pucanje. Zbog kompleksnosti su izbačeni itemi poput pancira i različitog oružja. Cilj neuronske mreže[3] je da na osnovu trenutne situacije igrača da njegovu sledeću akciju.



Slika 1: Schrodinger’s shooter

## 2 Opis rešenja

Program ima tri moda izvršavanja:

```
./Schshooter.out
```

koji obuhvata učitavanje rešenja u vidu neuronske mreže i igranje igrice uz grafički prikaz,

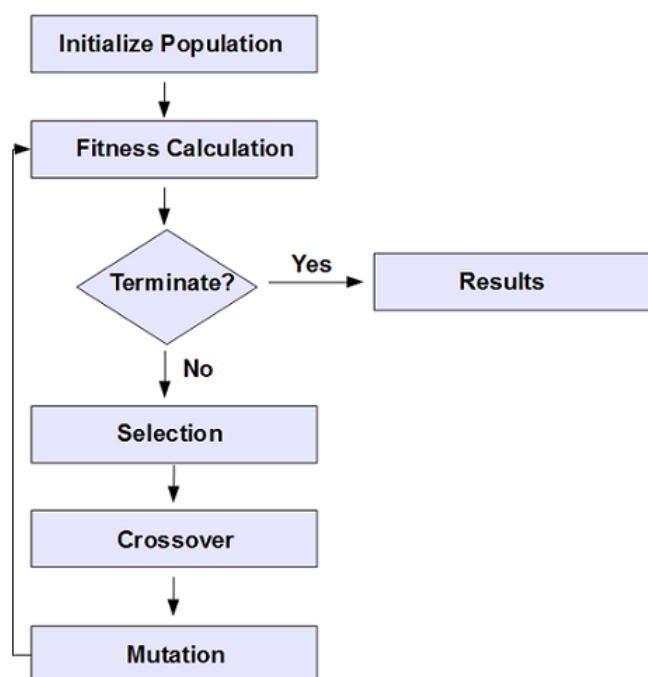
```
./Schshooter.out -t
```

koji vrši treniranje bez grafičkog prikaza i

```
./Schshooter.out -tv
```

koji vrši treniranje sa grafičkim prikazom.

Trening započinje kreiranjem početne generacije genetskog algoritma<sup>[1]</sup> (slika 2.<sup>1</sup>) kod koje svaki hromozom ima fitness jednak nuli. Hromozomi su predstavljeni sadržajem tj. nizom brojeva u pokretnom zarezu koji predstavlja težine svih veza mreže i jednim brojem u pokretnom zarezu koji predstavlja fitness tog hromozoma. Težine se uzimaju nasumično iz intervala  $(-10, 10)$  koji je takođe nasumično izabran. Broj hromozoma u generaciji kao i broj generacija su izabrani uzimajući u obzir veličinu hromozoma te iznose nekoliko stotina u prvom rešenju a nekoliko hiljada u drugom. O ulazu će više biti rečeno u narednom poglavlju.



Slika 2: Genetski algoritam

Nakon kreiranja generacije redom se uzimaju hromozomi i težine veza neuronske mreže se postavljaju na sadržaj hromozoma. Veličina ulaznog sloja u prvom rešenju je 191 a 10 u drugom, oba rešenja imaju samo jedan skriven sloj koji je veličine 10 u prvom zbog ogromnog broja veza usled veličine ulaza i 15 u drugom. Za aktivacionu funkciju je korišćena linearna aproksimacija sigmoidne funkcija. Sigmoidna funkcija je oblika<sup>2</sup>

<sup>1</sup>Slika preuzeta sa: <https://apacheignite.readme.io/docs/genetic-algorithms>

<sup>2</sup>Preuzeto iz dokumentacije fann biblioteke. [http://leenissen.dk/fann/html/files/fann\\_cpp-h.html](http://leenissen.dk/fann/html/files/fann_cpp-h.html)

ISPRAVI  
VE-  
LIČINE

x - ulaz  
 y - izlaz  
 s - nagib i  
 d - derivacija  
 raspon:  $0 < y < 1$

$$y = \frac{1}{1 + e^{-2 \times s \times x}} \quad (1)$$

$$d = 2 \times s \times y \times (1 - y) \quad (2)$$

dok stopa učenja nije podešena pošto nema nikakvog uticaja na dati problem. Izlazni sloj sadrži pet neurona čiji izlazi imaju vrednosti u intervalu  $[0,1]$  a koji regulišu kretanje gore-dole, levo-desno, treći ugao pod kojim igrač nišani, da li da puca i da li da dopuni municiju. Prva dva izlaza su podeljena u tri intervala  $[0,0.33]$ ,  $(0.33-0.66]$  i  $(0.66-1]$  koji redom odgovaraju kretanju gore(levo), bez kretanja i dole(desno), treći izlaz se skalira do 360 i ugao igrača se postavlja na datu vrednost a četvrti i peti daju potvrđan odgovor za vrednosti manje od 0.5 a negativan u suprotnom. U svakoj iteraciji programa se ažuriraju pozicija i akcije igrača u zavisnosti od izlaza mreže. Fitness svakog hromozoma se računa na osnovu broja eliminisanih protivnika i količine štete koje su naneli protivnicima po narednoj formuli a najbolje ocenjen hromozom se čuva.

$$\text{fitness} = \text{eliminacije} * 50 + \text{šteta} * 0.5$$

## 2.1 Selekcija, ukrštanje i mutacija

Po obradi svih hromozoma generacije, ukoliko ima još iteracija, vrši se selekcija hromozoma koji će učestvovati u izradi nove generacije. Selekcija se vrši ruletskim pristupom, tako što se računa zbir fitness-a svih hromozoma i svaki ima šansu da bude izabran srazmerno odnosu njegovog i celokupnog fitnessa. Nakon izbora hromozoma koji učestvuju u reprodukciji nasumično se biraju dva roditelja i vrše se ukrštanje i mutacija koji su implementirani na različite načine u prvom i drugom rešenju. Svaki par roditelja kreira dva deteta i to se vrši sve dok ne bude ispunjena nova populacija.

Ukrštanje u prvom rešenju:

i - nasumičan broj od 0 do veličine sadržaja hromozoma;  
 detel = sadržaj roditelja1 do i + sadržaj roditelja2 od i do kraja;  
 dete2 = sadržaj roditelja2 do i + sadržaj roditelja1 od i do kraja;

Mutacija u prvom rešenju:

t - nasumičan broj u pokretnom zarezu iz intervala  $[0,1]$ ;  
 Ukoliko je t manje od stope mutacije:  
 Promeni nasumičan element sadržaja hromozoma;

Ukrštanje u drugom rešenju:

Za svaki element  $i$  sadržaja hromozoma uradi:

$t$  - nasumičan broj u pokretnom zarezu iz intervala  $[0,1]$ ;

Ako je  $t < 0.5$ :

$dete1[i] = rodite1[i]$ ;

$dete2[i] = rodite2[i]$ ;

Inače:

$dete1[i] = rodite2[i]$ ;

$dete2[i] = rodite1[i]$ ;

Mutacija u drugom rešenju:

Za svaki element  $i$  sadržaja hromozoma uradi:

$t$  - nasumičan broj u pokretnom zarezu iz intervala  $[0,1]$ ;

Ukoliko je  $t$  manje od stope mutacije:

Promeni  $i$ -ti element sadržaja hromozoma;

Celokupan navedeni postupak se zatim obavlja dok nije zadovoljen kriterijum zaustavljanja, tj. dok se ne premaši zadati broj iteracija. Najbolja jednika kao i svi hromozomi poslednje generacije se čuvaju u vidu neuronskih mreža u tekstualnim datotekama radi čuvanja progressa i nastavka treniranja.

### 3 Uporedjivanje rešenja

## 4 Zaključak

### Literatura

- [1] dr Predrag Janičić and dr Mladen Nikolić. *Veštačka Inteligencija*. Beograd, Srbija, 2019.
- [2] Stanley O. Kenneth, Clune Jeff, Lehman Joel, and Risto Miikkulainen. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 2019. on-line at: <http://www.evolvingai.org/files/s42256-018-0006-z.pdf>.
- [3] Kantardzic Mehmed. *Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition: Concepts, Models, Methods, and Algorithms, Second Edition*. 2011.