

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

# Učenje težina neuronske mreže genetičkim algoritmom

*Luka Zuanović*

Voditelj: *prof. dr. sc. Domagoj Jakobović*

Zagreb, lipanj 2014.

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Neuronske mreže</b>	<b>3</b>
2.1. Osnovni model neurona . . . . .	3
2.2. Osnovni model neuronske mreže . . . . .	4
2.3. Podjela neuronskih mreža . . . . .	6
2.4. Učenje neuronske mreže (algoritam <i>Backpropagation</i> ) . . . . .	6
<b>3. Genetički algoritam</b>	<b>9</b>
3.1. Početna populacija . . . . .	10
3.2. Prikaz rješenja . . . . .	10
3.3. Operator mutacije . . . . .	10
3.4. Operator križanja . . . . .	10
3.5. Operator selekcije . . . . .	12
3.5.1. Proporcionalna selekcija . . . . .	12
3.5.2. Turnirska selekcija . . . . .	12
<b>4. Učenje neuronske mreže genetičkim algoritmom</b>	<b>14</b>
4.1. Motivacija . . . . .	14
4.2. Usporedba algoritama . . . . .	15
4.2.1. Vrsta mreže . . . . .	15
4.2.2. Prikaz rješenja . . . . .	15
4.2.3. Funkcija greške . . . . .	15
4.2.4. Početna populacija . . . . .	15
4.2.5. Genetski operatori . . . . .	16
4.2.6. Rezultati . . . . .	16
4.3. Ostali primjeri . . . . .	17
<b>5. Zaključak</b>	<b>18</b>

<b>6. Literatura</b>	<b>19</b>
<b>7. Sažetak</b>	<b>20</b>

# 1. Uvod

S problemom optimizacije susrećemo se praktički u svim područjima života, bilo da tražimo najbrži put do mora ili da izrađujemo satnicu predavanja na fakultetu itd. Problem optimizacije sastoji se u pronalaženju najboljeg rješenja iz prostora mogućih rješenja, što se svodi na traženje ekstrema kriterijske funkcije uz zadana ograničenja. Prvo što nam pada napamet kada smo suočeni s takvim problemom je isprobavati jedno po jedno rješenje dok ne iscrpimo sva moguća rješenja, tj. provodimo iscrpnu pretragu. No kako s porastom broja varijabli broj rješenja strelovito raste iscrpna je pretraga često neizvediva u razumnom vremenu, te se stoga ispomazemo heurističkim metodama.

Heurističke metode pronalaze rješenja, uz nisku računsku složenost, koja su dovoljno dobra ali ne i nužno optimalna. Jedna od takvih metoda je genetički algoritam (Holland, 1975). Genetički algoritam je univerzalna optimizacijska metoda nadahnuta evolucijom života. U evoluciji života osnovni je pojam prirodnog odabira, što znači da preživljavaju jedinke koje su prilagođenije okolini te se onda one razmnožavaju prosljeđujući svoje osobine na potomke uz moguće slučajne mutacije. Dakle prirodni odabir nije “slučajan” kako se to obično misli, nego je to proces koji čuva dobiti a briše pogreške. Možemo reći da se evolucija sastoji od slučajne mutacije te neslučajnog kumulativnog odabiranja.

Još jedan primjer nadahnuća prirodom u rješavanju problema su i neuronske (živčane) mreže koje oponašaju životinjski živčani sustav, tj. mozak. Primjenjive su na mnoge zadatke strojnoga učenja. Najpoznatiji algoritam za učenje neuronskih mreža je *Backpropagation*, no on nije primjenjiv ili prikladan za sve mreže. Druga mogućnost jest pozvati u pomoć evolucijske algoritme, poput spomenutih genetičkih, koji su univerzalni optimizatori, za učenje mreže.

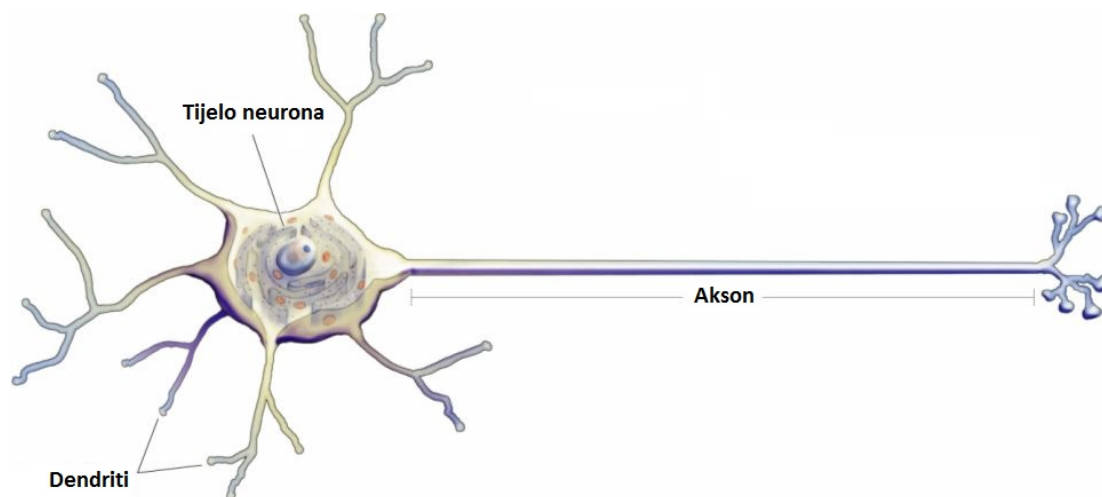
U drugom poglavlju dan je kratak osvrt na umjetne neuronske mreže, njihova podjela, algoritam *Backpropagation* te je pokazana veza između biološkog i umjetnog neurona. Nadalje, u trećem se poglavlju obrađuju genetički algoritmi. Načini prikaza rješenja i genetski operatori mutacije, križanja i selekcija obrađeni

su u zasebnim potpoglavljima. Na kraju, u četvrtom se poglavlju govori o učenju neuronskih mreža genetičkim algoritmom, te ga se uspoređuje s algoritmom *Backpropagation* i to kroz analizu eksperimenta opisanog u (Montana i Davis, 1989).

## 2. Neuronske mreže

Neuronske mreže su strukture sačinjene od više povezanih neurona, stoga je logično da izlaganje o neuronskim mrežama započnemo prikazom osnovnog modela neurona.

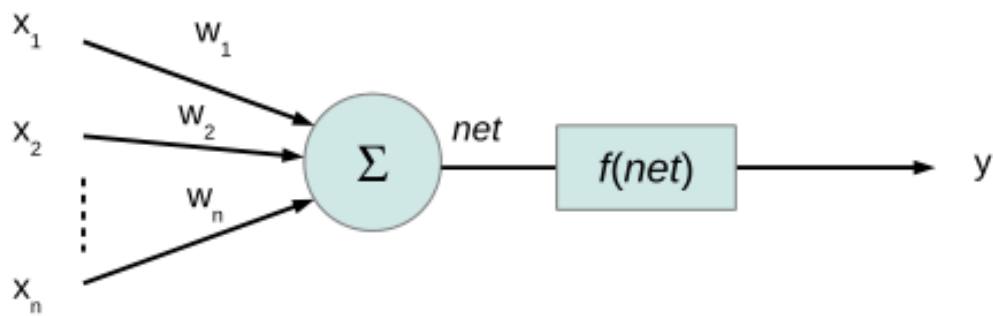
### 2.1. Osnovni model neurona



Slika 2.1: Osnovni dijelovi neurona

Neuron se sastoji od nekoliko dendrita, tijela i aksona (slika 2.1). Neuron u svom tijelu akumulira impulse primljene preko dendrita i kada se u njegovom tijelu akumulira dovoljan naboj (određen pragom paljenja), neuron akumulirani naboj šalje kroz svoj akson drugim neuronima (tj. neuron *pali*).

Na temelju ovog modela prirodnog neurona definiran je osnovni model umjetnog neurona (slika 2.2). Neuron se sastoji od  $n$  ulaza od kojih svaki ima svoju težinu  $w$ , tijela neurona koje računa ukupnu pobudu, označenu s  $net$  te prienosne funkcije  $f(net)$  kojom je modeliran akson neurona (Čupić et al., 2013). Ukupna

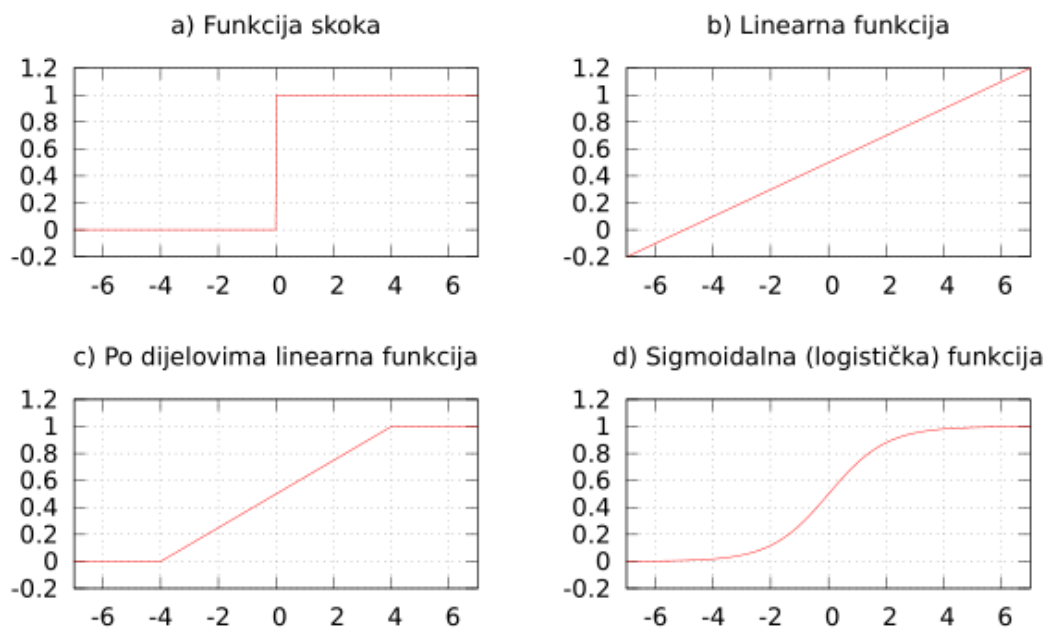


Slika 2.2: Osnovni model neurona, preuzeto iz (Čupić et al., 2013)

pobuda neurona računa se kao

$$net = \sum_{i=1}^n w_i \cdot x_i - \theta$$

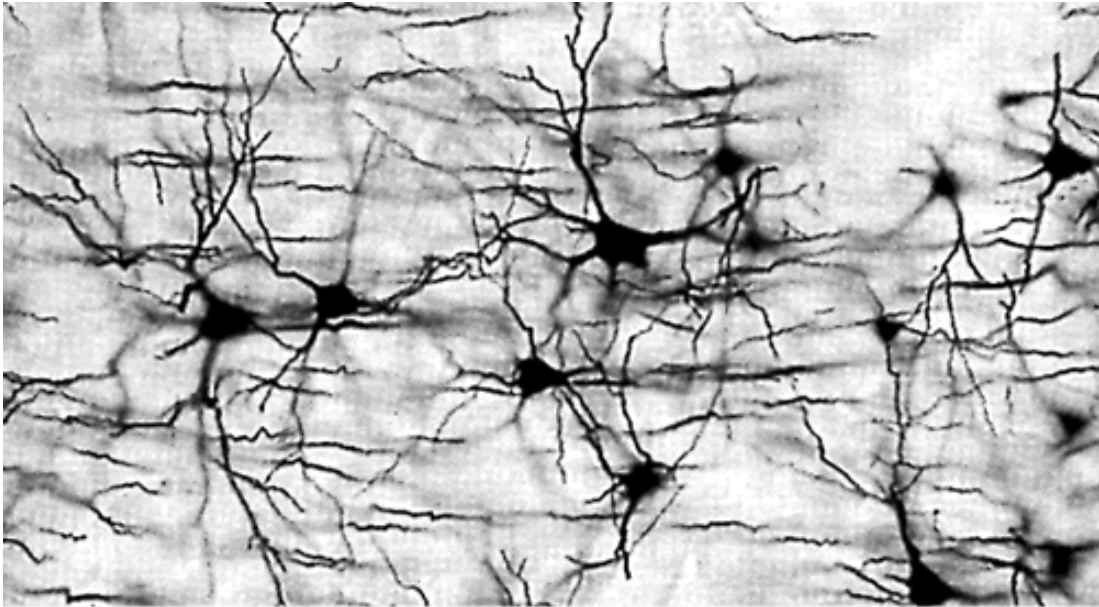
gdje  $\theta$  označava prag okidanja neurona. Tipične prijenosne funkcije prikazane su slikom 2.3



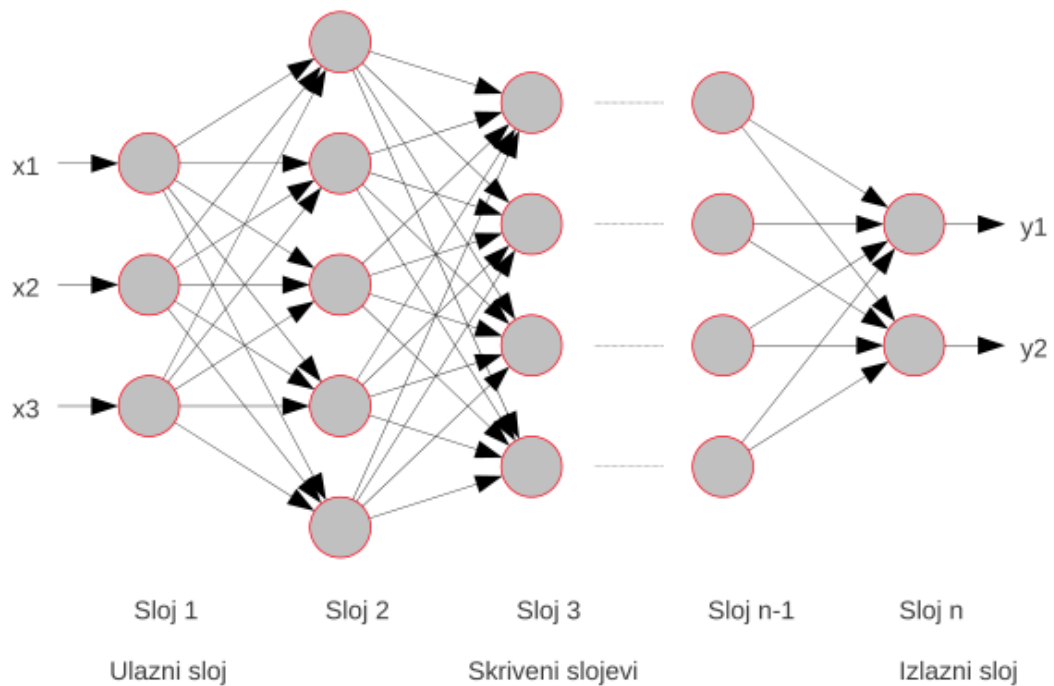
Slika 2.3: Tipične prijenosne funkcije, preuzeto iz (Čupić et al., 2013)

## 2.2. Osnovni model neuronske mreže

Povezivanjem više neurona nastaje neuronska mreža. Povezivanje je nužno želimo li obavljati imalo složeniju funkciju, a kako možemo povezati proizvoljan broj



Slika 2.4: Biološka neuronska mreža



Slika 2.5: Model umjetne neuronske mreže, preuzeto iz (Čupić et al., 2013)

neurona nije nelogično da su se neuronske mreže pokazale kao uspješan model za primjenu u:

- raspoznavanju uzoraka
- aproksimaciji funkcija



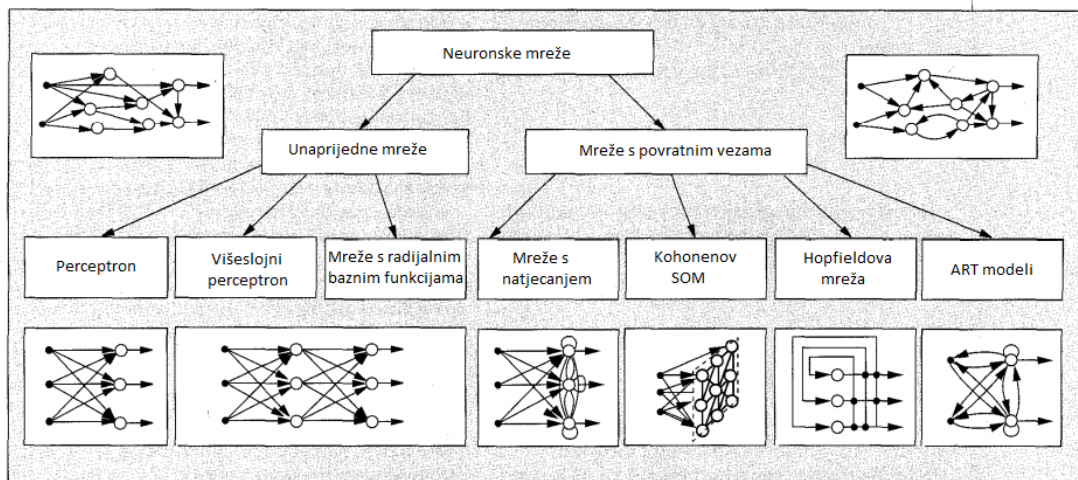
- klasifikaciji podataka
- predviđanju trendova
- itd.

Neuronske mreže sastoje se od tri tipa neurona:

- ulazni; Neuroni na koje se dovodi informacija koju je potrebno obraditi, ne obavljaju nikakvu obradu nego ulaz prosljeđuju ostalim neuronima u mreži.
- izlazni; Neuroni koji rezultate obrade odašilju u okolinu
- skriveni; Svi ostali neuroni koji se izvana ne vide

## 2.3. Podjela neuronskih mreža

Na slici 2.6 dana je jedna od podjela umjetnih neuronskih mreža. Za malo drugačiju podjelu te dodatnu literaturu čitatelja se upućuje na (Čupić et al., 2013).

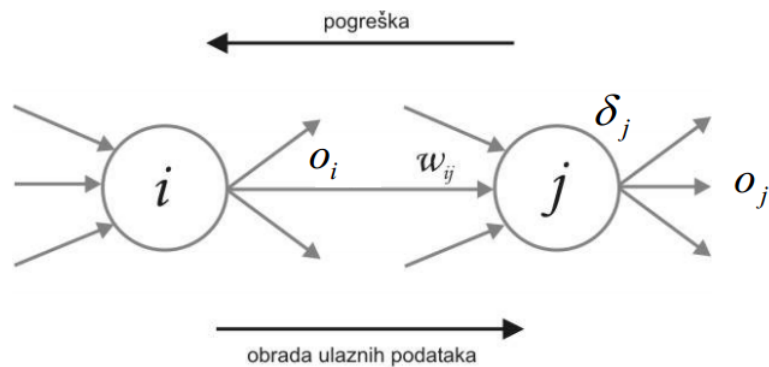


Slika 2.6: Podjela unaprijednih i rekurentnih mreža, preuzeto iz (Jain et al., 1996)

## 2.4. Učenje neuronske mreže (algoritam *Back-propagation*)

Najpoznatiji algoritam za učenje neuronskih mreže je algoritam *Backpropagation* (*backward propagation of error*, unazadno rasprostiranje pogreške). Kako bismo

definirali postupak učenja prvo se definira kriterijska funkcija koja mjeri kakvoću neuronske mreže. Obično se za kriterijsku funkciju uzima srednje kvadratno odstupanje između svakog željenog izlaza mreže i stvarne vrijednosti izlaza promatrano nad svim primjerima za učenje. Kriterijska funkcija je funkcija primjera za učenje (nepromjenjivi u postupku učenja) i težina mreže (pretpostavljajući da je struktura mreže fiksna) pa na njen iznos možemo utjecati samo promjenom težina. Zadaća je dakle postupka učenja pronaći one težine uz koje će iznos funkcije biti minimalan. Za minimizaciju funkcije primjenjuje se ideja gradijentnog spusta, zato se moraju izračunati parcijalne derivacije funkcije za svaku od težina. Skup primjera za učenje  $D$  su uređeni parovi  $(x, t)$  gdje  $x$  predstavlja ulaz, a  $t$  točan izlaz. Pseudokod algoritma dan je na slici<sup>1</sup> 2.8, dok su na slici 2.7 oznake korištene u pseudokodu, a na slici 2.9 je jasnije pokazano na što se odnose pojedine oznake unutar mreže.



**Slika 2.7:** Notacija za BP

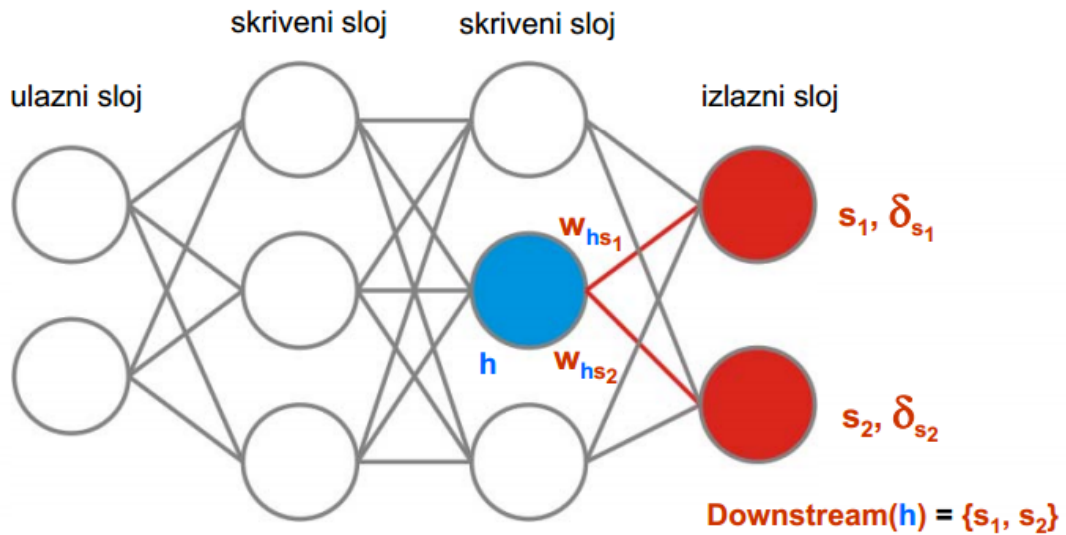
<sup>1</sup>Slike su preuzete s [http://www.fer.unizg.hr/\\_download/repository/UI\\_12\\_UmjetneNeuronskeMreze\[1\].pdf](http://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf)

```

Inicijaliziraj težinske faktore na slučajne vrijednosti
Dok nije ispunjen uvjet zaustavljanja čini
  Za svaki  $(\mathbf{x}, \mathbf{t})$  iz  $D$  čini
    Izračunaj izlaz  $o_u$  za svaku jedinicu  $u$ 
    Za svaku izlaznu jedinicu  $k$  izračunaj pogrešku  $\delta_k$ 
       $\delta_k \leftarrow o_k(1-o_k)(t_k - o_k)$ 
    Za svaku skrivenu jedinicu izračunaj pogrešku
       $\delta_h \leftarrow o_h(1-o_h) \sum_{s \in \text{Downstream}(h)} \omega_{hs} \delta_s$ 
    Ugodi svaki težinski faktor  $w_{ij}$ 
       $\omega_{ij} \leftarrow \omega_{ij} + \Delta\omega_{ij}$ 
    gdje je  $\Delta\omega_{ij} = \eta\delta_j o_i$ 
  Kraj
Kraj

```

Slika 2.8: BP algoritam



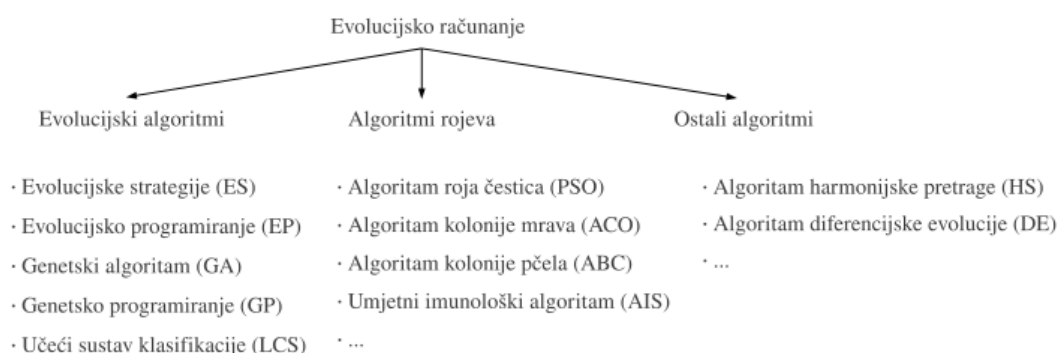
$$\delta_{s_1} = o_{s_1} (1 - o_{s_1}) (t_{s_1} - o_{s_1})$$

$$\delta_{s_2} = o_{s_2} (1 - o_{s_2}) (t_{s_2} - o_{s_2})$$

$$\delta_h = o_h (1 - o_h) (w_{hs_1} \delta_{s_1} + w_{hs_2} \delta_{s_2}) \quad \delta_h \text{ će služiti za ugađanje ulaza u } h$$

Slika 2.9: Mreža uz BP

### 3. Genetički algoritam



**Slika 3.1:** Hijerarhija evolucijskog računanja, preuzeto iz (Čupić et al., 2013)

Hijerarhija evolucijskog računanja dana je na slici 3.1, dok se čitatelja željnog saznanja o potankostima vrsta evolucijskog računanja upućuje na (Čupić et al., 2013) (Čupić, 2013), dok će se u ovom poglavlju obrađivati genetički algoritam.

Genetički algoritam je heuristička metoda slučajnog i usmjerenog pretraživanja prostora rješenja koje imitira prirodni evolucijski proces (Čupić et al., 2013). Genetički algoritam radi s populacijom jedinki, gdje svaka jedinka predstavlja jedno od mogućih rješenja problema. Kada se govori o genetičkim algoritmima jedinke se nazivaju kromosomima. Jedinke su međusobno usporedive po svojoj dobroti koju određuje funkcija dobrote koja ovisi o problemu koji se rješava. Dobrota govori koliko je ta jedinka dobro rješenje problema. Nekim od operatora selekcije biraju se one jedinke koje kreću u reprodukciju (roditelji). Reprodukcija stvara novu populaciju operatorima križanja i mutacije (slika 3.2).

Iz zadanog pseudokoda lako je vidljivo da genetički algoritam mora imati definirane stavke opisane u narednim potpoglavljima.

```

1  Geneticki_algoritam(){
2      generiraj_pocetnu_populaciju();
3      dok (nije_zadovoljen_uvjet_zavrsetka_evolutionarnog_procesa){
4          selektiraj_bolje_jedinke_za_reprodukciju();
5          reprodukcijom_generiraj_novu_populaciju();
6      }
7  }

```

**Slika 3.2:** Pseudokod genetičkog algoritma, preuzeto iz (Čupić et al., 2013)

### 3.1. Početna populacija

Genetički algoritma započinje rad generiranjem početne populacije. Najčešći način generiranja početne populacije je slučajnim odabirom vrijednosti iz domene, ali ju je moguće generirati i nekim drugim algoritmima.

### 3.2. Prikaz rješenja

Od svih načina prikaza rješenja, najjednostavniji koji se koristi kod genetičkih algoritama jest prikaz nizom bitova, u kojem se svaki kromosom sastoji od jednakog i nepromjenjivog broja bitova. Na primjeru ovog prikaza rješenja ukratko će biti objašnjeni operatori križanja i mutacije, dok se za ostale (prikaz poljem realnih brojeva, prikaz permutacijama i matricama, prikaz složenijim strukturama podataka (programima) te prikaz stablima) čitatelja se upućuje na (Čupić, 2013).

### 3.3. Operator mutacije

Kod prikaza rješenja nizom bitova operator mutiranja je vrlo jednostavan, za svaki bit se generira slučajna vjerojatnost mutacije (broj između 0 i 1) te se na temelju nje i zadane fiksne vjerojatnosti mutacije  $p_m$  bit izmjeni. Operator je opisan pseudokodom na slici 3.3.

### 3.4. Operator križanja

U križanju sudjeluju dvije jedinke (roditelji) te križanjem nastaju jedna ili dvije jedinke (djeca). Operatori križanja koji se koriste kod prikaza nizom bitova vrlo su jednostavni te su prikazani na slici 3.4, treba samo napomenuti da je kod

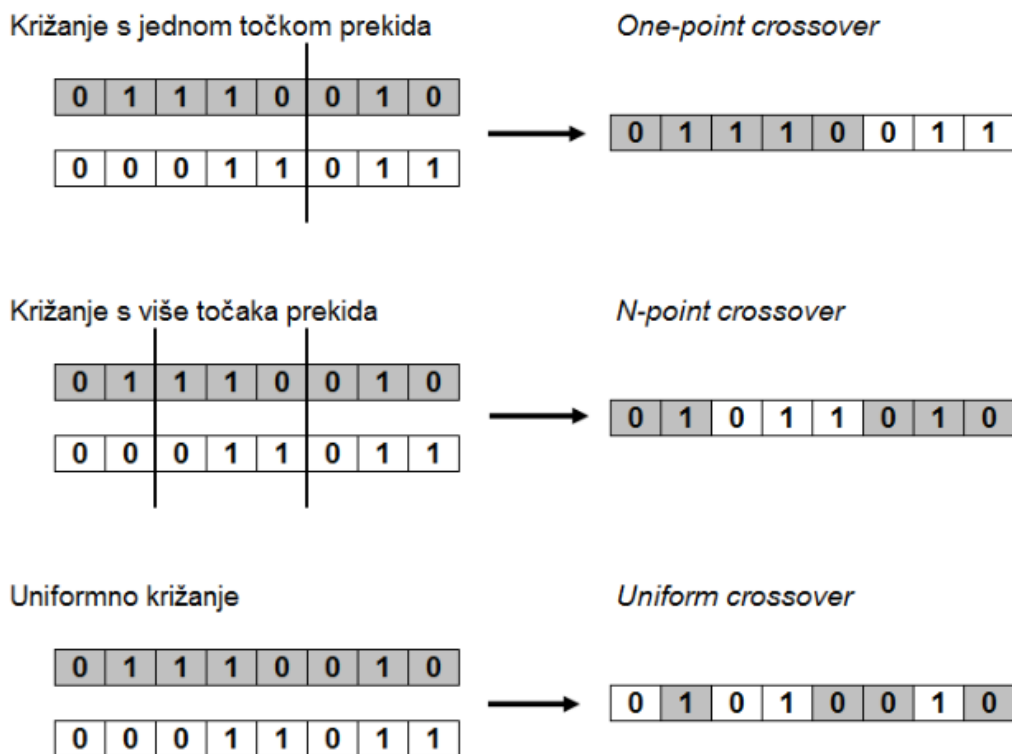
```

mutiraj(rjesenje[1 do n]: bit, pm: double)
ponavljaj za i iz 1 do n
    double r = slucajni_broj(0,1);
    ako r < pm tada
        rjesenje[i] = not rjesenje[i];
    kraj ako
kraj

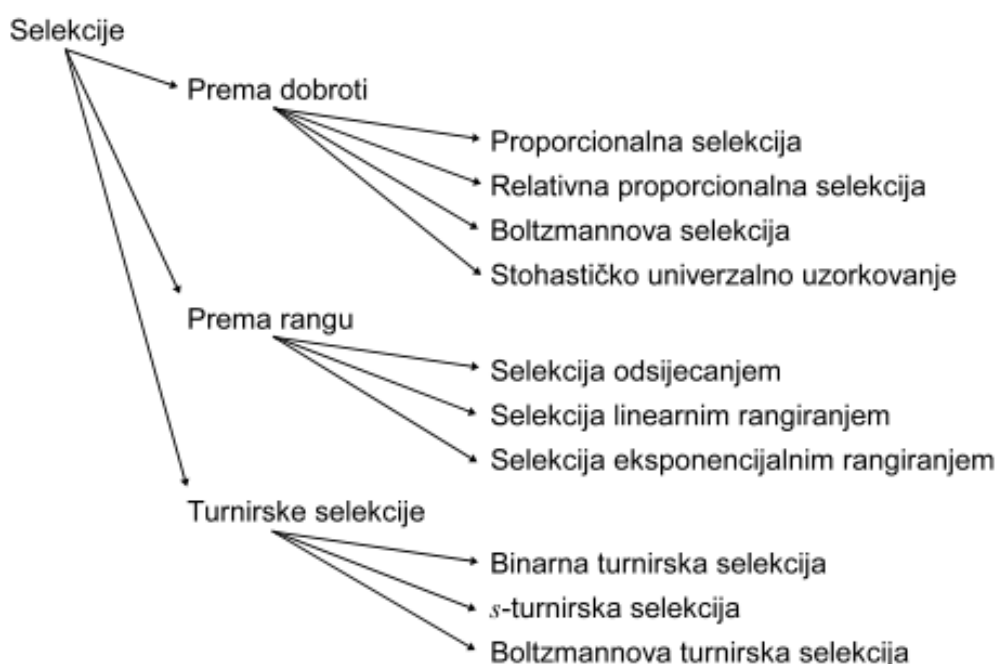
```

Slika 3.3: Pseudokod operatora mutacije, preuzeto iz (Čupić, 2013)

uniformnog križanja vjerojatnost da dijete ima pola gena od jednog a pola od drugog roditelja iznosi 0,5.



Slika 3.4: Operatori križanja, preuzeto iz (Čupić et al., 2013)



Slika 3.5: Podjela operatora selekcije, preuzeto iz (Čupić, 2013)

## 3.5. Operator selekcije

Selekcija je mehanizam odabira jedinki za sudjelovanje u reprodukciji, koji omogućava prenošenje dobrih svojstava iz generacije u generaciju, jer što je jedinka bolja veća je vjerojatnost njenog odabira.

Od brojnih vrsta selekcija (slika 3.5) ovdje ćemo ukratko opisati samo dvije najčešće dok se za ostale čitatelja upućuje na (Čupić, 2013).

### 3.5.1. Proporcionalna selekcija

Selekcija je poznata pod nazivom *Roulette-wheele selection*. Ideja iza algoritma je sljedeća: svakoj jedinki se pridruži kružni isječak proporcionalan njenoj dobroti, zavrtimo li više puta krug jedinke s većom dobrotom imaju veće šanse da budu odabrane.

### 3.5.2. Turnirska selekcija

Turnirska se selekcija obavlja uzimanjem  $k$  slučajnih jedinki iz populacije te se najbolja od uzetih jedinki odabire. Takva selekcija se naziva  $k$ -turnirska selekcija.  $K$ -turnirska selekcija dolazi u dvije varijante:

- svaki put se odabire k jedinki iz cijele populacije što omogućava ponavljanje jedinki
- k jedinki se odabire iz podskupova populacije nastalih uklanjanjem dotad odabranih jedinki



## 4. Učenje neuronske mreže genetičkim algoritmom

Do sada su u kratko opisane neuronske mreže i genetički algoritmi, te je rečeno da je najčešći način učenja neuronske mreže algoritmom *backpropagation* no iz činjenice da se genetički algoritmi<sup>1</sup> koriste za rješavanje optimizacijski problema, a podešavanje mreže za rješavanje zadanog zadatka svakako spada u tu kategoriju problema, prirodno je probati kombinirati genetičke algoritme i razvoj neuronske mreže.

Evolucijski algoritmi<sup>2</sup> mogu se primijeniti na razvoj neuronskih mreža na tri načina:

- Evolucija težinskih faktora
- Evolucija arhitektura
- Evolucija pravila učenja

U nastavku poglavlja obrađuje se evolucija težinskih faktora.

### 4.1. Motivacija

Klasični algoritmi učenja izvedeni su za pojedine tipove neuronskih mreža, tako se ne može samoorganizirajuća neuronska mreža učiti algoritmom *Backpropagation* dok se evolucijski algoritmi mogu koristiti na evoluciju značajki neograničenog broja različitih mreža zbog njihove univerzalnosti.

Nedostaci algoritma *Backpropagation* i njegovih inačica su:

- baziraju se na gradijentu a takvi algoritmi često zapinju u lokalnim optimumima

---

<sup>1</sup>Općenitije evolucijski algoritmi

<sup>2</sup>Stohastički optimizacijski algoritmi, robusni na lokalne optimume

- algoritmi koji se temelje samo na informacijama o gradijentu vrlo su neefikasni te zahtijevaju velik broj iteracija
- neprimjenjivi su na probleme minimizacije funkcije pogreške koja nije derivabilna

Nasuprot tome genetički algoritam stohastički je optimizacijski algoritam koji je otporan na lokalne optimume i ne zahtijeva da funkcija koja se optimira bude derivabilna.

No ipak radovi na temu usporedbe algoritma *Backpropagation* i genetičkog ne daju konačan odgovor na to koji od njih je efikasniji. U nastavku ćemo se osvrnuti na rezultate iskazane u (Montana i Davis, 1989).

## 4.2. Usporedba algoritama

David J. Montana i Lawrence Davis su u svom članku iz 1989 proveli nekoliko eksperimenata nad neuronskom mrežom primjenjujući vlastiti genetički algoritam i došli do zanimljivih rezultata od kojih je nama najinteresantniji onaj o usporedbi genetičkog algoritma i algoritma *Backpropagation*. U nastavku su ukratko opisani parametri pokusa.

### 4.2.1. Vrsta mreže

Unaprijedna neuronska mreža s 4 ulazna neurona i jednim izlaznim. Uz to sadrži 2 skrivena sloja s po 7 i 10 neurona redomice. Mreža sveukupno ima 126 težina.

### 4.2.2. Prikaz rješenja

Rješenja se prikazuju poljem od 126 realnih brojeva.

### 4.2.3. Funkcija greške

Koristili su standardnu funkciju greške gdje se zbrajaju kvadrati grešaka svakog od ulaznih primjera.

### 4.2.4. Početna populacija

Težine imaju distribuciju vjerojatnosti zadanu s  $t^{|r|}$  koju su odabrali na temelju rezultata prijašnjih istraživanja.

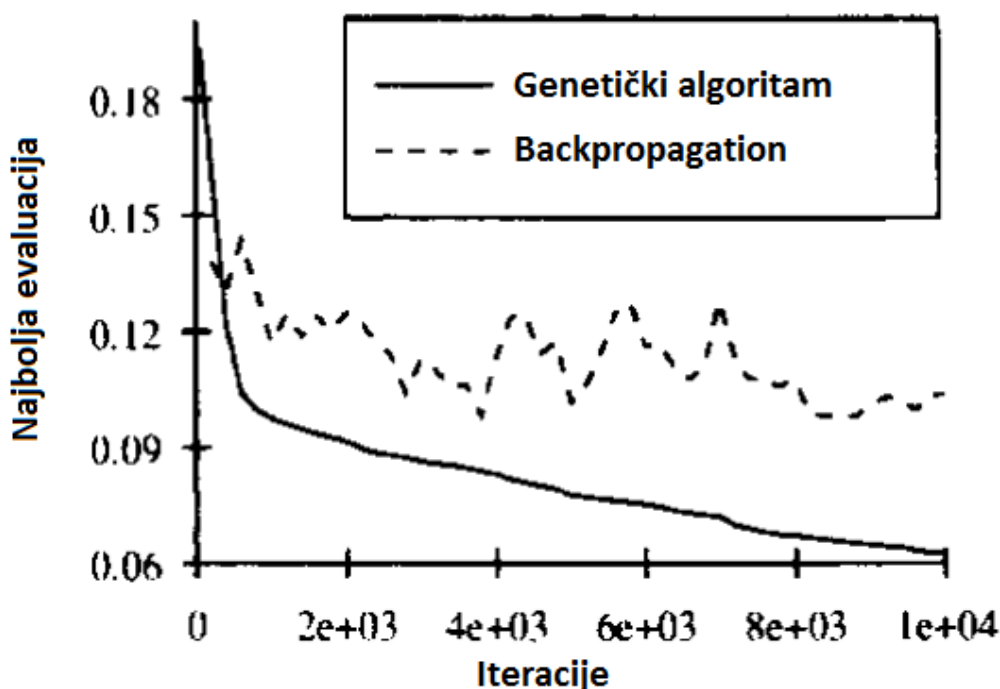
### 4.2.5. Genetski operatori

Mutacija neurona: Odabire se  $n$  neulaznih neurona, svim ulaznim težinama svih odabranih neurona se nadoda nasumična vrijednost, nove težine se zatim ukodiraju u kromosom djeteta.

Križanje neurona: Za svaki neulazni kromosom nasumično se odabire jedan od roditelja i od njega se preuzimaju sve ulazne težine;naime, kako je pretpostavka da je svaki neuron specijaliziran za određeno područje, u dijete se prenosi kompletan skup težina neurona izabranog roditelja čime se ta specijalizacija prenosi u dijete.

### 4.2.6. Rezultati

Kako bi evaluirali algoritam sa zadanim operatorima i parametrima, provedeno je 10 nezavisnih testiranja po broju iteracija i kao funkcija broja iteracija uzeta je vrijednost evaluacije najbolje jedinke. Rezultati prikazani slikom 4.1 su nedvojbeno pokazali superiornost genetičkog algoritma za učenje težina prethodno definirane neuronske mreže.



Slika 4.1: Usporedba efikasnosti algoritma *Backpropagation* i genetičkog, preuzeto iz (Montana i Davis, 1989)

### 4.3. Ostali primjeri

U ovom odlomku dan je kratak pregled još nekoliko radova slične tematike.

U svom magistarskom radu (Koehn, 1994) Philipp Koehn donosi detaljan pregled različitih načina kodiranja, onih temeljenih na vezama, onih temeljenih na čvorovima, onih temeljenih na slojevima te onih temeljenih na putevima u mreži te se čitatelja željnog detalja o svim ovim metodama upućuje na spomenuti rad. Također je dan pregled složenih metoda indirektnog kodiranja te opis najčešćih problema koji se javljaju u učenju neuronskih mreža. Autor je također izveo brojne pokuse sa sljedećim zanimljivim rezultatima:

- genetički algoritam je mnogo bolji za učenje težina od nasumične potrage
- korištenje i operatora križanja i operatora mutacije je bolje od korištenja nekog od njih zasebno, dok se zasebno boljim izborom pokazala mutacija
- optimalan raspon težina mreže je od -20 do 20 za aproksimaciju sinusne ali i nekih drugih realnih funkcija
- korištenje paralelnog genetičkog algoritma nad velikom populacijom se pokazalo uspješnim te je smanjilo mogućnost zapinjanja u lokalnim optimumima i očuvalo je raznolikost jedinki
- kodiranje jedinki Grayevim kodom nije urodilo plodom kako za logičke funkcije tako i funkcije realnih brojeva

Autori su u članku (Che et al., 2011) usporedili genetički (GA) i *Backpropagation* (BP) algoritam na tri skupa podataka (sinusna funkcija, klasifikacija perunika i određivanje bolovanja od dijabetesa) te došli do sljedećih zaključaka:

- BP je brži od GA u učenju neuronske mreže te troši manje procesorskog vremena po ciklusu
- BP ima problema s prenaučenošću dok GA ne
- BP stabilnije konvergiraju prema najboljem rješenju ali zahtijevaju veći broj ciklusa učenja od GA, dok je GA potrebno manje ciklusa učenja za pronalazak najboljeg rješenja ali mu je potrebno dulje vrijeme

## 5. Zaključak

Uz sve veću primjenu neuronskih mreža pri rješavanju raznih problema te uzevši u obzir koliko su efikasne u tome zasigurno je da će neuronske mreže u budućnosti biti sve češće korišten alat. Genetički algoritmi kao univerzalna metoda rješavanja optimizacijskih problema već su naišli na široku primjenu, od izrade prezentacijskih grupa za predmet seminar do osmišljavanja antena za svemirske letjelice, tj. u velikom broju različitih područja nalazi svoju primjenu. Efikasnost genetičkog algoritma uvelike ovisi o parametrima te se za daljnji razvoj treba usredotočiti na postupke pronalaženja optimalnih parametara algoritma. Učenje težina neuronske mreže spada u klasu optimizacijskih problema pa je prikladno za njegovo rješavanje koristiti genetički algoritam. Iako se može činiti da je genetički algoritam uvijek efikasniji od *Backpropagation* algoritma to nije tako. Njihov međusobni odnos s obzirom na to koliko su efikasni u rješavanju problema evolucije težina mreže ovisi isto tako i o samoj mreži koju uče. Za potrebe učenja težina neuronske mreže u (Montana i Davis, 1989) također su prikazani rezultati usporedbe različitih genskih operatora te će se razvoj genetičkih algoritama za ovu primjenu vjerojatno bazirati na definiranju optimalnih operatora.

## 6. Literatura

Zhen-Guo Che, Tzu-An Chiang, i Zhen-Hua Che. Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm. *Int J Innov Comput Inf*, 7:5839–5850, 2011.

John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.

Anil K Jain, Jianchang Mao, i K Moidin Mohiuddin. Artificial neural networks: A tutorial. *IEEE computer*, 29(3):31–44, 1996.

Philipp Koehn. Combining genetic algorithms and neural networks: The encoding problem. Magistarski rad, The University of Tennessee, Knoxville, 1994.

David J Montana i Lawrence Davis. Training feedforward neural networks using genetic algorithms. U *IJCAI*, svezak 89, stranice 762–767, 1989.

Marko Čupić. *Prirodom inspirirani optimizacijski algoritmi. Metaheuristike*. 2013. URL <http://java.zemris.fer.hr/nastava/pioa/knjiga-0.1.2013-07-12.pdf>.

Marko Čupić, Bojana Dalbelo Bašić, i Marin Golub. *Neizrazito, evolucijsko i neuroračunarstvo*. 2013. URL <http://java.zemris.fer.hr/nastava/nenr/knjiga-0.1.2013-08-12.pdf>.

## 7. Sažetak

Umjetna neuronska mreža je skup međusobno povezanih neurona. Umjetne neuronske mreže rabe se za raspoznavanje uzoraka, aproksimacije funkcija, klasifikaciju podataka, predviđanje trendova i druge zadatke strojnoga učenja. Genetički algoritam je heuristička metoda slučajnog i usmjerenog pretraživanja prostora rješenja koje imitira prirodni evolucijski proces. Genetički algoritam sastoji se od 5 koraka:

1. Generiranje početne populacije.
2. Računanje dobrote jedinki.
3. Primjena operatora selekcije.
4. Reprodukcijska (križanje + mutacija)
5. Ako uvjet završetka nije zadovoljen, ponovi korake 2-5.

Algoritam *Backpropagation* je algoritam za učenje težina neuronske mreže. Primjenom genetičkog algoritma na učenje težina neuronske mreže moguće je dobiti bolje rezultate nego primjenom algoritma *Backpropagation*.