

MOGUĆNOSTI GENETSKIH ALGORITAMA U MAŠINSKOM UČENJU

Milan Vugdelija¹
mr Jozef Kratica¹
Vladimir Filipović²
mr Slobodan Radojević¹

REZIME

U prvom delu članka daju se opšte ideje koje se koriste u genetskim algoritimima. Nakon toga, prikazane su neke primene tih ideja u kreiranju sistema koji poseduju veštačku inteligenciju.

1. GENETSKI ALGORITMI

Genetski algoritam (GA) je model mašinskog učenja u kome se događaji dešavaju po analogiji sa procesima koje izučava genetika, iz koje je i peuzet veliki broj termina. Opšta ideja se sastoji u sledećem:

Pretpostavimo da je data neka funkcija. Potrebno je (bar približno) odrediti vrednosti argumenata u kojima funkcija dostiže maksimum. U velikom broju takvih problema skup mogućih vrednosti argumenata je beskonačan ili previše veliki, tako da isprobavanje svih mogućnosti ne dolazi u obzir, a nije poznat ni algoritam koji bi sa dovoljnom sigurnošću dovodio do rešenja. U takvim situacijama može se primeniti sledeći postupak [1]:

- Nekoliko početnih vrednosti za argumente se zadaju na slučajan način, ili pomoću nekog jednostavnog postupka koji daje rezultate bolje od slučajnih.

- Sračunaju se odgovarajuće vrednosti funkcije.

- One vrednosti argumenata za koje se dobijaju najveće vrednosti funkcije se čuvaju, a argumenti kojima odgovaraju male vrednosti funkcije se odbacuju. Što je dobijena vrednost funkcije veća, to se odgovarajući argumenti čuvaju u više primeraka. Ovaj deo postupka zove se odabiranje (selection).

- Argumenti koji su 'preživeli' odabiranje, kombinuju se između sebe i na taj način se dobijaju sasvim nove vrednosti. Ovaj deo postupka zove se ukrštanje (crossover).

¹ Katedra za Matematiku, Mašinskog fakulteta u Beogradu, 27.marta 80.

² Matematički fakultet u Beogradu, Studentski trg 16.

- Uvodi se mala slučajna promena na vrednostima argumenata koji su dobijeni ukrštanjem prethodnih vrednosti. Ovaj deo postupka zove se mutacija (mutation).

- Pelazi se ponovo na sračunavanje vrednosti funkcije. Postupak se ponavlja sve dok se ne ispune uslovi za prekid, na primer da se dostigne izvestan nivo sličnosti među vrednostima argumenata koje treba ispitati, da se vrednosti najboljih argumenata razmnože do izvesne mere u okviru populacije, da se postupak izvede unepred zadati broj puta, itd.

Svaki postupak koji radi po ovoj opštoj ideji, naziva se genetski algoritam. Pri tome reč funkcija u gornjem tekstu treba shvatiti u najopštijem smislu. Navedimo samo dva primera:

- Neka je potrebno sprovesti cevovod između dve tačke u prostoru, tako da se minimizuje otpor pri protoku fluida kroz cevi. Dužine pojedinih cevi i veličine uglova koje one međusobno zaklapaju možemo smatrati za argumente funkcije, a ukupni otpor koji nastaje proticanjem nekog fluida kroz taj sistem cevi za vrednost funkcije. U cilju pronalaženja što optimalnijeg položaja cevi može se primeniti genetski algoritam.

- Banka želi da dobije formulu po kojoj će, na osnovu podataka o klijentu da donese odluku da li da mu odobri kredit ili ne. Na osnovu dosadašnjeg poslovanja, banka raspolaže velikim brojem informacija o (poslovnoj) ispravnosti odluka u prethodnim situacijama. Razne formule koje barataju podacima o klijentima možemo smatrati argumentima, a kvalitet takvih formula, dobijen testiranjem formule na prethodnom iskustvu, možemo smatrati vrednošću funkcije. Radi dobijanja jednog takvog pravila za odlučivanje može se primeniti genetski algoritam.

Strogo govoreći, postoji razlika između genetskih algoritama i evolucionih algoritama, pre svega u reprezentaciji podataka, ali opšta ideja je vrlo slična i preuzeta je iz prirode, a to je princip na kome funkcioniše i prirodna evolucija - najbolje jedinke dobijaju mogućnost da svoje kvalitete prerasporede na novi način, čime se mogu dobiti još bolje jedinke, dok lošije jedinke takvu mogućnost ne dobijaju.

2. MAŠINSKO UČENJE

U praksi se veoma često javlja potreba za donošenjem odluke (na pr. u gore navedenom primeru potrebno je odlučiti da li se kredit odobrava). Da bi mašina vršila takvo odlučivanje, potrebno je da joj se zada postupak odlučivanja. Međutim, čovek se u svom odlučivanju uglavnom ne koristi formulom, nego akumuliranim znanjem koje nije lako formalizovati. Stoga je od interesa omogućiti mašini da na osnovu primera ispravnog i neispravnog odlučivanja (ili na neki drugi način) sama generiše postupak za odlučivanje, koji će kasnije kroz testiranje ili kroz povratne informacije u radu korigovati i usavršavati. Ovo formiranje postupka za odlučivanje i njegovo usavršavanje od strane mašine naziva se mašinskim učenjem [2].

3. POSTOJEĆE APLIKACIJE

Ovde ćemo navesti nekoliko primera, u kojima je primenom GA ostvaren u praksi mehanizam mašinskog učenja.

3.1. Softverski kontroler autonomnog vozila

Alan C. Shultz, John J. Grefenstette i Kenneth A. De Jong [3] izveštavaju o primeni GA na formiranje softverskog kontrolera za autonomno vozilo.

Autori koriste pojam scenario greške (fault scenario). Pod njim se podrazumeva opis grešaka koje mogu nastati pri funkcionisanju vozila, i uslova pod kojima može doći do takvih grešaka. Scenario može sadržavati i informacije o okruženju u kome

vozilo funkcioniše. Drugim rečima, scenario greške se sastoji od pravila koja govore pod kojim uslovima nastupa koji tip greške. Cilj je doći do takvih scenarija po kojima kontroler može da prepozna i izbegne situacije kritične po funkcionisanje vozila.

Koristi se posebna aplikacija, koja prati veličine koje opisuju stanje vozila (brzina, položaj, pravac kretanja itd.) i druge bitne parametre (udaljenost do cilja, brzina i pravac vetra itd.) i simulira kretanje vozila. Tokom rada tog programa se isprobavaju razni scenariji, i korisnost scenarija se ocenjuje na osnovu uspešnosti misije.

U genetskom algoritmu koji je upotrebljen, natprosečno korisni scenariji se ukrštaju i generišu nove, loši scenariji se odbacuju i zamenjuju generisanim, dok se ne dobiju scenariji po kojima se sa traženim stepenom sigurnosti misija može uspešno završiti.

3.2. BEAGLE

BEAGLE (bigl) je skraćenica od Biologic Evolutionary Algorithm Generating Logical Expressions - evolucioni algoritam koji generiše logičke izraze. Radi se o komercijalnom programskom paketu koji se sastoji od 6 odvojenih celina [4].

Programski paket BEAGLE ima za ulazne podatke kolekciju objekata i njihove opise. Svaki objekat ima svojstva - attribute koji čine opis datog objekta. Paket na osnovu tih podataka generiše logičke izraze koji operišu atributima datih objekata, a koji razvrstavaju objekte u odgovarajuće klase. Svaki logički izraz koji se kreira prolazi kroz testiranje na svim datim objektima, koristeći dodatno znanje o stvarnoj pripadnosti tih objekata pojedinim klasama. Na osnovu testiranja se ocenjuje kvalitet logičkog izraza, dajući poseban bonus kratkim izrazima. Natprosečni izrazi se dalje kombinuju, tako da se dobijaju novi izrazi, koji se dodatno ocenjuju, dok se ne dobije izraz koji sa zadovoljavajućom tačnošću razvrstava objekte, ili dok se ne pokaže da dalje nastavljanje postupka ne dovodi do poboljšanja.

Karakteristično je da ovde jedinke nad kojima se vrši genetski algoritam (logički izrazi) nisu kodirane stringovima fiksne dužine kao što je to obično slučaj, već pomoću binarnih stabala koja mogu biti različite dužine. Ukrštanje je realizovano tako što se razmene slučajno izabrani podizrazi koji se nakon razmene mogu povezati drugačijom operacijom.

Ovaj programski paket je već primenjen u veoma različite svrhe od ocenjivanja profitabilnosti u raznim vrstama ulaganja imovine, do prognoziranja rezultata konjskih trka.

Značajno je napomenuti da je evolucioni metod u jednoj od važnijih primena paketa BEAGLE (prepoznavanje raznih vrsta stakla, Evett and Spiehler, 1987) u svakom eksperimentu prevazišao standardni statistički metod, i da su logički izrazi dobijeni pomoću ovog programskog paketa, ugrađeni u ekspertni sistem za identifikaciju vrsta stakla, što nije moglo da se postigne upotrebom drugih metoda.

3.3. Sistem za klasifikovanje koji uči (Learning Classifier System)

David E. Goldberg izveštava o primeni GA na dva inženjerska problema, oba u vezi sa optimizacijom protoka prirodnog gasa kroz cevovod[5].

Drugi od ta dva rada odnosi se na sistem za klasifikovanje koji uči. Ovde se takođe koriste pravila oblika ako - onda. Takva pravila primaju poruke i na osnovu njih mogu da dejstvuju. Sistem se sastoji od tri dela:

- interpreter pravila koji šalje poruke
- sistem koji razdeljuje svim pravilima podjednake količine kredita
- genetski algoritam

Prvi deo sistema prihvata poruke iz spoljnog sveta, zapisuje ih u listu poruka u specijalnom obliku pogodnom za GA (ovde su to bili stringovi fiksne dužine koji su se sastojali od tri moguća znaka, 0, 1 i \emptyset). Poruke sadrže podatke o ulaznom pritisku, količini gasa koji pritiče, dobu dana, dobu godine, itd. u detaljnoj simulaciji ponašanja cevi.

Drugi deo sistema bazira se na ideji Džona Holanda, nazvanoj Bucket Brigade Algorithm [1]. Svako pravilo dobija na početku izvesnu sumu kredita. Pravilo koje dejstvuje, plaća za svoje dejstvovanje neku svotu. Ako je pravilo izazvalo uspešnu akciju, ono dobija nagradu u vidu još kredita. Deo nagrade dobija i prethodno pravilo koje je poruku prosledilo do pravila koje dobija nagradu. Na taj način kvalitetna pravila se "obogate", a loša pravila "bankrotiraju". Na mesto pravila koja su bankrotirala dolaze pravila koja se generišu ukrštanjem uspešnijih pravila.

4. ZAKLJUČAK

Od polovine osamdesetih godina do sada stalno se pojavljuju novi radovi o uspešnoj primeni GA na rešavanje raznih problema. Primera za to je svakog dana sve više. Primene su vrlo raznovrsne i ne odnose se samo na mašinsko učenje i prepoznavanje oblika, tj. klasifikaciju. Na primer, na Tehničkom Univerzitetu u Berlinu koristi se GA za dizajniranje oblika propelera, u Volkswagenu se GA primenjuje za optimizovanje oblika vozila, postoje razne primene u elektronici (za optimalno raspoređivanje komponenti u čipovima, za projektovanje elektronskih kola), u finansijama, itd [2].

Ideje koje su korišćene u osnovi su ideje evolucionih/genetskih algoritama, međutim već iz navedenih primera je jasno da u realizaciji tih ideja postoji velika raznolikost.

Na osnovu rečenog se može doneti nekoliko zaključaka:

- GA obezbeđuju jedan robustan i opšti metod za rešavanje problema, koji može biti koristan u slučajevima kada postoji funkcija za ocenu kvaliteta rešenja, a nema analitičkog rešenja.

- Oblast razvijanja i primene GA je u ekspanziji, pa se mogu očekivati novi izveštaji o dostignućima u praktičnoj primeni genetskih algoritama, kako u oblasti mašinskog učenja, tako i u drugim oblastima.

- U okviru razvoja GA postoje vrlo raznovrsne ideje koje nisu ni izbliza iscrpene, tako da je oblast pogodna i za primenu razvijenih ideja, ali i za originalne doprinose i teorijsko proučavanje.

5. LITERATURA

[1] David E. Goldberg, *Algorithms in Search, Optimization and Machine Learning*, Addison-Westey, Reading, MA, 1989.

[2] Richard Forsyth, "The evolution of intelligence", *Machine Learning: Principles and techniques*, pp. 65-82, Chapman and Hall, New York, 1989.

[3] Alan C. Schultz, John J. Grefenstette, Kenneth A. De Jong, "Test and Evaluation by Genetic Algorithms", *IEEE Expert*, October 1993. pp. 9-14

[4] Richard Forsyth, *PC BEAGLE User Guide*, Warm Boot Ltd., Nottingham, 1987

[5] David E. Goldberg, "Genetic algorithms and rule learning in dynamic system control", *Proc. Int. Conf. on Genetic Algorithms and their Applications*, Carnegie Mellon University, Pittsburgh, 1985.

ABSTRACT

In this paper we introduce the basic idea of genetic algorithms, and then we give some illustrations of applying genetic algorithms to machine learning.