

## GA

Algoritmii genetici is inspirati din procesul de evolutie descris de Darwin. Fata de alte tehnici, unde o solutie la problema se construiesc iterativ (vezi greedy), aici se pleaca cu solutii gata facute, care se numesc cromozomi si constituie asa numitul material genetic pentru procesul de evolutie si de selectie naturala. Algoritmii genetici se folosesc pentru probleme unde se doreste maximizarea/minimizarea unei functii si unde spatiul solutiilor (aka toate solutiile posibile) sunt foarte nasol de explorat prin brute force (aka backtracking). Ma refer aici la probleme NP-complete, cum ar fi problema comis voiajorului (aka travelling salesman problem). Noah bun, sa aveti in minte un grafic cu vai si dealuri. GA-ul incearca sa se „plimbe” pe el si sa gaseasca cea mai adanca vale. Sa aveti in vedere ca denumirea din GA a functiei de minimizat este „fitness”. Deci orice solutie/cromozom are asociata o functie de fitness, cat de bun ii respectivul cromozom fata de restul dintr-o populatie. Acuma avem 2 mecanisme care trebuie echilibrate: explorare si exploatare. Explorarea ii aia care ma scoate dintr-o vale (atractor local) care se poate sa nu fie tot una cu minimul global. Practic imaginati-va o bila care o picat intr-o groapa si ii trebe un sut undeva s-o scoata de acolo ☺. Asta se face mai ales prin mutatie, care sa zicem ca mere asa: alegem o gena random (o bucata din cromozom) si o alteram. Explorarea se poate face si prin crossover intr-o masura, care zice asa: avem 2 cromozomi, alegem un punct de taietura (single cut point crossover) si interschimbam genele celor 2 dinainte si de dupa taietura. Remarcam ca mutatia ia un cromozom si scoate un cromozom. Crossover-ul ia 2 cromozomi (parents) si scoate 2 cromozomi (offspring).

La finalul unei epoci evolutive (dupa ce ne-am facut damblaua cu mutatie/crossover), o sa avem mai multi cromozomi in „pool” decat am avut la inceput. Aici intervine exploatarea. Asta ii partea care, fiind cu bila pe o panta, ne duce catre cel mai jos punct al vail. Se sorteaza populatia dupa fitness si se arunca la gunoi excesul de cromozomi. Raman numai cei mai puternici, ca in natura. Asta se cheama selectie elitista. Repetam procedeul de N ori si la final primul element din populatie este solutia problemei.

### 1.perceptronul

cand? -> il folosim in probleme de clasificare binara

$$\begin{array}{c} \backslash \quad + \\ - \quad \backslash \quad + \quad + \\ - \quad - \quad \backslash \quad + \quad + \quad + \\ - \quad - \quad - \quad \backslash \\ \wedge \\ | \\ (x_1, x_2) \end{array}$$

$$= w_1x_1 + w_2x_2 + \text{bias} \quad (ax + by + c) \rightarrow x = x_1 \quad y = x_2 \quad c = \text{bias}$$

bias e translatia cat de la stanga sau dreapta e dreapta

XOR nu e liniar separabila

$$0 \wedge 0 = 0$$

$$1 \wedge 0 = 1$$

$$0 \wedge 1 = 1$$

$$1 \wedge 1 = 0$$

Nu exista linie sa le separe pe astea

2.retea neuronală

3 layers

O O O } FULLY CONNECTED

O O }

cand folosim?

in probleme de clasificare

exemplu detectare de cifre

aveam o matrice de pixeli de 8x8

si aveai 255 valoarea in pixelii care aveau culoarea alba si in rest 0 ca sa formeze o cifra

cati neuroni de input trebuie sa am?

retea are un layer flat de neuroni 0 0 0 0 0 0 0 0

facem flatten la matrice

[]

[]

luam coloana de jos si o concatenam mai sus

si repetam pentru toate randurile

si optinem o linie de 64 de elemente deci 64 de elemente(neuroni) de input

cati hidden? aprox 100 dar dupa gust in general

si cati neuroni de output? 10 -ca 10 cifre

retelele au valori intre [0,1]

cum le aducem in intervalul ala?  $(v-v_{min})/(v_{max}-v_{min})$

la noi o sa fie  $(v-0)/(255-0)$

output cifra 4 se encoda intr un vector de lungime 10 unde era valoarea 1 pe pozitia 5

( 0 0 0 0 1 0 0 0 0 0 )

^

(0,25 0,4 0,78 ...)

se face o predictie iar numarul (0,1) sa fie apropiat de 1

N epoci

s ori

o epoca inseamna parcursul datelor de antrenament

### 3.Algoritmi genetici

cand? probleme de minim sau maxim cand spatiul solutiilor este foarte mare

de obicei se folosesc pentru complete NP-complete adica nu exista alg pol

-> pornim direct cu solutia

avem entitatea numita chromosome  $\Leftrightarrow$  solutie

acest chromosome are asociata o functie de fitness:  $R^n \rightarrow R$

acesti algoritmi au un tradeoff

facem backtracking si gasim cea mai buna dar dureaza foarte mult

acesti algoritmi genetici sunt intre brute force si random search

daca ai un grafic cu minim local si minim global punctul din minimul local

trebuie sa vrea sa exploreze alte gropi pentru a gasi cel global

EXPLOITATION -> SELECTION

EXPLORATION -> MUTATIE & CROSSOVER -> introduce new gen material

for epoch in EPOCHS

|MUT&CROSSOVER

|SORT BY FITNESS

|SELECTION

-

RETURN POP[0]

MUTATIE(c)->offspring (primeste ca param un chromosom si returneaza un offspring)

la CROSSOVER(c1,c2)->o1,o2 avem 2 chromosomi si 2 offsprings returnati

un exemplu de crossover

0 1 0 1 1 0 0

1 1 1 1 0 1 0

prima bucata din primul se concateneaza cu ultima din a doua

si a doua din primu cu prima din a doua

pentru a introduce element genetic nou

#### 4.PARTICLE SWARM OPTIMIZATION

cand? determinare minimului sau maximului unei functii continue pe un interval  $[i1,i2]$

$f(x,y)=x^2+y^2$  -> minimize -> are minim in (0,0)

$=x^2-y^2$  ->asta nu are minim, asta e saddle point ala de mai sus arata ca un U in 3d

Astazi recunoastere de fete