Temat: Algorytm gradientu prostego

Wykonujący ćwiczenie: Lena Zubik

Parametry:

x0 – punkt startowy algorytmu

min_x, max_x – zakres dziedziny funkcji, na której ilustrowane jest działanie algorytmu

eps - dokładność, z jaką algorytm ma zlokalizować minimum

learn_rate - współczynnik uczenia się (przemnożony przez wartość gradientu/pochodnej w danym punkcie daje długość kroku d)

max_iter - maksymalna liczba iteracji

Wybrane przeze mnie funkcje:

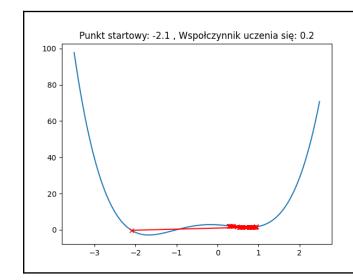
- Wielomian 2 stopnia 2*x^2
- Wielomian 4 stopnia $1.5*(x^4) + 2.2*(x^3) 3.3*(x^2) 1.3*x + 2.8$

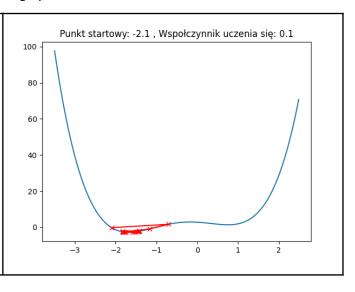
Krótki opis działania algorytmu:

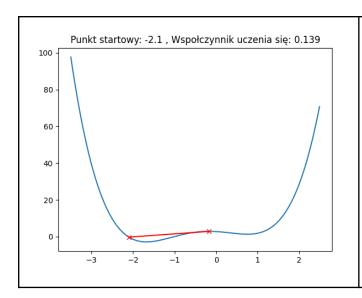
Startujemy w zadanym punkcie początkowym i w każdej iteracji obliczamy długość i kierunek (uwzględniany jako znak w długości) kroku. Jest on równy iloczynowi współczynnika uczenia się oraz wartości pochodnej w danym punkcie (informacji o nachyleniu krzywej w danym miejscu). Sprawdzamy czy długość tego kroku nie będzie mniejsza od zadanej dokładności eps (sprawdzenie warunku czy odległość między kolejnym punktem a bieżącym nie jest mniejsza niż zadana dokładność) i jeżeli jest on dłuższy to kontynuujemy obliczając współrzędne kolejnego punktu (ten w którym jesteśmy + krok) lub przerywamy i kończymy działanie algorytmu.

Eksperymenty:

Prezentacja wpływu doboru współczynnika uczenia się na wynik algorytmu:



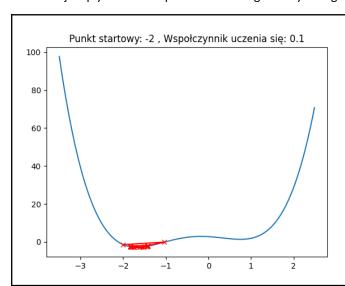


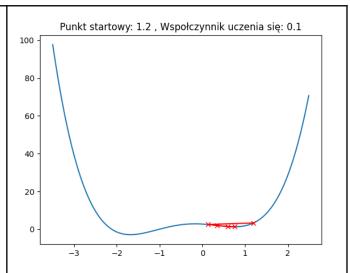


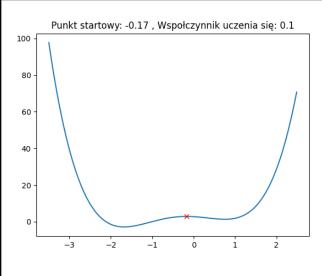
Obserwacje: Jak widać w zależności od wielkości współczynnika uczenia się uzyskujemy różne rezultaty. Raz algorytm utyka w prawym, raz w lewym minimum. Może się zdarzyć też, że algorytm nie odnajdzie minimum, przemieszcza się od jednego punktu do drugiego.

Wnioski: Należy dla każdego przypadku sprawdzić rezultat algorytmu dla różnych wielkości współczynnika uczenia się. Zbyt duża jego wartość powoduje, że krok się znacznie wydłuża i algorytm może przeskoczyć do kolejnego minimum. Zbyt mała jego wartość znacznie wydłuża działanie algorytmu i zwiększa liczbę potrzebnych iteracji do osiągnięcia wyniku końcowego. Warto też zdawać sobie sprawę, że ograniczenie na liczbę iteracji jest konieczne, ponieważ algorytm przy złośliwie wybranym współczynniku może utknąć w między dwoma punktami.

Prezentacja wpływu doboru punktu startowego na wynik algorytmu:







Obserwacje: Jak widać w zależności od tego jaki punkt startowy zostanie wynik działania algorytmu będzie różny. Raz znajdziemy prawe, raz lewe minimum. Czasami może się okazać, że nie otrzymamy minimum (bardzo złośliwych przypadkach algorytm utknie w punkcie startowym).

Wnioski: W zależności od wybranego punktu startowego możemy otrzymać różne minima funkcji. Przy rozsądnie małej wielkości współczynnika uczenia (brak gwałtownych przeskoków) algorytm obiera kierunek poruszania się bazując głównie na pochodnej funkcji w danym punkcie (nachyleniu krzywej). Porusza się zgodnie z jej spadkiem, dlatego w przypadku punktu startowego obranego na zboczu krzywej, algorytm wybierze minimum zlokalizowane w dolinie, do której ona prowadzi. Niebezpieczną sytuacją jest obranie punktu startowego w maksimum lokalnym danej funkcji. Tam pochodna, tak jak i w minimach, się zeruje, co skutkuje zerowym krokiem. Aby zapobiec tworzeniu się nieskończonej pętli w takich wypadkach, należy zawsze ograniczać liczbę możliwych iteracji.