

# Using evolutionary algorithm to optimize a structure of neural network for a specific ML task

Lazar Cvijić 200/2020

Septembar 2023

Seminarski rad u okviru kursa Računarska inteligencija

Matematički fakultet

## Sadržaj

1. Uvod.....	2
2. Ideja rešenja.....	3
2.1 Populacija.....	3
2.2 Elitizam.....	3
2.3 Selekcija.....	4
2.4 Ukrštanje.....	4
2.5 Mutacija.....	4
2.6 Tok algoritma.....	4
3. Rezultati.....	4
4. Zaključak.....	5
5. Literatura.....	5

# 1. Uvod

U ovom radu predstavljen je problem optimizacije strukture neuronske mreže za dati zadatak mašinskog učenja. Izabrani zadatak u ovom slučaju je višeklasna klasifikacija ruku karata u pokeru na osnovu prethodnih rešenja bez eksplicitnog navodjenja pravila same igre. Skup podataka koji se koristio preuzet je sa zvaničnog UC Irvine Machine Learning Repository sajta čiji link se nalazi u literaturi. Svaki element trening skupa sačinjen je od deset celih brojeva od kojih svaka dva predstavljaju jednu kartu tj. njen znak i njenu vrednost na sledeći način:

Za ruku karata S1 C1 S2 C2 S3 C3 S4 C4 S5 C5 C

Si : Ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}

Ci : Numerical (1-13) representing (Ace, 2, 3, ... , Queen, King)

C : class PokerHand (0-9) :

0: Nothing in hand; not a recognized poker hand

1: One pair; one pair of equal ranks within five cards

2: Two pairs; two pairs of equal ranks within five cards

3: Three of a kind; three equal ranks within five cards

4: Straight; five cards, sequentially ranked with no gaps

5: Flush; five cards with the same suit

6: Full house; pair + different rank three of a kind

7: Four of a kind; four equal ranks within five cards

8: Straight flush; straight + flush

9: Royal flush; {Ace, King, Queen, Jack, Ten} + flush

Za rešavanje ovog problema jedan od načina je da koristimo potpuno povezanu neuronsku mrežu koja će kao ulaz imati 10 brojeva koji će predstavljati ruku karata, a kao izlaz 10 neurona koji predstavljaju verovatnocu da se ta ruka nalazi u i-toj klasi.

## 2. Ideja rešenja

Kao što glasi u samom naslovu koristićemo evolutivni algoritam da napravimo sto bolju strukturu neuronske mreže koje će biti što preciznija za klasifikaciju naših ruku poker karata. Koristićemo genetski algoritam da bismo to postigli. Genetski algoritam na početku inicijalizuje populaciju nasumično napravljenih jedinki, koje kroz više generacija prolaze kroz proces selekcije, ukrštanja i mutacije. Nakon čega vođen fitnes funkcijom algoritam iz generacije u generaciju eliminise lošije jedinke i pušta bolje da ostave potomstvo. U našem slušaju jedinka će nam biti objekat klase Individual koji u sebi sadrži model neuronske mreže, niz brojeva neurona u svakom sloju, broj slojeva i precznost modela koji nam zapravo predstavlja fitnes funkciju.

### 2.1 Populacija

Na početku inicijalizovaćemo populaciju od  $n$  jedinki koja će kroz generacije dobijati nove članove ali neki dobri će preživljavati i ostajati u novim populacijama. Veličina populacije će nam uticati na krajnje rešenje tako što, što je veća populacija veći je domen mogućih novih jedinki koje mogu da nastanu ukrštanjem.

### 2.2 Elitizam

Elitizam označava broj prvih najboljih jedinki koje će preživeti tj. ostaviti potomstvo u sledećoj generaciji. Na ovaj način ako smo na početku imali sreće I dobili dobre jedinke koristimo elitizam kako bi ih zadržali što duže ako nastave da budu najbolje. Najbolje jedinke se naravno određuju po fitnes funkciji kao i do sada.

### 2.3 Selekcija

U ovom algoritmu korišćena je turnirska selekcija u kojoj iz nasumičnog uzorka jedinki uzima se najbolja i ona sigurno daje potomstvo za sledeću generaciju.

### 2.4 Ukrštanje

Na ovaj način kreiramo nove jedinke ukrštajući attribute za sada najboljih jedinki pod pretpostavkom da ćemo napraviti još bolje. Dva roditelja se ukrštaju i od njih nastaju dva deteta koja će biti jedni od predstavnika sledeće generaije. Korišćeno je

jednopoloziciono ukrštanje po indeksu koji se dobija slučajnim izborom u rasponu manjeg roditelja.

## 2.5 Mutacija

Mutacija je proces koji pod određenom verovatnoćom menja jedinku kako bismo dozvolili pretrazi da razmatra i jedinke koje nisu u početnom domenu. Ovde to radimo tako što prođemo kroz sve slojeve jedinke i pod određenom verovatnoćom tom sloju smanjimo broj neurona za jedan.

## 2.6 Tok algoritma

U glavnoj petlji idemo za svaku generaciju prvo radimo elitizam kako bismo odmah izvukli najbolje jedinke i odmah ih prebacili u novu generaciju. Nakon toga radimo selekciju gde biramo dva po dva roditelja, ali pazimo da ne izaberemo dva roditelja istog indeksa jer onda ukrštanje nema smisla. Takođe to je jedna od optimizacija primenjenih na algoritam kako bi se uštedelo vreme rada samog algoritma koje je u startu veoma veliko gledajući prirodu problema. Pošto svaki put kada napravimo nove jedinke ukrštanjem koje sledi odmah posle selekcije, nakon čega ide mutacija, moramo ponovo da treniramo te modele što je veoma zahtevan zadatak za računar. Kada prođemo sve generacije vraćamo niz najboljih jedinki kroz generacije.

## 3. Rezultati

Algoritam je testiran sa više različitih skupova hiperparametara ali zbog velike vremenske složenosti i uslova za rad (potrebni su mnogo jaki i brzi računari) nije bilo moguće istrenirati na prevelikoj populaciji ili na prevelikom broju generacija. Ali pronadjen je lep skup parametara gde se vidi nemali napredak u performansama. Inicijalizovana je populacija od 32 jedinke, veličina turnira je bila 8, broj generacija 5, elitizam 2, verovatnoća mutacije u sloju 5% i 15 epoha za svaku mrežu. Nakon treniranja ove populacije na 25 000 podataka iz treninga preciznost je rasla iz generacije u generaciju i došlo je do napretka od skoro 10%, najbolja jedinka iz prve generacije imala je 'accuracy' 57%, a najbolja iz poslednje generacije 66.3%! Što nije mala razlika za isti broj epoha pri treniranju obe i razlici samo u broju slojeva i broju neurona u svakom sloju. Kasnije je model i evaluiran na test skupu podataka koje se sastoji od 1 000 000 ruku karata i pokazan je sličan napredak!

## 4. Zaključak

Sama priroda rešavanja ovog problema uslovljava da moramo iznova da treniramo veliki broj neuronskih mreža što je samo po sebi veliki teret za računar. Ali biranjem pravih hiperparametara došlo je do značajnog napretka I ovaj algoritam može da se primeni na još teže probleme mašinskog učenja jer će im pomoći pri precinosti za manji broj treninga. Takođe pri prisustvu jačih računara za ovaj isti problem može da se nađe još bolje rešenje ako povećamo populaciju i broj epoha treniranja neuronske mreže kao i uvođenjem novih metrika i parametara za optimizaciju, ovo je samo uvod u veliku temu neuroevolucije koja će u budućnosti biti još značajnija.

## 5. Literatura

1. <https://archive.ics.uci.edu/dataset/158/poker+hand>