Metoda gradientu prostego

Karol Korszun

March 2024

1 Opis algorytmu

Podczas poszukiwania lokalnego minimum funkcji, metoda gradientu prostego polega na iteracyjnym przemieszczaniu się w kierunku przeciwnym do gradientu funkcji w punkcie startowym. Najpierw obliczany jest gradient funkcji w aktualnym punkcie, który wskazuje kierunek najszybszego wzrostu funkcji. Następnie, dokonuje się przesunięcia punktu startowego w przeciwnym kierunku gradientu, o pewną stałą wielkość kroku. Proces ten powtarzany jest aż do osiągnięcia pewnego warunku zatrzymania, na przykład, gdy różnica między kolejnymi punktami osiągnie wartość mniejszą niż ustalona, lub gdy zostanie przekroczona maksymalna liczba iteracji.

Każdy kolejny punkt dany jest wzorem

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla F\left(\mathbf{x}_n\right), \quad n \geq 0$$

gdzie γ_n to długość kroku.

1.1 Opis testowanej funckji

Funckja na której były wykonywane wszystkie eksperymenty wygląda następująco:

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

$$\text{gdzie } \mathbf{x} \in [-100, 100]^n \subset \mathbb{R}^n, \quad n = 10,$$

Testy zostały przeprowadzone dla trzech różnych wartości parametru $\alpha = 1, 10, 100$.

2 Badanie wpływu długości kroku na zbieżność algorytmu

W ramach eksperymentów przeprowadzonych dla funkcji optymalizowanej zbadano wpływ różnych wartości parametru kroku (learning rate) na zbieżność algorytmu. Parametry kroku były wybierane z listy ustalonych wartości, która

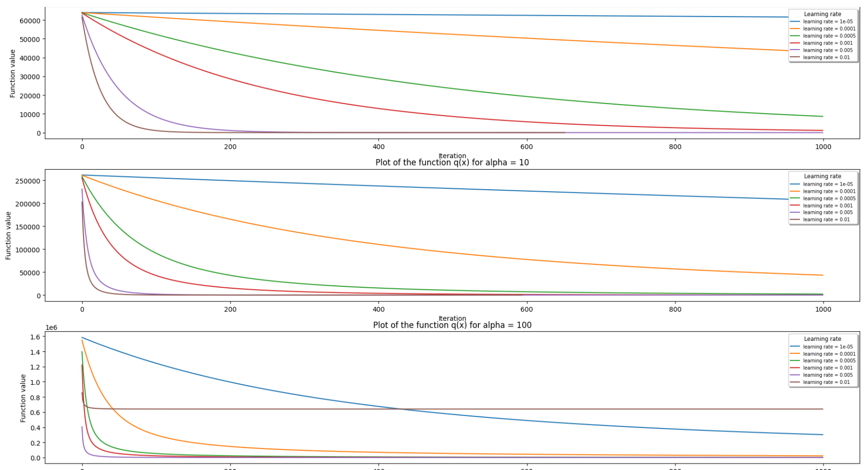


Figure 1: Wpływ długości kroku na zbieżność

obejmowała zakres od 0.00001 do 0.1. Celem eksperymentów było zrozumienie, w jaki sposób różne wartości parametru kroku wpływają na szybkość zbieżności algorytmu optymalizacyjnego.

W trakcie eksperymentów zauważono, że dla większych wartości parametru α, mniejsze wartości parametru kroku prowadziły do szybszej zbieżności algorytmu. Jednakże, należy zaznaczyć, że niektóre wartości parametru kroku nie doprowadziły do zbieżności algorytmu w ogóle. W przypadku stosowania zby t dużych wartości parametru kroku, algorytm mógł nie osiągnąć stanu zbieżności w granicy maksymalnej liczby iteracji, która wynosiła 1000.

3 Porównanie czasów

Zbadano czas, jaki algorytm potrzebował na wykonanie zadania, używając różnych wartości kroku od 0.00001 do 0.01. Próbki były równomiernie rozłożone w tym zakresie. Na osi y wykorszystano skalę logarytmiczną. Możemy zauważyć, że parametr alfa ma wielki wpływ na czas trwania algorytmu. Dla mniejszych rozmiarów kroku wszystkie czasy wyglądają podobnie, jednak gdy długość kroku zostanie zwiększona widać dużą dysproporcję w efektywności metody zależnie od parametru alfa.

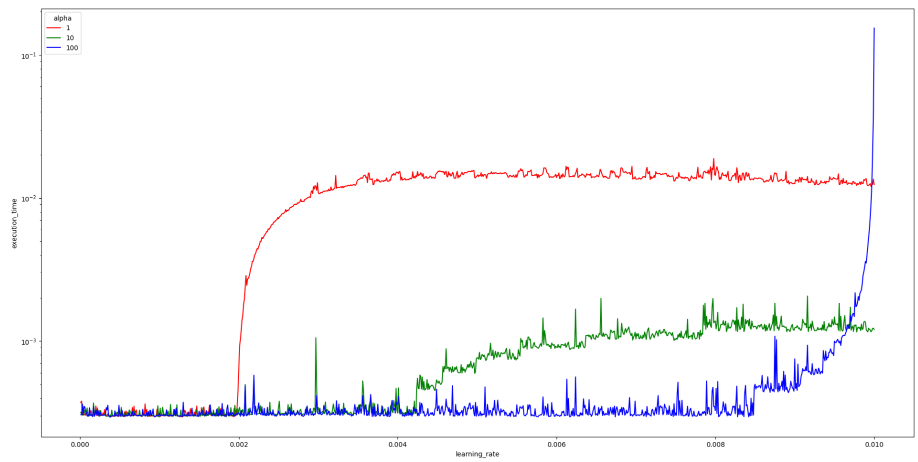


Figure 2: Wpływ długości kroku na zbieżność

4 Wizualizacja działania algorytmu

Zostały przeprowadzone eksperymenty w celu sprawdzenia zbieżności algorytmu dla różnych punktów początkowych x_0. Otrzymane wyniki zostały przedstawione poniżej.

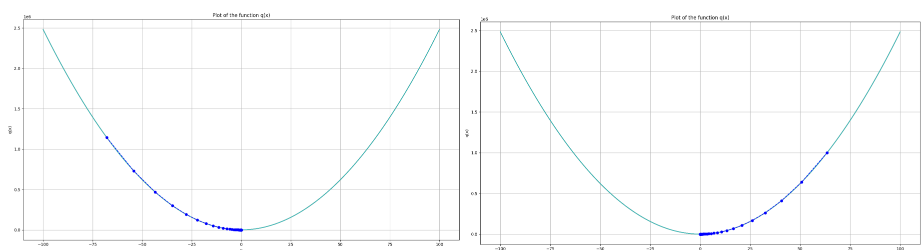
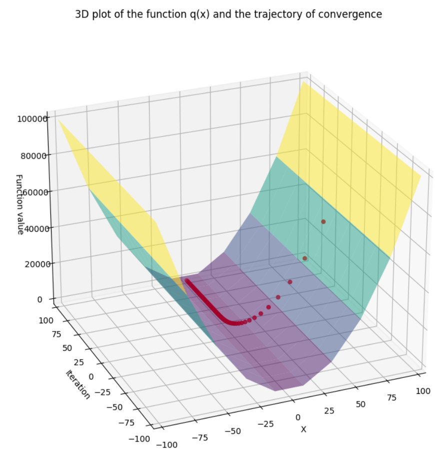


Figure 3: Zbieżność algorytmu w dwóch wymiarach

Poniżej przedstawiono wizualizację trajektorii zbieżności w trzech wymiarach. Jedną z osi na wykresie reprezentuje liczba iteracji, natomiast pozostałe dwie osie odnoszą się do współrzędnych przestrzeni wielowymiarowej. Obserwujemy, że wraz z postępem iteracji algorytm dąży do minimalizacji funkcji celu, dążąc do osiągnięcia lokalnego lub globalnego minimum.



(a)

Figure 4: Wykres zbieżności w trzech wymiarach

5 Wnioski i obserwacje

Z przeprowadzonych doświadczeń wynika, że długość kroku (learning rate) odgrywa kluczową rolę w zbieżności algorytmu gradientu prostego, więc sukces naszego algorytmu optymalizacji w dużej mierze zależy od prawidłowego doboru tego parametru, Dodatkowo, wyniki przeprowadzonych eksperymentów pokazały, że parametr alfa ma istotny wpływ na czas potrzebny do osiągnięcia optymalnego rozwiązania. Należy zauważyć, że dla większych wartości funkcji celu, mniejsze kroki, mogą prowadzić do szybszej konwergencji.