

# Optimizacija arhitekture konvolutivne neuronske mreže uz pomoć genetskog programiranja (klasifikacija slika)

Projekat u okviru kursa Računarska inteligencija  
Matematički fakultet  
Univerzitet u Beogradu

Andrijana Aleksić  
[mi18099@alas.matf.bg.ac.rs](mailto:mi18099@alas.matf.bg.ac.rs)  
Svetlana Bićanin  
[mi18028@alas.matf.bg.ac.rs](mailto:mi18028@alas.matf.bg.ac.rs)

Septembar 2022

# Sadržaj

<b>1</b>	<b>Opis problema</b>	<b>3</b>
<b>2</b>	<b>Implementacija</b>	<b>3</b>
2.1	Parametri slojeva CNN-a . . . . .	3
2.2	Ulazni podaci i parametri GA . . . . .	3
2.3	Genetski algoritam . . . . .	4
2.3.1	Selekcija . . . . .	4
2.3.2	Ukrštanje . . . . .	4
2.3.3	Mutacija i elitizam . . . . .	4
2.4	Izlaz programa . . . . .	4
<b>3</b>	<b>Rezultati</b>	<b>4</b>
<b>4</b>	<b>Zaključak</b>	<b>5</b>
	<b>Literatura</b>	<b>6</b>

# 1 Opis problema

Konvolutivne neuronske mreže (CNN) postižu izuzetan uspeh na polju klasifikacije slika poslednjih godina. Međutim, učinak CNN-a u velikoj meri zavisi od njihove arhitekture. Za većinu najsavremenijih CNN-a, njihove arhitekture se često ručno projektuju. Stoga je teško za korisnike, koji nemaju adekvatno prošireno znanje u oblasti CNN-a, da dizajniraju optimalne CNN arhitekture za sopstvenu klasifikaciju slika. U ovom radu predlažemo metod automatskog projektovanja CNN arhitekture korišćenjem genetskih algoritama, kako bi se efikasno rešili zadaci klasifikacije slika.

U nastavku je dat opis naše implementacije rešenja pomoću Genetskog algoritma.

## 2 Implementacija

Svaka jedinka u Genetskom programiranju će biti predstavljena kao niz slojeva CNN-a, odnosno niz tipova slojeva CNN-a. Ti slojevi mogu biti: Convolution (Conv2D), Pooling (MaxPooling, AveragePooling), Regularization (Dropout - da ne bi doslo do preprilagođavanja), Reshaping (Flatten), Core (Dense). Svaka jedinka poseduje određena ograničenja (npr. prvi sloj jedinice mora biti konvolucija, poslednji sloj mora biti potpuno povezani sloj, mora postojati sloj Flatten i sl.). Jedinke mogu biti različitih dužina. Ono što dodatno proširuje model je nasumičan izbor različitih parametara navedenih slojeva CNN-a. Naime, u trenutku kada se na osnovu tipova iz jedinice pravi konkretan model (CNN), pozivaće se konstruktorske funkcije za odgovarajuće slojeve sa nasumično odabranim parametrima. Ukrštanjem i mutacijom ćemo obezbediti da se unutrašnji slojevi CNN-a menjaju tako da na kraju dobijemo kvalitetnu arhitekturu modela. Kvalitet svake jedinice određuje fitnes koji predstavlja preciznost (accuracy) modela klasifikacije slika iz trening skupa.

### 2.1 Parametri slojeva CNN-a

Inicijalna populacija je napravljena tako što je za prvi sloj (nulti) uzet sloj Conv2D sa randomizovanim parametrima, osim input shape paramtera koji je uzet na osnovu skupa nad kojim se trenira. Pored toga, poslednja dva sloja su uvek Flatten bez parametara, a zatim Dense kom su poslali kao parametri units sa vrednošću 10 i activation sa vrednošću 'softmax'. Između prvog i poslednja dva sloja, random odlukom je pravljeno između jednog i pet unutrašnjih slojeva. Svaki od njih ne samo da je bio nasumično biran, već su i njegovi parametri bili random birani. Ukoliko je u izbor ušao Conv2D, uzimali smo sledeće parametre u obzir : filters koji je uzimao ili vrednost 32 ili 64; kernel size koji je uzimao vrednost ili (3,3) ili (2,2); strides koji je uzimao ili (1,1) ili (2,2); activation koji je uzimao uvek 'relu'; padding koji je uzimao ili 'same' ili 'valid' Zatim, ukoliko je izabran sloj MaxPooling2D, pool size parametar je uziman kao (2,2), isto važi i za AveragePooling2D. U slučaju Dropout sloja, parametar rate je uzeo neku vrednost iz uniforme raspodele od 0 do 1. Ono što je poboljšalo preciznost modela u novonastalim generacijama je što su zapamćeni parametri roditelja i deca su ih nasledivala.

### 2.2 Ulazni podaci i parametri GA

Kao ulazni skup podataka koristimo skup CIFAR-10. Skup podataka se sastoji od 60000 slika u boji veličine 32x32 u 10 klasa, sa 6000 slika po klasi. Postoji 50000 slika za obuku i 10000 slika za testiranje. Ovaj skup se pokazao kao dovoljno složen da iz njega možemo da izvučemo neke bitne karakteristike naseg modela.

Ulazni parametri Genetskog algoritma:

1. broj generacija - 6
2. broj jedinki populacije - 18
3. parametar elitizma - 18 // 3
4. veličina turnirske selekcije - 4
5. verovatnoća da dođe do mutacije - 0.1

## 2.3 Genetski algoritam

### 2.3.1 Selekcija

Za *selekciju* koristimo turnirsku selekciju veličine 4. Jedinke za selekciju biramo nasumično i uzimamo najbolju jedinku tj. onu koja ima najveći fitnes među odabranim. Fitnes jedinke predstavlja preciznost klasifikacionog modela na trening skupu.

### 2.3.2 Ukrštanje

Populacija je napravljena tako da jedinke imaju između 4 i 9 slojeva. Omogućeno je da novonastale generacije nastaju od roditelja sa različitim brojem slojeva. Takođe, deca su uzimala nasumičan broj slojeva pri čemu je minimalan broj bio manji od broja slojeva roditelja1 i roditelja2, a maksimalan broj slojeva je maksimum između broja slojeva roditelja1 i roditelja2. Potom je napravljen niz koji je nastao spajanjem svih slojeva oba roditelja i iz njega je izabrano nasumično onoliko slojeva koliko smo odredili za to dete. Napomena je da i dalje ostaje arhitektura u kojoj je prvi sloj uvek Conv2D, a poslednja dva Flatten i zatim Dense

### 2.3.3 Mutacija i elitizam

*Mutaciju* radimo sa verovatnoćom 0.1. Svaki od srednjih slojeva jedinke ćemo potencijalno mutirati. Mutaciju vršimo nasumičnim izborom novog sloja iz skupa slojeva za koje se podrazumeva da mogu stajati na toj poziciji.

U našoj implementaciji koristimo i *elitizam* kako bismo sačuvali najbolje jedinke u populaciji. Elitizmom čuvamo 20% najboljih jedinki.

## 2.4 Izlaz programa

Program se zaustavlja kada preciznost modela pređe vrednost 0.95 ili ako to nije slučaj, kada genetski algoritam prođe broj generacija koji mu je zadat. Kao izlaz dobijamo niz arhitektura najboljih modela, a zatim računamo njihove preciznosti na test podacima, jer su prvobitno računane na skupu za validaciju

## 3 Rezultati

U ovom odeljku prikazacemo naše rezultate na osnovu statistika dobijenih pokretanjem programa nekoliko puta.

Naime, ustanovljene su izvesne analogije:

Modeli koji su se losije pokazali imali su vecinski slojeve MaxPooling2D i AveragePooling2D u okviru srednjih slojeva. Modeli koji su imali znatno bolje performanse su imali vecinski Conv2D slojeve, ali takodje i manji broj slojeva CNN-a.

Pored toga, izvucene su razne statistike, i za veci i za manji broj slojeva.

Modeli koji sadrze i Dropout sloj, sprecavali su overfitovanje, samim tim pokazivali su vecu prezicnost.

Takodje, uoceno je da ukoliko je medju poslednjim slojevima bilo vise od jednog Dense sloja, model je bio nesto optimalniji.

## 4 Zaključak

Projekat je prosao kroz nekoliko verzija. Krajnja verzija projekta koja pronalazi najbolju arhitekturu CNN-a, uz to pamti optimalne parametre roditelja i prenosi ih na nove generacije pokazala se kao dobar i praktican odgovor na temu iz dva razloga.

Modeli brze konvergiraju arhitekturi koja najpreciznije klasifikuje slike u odnosu na Grid Search algoritam, koji ima isti zadatak, ali ga sprovodi potpuno posptuno, bez ucenja.

Ovakav nacin modelovanja arhitekture moze omoguciti i laicima u sveri CNN-a da pronadju model za klasifikaciju adekvatan njihovom skupu podataka.

## Literatura

- [1] [Automatically Designing CNN Architectures using GA](#)
- [2] [Vestacka inteligencija](#)