

# **Neuroevolution**

## **Snake game**

Projekat iz Računarske inteligencije  
Matematički fakultet  
Univerzitet u Beogradu

Jovan Škorić 362/2020  
Nikola Kuburović 369/2020

# Sadržaj

<b>1 Uvod</b>	<b>3</b>
1.1 Opis problema . . . . .	3
1.2 Pregled literature . . . . .	3
<b>2 Opis rešenja</b>	<b>3</b>
2.1 Opšti rad algoritma . . . . .	3
2.2 Način kodiranja rešenja . . . . .	3
<b>3 Neuronska mreža</b>	<b>4</b>
3.1 Opis . . . . .	4
3.2 Struktura neuronske mreže . . . . .	4
3.3 Rad mreže . . . . .	5
3.4 Trening mreže . . . . .	5
<b>4 Genetski algoritam</b>	<b>5</b>
4.1 Generisanje rešenja . . . . .	5
4.2 Selekcija . . . . .	6
4.3 Reprodukција . . . . .	6
4.4 Evolucija . . . . .	7
<b>5 Eksperimentalni rezultati</b>	<b>7</b>
5.1 Eksperimentalno okruženje . . . . .	7
5.2 Metodologija . . . . .	7
<b>6 Zaključak</b>	<b>7</b>
6.1 Kritički osvrt . . . . .	7
6.2 Pravci daljeg istraživanja . . . . .	7
<b>7 Literatura</b>	<b>8</b>

# 1. Uvod

## 1.1 Opis problema

Projekat se bavi razvojem veštačke inteligencije koja uči da igra igricu "Zmijica" koristeći neuronske mreže i genetske algoritme. "Zmijica" je klasična video igra u kojoj igrač kontroliše zmiju koja raste kada jede jabuke i mora izbegavati zidove i sopstveno telo kako bi nastavila igru. Problem koji se rešava je kako optimizovati kretanje zmije da maksimalno poveća broj poena pre nego što udari u zid ili sopstveno telo.

## 1.2 Pregled literature

Mnoge studije su se bavile optimizacijom agenata za igranje igara koristeći različite metode mašinskog učenja. Istraživanja pokazuju da su neuronske mreže i genetski algoritmi efikasni za ovaj tip problema. Na primer, radovi poput "Neuroevolution of Augmenting Topologies" (NEAT) pokazuju kako evolucija neuronskih mreža može proizvesti složena ponašanja agenata u igrama.

# 2. Opis rešenja

## 2.1 Opšti rad algoritma

Algoritam koristi kombinaciju neuronske mreže i genetskog algoritma kako bi optimizovao ponašanje zmije. Neuronska mreža prima podatke o onome što zmija trenutno vidi na tabli (udaljenost do zida, jabuke i tela zmije) i odlučuje o sledećem potezu zmije. Genetski algoritam optimizuje težine neuronske mreže kroz selekciju, ukrštanje i mutaciju najboljih jedinki u populaciji.

## 2.2 Način kodiranja rešenja

Neuronska mreža je implementirana u fajlu ***neural\_network.py*** i sastoji se od četiri sloja: ulaznog sloja sa 24 neurona, skrivenog sloja sa 16 neurona, skrivenog sloja sa 8 neurona i izlaznog sloja sa 4 neurona, sa aktivacionim funkcijama ReLU, Sigmoid i Softmax.

Genetski algoritam je implementiran u fajlu **ga.py**. On kreira populaciju zmija, procenjuje njihovu fitnes funkciju, selektuje najbolje jedinke i primenjuje operacije ukrštanja i mutacije za generisanje nove populacije.

Zmijica je definisana u fajlu **snake.py**, gde su implementirane funkcije za kretanje zmije, generisanje jabuka i procenu udaljenosti do prepreka. Fitnes funkcija nagrađuje zmiju za dužinu života i broj pojedenih jabuka.

### 3. Neuronska mreža

#### 3.1 Opis

Neuronska mreža korišćena u ovom projektu služi za donošenje odluka o kretanju zmije na osnovu trenutne situacije na tabli. Mreža je trenirana korišćenjem podataka o onome sto zmija vidi u odredjenom trenutku. Vid zmije prati udaljenosti zmije od zida, svog tela i jabuke, a njena struktura omogućava da se prilagodi i optimizuje kroz genetski algoritam.

#### 3.2 Struktura neuronske mreže

1. **Ulazni sloj:** Sadrži 24 neurona koji predstavljaju različite aspekte trenutnog stanja igre, kao što su udaljenosti do zidova, jabuke i tela zmije u različitim pravcima.
2. **Skriveni slojevi:**
  1. **Prvi skriveni sloj:** Sadrži 16 neurona koji koriste ReLU aktivacionu funkciju za modelovanje složenih odnosa između ulaznih podataka i odgovora mreže.
  2. **Drugi skriveni sloj:** Sadrži 8 neurona koji koriste Sigmoid aktivacionu funkciju.
3. **Izlazni sloj:** Sadrži 4 neurona, od kojih svaki predstavlja jedan od četiri moguća smera zmije (gore, dole, levo, desno). Ovi neuroni koriste Softmax aktivacionu funkciju kako bi se dobile verovatnoće za svaku akciju.

### 3.3 Rad mreže

1. **Ulazni podaci:** Neuronska mreža prima vektor od 24 vrednosti kao ulaz. Ove vrednosti predstavljaju vid zmije u osam razlicitih smerova, pocevsi s leva i redom u krug u smeru kazaljke na satu. Svaki pogled u odredjenom smeru prati tri stvari: udaljenost od zida, jabuke i tela.
2. **Propagacija unapred:** Ulazni podaci prolaze kroz skriveni sloj, gde se primenjuje ReLU aktivaciona funkcija. Iz skrivenog sloja podaci prolaze do izlaznog sloja, gde se primenjuje Softmax funkcija kako bi se dobile verovatnoće za svaku od mogućih akcija.
3. **Izlaz:** Izlaz neuronske mreže je vektor od četiri verovatnoće, od kojih svaka predstavlja verovatnoću za određeni smer. Smer sa najvećom verovatnoćom se bira kao sledeći potez zmije.

### 3.4 Trening mreže

Neuronska mreža se trenira pomoću genetskog algoritma koji optimizuje težine i pristranosti mreže na osnovu uspešnosti zmije u igri.

1. **Inicijalizacija težina:** Na početku, težine i pristranosti mreže su inicijalizovane nasumično.
2. **Procena performansi:** Performanse mreže se procenjuju na osnovu fitnes funkcije koja uzima u obzir dužinu života zmije i broj pojedenih jabuka.
3. **Genetski operatori:** Težine i pristranosti najboljih mreža se kombinuju i mutiraju kako bi se generisale nove generacije mreža sa poboljšanim performansama.

## 4. Genetski algoritam

### 4.1 Generisanje rešenja

1. **Reprezentacija jedinki:** Svaka jedinka u populaciji predstavlja jednu zmiju. Jedinke su predstavljene kao težine i pristranosti neuronske mreže koja kontroliše kretanje zmije.
2. **Generisanje početne populacije:** Na početku algoritma se generiše početna populacija zmija.
  - **Ulazni parametri:**

- Veličina populacije
  - Dimenzije table
  - **Inicijalizacija jedinki:** Svaka jedinka (zmija) dobija nasumične težine i pristranosti neuronske mreže.
3. **Ocena jedinki (fitnes funkcija):** Fitnes funkcija procenjuje učinak svake zmije na osnovu dužine života i broja pojedenih jabuka.

## 4.2 Selekcija

1. **Izbor roditelja:** Roditelji se biraju na osnovu njihovih fitnes vrednosti. Jedinke sa većim fitnesom imaju veću verovatnoću da budu izabrane.
- **Ulazni parametri:**
    - Populacija: Lista svih zmija u trenutnoj populaciji
    - Broj učesnika turnira (*tournament\_size*)
    - Broj jedinki koje su pobednici turnira (*num\_individuals*)
  - **Biranje roditelja:** Koristi se metoda turnirske selekcije. U petlji sa *num\_individuals* iteracija se izabere grupa od *tournament\_size* jedinki, a zatim se od njih odabere jedna koja ima najveći fitnes.

## 4.3 Reprodukција

1. **Ukrštanje:** Kombinovanje delova roditeljskih jedinki kako bi se generisala nova jedinka.
- **Ulazni parametri:**
    - Roditelji: Neuronske mreže oba roditelja
  - **Proces ukrštanja:** Ukrštaju se odgovarajući slojevi iz oba roditelja. Preciznije, vrši se jednopoziciono ukrštanje, primenjeno na pojedinačne odgovarajuće redove slojeva.
2. **Mutacija:** Menjanje nekih delova jedinke na nasumičan način.
- **Ulazni parametri:**
    - Neuronska mreža jedinke koja se mutira
    - Verovatnoća mutacije: Verovatnoća da dođe do mutacije
  - **Proces mutacije:** Za svaku težinu i pristranost u jedinki proverava se da li će biti mutirana na osnovu verovatnoće mutacije. Ako je generisani broj manji od verovatnoće mutacije, težina ili pristranost se nasumično menja.

## 4.4 Evolucija

1. **Generisanje nove populacije:** Nakon ukrštanja i mutacije generiše se nova populacija zmija.
2. **Kriterijum zaustavljanja:** Algoritam se zaustavlja nakon unapred definisanog broja generacija ili kada fitnes vrednost dostigne određeni prag.
3. **Rezultat algoritma:** Algoritam vraća rezultat i fitnes za najbolju jedinku u trenutnoj generaciji, a takođe prati iste stvari za sve generacije globalno.

## 5. Eksperimentalni rezultati

### 5.1 Eksperimentalno okruženje

- **Hardver:** Intel i7 CPU, 16GB RAM
- **Operativni sistem:** Ubuntu 20.04
- **Kompajler:** Python 3.8, PyTorch 1.7.1

### 5.2 Metodologija

Testiranje je izvršeno na tabli dimenzija 10x10. Algoritam je treniran na 200 generacija sa populacijom od 1000 zmija. Za poređenje performansi, testirane su različite veličine table, različiti algoritmi selekcije

## 6. Zaključak

### 6.1 Kritički osvrt

Implementacija veštačke inteligencije za igricu "zmijica" pokazala je potencijal neuronskih mreža i genetskih algoritama. Ipak, postoje oblasti za unapređenje, kao što su optimizacija hiperparametara i istraživanje drugih metoda.

### 6.2 Pravci daljeg istraživanja

- Integracija naprednijih tehnika mašinskog učenja.
- Eksperimentisanje sa alternativnim algoritmima za optimizaciju.

- Optimizovanje fitness funkcije
- Eksperimentisanje sa različitim brojem skrivenih slojeva u neuronskoj mreži

## 7. Literatura

- Stanley, K. O., & Miikkulainen, Evolving Neural Networks through Augmenting Topologies - "<https://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf>"
- Neuroevolution of augmenting topologies (wikipedia) - "[https://en.wikipedia.org/wiki/Neuroevolution\\_of\\_augmenting\\_topologies](https://en.wikipedia.org/wiki/Neuroevolution_of_augmenting_topologies)"
- "<https://github.com/Chrispresso/SnakeAI>"
- "<https://github.com/ygutgutia/Snake-Game-Genetic-Algorithm>"
- "<https://github.com/greerviau/SnakeAI>"