Neuroevolucija

Projekat u okviru kursa Računarska inteligencija
Matematički fakultet
Univerzitet u Beogradu

Bogdan Marković
mi18099@alas.matf.bg.ac.rs

Septembar, 2022.

Sadržaj

1 Opis problema

Genetski algoritmi pružaju mogućnost optimizacije različitih vrsta problema. Jedna od mogućih primena je i razvoj neuronskih mreža. Postoje razni pristupi u razvoju neuronske mreže, na primer: razvoj topologije mreže, razvoj težina koje koriste slojevi neuronske mreže, kao i kombinacija ova dva pristupa. Neuronske mreže se mogu upotrebiti za različite vrste problema, ovde je odabrano da se koristi za problem regresije, na dva odabrana skupa: boston housing i Life expectancy skup.

U nastavku je dat opis implementacije rešavanja ovog problema, to jest razboja neuronske mreže genetskim algoritmom za potrebe regresije.

2 Implementacija

Svaka jedinka u genetskom algoritmu predstavlja jednu neuronsku mrežu. Hromozom jedinke predstavlja niz realnih brojeva koje predstavljaju težine koje se koriste u okviru slojeva neuronske mreže.

2.1 Klasa DenseLayer

Implementirana je klasa DenseLayer koja predstavlja jedan sloj neuronske mreže. Na nivou te klase vrše se izračunavanja koja kao rezultat daju izlaz iz određenog sloja. Aktivaciona funkcija koja se koristi na svim slojevima (osim na poslednjem) je ReLu.

2.2 Klasa NeuralNetwork

Klasa NeuralNetwork sadrži skup instanci klase DenseLayer, kao i metodu za računanje računanje srednje apsolutne greške koju daje neuronska mreža kada se upotrebi na određenim podacima za regresiju. Kao jedinka genetskog algoritma, NeuralNetwork sadrži i funkciju fitnesa, koja je jednaka negaciji srednje apsolutne greške: $f(x_i) = -MAE(xi)$ gde je x_i jedna jedinka populacije, tj. jedna instanca neuronske mreže. Neuronska mreža dobija kao parametre i atribute trening skupa, na osnovu kojih vrši regresiju.

2.3 Skupovi za treniranje i testiranje mreže

Kao trening i test skupovi odabrani su skupovi boston housing i Life expectancy. Prvi sadrži 13 atributa na osnovu kojih se procenjuje cena kuće, a drugi 18 atributa na osnovu kojih se procenjuje očekivani prosečni životni vek u različitim zemljama, od 2000. do 2015. godine.

2.4 Genetski algoritam

2.4.1 Inicijalna populacija i broj epoha

Inicijalnu populaciju u genetskom algoritmu čine jedinke sastavljene od slučajno odabranih težina iz uniformne raspodele U(-1.2, 1.2). Odabrana veličina populacije za boston housing je 50, a za Life expectancy 100. Odabran broj epoha za prvi skup je 100, a za drugi 300.

2.4.2 Elitizam

Za oba skupa broj najboljih jedinki koje čuvamo za sledeću generaciju je oko 20 posto ukupne populacije.

2.4.3 Selekcija

Za veličinu selekcije uzeto je oko 20 posto veličine populacije. Koritsti se turnirska selekcija kojom iz skupa od oko 20 posto najboljih jedinki u tekućoj populaciji odaberu 2 koje će se koristiti za ukrštanje.

2.4.4 Ukrštanje

Radi dobijanja jedinki koje će činiti narednu populaciju, testirane su 3 vrste ukrštanja 2 roditelja: jednopoziciono ukrštanje, dvopoziciono i uniformno. U oba skupa sva 3 tipa ukrštanja daju slične rezultate, sa malom prednošću uniformnog ukrštanja.

2.4.5 Mutacije

Mutacija je testirana sa verovatnoćom 0.005. Malo bolji rezultati dobijaju se ako se verovatnoća mutacije izračuna na osnovu sledeće formule: $mutation_p = 0.005 * \frac{f(xi)}{f(pi)}^2$. Ovom formulom se podstiče diverzifikacija, zato što je verovatnoća mutacije kod bolje prilagođenih jedinki veća, čime se potencijalno izbegava zaglavljivanje u lokalnom minimumu.

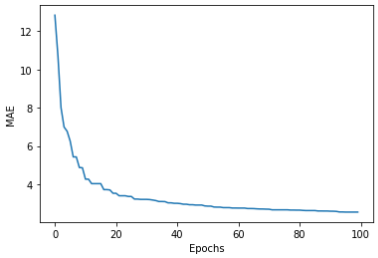
2.4.6 Zaustavljanje i izlaz algoritma, testiranje

Algoritam se zaustavlja nakon odabranog broja epoha i kao rezultat vraća hromozom najbolje jedinke iz poslednje generacije. Na osnovu tog koda kreira se neuronska mreža. Zatim se ta mreža koristi za dobijanje rezultata regresije nad test skupom.

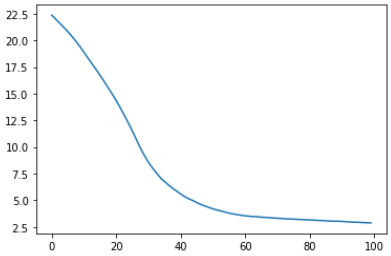
3 Rezultati, poređenje sa propagacijom unazad

Osnovna mera rezultata je srednja apsolutna greška najbolje jedinke, kada se primeni na trening i test skup. Za boston housing skup podataka, ako se odabere arhitektura mreže sa jednim skrivenim slojem i 40 neurona u okviru njega, rezultati su sledeći: srednja apsolutna greška na trening skupu je 2.59, a na test skupu 3.10, što su prihvatljivi rezultati. Kada se to uporedi sa rezultatom biblioteke keras (koja koristi backpropagation uz adam optimizaciju, uz identičnu arhitekturu sa 40 neurona, i 100 epoha), rezultati genetskog algoritma su nešto bolji. Keras daje SAG na trening skupu 2.87, odnosno 3.65 na test skupu.

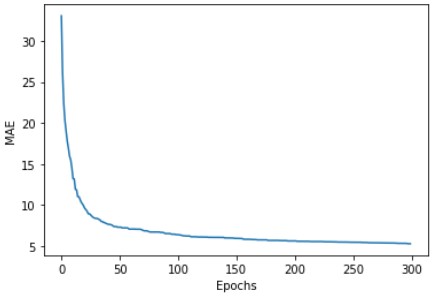
Za life expectancy skup podataka, ako se odabere arhitektura mreže sa dva skrivena sloja sa po 40, odnosno 30 neurona u okviru njega, rezultati su sledeći: srednja apsolutna greška na trening skupu je 4.8372, a na test skupu 5.1, što su nešto vidno lošiji rezultati u odnosu na prethodni skup. Kada se to uporedi sa rezultatom biblioteke keras (koja koristi backpropagation uz adam optimizaciju, uz identičnu arhitekturu sa 40 neurona, i 100 epoha), rezultati tog algoritma su primetno bolji. Keras daje SAG na trening skupu 2.32, odnosno 2.48 na test skupu.



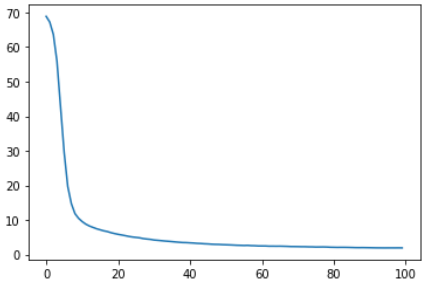
Slika 1: SAG (genetski algoritam, boston housing)



Slika 2: SAG (backpropagation, boston housing)



Slika 3: SAG (genetski algoritam, life expectancy)



Slika 4: SAG (backpropagation, life expectancy)

4 Zaključak

Testiranjem genetskog algoritma sa različitim veličinama populacije, različitim brojem epoha, promenama različitih parametara i upotrebom različitih funkcija za ukrštanje i mutaciju, prethodni rezultati su najbolji do kojih je genetski algoritam došao u dva navedena skupa podataka. Kada se uporedi sa metodom propagacije unazad, za jednostavnije skupove genetski algoritam može da da zavidne rezultate, nekad čak i bolje od propagacije unazad, za isti broj epoha. Međutim, problem nastaje kada se pređe na nešto kompleksniji skup, gde genetski algoritam radi skoro duplo lošije od propagacije unazad. Zaključuje se da je genetski algoritam za razvoj neuronskih mreža u redu koristiti za jednostavnije probleme i probleme manjih dimenzija. U drugim situacijama, propagacija unazad generalno daje dosta bolje rezultate, uz korišćenje manje vremena.