

Klasyfikacja przy użyciu drzewa decyzyjnego

Daria Bartkowiak

10 grudnia 2025

1 Implementacja algorytmu

Zaimplementowałem klasyczny algorytm gradientu prostego dla funkcji $f : \mathbb{R}^n \rightarrow \mathbb{R}$ z stałą długością kroku α . Dla danego punktu startowego $x_0 \in \mathbb{R}^n$ generujemy ciąg

$$x_{k+1} = x_k - \alpha \nabla f(x_k),$$

a iteracje kończymy, gdy norma gradientu jest wystarczająco mała,

$$\|\nabla f(x_k)\| \leq \varepsilon,$$

lub po przekroczeniu maksymalnej liczby iteracji `max_iter`. Algorytm jest zaimplementowany w sposób ogólny: przyjmuje jako argumenty funkcję `f`, jej gradient `grad_f` oraz wektor startowy `x0` i zwraca przybliżone minimum, punkt dla, którego to minimum jest osiągane oraz wartości funkcji w kolejnych iteracjach.

2 Zastosowanie

Algorytm zastosowałem do znalezienia minimum funkcji Ackleya w wersji jedno- i dwuwymiarowej. Ustawiłem następujące parametry:

$$\text{max_iter} = 10000, \quad \alpha = 0,01, \quad x_0 = [1] \text{ lub } x_0 = [1, -1], \quad \text{tol} = 10^{-6}.$$

A o to otrzymane wyniki:

Dla funkcji 1D:

$$\text{iteracje} = 5, \quad x^* \approx 0,9685, \quad f(x^*) \approx 3,5745,$$

Dla funkcji 2D:

$$\text{iteracje} = 23, \quad x^* \approx 0,9685, \quad f(x^*) \approx 3,5745.$$

Podejrzańskie może wydawać się wskazane minimum funkcji, ponieważ z definicji wiemy, że funkcja Ackleya osiąga minimum globalne w punkcie $x = 0$ (z wartością $f(0) = 0$). Otrzymany punkt $x^* \approx 0,9685$ z wartością $f(x^*) \approx 3,5745$ jest jednak minimum lokalnym. Wynika to z faktu, że funkcja Ackleya jest silnie niewypukła i posiada wiele minimów lokalnych, a metoda gradientu prostego przy zadanym punkcie startowym oraz stałej długości kroku może utknąć w takim minimum (w "pułapce"), zamiast dojść do właściwego minimum globalnego.

3 Eksperymenty

3.1 Ustawienia eksperymentu

W przypadku funkcji jednowymiarowej przetestowałem następujące wartości kroku:

$$\alpha \in \{0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1, 1.5, 2\}.$$

Dla każdej z nich uruchamiano algorytm z dwoma różnymi punktami startowymi:

$$x_0 \in \{0.01, 1\}.$$

Dla przypadku dwuwymiarowego użyłam tych samych wartości kroku α i dwóch punktów startowych

$$x_0 \in \{[0.01, -0.01], [1, -1]\}.$$

3.2 Wyniki

Tabela 1: Ackley 1D, punkt startowy $x_0 = 1$.

α	iteracje	x^*	$f(x^*)$
0.001	145	0.9685	3.5745
0.01	5	0.9685	3.5745
0.05	10000	0.0177	0.0872
0.1	10000	0.0942	0.7988
0.2	10000	-0.8070	4.2793
0.3	10000	5.3765	15.4045
0.5	10000	-390.3249	22.0829
0.8	10000	319.0637	20.2067
1.0	10000	272.6765	22.0779
1.5	10000	409.1202	20.6469
2.0	10000	-1551.2759	21.8680

Tabela 2: Ackley 2D, punkt startowy $x_0 = (1, -1)$.

α	iteracje	x^*	$f(x^*)$
0.001	289	[0.9685, -0.9685]	3.5745
0.01	23	[0.9685, -0.9685]	3.5745
0.05	10000	[1.0208, -1.0208]	3.7165
0.1	10000	[-0.3894, 0.3894]	3.8023
0.2	10000	[-0.7496, 0.7496]	4.5052
0.3	10000	[-114.3171, 114.3171]	22.0540
0.5	10000	[-29.8514, 29.8514]	20.8540
0.8	10000	[26.0336, -26.0336]	19.9500
1.0	10000	[136.7692, -136.7692]	21.5096
1.5	10000	[-96.3519, 96.3519]	22.1679
2.0	10000	[1022.7170, -1022.7170]	21.9041

Tabela 3: Ackley 1D, punkt startowy $x_0 = 0.01$.

α	iteracje	x^*	$f(x^*)$
0.001	10000	0.0021	0.0087
0.01	10000	-0.0408	0.2498
0.05	10000	-0.3364	3.4227
0.1	10000	-1.1537	5.0728
0.2	10000	-0.9830	3.5851
0.3	10000	203.0658	20.2197
0.5	10000	-651.5819	22.2995
0.8	10000	-771.3657	22.2038
1.0	10000	-892.7579	21.6672
1.5	10000	717.7237	21.8701
2.0	10000	-1336.4212	22.3035

Tabela 4: Ackley 2D, punkt startowy $x_0 = (0.01, -0.01)$.

α	iteracje	x_1^*	x_2^*	$f(x^*)$
0.001	10000	0.0010	-0.0010	0.0042
0.01	10000	0.0136	-0.0136	0.0643
0.05	10000	0.2044	-0.2044	2.1927
0.1	10000	-0.4286	0.4286	3.9550
0.2	10000	-1.2637	1.2637	6.2672
0.3	10000	120.2817	-120.2817	21.8977
0.5	10000	-28.8525	28.8525	20.8328
0.8	10000	65.6963	-65.6963	22.0001
1.0	10000	-69.2642	69.2642	21.8036
1.5	10000	123.3812	-123.3812	22.2384
2.0	10000	34.9954	-34.9954	19.9829

3.3 Wykres zbieżności

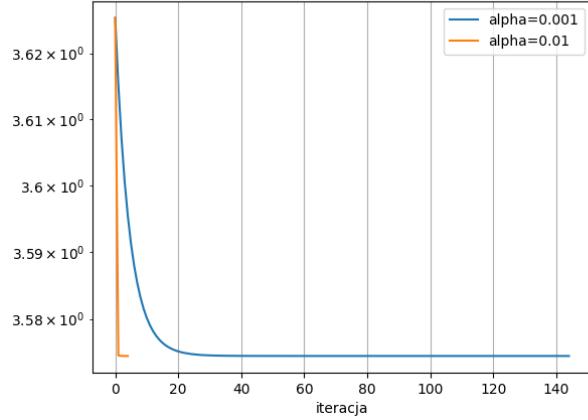
Dodatkowo dla wybranego punktu startowego $x_0 = 1$ w wymiarze 1D postanowiłem zilustrować zbieżność na wykresach w skali logarytmicznej dla:

$$\alpha \in \{0.001, 0.01\}$$

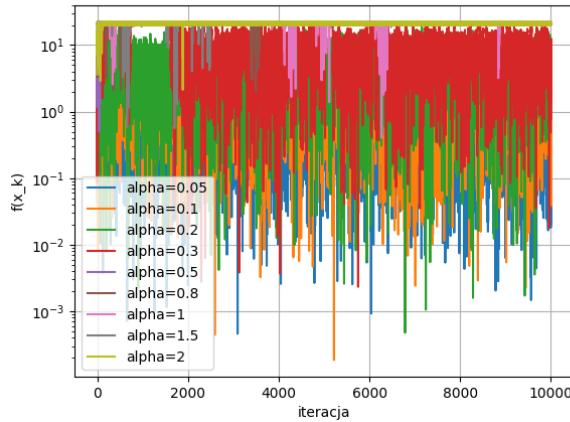
(„małe” kroki) oraz

$$\alpha \in \{0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1, 1.5, 2\}$$

(„większe” kroki).



Rysunek 1: Zbieżność metody gradientu prostego dla funkcji Ackleya 1D z małymi krokami $\alpha = 0.001$ i $\alpha = 0.01$.



Rysunek 2: Przebieg wartości $f(x_k)$ dla większych kroków $\alpha \in \{0.05, \dots, 2\}$ dla funkcji Ackleya 1D.

3.4 Wnioski

- Dla małych kroków, takich jak $\alpha = 0.001$ oraz $\alpha = 0.01$, metoda gradientu prostego zachowuje się stabilnie: wartości $f(x_k)$ na wykresie szybko maleją, a następnie ustalają się na stałym poziomie.
- Dla punktu startowego $x_0 = 0,01$ widać, że metoda nie wpada w to samo minimum lokalne co dla $x_0 = 1$ (wartości funkcji są znacznie bliższe minimum globalnemu), ale jednocześnie nie zbiega w rozsądny czasie.
- Dla niektórych konfiguracji (np. $x_0 = 1$, $\alpha = 0.05$) udaje się zejść bardzo blisko minimum globalnego, ale warunek stopu związany z normą gradientu nie jest spełniony przed osiągnięciem `max_iter`.
- Wraz ze wzrostem długości kroku (np. $\alpha \geq 0.2$) przebiegi na wykresach stają się coraz bardziej chaotyczne: wartości $f(x_k)$ oscylują w szerokim przedziale, a na wykresie w skali logarytmicznej widać liczne skoki. Tabelki pokazują, że końcowe punkty x^* mają bardzo dużą wartość bezwzględną, a wartości funkcji są bliskie poziomom $f(x^*) \approx 20-22$.
- Dla dużych kroków (np. $\alpha = 0.5, 0.8, 1, 1.5, 2$) metoda w praktyce nie zbiega: iteracje kończą się dopiero po osiągnięciu `max_iter`, a trajektorie w tabelach wskazują na „rozbieganie się” w odległe rejony przestrzeni.

- Wyniki dla funkcji dwuwymiarowej odzwierciedlają te same zjawiska: dla małych α otrzymujemy przyzwoite wartości funkcji (blisko minimum globalnego lub lokalnego), natomiast dla większych kroków kolejne iteraty x_k mają bardzo duże współrzędne, a wartości $f(x_k)$ utrzymują się na wysokim poziomie.
- Silna zależność wyników od punktu startowego (porównanie $x_0 = 0.01, 1$) potwierdza, że dla funkcji o wielu minimach lokalnych, takich jak Ackley, metoda gradientu prostego ze stałą długością kroku nie gwarantuje znalezienia minimum globalnego i wymaga starannego doboru zarówno x_0 , jak i parametru α .