Dokumentacja – laboratorium nr 1
Wprowadzenie do Sztucznej Inteligencji

Dominika Wyszyńska 318409

29 października 2023

1 Opis treści zadania

Celem zadania było zaimplementowanie algorytmu gradientu prostego dla podanej w zadaniu funkcji celu, gdzie parametr α przyjmował wartości 1, 10 lub 100.

$$q(x) = \sum_{i=1}^{n} \alpha^{\frac{i-1}{n-1}} * x_i^2$$

$$x \in [-100, 100]^n, \quad x \in R^n$$

Funkcja powinna być w stanie minimalizować wartość funkcji celu poprzez aktualizację parametrów w każdej iteracji na podstawie obliczonego gradientu. Algorytm powinien mieć możliwość obsługi różnych parametrów oraz warunków stopu, takich jak maksymalna liczba iteracji lub epsilon (minimalna akceptowalna różnica pomiędzy kolejnymi krokami algorytmu). Implementacja solwera powinna być w stanie zoptymalizować każdą zadaną funkcję dla różnych wymiarów przestrzeni R^n.

Należało także zbadać wpływ wartości parametru kroku na zbieżność metody oraz czas jej działania. W związku z tym należało zamieścić wykresy zależności wartości funkcji w punkcie w danym kroku od numeru iteracji/kroku.

Implementacja programu została umieszczona na gitlabie wydziałowym.

2 Opis planowanych eksperymentów numerycznych

Na potrzeby eksperytmentów miało zostać przyjęte n=10, czy przestrzeń 10-wymiarowa oraz także stały punkt startowy – przykładowo wektor $x_0 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$.

Solwer o nazwie *gradient_descent* działa jako mechanizm rozwiązujący, który przyjmuje trzy główne wejścia: funkcję celu, punkt początkowy oraz obiekt *params_t*. Obiekt ten przyjmuje wartości parametrów tj. maksymalna liczba iteracji, epsilon, długość kroku oraz zmienne aktywujące warunki stopu. W związku z tym, eksperymenty zostały przeprowadzone, dla różnych wartości atrybutów obiektu *params_t* oraz parametru alpha.

Eksperymenty zostały zaprojektowane w następujący sposób:

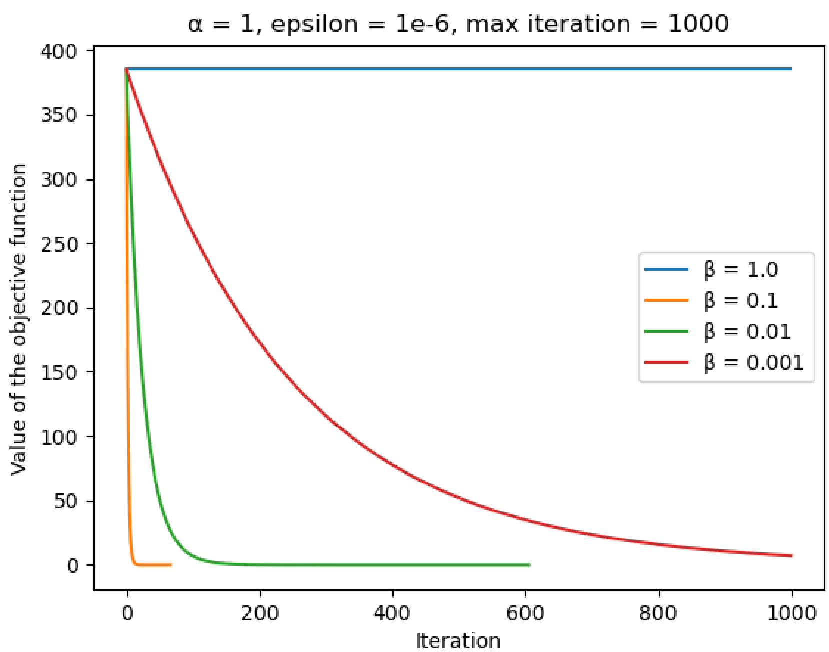
- Badanie wpływu parametru alpha dla trzech różnych wartości (0, 10 bądź 100).
- Analiza wpływu długości kroku (beta).
- Badanie różnych ilości iteracji: Eksperymenty zostaną przeprowadzone dla różnych maksymalnych liczb iteracji, gdy warunek stopu związany z liczbą iteracji jest aktywny.
- Analiza różnych wartości epsilon: Badanie zostanie przeprowadzone dla różnych wartości epsilon w celu zrozumienia wpływu tego parametru na dokładność i zbieżność algorytmu.

3 Opis uzyskanych wyników

3.1 Analiza wpływu długości kroku (beta)

Eksperymenty zostały przeprowadzone dla stałej wartości liczby maksymalnych iteracji równej 1000, epsilon równe $1e-6$ oraz przy wszystkich aktywnych warunkach stopu.

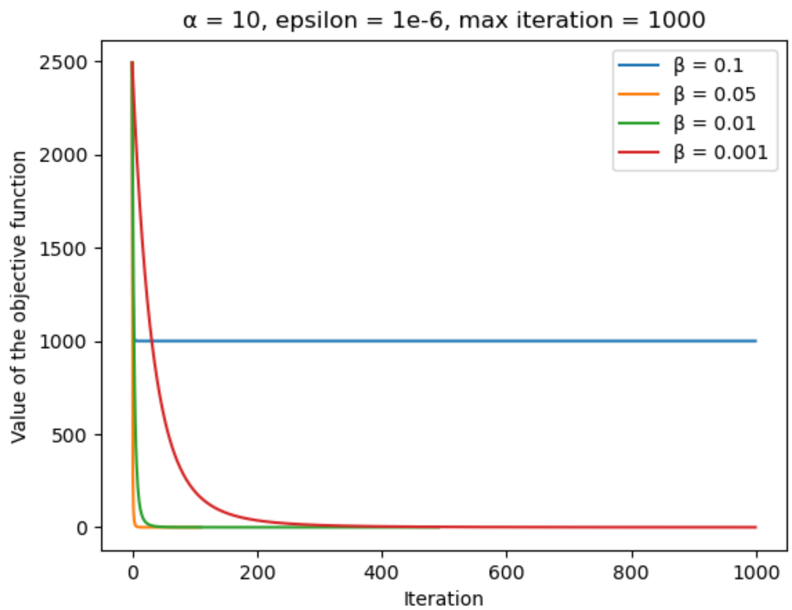
(a) Gdy $\alpha = 1$



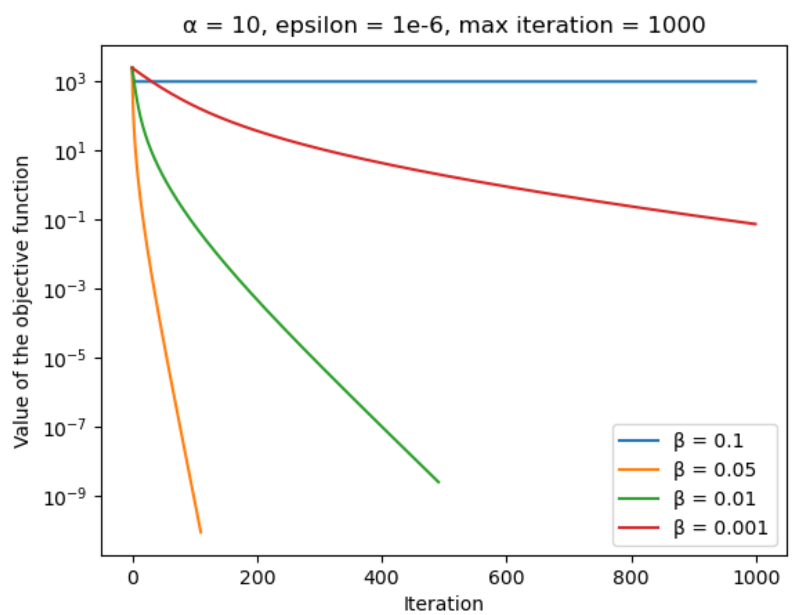
Rysunek 1: Wykres zależności wartości funkcji celu od iteracji dla różnych wartości beta.

Jak widać na wykresie powyżej, gdy weźmiemy zbyt dużą wartość skoku, czyli dla $\alpha = 1$, $\beta = 1$ wartość funkcji nie zbiega do najmniejszej wartości, czyli do 0. Natomiast gdy weźmiemy wartość kroku zbyt małą, to dłużej zajmuje znalezienie optimum. Ograniczeniem także jest maksymalna liczba iteracji. W przypadku $\alpha = 0.1$ najlepszym rozwiązaniem wydaje się $\beta = 0.1$.

(b) Gdy $\alpha = 10$



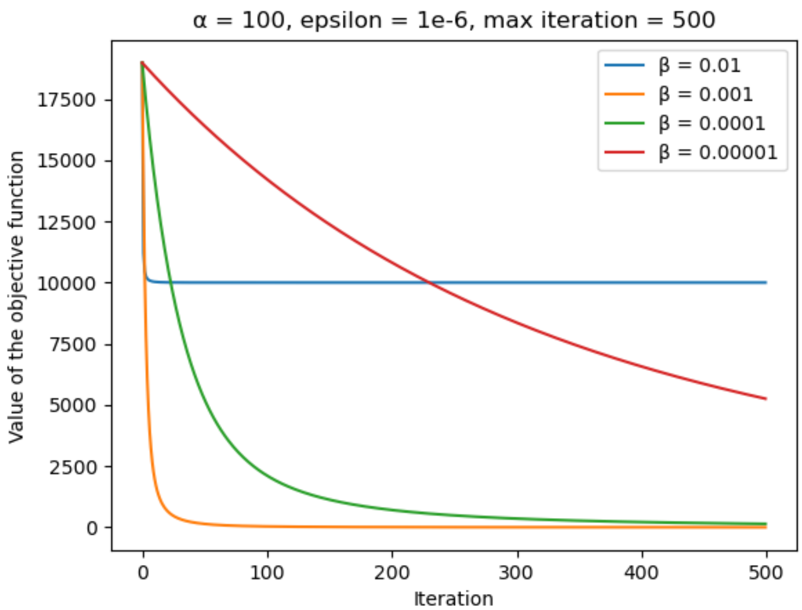
Rysunek 2: Wykres zależności wartości funkcji celu od iteracji dla różnych wartości beta.



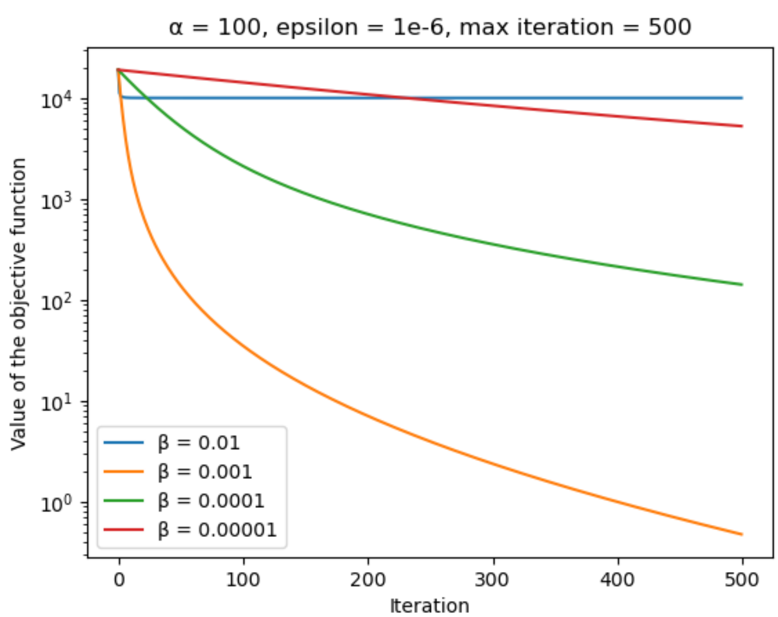
Rysunek 3: Logarytmiczny wykres zależności wart. funkcji celu od iteracji dla różnych wartości beta.

Dla $\alpha = 10$ zostały przedstawione dwa wykresy: liniowy oraz logarytmiczny, gdyż logarytmiczny lepiej prezentuje informacje o wartości funkcji celu w punkcie dla różnych β. Jak widać na wykresie, duże znaczenia ma dobór odpowiedniej wartości β, gdyż wpływa ono na szybkość zbieżności oraz czy w ogóle znajdzie optimum. Dla $\alpha = 10$ najlepszą wartością wydaje się $\beta = 0.05$.

(c) Gdy $\alpha = 100$



Rysunek 4: Wykres zależności wartości funkcji celu od iteracji dla różnych wartości beta.



Rysunek 5: Logarytmiczny wykres zależności wartości funkcji celu od iteracji dla różnych wartości beta.

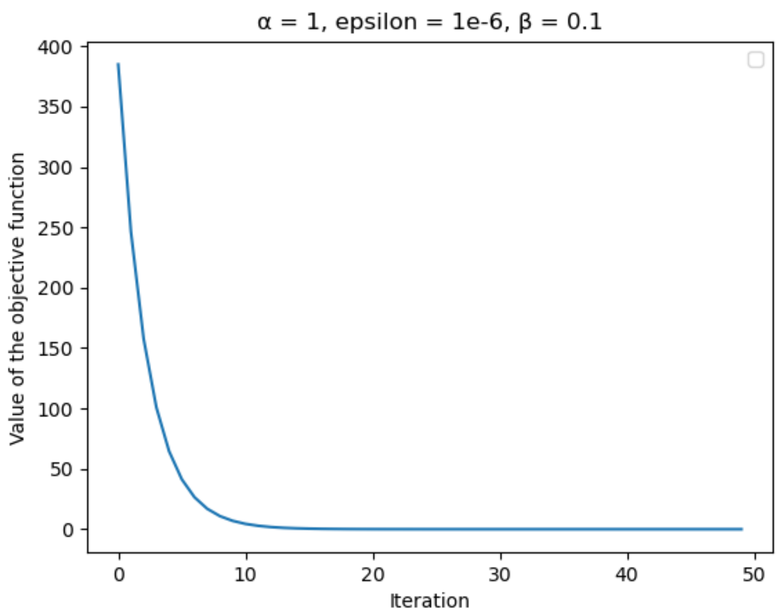
Dla $\alpha = 100$ także zostały przedstawione dwa wykresy: liniowy oraz logarytmiczny. Identycznie jak dla poprzednich wartości α, gdy dobrana została $\beta = 0.01$ lub wyższe wartości, wykres nie zbiega do optimum,a zatrzymuje się na wartości funkcji rzędu 4. Natomiast dla zbyt małej $\beta = 0.00001$ pięćset iteracji nie wystarcza, aby znalezione zostało optimum. W związku z tym, optymalne wydaje się $\beta = 0.001$.

3.2 Badanie różnych wartości maksymalnych iteracji

Eksperymenty zostały przeprowadzone dla stałej wartości epsilon równe $1e-6$, bety stałej dla każdego α oraz przy wszystkich aktywnych warunkach stopu. W każdym podpunkcie został przedstawiony tylko jeden przykładowy wykres, gdyż wykresy dla różnych wart. maksymalnych iteracji nakładałyby się na siebie.

(a) Gdy $\alpha = 1$ (przykładowo β=0.1)

Liczba iteracji	Wartość funkcji celu w punkcie końcowym (x_k)
5	41.339060
10	4.438748
25	0.005495
50	$7.842589e-8$
66	$3.976665e-11$

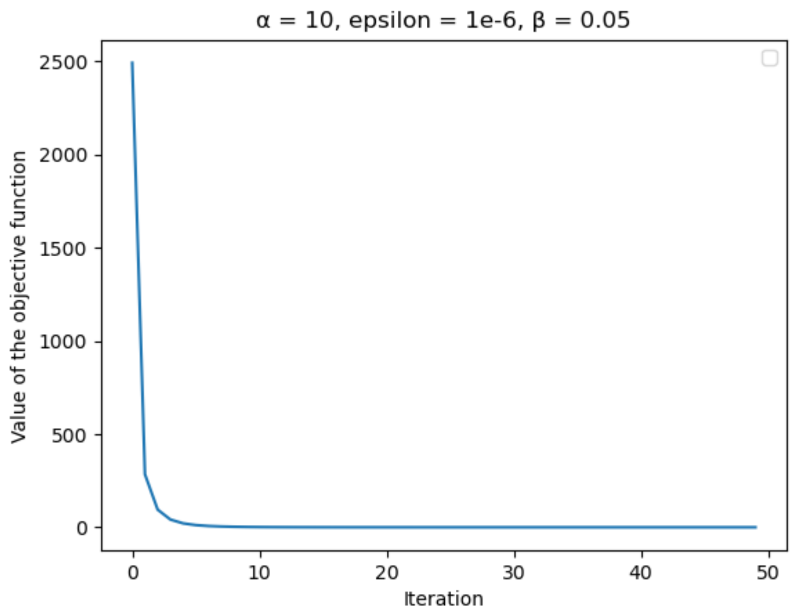


Rysunek 6: Wykres zależności wartości funkcji celu od iteracji dla maks. liczby iteracji równej 50.

Dla $\alpha = 1$ funkcja znajduje optimum po 66 iteracjach przy ustalonych parametrach. Funkcja dość szybko zbiega do niskich wartości w ciągu zaledwie 10 iteracji (znaczenie ma ustalenie odpowiedniego β). Na podstawie tabeli można stwierdzić, iż ograniczenie liczby iteracji sprawia, że funkcja może nie zdążyć znaleźć optimum. Im większa liczba iteracji, tym większa szansa, że znajdzie optimum.

(b) Gdy $\alpha = 10$ (przykładowo β=0.05)

Liczba iteracji	Wartość funkcji celu w punkcie końcowym (x_k)
5	5.697260
10	0.773981
50	$3.061460e - 5$
100	$7.094017e - 10$
110	$6.966214e - 11$

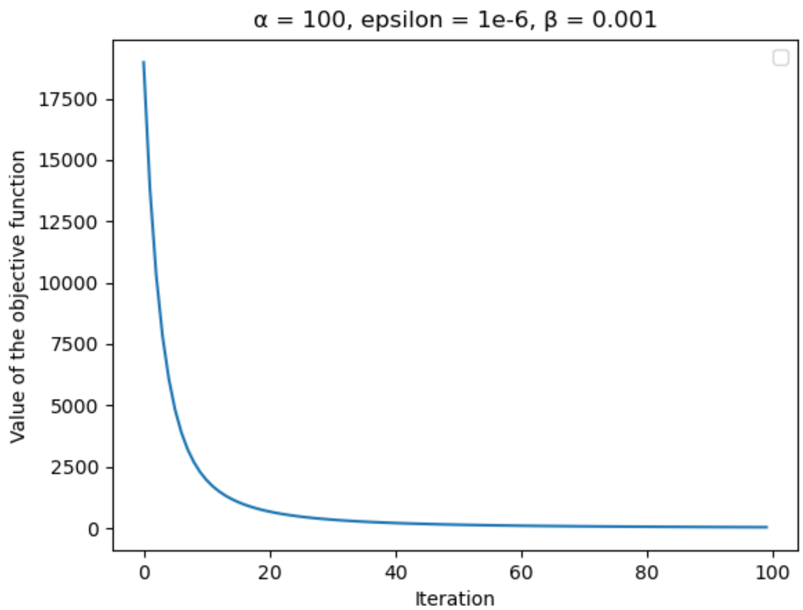


Rysunek 7: Wykres zależności wartości funkcji celu od iteracji dla maks. liczby iteracji równej 50.

Dla $\alpha = 10$ funkcja znajduje optimum po 110 iteracjach przy ustalonych parametrach. Osiąga wtedy wartość równą około $6.966214e-11$. Funkcja także dość szybko zbiega do niskich wartości.

(c) Gdy $\alpha = 100$ (przykładowo β=0.001)

Liczba iteracji	Wartość funkcji celu w punkcie końcowym (x_k)
10	107.810418
50	23.643940
100	9.447524
500	0.311920
1000	0.023376
3797	$2.486896e - 7$



Rysunek 8: Wykres zależności wartości funkcji celu od iteracji dla maks. liczby iteracji równej 100.

Dla $\alpha = 100$ funkcja znajduje optimum po aż 3797 iteracjach przy ustalonych parametrach. Osiąga wtedy wartość równą około $2.486896e-7$. Dość długo zajmuje funkcji znalezienie optimum. Gdyby ograniczyć liczbę iteracji do 100, wartość funkcji w punkcie końcowym będzie wynosić około 9.447524, co sprawia, że nie jest dość blisko optimum.

3.3 Analiza różnych wartości epsilon

Epsilon odgrywa istotną rolę w algorytmie gradientu prostego, wpływając bezpośrednio na zbieżność i stabilność procesu optymalizacji. Wartość epsilon określa minimalną akceptowalną różnicę pomiędzy kolejnymi krokami algorytmu. Gdy epsilon jest zbyt duże, istnieje ryzyko, że algorytm może "przeskoczyć"minimum lokalne, co może spowodować niestabilność i brak zbieżności. Z drugiej strony, gdy epsilon jest zbyt małe, może to spowodować wolniejszą zbieżność, co prowadzi do dłuższego czasu obliczeń. Optymalna wartość epsilon zależy od specyfiki danej funkcji celu.

4 Podsumowanie i wnioski

Celem tych eksperymentów było zrozumienie wpływu różnych parametrów na zachowanie się algorytmu gradientu prostego. Przeanalizowane wyniki pozwolą na lepsze zrozumienie zachowania się algorytmu oraz umożliwią optymalizację jego parametrów dla różnych zastosowań.

Na podstawie eksperymentów opisanych w sprawozdaniu, możliwe jest wysnucie odpowiednich wniosków. Długość kroku (β) wpływa na zbieżność algorytmu, więc należy z rozwagą dobierać dany parametr. Jeżeli weźmiemy zbyt dużą wartość, funkcja może nie znaleźć optimum i oscylować przy wartościach dużo większych. Natomiast dla zbyt małej długości kroku, znalezienie optimum będzie trwało o wiele dłużej. Zmniejszając długość kroku, zwiększamy dokładność przybliżenia, co zazwyczaj wiąże się z koniecznością zwiększenia maksymalnej liczby iteracji. Przy ograniczaniu funkcji liczbą iteracji może wystąpić sytuacja, gdy algorytm nie zdąży znaleźć optimum. Dlatego im dłużej algorytm się wykonuje, tym większa szansa na znalezienie optimum. Jednakże zwiększając liczbę maksymalnych kroków, zwiększamy czas wykonywania programu. Natomiast epsilon wpływa bezpośrednio na zbieżność i stabilność procesu optymalizacji.

Podsumowując, dzięki tym parametrom, algorytm jest w stanie dostosować się do specyfiki konkretnej funkcji celu, umożliwiając kontrolę nad dokładnością rozwiązania, szybkością zbieżności oraz warunkami stopu. Możliwość aktywacji lub dezaktywacji warunku stopu daje elastyczność w zależności od konkretnych potrzeb i charakterystyki funkcji celu.