

Primena neuronske mreže i genetskog algoritma u video igri

Miloš Tepić & Miloš Ribar

Uvod

- Ideja je da napravimo simulaciju trke različitih karaktera u našoj video igri, i da vidimo koji od njih će imati najbolje rezultate, koristeći neuronske mreže i genetski algoritam. Svaka familija karaktera ima različite parametre. Ti parametri su brzina kretanja, sila skoka, težina i uticaj gravitacije na njih. Naši karakteri poseduju 6 senzora koji im omogućavaju da donose odluke o kretanju.



Moguća kretanja su levo, desno i skok.

Opisi algoritama

- Neuronska mreža je jedan od načina implementacije veštacke inteligencije često korišćen u mašinskom učenju. Sastoji se od neurona gde svaki neuron ima n ulaza i jedan izlaz. Naša neuronska mreža poseduje 6 neurona u ulaznom sloju, i svaki ima 6 ulaza, i 2 skrivena sloja. U izlaznom sloju poseduje 3 neurona.

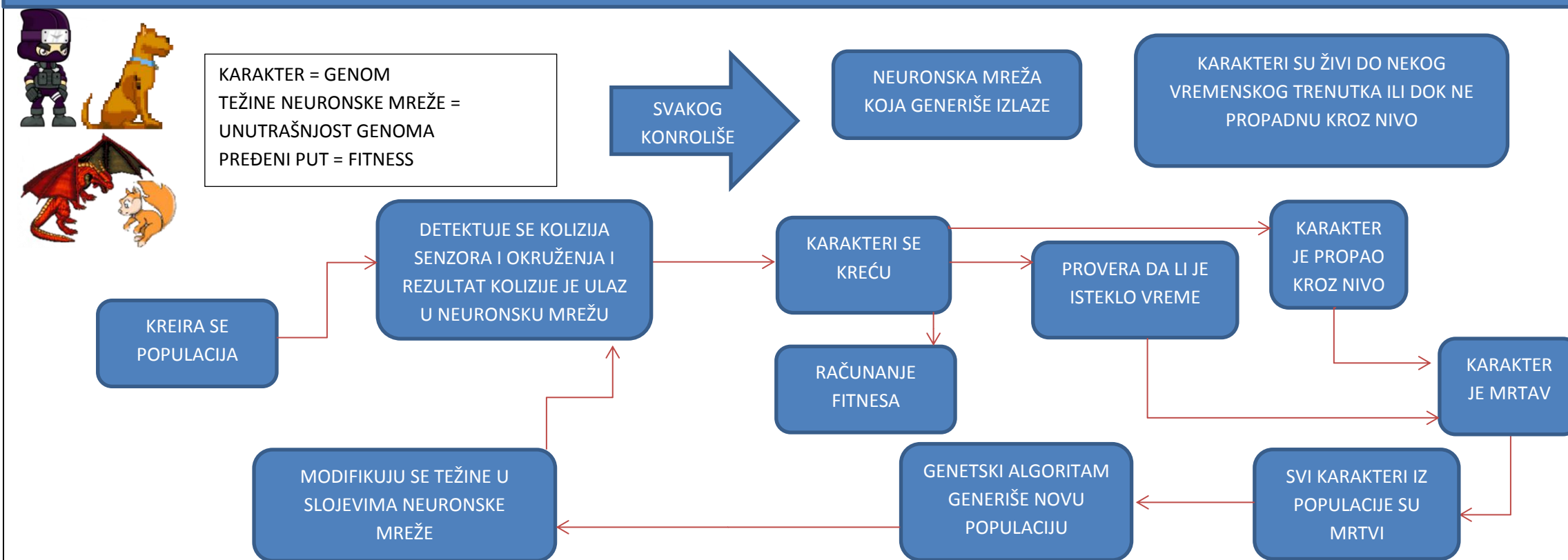
- Genetski algoritam se često koristi u oblasti veštačke inteligencije pri vršenju pretraga i/ili optimizovanju. Oponaša proces prirodne selekcije tako što koristi tehnike kao što su selekcija, mutacija, krosing-over.

Razlog korišćenja genetskog algoritma

- Ideja je bila da vidimo da li je moguće obučiti neuronsku mrežu bez korišćenja back propagation algoritma. Njene težine popravljamo pomoću genetskog algoritma.

- Hromozom genoma je lista težina iz neuronske mreže karaktera, a fitness smo određivali pomoću pređenog puta jedinke.

Kako karakteri uče da se kreću



Rezultati i zaključak



Karakter	Najbolji rez	Najgori rez	Najbolja generacija	Ukupno generacija
Nindža	471.79	-4.26	5	153
Pas	468.57	-2.45	121	135
Veverica	469.41	-2.16	38	147
Zmaj	416.35	-0.36	70	129

Po teoriji bi trebali da dobijemo dobre rezultate brže nego korišćenjem back propagation algoritma, ali nam se dešavalo da nam rešenja dosta variraju i da se desi overfitting. Više puta se desilo da nam karakter samo skače i da to bude jedini način na koji nauči da se kreće jer mu je tako veća verovatnoća da će doći do cilja i izbeći prepreke za razliku od običnog kretanja.

Alternativni algoritmi koji bi bolje radili u rešavanju ovog problema:

- Neuroevolution of Augmentet Topologies (NEAT algoritam)
- Reinforced Learning Algoritam