Fakultet elektrotehnike i računarstva

Zavod za primjenjeno računarstvo

**Napredni algoritmi i strukture podataka**

2. laboratorijska vježba

Vinko Kodžoman 0036470625

Zagreb, 7.12.2015

**Zadatak**

1. Riješiti zadatak s neuronskom mrežom iz skupine za 5 bodova, ali pisanjem vlastitog programa (poželjno C, C++, C# ili Java). Pri tome nije potrebno grafički prikazati izlaz mreže (naravno, nije ni zabranjeno), već je kao dokaz uspješnosti dovoljno izračunati neku objektivnu mjeru (brojčani pokazatelj).
2. Genetskim algoritmom riješiti neki složeniji problem koji će zahtijevati pomniju razradu osnovnih mehanizama (križanje, mutacije, itd.). Poželjno je napraviti više različitih modela (mehanizama) i usporediti rezultate.

**Napomena**: Spoj 1) i 3) zadatka. Genetski algoritmo (HGA) se treniraju neuronske mreže za aproksimacije funkcije.

**Rješenje zadatka**

**Teorijski uvod**

Automatizirana obrada podataka se danas uglavnom obavlja u računalima. Naspram računala živi organizmi imaju močnu obradu podataka putem živčanog sustava. Pristup obradi podacima visokom paralelizacijom sličnom obradi podataka u živčanom sustavu (mozak). Mozak se sastoji od velikog broja neurona koji rade paralelno. Grana računarstva koji riješava probleme kopirajući živćani sustav živih organizama se zove Neuro-računarstvo, i glavna tema su umjetne neuronske mreže.

Neuronske mreže su skup međusobno povezanih jednostavnih procesnih jedinica (neurona) čija se funkcija temelji na biološkom neuronu i služe za distribuiranu obradu podataka. Glavna zadaće neuronski mreža su klasifikacija, raspoznavanje uzoraka i predviđanje. Velika prednost neuronskih mreža je robusnos na pogreške, rad sa nejasnim ili manjakvim podacima.

Genetski algoritmi slično kao i neuronske mreže pokušavaju modelirati stvari iz prirode. Glavna jezgra genetski algoritama je proces evolucije. Evolucija koja kroz dug period mjenja žive organizme procesom selekcije i genetičkim driftom. Jedinke sa većim reproduktivnim uspjehom imaju veći fitnes i ujedno veće šanse za prenjeti svoje gene u sljedeću populaciju (potomke). U računalima svaka jedinka ima neka svojstva nad kojima djeluju sile mutacije i križanja. Dvije jedinke iz populacije se križaju kako bi napravili potomka, a mutacija se izvodi na jedinci nakon križanja kako bi bolje modelirala evolucijski proces (genetički drift), te izvadila jedinke iz lokalnih minimuma.

**Implementacija**

Program je napisan u programskom jeziku Java u IntelliJ IDEA 15 (Jet Brains). Niti jedna vanjska biblioteka nije korištena, osim onih ugrađenih u JE-08 (kolekcije, rad sa ulazom, izlazom, ...)

1. **IOEngine**

Parser učitava točke funkcija iz datoteke u obliku ulaz izlaz. Učitane podatke modelire u funkciju koja se sastoji od x i y podataka (double). Nakon što su svi podaci učiani korisnik bira vrstu rada algoritma i sve potrebne konfiguracije (broj neurona u srednjem sloju, vrsta rada (error ili iteracija) i broj jedinki u populaciji).

1. **Neuronske mreže**

Neuronska mreža se sastoji od tri sloja u obliku 1 x N x 1, gdje postoji jedan ulazni i izlazni neuron (funkcija sa jednom varijablom). N je broj neurona u srednjem sloju i njega korisnik zadaje.

1. **Genetički algoritam**

Genetički algoritam je hibridni, tj. jedinke su neuronske mreže opisane u cijelini iznad. Nad jedinkama se izvodne operacije **križanja** i **mutacije**. Genetički algoritam ima ugrađen elitizam koji osiugrava da se jedinka sa najboljim fitnessom prenosi u sljedeću generaciju. Fitness se računa kao razlika dobivene i prave vrijednosti iz ulazne datoteke.

**Križanje** je operacije koje kao razultat stvara novu neuronsku mrežu iz dvije roditeljske. Djetetu se dodjeluju težine na neuronskim vezama kao slučajan broj u uniformnoj razdiobi između težina svojih roditelja.



Slika 1 Kod koji prikazuje križanje dvije neuronske mreže, tj. nastanak djeteta

**Mutacija** je urnarna operacija i izvodi sa nad jednom neuronskom mrežom. Neuronske mreže imaju definiranu mutaciju kao slučajan broj iz normalne razdiobe . Broj K zadaje korisnik (npr. K = 0.1, K = 5, K = 10, ...) i algoritam mutacije se vrši nad svakom težinom neuronske mreže zasebno.

**Zaključak**

Genetički algoritam najčešće nije prvi izbor za rješavanje problema (upotrebljava se kada ne postoji neka druga bolja ne heuristička – egzaktna metoda), ali kao i većina metaheurističkih metoda njegov rezultat je dovoljno dobar da bi bio upotrebljiv i dobiven je u prihvatljivom vremenskom roku. Genetički algoritmi omogućuju veliku fleksibilnost u primjeni nad brojim problemima, kao što je koršitenje HGA za treniranje neuronski mreža. Neuronske mreže rješavaju probleme klasifikacije i raspoznavanja uzoraka čak i na skromnim računalima, to plus njihova otpornost na šumove.

**Literatura**

|  |  |
| --- | --- |
| **Vrsta** | **Format** |
| Knjiga | Domagoj Kusalić: Napredno programiranje i algoritmi u C-u i C++-u 5. nepromijenjeno izdanje, Element, Zagreb, 2014. |
| Prezentacije | UI 12 – Umjetne Neuronske Mreže |