

Podešavanje težina neuronske mreže upotrebom optimizacionog algoritma

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Nikola Stamenić, Lea Petković
mi16177@alas.matf.bg.ac.rs, mi16163@alas.matf.bg.ac.rs

9. juni 2020

Sažetak

U ovom tekstu je ukratko prikazana osnovna forma seminarskog rada.

Sadržaj

1	Uvod	2
2	Neuronske mreže	2
2.1	Keras Pajton biblioteka	2
3	Optimizacija rojem čestica	2
3.1	Originalni algoritam PSO	2
3.2	Druga generacija PSO algoritma	3
3.3	Novi model PSO-a	3
4	Podešavanje težina neuronske mreže pomoću optimizacije rojem čestica	3
4.1	Rezultati	3
5	Zaključak	4
	Literatura	4
A	Dodatak	4

1 Uvod

Ovde ide neki uvod

2 Neuronske mreže

Neuronska mreža (eng. *Artificial Neural Networks, ANN*) je sistem koji vrši mapiranje između ulaza i izlaza problema. Neuronske mreže zapravo predstavljaju parametrizovanu reprezentaciju koja se može koristiti za aproksimaciju raznih funkcija [3]. Matematičkom optimizacijom nekog od kriterijuma kvaliteta vrši se pronalaženje odgovarajućih parametara.

Neuronske mreže uče informacije kroz proces treniranja u nekoliko iteracija. Kada je proces učenja završen, neuronska mreža je spremna i sposobna da klasifikuje nove informacije, predvidi ponašanje, ili aproksimira nelinearnu funkciju problema. Njena struktura sastoji se od skupa neurona, predstavljenih funkcijama, koji su međusobno povezani sa ostalim neuronim organizovanim u slojevima.

Struktura neuronske mreže se razlikuje po broju slojeva. Prvi sloj jeste ulazni sloj, poslednji sloj jeste izlazni, a svi slojevi između se nazivaju skrivenim slojevima. Slojevi su međusobno potpuno povezani. Slojevi komuniciraju zahvaljujući tome što je izlaz svakog neurona, iz prethodnog sloja, povezan sa ulazima svih neurona iz narednog sloja. Jačina veza kojom su neuroni međusobno povezani se naziva težinski faktor (eng. *weight*). Najčešće ima 3 sloja.

Postoje različite vrste neuronskih mreža. Mozemo ih klasifikovati prema: broju slojeva (jednoslojne i višeslojne), vrsti veza između neurona, smeru prostiranja informacija (neuronske mreže sa propagacijom unapred ili unazad) [1], vrsti podataka itd.

Njihove primene su mnogobrojne, obzirom da predstavljaju najčešće primenjivanu metodu mašinskog učenja. Neke od primena su: kategorizacija teksta, medicinska dijagnostika, prepoznavanje objekata na slikama, autonomna vožnja, igranje igara poput igara na tabli ili video igara, mašinsko prevođenje prirodnih jezika, prepoznavanje rukom pisanih tekstova itd.

2.1 Keras Pajton biblioteka

Keras je biblioteka za neuronske mreže, napisana u Pajtonu. Keras radi na platformi za mašinsko učenje *TensorFlow*. Razvijena je sa ciljem da omogući brzo eksperimentisanje sa neuronskim mrežama [2], da bude razumljiva, modularna i proširiva.

Pored, već pomenute, *TensorFlow* platforme, ova biblioteka radi i na: *Microsoft Cognitive Toolkit, R, Theano, ili PlaidML* platformama. Nastala je kao deo istraživanja u okviru projekta ONEIROS (eng. *Open-ended Neuro-Electronic Intelligent Robot Operating System*), a njen autor i održavaoc je gugl inženjer - François Chollet.

3 Optimizacija rojem čestica

U ovoj sekciji biće objašnjen sam algoritam optimizacije rojem čestica. Najviše vremena biće posvećeno originalnom algoritmu PSO, a kao i drugoj generaciji algoritma PSO (eng. *The Secong Generation of PSO*), kao i novom modelu algoritma PSO (eng. *A New Model of PSO*).

3.1 Originalni algoritam PSO

Algoritam PSO (eng. *Particle Swarm Optimization*) je metod za optimizaciju neprekidne nelinearne funkcije, koji je predložio Eberhart. Sam algoritam je inspirisan posmatranjem socijalnog i kolektivnog ponašanja u kretanju jata ptica pri potrazi za hranom ili preživaljavanjem. PSO je nadahnut kretanjem najboljeg člana populacije i njegovog iskustva. Metafora govori da se skup rešenja kreće prostorom pretrage sa ciljem da nađe što bolju poziciju, rešenje [3]. Populacijom se smatra grupa čestica i gde svaka predstavlja poziciju $x_i \in R^D$, $i = 1, \dots, M$ u višedimenzionom prostoru. Čestice se evaluiraju u posebnoj funkciji optimizacije, kako bi se odredila njihova prilagođenost i sačuvala najbolja vrednost. Svaka čestica se kreće po prostoru pretrage u zavisnosti od funkcije brzine v_i koja u obzir uzima globalno najbolju poziciju u populaciji ($p_g \in R^D$ - socijalna komponenta) kao i najbolju poziciju date čestice ($p_g \in R^D$ - kognitivna komponenta). Čestice će se kretati u svakoj iteraciji na drugu poziciju, dok ne dostignu optimalnu poziciju. U svakom momentu t , brzina čestice i se ažurira koristeći:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(p_i(t) - x_i(t)) + c_2 r_2(p_g(t) - x_i(t))$$

gde je ω inertna težina i obično je postavljena da varira linearno od 1 do 0 tokom iteracije, c_1 i c_2 su koeficijenti ubrzanja, r_1 i r_2 su slučajni brojevi iz uniformne (0,1) raspodele. Ubrzanje v_i je ograničeno između $[v_{min}, v_{max}]$. Ažuriranjem ubrzanja na ovaj način dozvoljavamo jedinki i da traži najbolju poziciju $p_i(t)$, dok se najbolje globalno rešenje računa: [3]

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Optimizacija rojem čestica se može prikazati sledećim pseudokodom:

3.2 Druga generacija PSO algoritma

Druga generacija PSO algoritma je unapređenje originalnog PSO algoritma, koja u obzir uzima tri aspekta: lokalni optimum svih jedinki, globalno najbolje rešenje, i novi koncept - geometrijski centar optimalne populacije. Autor knjige *"Second Generation Particle Swarm Optimization"* objašnjava da ptice održavaju određenu distancu između centra jata (hrane). Jata ptica uvek ostaju u istom regionu neko vreme, tokom kojeg će centar jata ostati nepomeren u očima jedinki. Nakon toga, jato se kreće na sledeći region, tada sve jedinke moraju održati određenu distancu sa centrom jata.

3.3 Novi model PSO-a

Ovaj algoritam je predložen od strane Garoa (eng. *Garro*), a on ga je bazirao na osnovu ideja drugih autora koji su predlagali unapređenje originalnog PSO algoritma. Shi i Eberhart su predlagali linearno variranje inertnih težina kroz generacije, što je znatno unapredilo performanse originalnog PSO algoritma. Yu je razvio strategiju da kada se kroz generacije globalno najbolje rešenje ne poboljšava, svaka jedinka i biva izabrana sa predefinisanim verovatnoćom a zatim dodat slučajni šum svakom vektoru brzine dimenzije v_i izabrane jedinke i . Bazirano na nekim evolutivnim shemama Genetičkih Algoritama, nekoliko efektnih mutacija i ukrštanja su predložene za PSO.

4 Podešavanje težina neuronske mreže pomoću optimizacije rojem čestica

Treniranje neuronske mreže, odnosno podešavanje težina, izvršeno je pomoću optimizacije rojem čestica. Napravljena je troslojna neuronska mreža (ulazni, skriveni i izlazni sloj) pomoću Keras biblioteke u Pajtonu, pomoću koje se rešavaju klasifikacioni problemi. Dve trećine podataka korišćene su za trening podatke, a ostatak čini test podatke. Skriveni sloj sadrži 100 jedinica/čvorova, dok je broj izlaza ekvivalentan broju različitih klasa u skupu podataka koji se koristi. Kao metrika korišćena je preciznost (eng. *accuracy*).

Kao što je ranije napomenuto, ANN trenirana je pomoću PSO algoritma. PSO koristi 30 čestica, prolazi kroz 300 iteracija i šalju se trening podaci i neuronska mreža. Težine neuronske mreže predstavljaju pozicije čestica. Inicijalne težine biraju se nasumično unutar intervala $[-2, 2]$, a ažuriranje težina se vrši prema formuli: $x_i(t+1) = x_i(t) + v_i(t+1)$, objašnjenom u poglavlju 3.1.

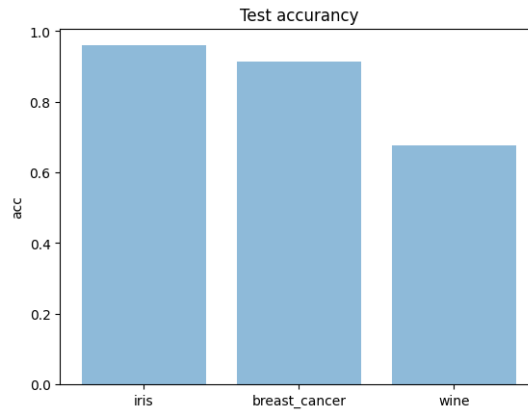
Kao funkcija cilja korišćena je preciznost. Takođe, u cilju izbegavanja zaglavljivanja u lokalnom minimumu algoritam se pokreće nekoliko puta i pozicije najgorih 5 čestica se nasumično menjaju. Najbolja čestica je upravo ona sa najvećom preciznošću.

Ovim poglavljem opišite sve relevantne aspekte vašeg rada: npr. opšti rad algoritma, način kodiranja rešenja i funkciju cilja u slučaju da radite optimizacionu tehniku, itd. Budite objektivni i poštenu u opisu svoje metode i jasno ukažite na to da ste reprodukovali pristup koji je neko već osmislio pre vas tako što ćete ubaciti odgovarajuće reference ka svim relevantnim radovima.

4.1 Rezultati

U ovoj sekciji biće predstavljeni eksperimentalni rezultati dobijeni nad skupovima: Iris, Breast Cancer i Wine, kao i upoređeni sa rezultatima drugih istraživača koji su se bavili ovom temom i koristili navedene skupove.

Ovo poglavlje sadrži tabele i grafike kojim upoređujete vaš pristup sa drugim pristupima iz literature. Pronađi relevantne test primere (instance) i testirati predloženu metodu (metode) na njima, ukoliko su dostupne. Ako ne možete da dođete do test primera iz literature, kreirajte na osnovu opisa iz radova test primere koji su im slični. Testirajte vaše metode pod



Slika 1: Preciznost na test podacima skupova: iris, breast cancer i wine.

sličnim uslovima kao i autori iz literature. Pri demonstraciji rada metode uvek pokazati kako vaša metoda radi na na jednostavnim test primerima koji se mogu vizualizovati. Specijalno, ako ne upoređujete rešenje sa drugim rešenjima, možete komentarisati efikasnost algoritma i procenjivati kvalitet tako što ga upoređujete sa drugim pristupima koje ste sami razvili (npr. algoritmom potpune pretrage, eng. brute-force u slučaju da radite optimizacionu tehniku). Budite objektivni i poštteni u prikazu svojih rezultata i jasno naznačite ako vaš pristup nije bolji (što je vrlo verovatno) od onih koji su predloženi u literaturi. U okviru ovog poglavlja opišite i eksperimentalno okruženje u kojem ste testirali vaš program, karakteristike hardvera, operativni sistem, kompajler, itd.

5 Zaključak

Ovde pišem zaključak.

Literatura

- [1] Internet Archive Wayback Machine. on-line at: <https://web.archive.org/web/20090215055110/http://learnartificialneuralnetworks.com/#Intro>.
- [2] Keras. on-line at: <https://keras.io/>.
- [3] Beatriz A. Garro and Roberto A. Vázquez. Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms. 2 June 2015.

A Dodatak

Ovde pišem dodatne stvari, ukoliko za time ima potrebe.