**Tehnici de Optimizare**

*1) Concepte generale*

*2) Optimizări*

*3) Algoritmi implementați*

În cazul algoritmilor de IA, acuratețea și minimizarea numărului de operații complexe efectuate de aceștia sunt factorii cei mai importanți care descriu un algoritm. În implementările noastre am încercat să creștem semnificativ acuratețea unor algoritmi de IA și apoi să îi optimizăm.

3.1) Optimizări pe KNN

Algoritmul KNN este unul din cei mai simpli algoritmi de clasificare. KNN (K-Nearest Neighbours) decide clasa unei intrări astfel: algoritmul caută cei K mai apropiați vecini ai intrării și îi asignează clasa cea mai comună dintre acești vecini.

!!! INSERT KNN IMAGE HERE (DIN CONSPECT 2, PAG. 3)

Deși algoritmul este simplu de înțeles și implementat, exista 2 factori importanți care pot influența acuratețea și performanța algoritmului: funcția aleasă pentru calcularea distanței și valoarea aleasă pentru K. Există multe funcții propuse pentru calcularea distanței, distanța Euclidiană fiind cea mai comună și cea aleasă de noi în implementări.

Avem trei implementări de KNN care rulează concomitent, cu optimizări alese pentru găsirea unei valori pentru K. Aceștia sunt antrenați pe același set de date de antrenare (numit A) și acuratețea lor la finală este măsurată pe același set de date de testare (numit T).

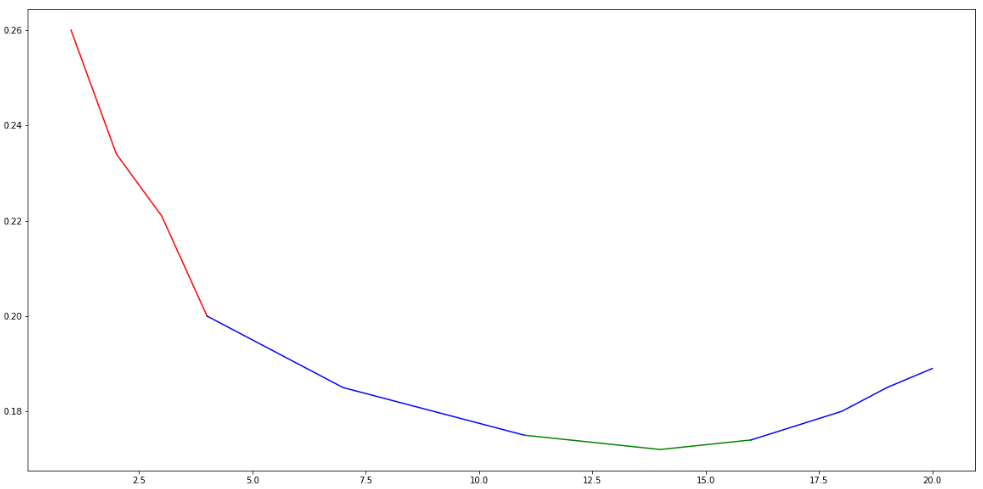
Prima implementare este un KNN simplu, fără optimizări, cu valoarea lui K aleasă ca 3 (majoritatea implementărilor aleg un K arbitrar, cel mai des 2 sau 3). Deși aceasta este cea mai rapidă dintre implementări, suferă din punct de vedere al acurateței.

Celelalte implementări împart setul de date de antrenare in 2: un set pe care se face antrenarea efectivă (90% din datele totale ale setului de antrenare, pe care îl numim α) și un set de testare pentru K (restul datelor de antrenare, pe care îl numim β). După selectarea unui K și antrenarea datelor pe α, se testează acuratețea algoritmului pe setul β.

Al doilea algoritm măsoară acuratețea modelului pentru toate valorile posibile ale lui K (de la 1 la N, unde N este numărul intrărilor din α) și alege K cu cea mai bună acuratețe. Această implementare devine foarte lentă și ineficientă pentru un set de date foarte mare.

Al treilea algoritm aplică o metodă „Divide et Impera” pentru a-l alege pe K, folosind tot acuratețea pentru alegerea sa. Se pornește de la intervalul 1, ..., N și se aplică recursiv până se găsește K optim. Pentru un interval ales, se măsoară acuratețea pentru valorile din capetele sale, intervalul restrângându-se la jumătate în funcție de valorile acestea. Algoritmul este mult mai eficient ca cel precedent și știm că găsește aceeași valoare optimă, cu excepția cazurilor în care seturile de date sunt foarte mici și acuratețea pe valorile lui K nu este stabilă. Noi am găsit un set de date relativ mic pentru antrenare si testare, totuși se observă că algoritmul funcționează și pe acesta, algoritmii 2 și 3 având aceeași acuratețe. Numărul mic de date de intrare pe care îl avem explică și acuratețea slabă în general a modelului nostru (57.70 pentru algoritmul 1 și 62.55 pentru algoritmii 2 și 3) deoarece KNN are nevoie de seturi de date mai mari decât alți algoritmi de clasificare pentru a ajunge la o acuratețe bună. Totuși, și în aceste condiții, se observă o creștere semnificativă a acurateței măsurate pe setul T după aplicarea algoritmilor.

Al treilea algoritm este optimizat și are cea mai bună acuratețe. Mai jos avem un grafic, reprezentativ KNN aplicat pentru seturi mari de date. Pe axa OX sunt valorile alese pentru K și pe OY este eroarea de clasificare (invers proporțională cu acuratețea).



După cum se observă, pentru valori mici ale lui K (primele 2 secțiuni) eroarea este foarte mare dar scade cu cât K crește, ajungând până la un interval de stabilizare (secțiunea verde) în care se află K-ul optim (cu eroarea cea mai mică) și apoi începe iar să crească (ultima secțiune).

3.2) Optimizări pe Gradient Descent

3.3) Optimizări pe Stochastic Gradient Descent

*4) Lucrarea „A Dual Coordinate Descent Method for Large-scale Linear SVM”*