GITHub: https://github.com/dragosprodan/tema_sii

Metoda gradient descent

Modelarea coeficienților β:
la iterația 0: valori random (sau 0)
laiterațiat+1(t=0,1,2,...)
$\beta k(t+1) = \beta k(t) - learning\_rate * error(t) * xk, k=1,2,\ldots,d$
$\beta 0(t+1) = \beta 0(t) - learning\_rate * error(t)$
Unde
error(t) = computed - realOutput
$error(t) = \beta 0(t) + \beta 1(t)*x1 + \beta 2(t)*x2 + ... + \beta d(t)*xd - y$

BATCH gradient descent:
Eroarea se calculează pentru fiecare exemplu de antrenament
Modelul se updatează dupa ce toate exemplele de antrenament au fost evaluate (la finalul unei epoci)

Algoritm evolutivi

Initializare populaţie P(0)
Evaluare P(0)
g := 0;
CâtTimp (not condiţie_stop) execută
Repetă
Selectează 2 părinţi p1 şi p2 din P(g)
Încrucişare(p1,p2) =>o1 şi o2
Mutaţie(o1) => o1*
Mutaţie(o2) => o2*
Evaluare(o1*)
Evaluare(o2*)
adăugare o1* şi o* în P(g+1)
Până când P(g+1) este completă
g := g + 1
Sf CâtTimp

Algoritm neural network

Se iniţializează ponderile
Cât timp nu este îndeplinită condiţia de oprire
Pentru fiecare exemplu
Se activează fiecare neuron al reţelei
Se propagă informaţia înainte şi se calculează ieşirea corespunzătoare fiecărui neuron al reţelei
Se ajustează ponderile
Se stabileşte şi se propagă eroarea înapoi
Se stabilesc erorile corespunzătoare neuronilor din stratul de ieşire
Se modifică ponderile între nodurile de pe stratul ascuns şi stratul de ieşire
Se propagă erorile nodurilor de pe stratul de ieşire înapoi în toată reţeaua -> se distribuie erorile pe toate conexiunile existente în reţea proporţional cu valorile ponderilor asociate acestor conexiuni
Se modifică ponderile între nodurile de pe stratul de intrare şi stratul ascuns

Algoritm AdaBoost+RegressionTree

Se importa datele
Se elimina parametri irelevanti
Se initializeaza crossvalidatorul
Cat timp N nu este 16
    Sa initializeaza un RegressionTree de adancime N
    Se antreneaza RegressionTree-ul
    Se calculeaza CrossValidatorScore-ul
    Daca este mai bun se inlocuieste adancimea maxima
Se initializeaza un AdaBoostRegressor
Se antreneaza AdaBoostRegressor-ul
Se calculeaza CrossValidatorScore-ul
Se calculeaza Eroare medie patratica

Algoritm LASSO

Se importa datele
Se elimina parametri irelevanti
Se impart datele in test si train
Se initializeaza modelul
Se antreneaza modelul
Se calculeaza Eroare medie patratica