Wydział Elektroniki
i Technik Informacyjnych

POLITECHNIKA WARSZAWSKA

Wstęp do Sztucznej Inteligencji

Ćwiczenie nr 1:
Zagadnienie przeszukiwania przestrzeni i podstawowe podejście do niego

Kaczmarski Robert
293377

Warszawa 2021

Polecenie

Zadanie polega na minimalizacji funkcji celu i porównanie wyników dla metody najszybszego spadku gradientu i metody Newtona. Funkcją celu będzie funkcja Himmelblau postaci:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

$$-5 \leq x \leq 5$$

$$-5 \leq y \leq 5$$

Należy zaobserwować różnice w działaniu metody spadku gradientu i metody Newtona dla różnych punktów początkowych i różnej wartości parametru Beta, oznaczającego szybkość uczenia się.

Użyte narzędzia

Do wykonania zadania użyty został Python w wersji 3.8 oraz następujące biblioteki:

- matplotlib
- numpy
- time

Wyniki

Zgodnie z informacjami zaczerpniętymi z Wikipedii [1] na temat funkcji Himmelblau możliwe do znalezienia minima lokalne to:

- f(3,0, 2,0) = 0,0,
- f(-2,805118, 3,131312) = 0,0,
- f(-3,779310, -3,283186) = 0,0,
- f(3,584428, -1,848126) = 0,0.

Wartości te posłużą do zweryfikowania poprawności otrzymanych wyników. Eksperymenty zostały przeprowadzone dla czterech różnych punktów początkowych (x, y):

- (4, 4),
- (-4, -4),
- (0, 0),
- (-4, 0).

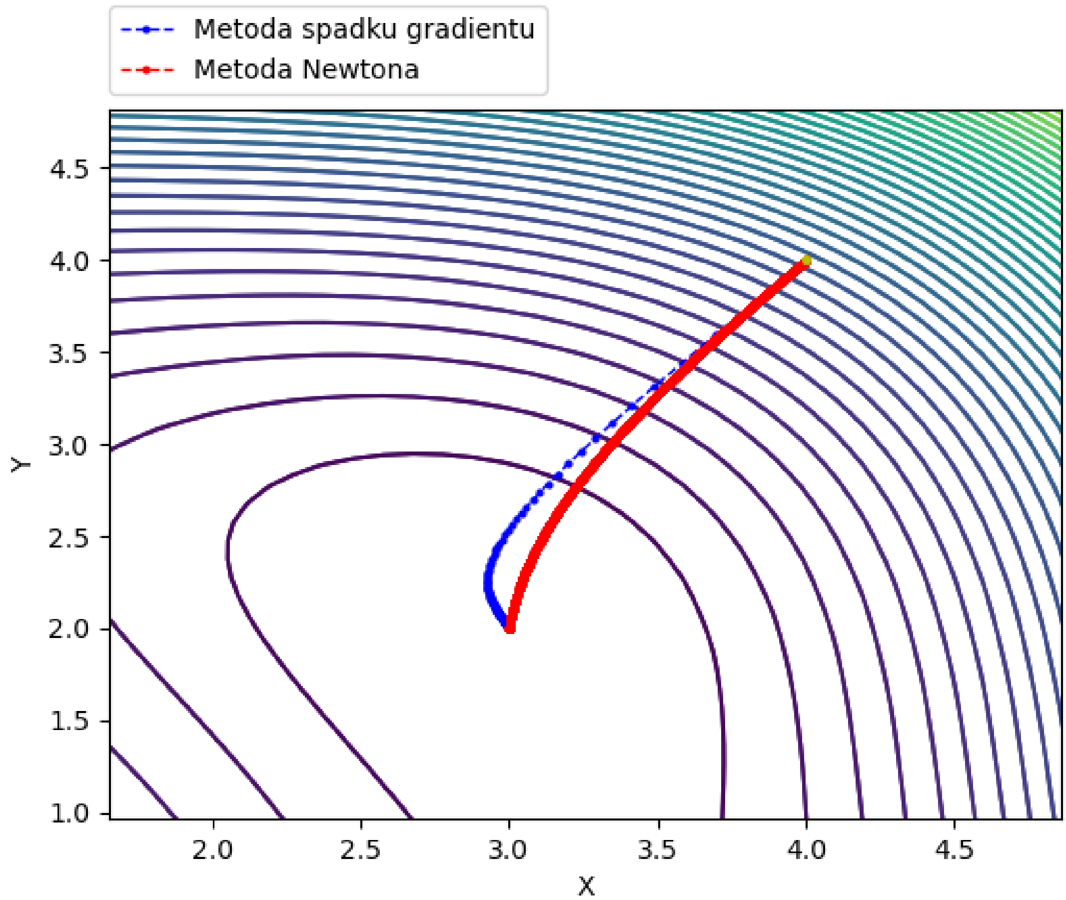
Warunkiem stopu algorytmów było przekroczenie maksymalnej liczby iteracji oraz mały epsilon

$\varepsilon = 10^{-12}$. Otrzymane wyniki zostały przedstawione w Tabelach 1-4, przy czym ‘nan’ oznacza brak znalezionego rozwiązania.

		Metoda spadku gradientu			Metoda Newtona		
Maks. liczba iteracji	β	Znaleziony punkt (x, y)	Liczba iteracji	Czas wykonania	Znaleziony punkt	Liczba iteracji	Czas wykonania
100000	1	(nan, nan)	100000	6,94 s	(3, 2)	8	0,02 s
100000	0,1	(nan, nan)	100000	6,91 s	(3, 2)	238	0,03 s
100000	0,01	(3, 2)	75	0,02 s	(3, 2)	2247	0,31 s
100000	0,001	(3, 2)	809	0,05 s	(3, 2)	20253	2,92 s
100000	0,0001	(3, 2)	7285	0,39 s	(3, 2)	100000	13,33 s
10000	0,001	(3, 2)	809	0,04 s	(3, 2)	10000	1,42 s
1000	0,001	(3, 2)	809	0,03 s	(3,4, 3,11)	1000	0,14 s
100	0,001	(2,98, 2,04)	100	0,01 s	(3,92, 3,89)	100	0,02 s
10	0,001	(3,16, 2,84)	10	0,00 s	(3,99, 3,98)	10	0,02 s
1	0,001	(3,83, 3,77)	1	0,00 s	(3,99, 3,99)	1	0,00 s

Tabela 1. Porównanie metod spadku gradientu i Newtona dla punktu startowego (4, 4).

W przypadku wyboru punktu startowego (4, 4) obie metody znalazły minimum lokalne f(3,0, 2,0), z tą różnicą, że potrzeba było dobrać odpowiednie wartości parametrów β oraz maksymalną liczbę iteracji ($iters_{maks}$). Dla metody spadku gradientu było to β ≥ 0.01 i $iters_{maks} > 100$, a dla metody Newtona β ≥ 1 i $iters_{maks} \geq 10000$. Przy zbyt dużym parametrze β metoda spadku gradientu nie znajduje rozwiązania, a metoda Newtona radzi sobie znakomicie. Parametr β wpływa na długość kroku, dlatego przy zbyt dużej wartości może dojść do przeskoczenia potencjalnego minimum, co skutkować może zapętleniem się algorytmu, próbującego powrócić na dobrą drogę. Pomijając eksperymenty, dla których metoda spadku gradientu nie dostarczyła rezultatów, czas wykonania metody Newtona jest dłuższy niż metody spadku gradientu. Jest to najpewniej spowodowane tym, że metoda Newtona wymaga bardziej skomplikowanych obliczeń, takich jak obliczenie hesjanu i jego odwrotności. Przykładowo na Rys. 1. przedstawiona została droga do minimum wyznaczana przez punkty (x, y) w kolejnych iteracjach.



Rys. 1. Wyznaczone punkty w kolejnych iteracjach obu metod dla β = 0.001, iter_maks = 100000.

Jak widać na Rys. 1. metoda Newtona zbiega niemalże bezpośrednio do minimum lokalnego, ponieważ obliczenie hesjanu informuje o kierunku prowadzącym do minimum, a nie tylko o monotoniczności funkcji jak w przypadku gradientu.

Wnioski płynące z otrzymanych wyników zbadanego punktu startowego (-4, -4) w Tabeli 2. pokrywają się wnioskami dla punktu (4, 4). Obie metody znalazły minimum lokalne f(-3,779310, -3,283186).

		Metoda spadku gradientu			Metoda Newtona		
Maks. liczba iteracji	β	Znaleziony punkt (x, y)	Liczba iteracji	Czas wykonania	Znaleziony punkt	Liczba iteracji	Czas wykonania
100000	1	(nan, nan)	100000	8,09 s	(-3,78, -3,28)	6	0,00 s
100000	0,1	(nan, nan)	100000	7,25 s	(-3,78, -3,28)	220	0,05 s
100000	0,01	(-3,78, -3,28)	24	0,02 s	(-3,78, -3,28)	2061	0,28 s
100000	0,001	(-3,78, -3,28)	300	0,02 s	(-3,78, -3,28)	18389	2,47 s
100000	0,0001	(-3,78, -3,28)	2767	0,13 s	(-3,78, -3,28)	100000	13,42 s
10000	0,001	(-3,78, -3,28)	2767	0,14 s	(-3,87, -3,6)	10000	1,34 s
1000	0,001	(-3,78, -3,28)	1000	0,06 s	(-3,98, -3,95)	1000	0,13 s
100	0,001	(-3,91, -3,6)	100	0,02 s	(-3,99, -3,99)	100	0,02 s
10	0,001	(-3,99, -3,93)	10	0,00 s	(-3,99, -3,99)	10	0,02 s
1	0,001	(-3,99, -3,99)	1	0,00 s	(-3,99, -3,99)	1	0,00 s

Tabela 2. Porównanie metod spadku gradientu i Newtona dla punktu startowego (-4, -4).

Ciekawe zjawisko można zaobserwować patrząc na wyniki zamieszczone w Tabeli 3 i 4. Otrzymane minimum lokalne dla metody Newtona rożni się od tej znalezionej poprzez metodę spadku gradientu. Dla punktu startowego metoda spadku gradientu odnalazła minimum w (3, 2), a metoda Newtona zatrzymała się w punkcie (-0,27, -0,92), który nie należy do wymienionych wcześniej minimów lokalnych.

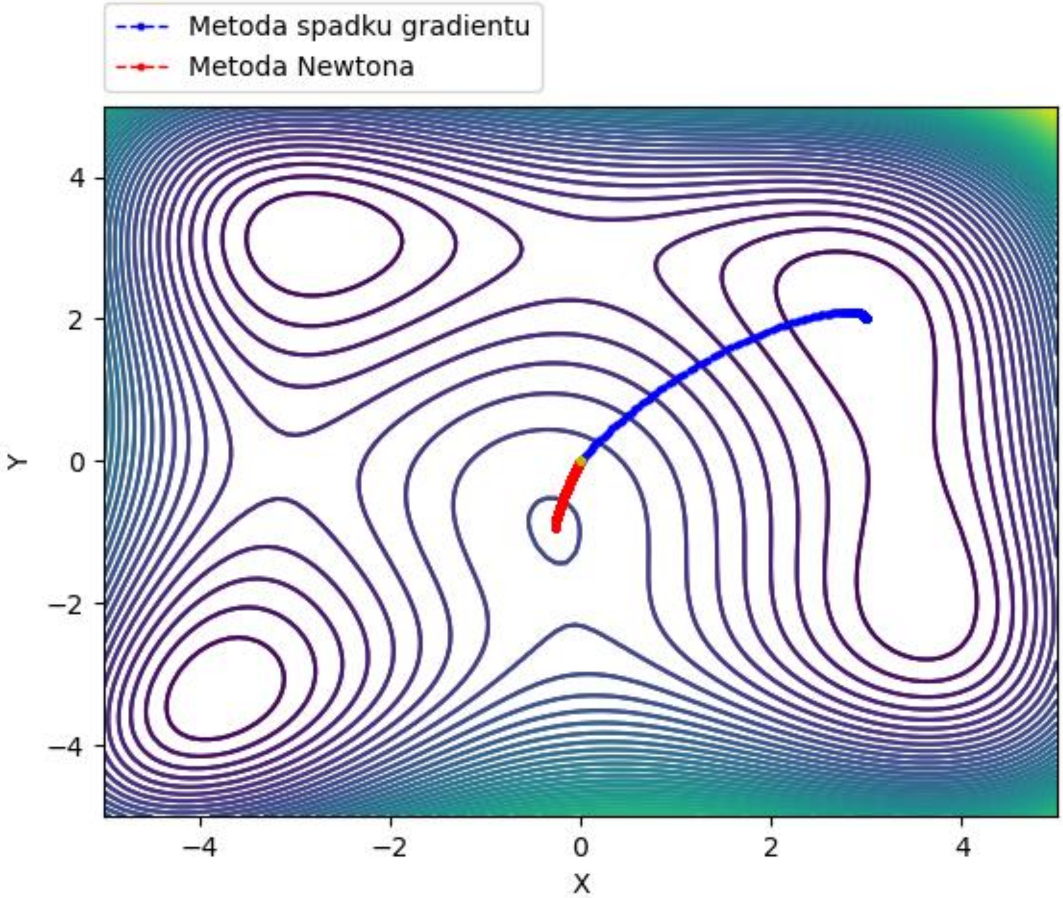
		Metoda spadku gradientu			Metoda Newtona		
Maks. liczba iteracji	β	Znaleziony punkt (x, y)	Liczba iteracji	Czas wykonania	Znaleziony punkt	Liczba iteracji	Czas wykonania
100000	1	(nan, nan)	100000	7,47 s	(-0,27, -0,92)	6	0,00 s
100000	0,1	(nan, nan)	100000	6,78 s	(-0,27, -0,92)	222	0,03 s
100000	0,01	(3, 2)	86	0,02 s	(-0,27, -0,92)	2086	0,28 s
100000	0,001	(3, 2)	847	0,03 s	(-0,27, -0,92)	18640	2,72 s
100000	0,0001	(3, 2)	7675	0,47 s	(-0,27, -0,92)	100000	13,58 s
10000	0,001	(3, 2)	847	0,05 s	(-0,27, -0,92)	10000	1,42 s
1000	0,001	(3, 2)	847	0,05 s	(-0,19, -0,54)	1000	0,16 s
100	0,001	(2,91, 2,09)	100	0,02 s	(-0,03, -0,08)	100	0,03 s
10	0,001	(0,17, 0,25)	10	0,00 s	(-0,00, -0,01)	10	0,00 s
1	0,001	(0,01, 0,02)	1	0,00 s	(-0,00, -0,00)	1	0,00 s

Tabela 3. Porównanie metod spadku gradientu i Newtona dla punktu startowego (0, 0).

		Metoda spadku gradientu			Metoda Newtona		
Maks. liczba iteracji	β	Znaleziony punkt (x, y)	Liczba iteracji	Czas wykonania	Znaleziony punkt	Liczba iteracji	Czas wykonania
100000	1	(nan, nan)	100000	7,45 s	(-3,07, -0,08)	6	0,02 s
100000	0,1	(nan, nan)	100000	7,25 s	(-3,07, -0,08)	224	0,03 s
100000	0,01	(-3,78, -3,28)	42	0,00 s	(-3,07, -0,08)	2097	0,30 s
100000	0,001	(-3,78, -3,28)	456	0,02 s	(-3,07, -0,08)	18750	2,48 s
100000	0,0001	(-3,78, -3,28)	4297	0,20 s	(-3,07, -0,08)	100000	13,42 s
10000	0,001	(-3,78, -3,28)	456	0,02 s	(-3,07, -0,08)	10000	1,38 s
1000	0,001	(-3,78, -3,28)	456	0,03 s	(-3,49, -0,03)	1000	0,14 s
100	0,001	(-3,15, -0,72)	100	0,00 s	(-3,93, -0,00)	100	0,02 s
10	0,001	(-3,41, -0,06)	10	0,00 s	(-3,99, -0,00)	10	0,00 s
1	0,001	(-3,90, -0,01)	1	0,00 s	(-3,99, -0,00)	1	0,00 s

Tabela 4. Porównanie metod spadku gradientu i Newtona dla punktu startowego (-4, 0).

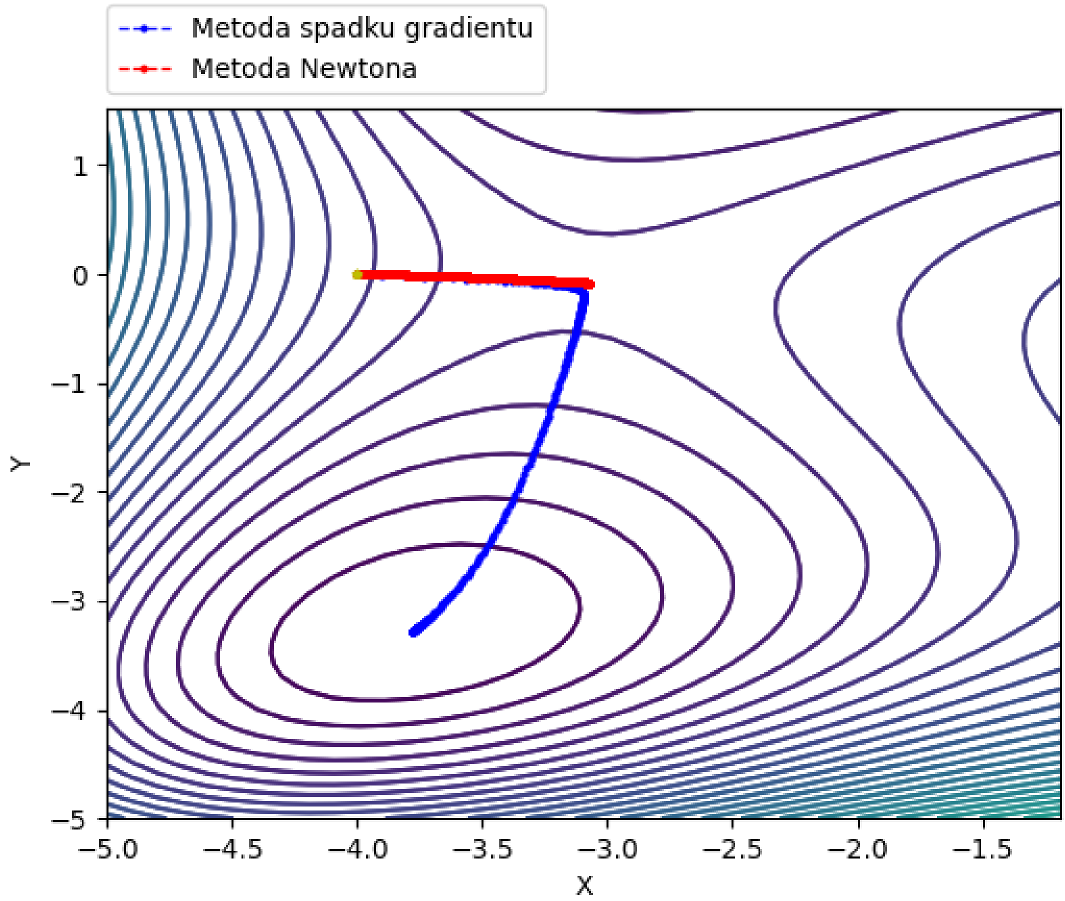
Na Rys. 2. Zobrazowana została droga powstała z połączenia punktów w kolejnych iteracjach algorytmów dla punktu początkowego (0, 0).



Rys. 2. Wyznaczone punkty w kolejnych iteracjach dla B = 0.001, iter_maks = 100000.

Jak widać na Rys. 2. metoda Newtona nie pozwoliła na wyznaczenie minimum lokalnego. Można by rzec, że skusiła się najbliższym punktem przypominającym minimum lokalne. (Nie mam pewności dlaczego tak się dzieje, ale możliwe, że jest to tak zwany punkt siodłowy określany w analizie matematycznej przy funkcji dwóch zmiennych. Jeśli tak to znaczyłoby to, że metoda Newtona nie jest na nie odporna). Podobnie dzieje się dla punktu startowego (-4, 0), co przedstawia Rys. 3.

Komentarze w nawiasach ( ) oznaczają niepewne wnioski😊



Rys. 3 Wyznaczone punkty w kolejnych iteracjach dla B = 0.001, iter_maks = 100000.

Porównując Rys. 2 i Rys. 3, można zauważyć, że początkowy kierunek metody spadku gradientu pokrywa się z kierunkem metody Newtona. Pomimo napotkania potencjalnego minimum lokalnego metoda spadku gradientu znalazła poprawne minimum lokalne w punkcie (-3,78, -3,28). (Może to oznaczać, że metoda gradientu jest odporna na napotkane punkty siodłowe.

Podsumowanie

Z przeprowadzonych eksperymentów wynika, że dla badanej funkcji w zależności od dobranych parametrów (β, iter_maks) jedna metoda jest lepsza od drugiej. Dla dużych parametrów β, iter_maks metoda Newtona znalazła minimum lokalne, a metoda spadku gradientu nie. Dla małego parametru β i dużej maksymalnej liczbie iteracji szybsza jest metoda spadku gradientu i wymaga mniejszej liczby iteracji. (Metoda Newtona nie jest odporna na punkty siodłowe i nadaje się lepiej do płaskich funkcji.)

Bibliografia

[1] https://en.wikipedia.org/wiki/Himmelblau%27s_function