

Comparație între Metode de optimizare folosite in clasificare

1 Sarcina de învățare

Sarcina de învățare abordată în acest studiu este una de clasificare multi-class. Scopul este de a evalua și compara performanța a două metode de optimizare distincte pentru antrenarea unui model de clasificare: Gradient Descent (GD) și o metodă alternativă de optimizare (in plus, Metoda Newton este si aceasta implementata in sa acuratetea si performanta acesteia sunt scazute).

2 Baza de date utilizată

Baza de date utilizată conține 9 clase distincte, fiecare cu caracteristici specifice unei afecțiuni ale frunzelor roșiilor. Pozele au fost împartite in 9 categorii, una in care sunt prezentate frunze ale plantelor sanatoase, iar celelalte 8 frunze cu diferite afecțiuni, de la pete galbene, pana la frunze infestate de insecte. Datele au fost împărțite în seturi de antrenare și testare pentru evaluarea performanței modelului(setul de date provine de pe site-ul <https://www.kaggle.com/>).

Fiecare clasă conține un număr mare de exemple, cu distribuția variată pentru a putea detecta orice defect al frunzelor și de a detecta sănătatea plantei.

3 Functia de activare

Algoritmul este bazat pe funcția de activare logaritm din sigmoid, conform cerințelor temei, de la numărul 19. Aceasta a fost implementată separat și folosită ca funcția "log_sig()".

$$g(z) = z \ln(1 + \sigma(z)) \quad (1)$$

unde:

- $\sigma(z)$ reprezintă funcția sigmoid $\sigma(x) = 1/(1 + e^{-x})$

4 Algoritmi de optimizare implementați

4.1 Gradient Descent (GD)

Algoritmul clasic de Gradient Descent implementat utilizează următoarea regulă de actualizare a parametrilor:

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t) \quad (2)$$

unde:

- x_t reprezintă parametrii modelului la iterația t
- α_t este rata de învățare
- $\nabla f(x_t)$ este gradientul funcției de cost la iterația t

Algoritmul de Gradient Descent este implementat cu următorii hiperparametri:

- Număr de iterații: 300
- Rata de învățare: 0.5

4.2 Metoda Alternativă de optimizare

Algoritmul Adam (Adaptive Moment Estimation) combină avantajele Momentum și RMSProp pentru actualizarea parametrilor, adaptând automat rata de învățare pentru fiecare parametru.

Regula de actualizare este:

$$x_{t+1} = x_t - \alpha \frac{y_t}{\sqrt{z_t} + \epsilon} \quad (3)$$

unde:

- y_t este estimarea corectată a primului moment (media gradientelor)
- z_t este estimarea corectată a celui de-al doilea moment (media pătratelor gradientelor)
- α este rata de învățare
- ϵ este o constantă pentru stabilitate numerică

Algoritmul Adam este implementat cu următorii hiperparametri:

- $\beta_1 = 0.9, \beta_2 = 0.999$ rata de scadere exponențială estimată la momentul k
- Rata de învățare: 0.05
- Număr de iterații: 300

4.3 Metoda Newton

Al treilea algoritm implementat este metoda Newton, care utilizează informații de ordinul doi (matricea Hessian) pentru a determina direcția și mărimea pasului de optimizare. Regula de actualizare pentru metoda Newton este:

$$x_{t+1} = x_t - \alpha_t (\nabla^2 F(x_t))^{-1} \nabla F(x_t) \quad (4)$$

unde:

- x_t reprezintă parametrii modelului la iterația t

- $\nabla^2 F(x_t)$ este matricea Hessian a funcției de cost la iterația t
- $\nabla F(x_t)$ este gradientul funcției de cost la iterația t

Algoritmul Newton este implementat cu următorii hiperparametri:

- Număr de iterații: 300

5 Rezultate

5.1 Performanță temporală

Timpul de antrenare pentru cele trei metode sunt:

- Gradient Descent: 47,45 secunde
- Metoda Alternativă: 43,91 secunde
- Metoda Alternativă: 689.93 secunde

Metoda alternativă este nu doar mai eficientă în termeni de convergență și acuratețe, dar și ușor mai rapidă din punct de vedere computațional când numărul de iterații este unul ridicat.

5.2 Convergența funcției de cost

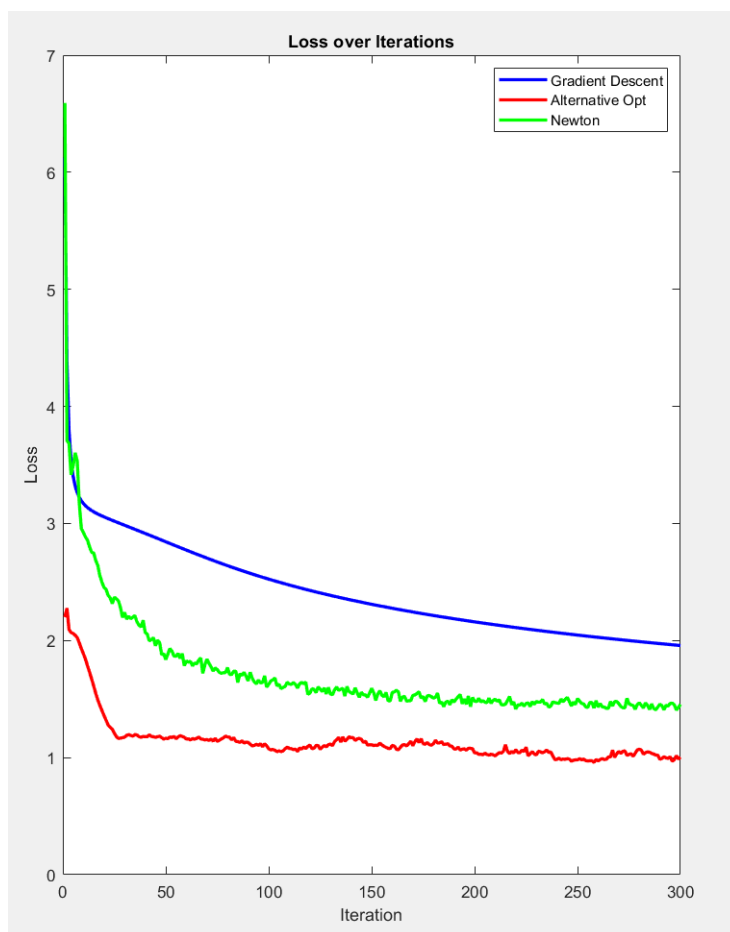


Figure 1: Evoluția funcției de cost pentru cele trei metode de optimizare

După cum se poate observa în Figura 1, metoda alternativă atinge o valoare a funcției de cost semnificativ mai mică comparativ cu Gradient Descent după 300 de iterații. De asemenea, convergența este mult mai rapidă pentru metoda alternativă, atingând un platou după aproximativ 50 de iterații.

5.3 Evoluția normei gradientului

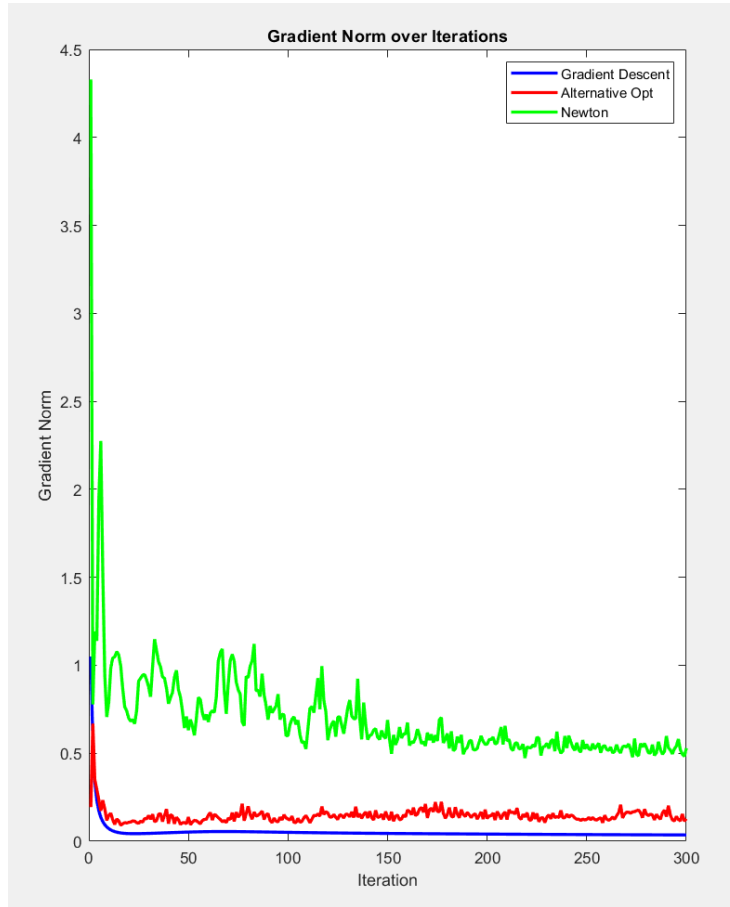


Figure 2: Evoluția normei gradientului pentru cele trei metode de optimizare

Figura 2 ilustrează comportamentul normei gradientului pentru cele trei metode. Gradient Descent prezintă o scădere monotonă a normei gradientului, stabilizându-se la valori foarte mici. Metoda alternativă prezintă o scădere rapidă inițială, urmată de oscilații ușoare, ceea ce sugerează că această metodă explorează mai dinamic spațiul parametrilor. În comparație cu acestea, Metoda Newton prezintă foarte multe oscilații de-a lungul procesului, având o valoare inițială mai mare chiar și decât metoda Gradient, însă are o scădere foarte rapidă către valorile metodei Adam.

5.4 Acuratețea și matricea de confuzie

Acuratețea obținută pentru cele două metode este:

- Gradient Descent: 66.10%
- Metoda Alternativă: 71.75%
- Metoda Newton: 69,87%

Matricele de confuzie pentru cele două metode sunt prezentate mai jos:

5.4.1 Matrice de confuzie pentru Gradient Descent

$$\begin{bmatrix} 180 & 1 & 9 & 21 & 2 & 0 & 2 & 2 & 23 \\ 71 & 25 & 6 & 3 & 26 & 6 & 17 & 39 & 27 \\ 2 & 0 & 183 & 3 & 4 & 4 & 2 & 22 & 0 \\ 17 & 7 & 36 & 74 & 29 & 10 & 16 & 15 & 16 \\ 2 & 0 & 8 & 7 & 158 & 16 & 14 & 3 & 12 \\ 0 & 0 & 0 & 1 & 2 & 215 & 2 & 0 & 0 \\ 8 & 2 & 7 & 2 & 0 & 14 & 157 & 17 & 13 \\ 10 & 1 & 18 & 1 & 3 & 9 & 37 & 140 & 1 \\ 19 & 1 & 0 & 1 & 0 & 0 & 9 & 0 & 190 \end{bmatrix}$$

Table 1: Matrice de confuzie pentru modelul antrenat cu Gradient Descent

5.4.2 Matrice de confuzie pentru Metoda Alternativă

$$\begin{bmatrix} 175 & 15 & 4 & 12 & 2 & 0 & 2 & 5 & 25 \\ 20 & 106 & 2 & 28 & 11 & 0 & 12 & 29 & 12 \\ 1 & 4 & 176 & 8 & 5 & 2 & 7 & 16 & 1 \\ 7 & 39 & 21 & 119 & 20 & 2 & 5 & 5 & 2 \\ 1 & 13 & 9 & 10 & 165 & 2 & 7 & 5 & 8 \\ 0 & 2 & 0 & 0 & 3 & 206 & 2 & 6 & 1 \\ 3 & 9 & 4 & 7 & 7 & 5 & 151 & 22 & 12 \\ 4 & 13 & 9 & 3 & 1 & 5 & 27 & 154 & 4 \\ 19 & 7 & 0 & 5 & 0 & 0 & 6 & 0 & 183 \end{bmatrix}$$

Table 2: Matrice de confuzie pentru modelul antrenat cu Metoda Alternativă

5.4.3 Matrice de confuzie pentru Metoda Newton

108	3	31	4	11	3	20	19	41
42	10	49	1	35	8	25	18	32
10	0	175	1	8	10	6	7	3
19	6	93	12	30	18	12	12	18
6	3	33	4	98	59	7	2	8
1	2	21	2	24	140	15	7	8
12	0	20	0	1	32	91	32	32
26	5	41	2	2	18	27	75	24
18	0	13	0	4	7	10	1	167

Table 3: Matrice de confuzie pentru modelul antrenat cu Metoda Newton

6 Comentarii și concluzii

6.1 Comparație generală

Rezultatele experimentale demonstrează că metoda alternativă de optimizare obține cea mai bună performanță pentru sarcina de clasificare analizată, cu o acuratețe de 71.75%. Gradient Descent, deși necesită un număr mai mare de iterații, ajunge la o acuratețe rezonabilă de 66.10%. Am testat și metoda Newton care a obținut o acuratețe de 69.87%, însă rezultatele acesteia ar trebui interpretate cu precauție din cauza unor posibile probleme de stabilitate numerică în implementarea curentă.

6.2 Analiza funcției de cost și a normei gradientului

Evoluția funcției de cost arată diferențe semnificative între cele două metode principale:

- Metoda Alternativă ajunge la o valoare foarte mică a funcției de cost (1.003652), indicând o adaptare excelentă la datele de antrenare.
- Gradient Descent prezintă o scădere constantă dar mai lentă a funcției de cost, ajungând la o valoare de 1.957334 după 300 de iterații.

În ceea ce privește norma gradientului, observăm că:

- Gradient Descent atinge o normă a gradientului foarte mică (0.035683), indicând o convergență precisă către un punct staționar.
- Metoda Alternativă menține o normă a gradientului moderat mai mare (0.127583), ceea ce sugerează o explorare mai dinamică a spațiului parametrilor, potențial benefică pentru evitarea minimelor locale.

6.3 Performanța pe clase

Analizând matricele de confuzie pentru cele două metode principale, observăm că:

- Metoda Alternativă obține performanțe mai echilibrate pe toate clasele, cu valori crescute pe diagonala principală.
- Gradient Descent performează bine pe anumite clase, dar are dificultăți cu clasele 2 și 4.

6.4 Concluzii finale

Rezultatele obținute conduc la următoarele concluzii:

1. **Superioritatea metodei alternative:** Metoda alternativă de optimizare (o metodă de tip Adam) demonstrează performanțe superioare atât în termeni de convergență, cât și de acuratețe finală. Aceasta ajunge la o funcție de cost semnificativ mai mică și obține o acuratețe cu aproximativ 5.65 puncte procentuale mai mare decât Gradient Descent.
2. **Eficiența Gradient Descent cu suficiente iterații:** Deși mai lent în convergență, Gradient Descent ajunge la performanțe rezonabile (66.10%) după un număr suficient de iterații, demonstrând robustețea acestei metode clasice.
3. **Compromisul între viteza de convergență și complexitatea implementării:** Metoda alternativă oferă o convergență mai rapidă la cost de o implementare potențial mai complexă, în timp ce Gradient Descent este mai simplu de implementat dar necesită mai multe iterații pentru rezultate competitive.

Deși metoda Newton a arătat potențial prin acuratețea obținută, instabilitatea sa numerică și complexitatea implementării ridică semne de întrebare privind aplicabilitatea sa practică pentru această sarcină de clasificare.

Aceste rezultate subliniază importanța alegerii metodei potrivite de optimizare pentru problema specifică abordată și demonstrează avantajele metodelor alternative de optimizare față de Gradient Descent clasic pentru sarcini de clasificare complexe.