WSI

Ćwiczenie 1

Prowadzący : mgr inż. Mikołaj Markiewicz

Wykonał: Jan Kaniuka

Numer indeksu: 303762

Treść zadania – zagadnienie przeszukiwania i podstawowe podejście do niego.

Zaimplementować metodę gradientu prostego dla funkcji jednej zmiennej. Zbadać działanie metody w zależności od parametrów wejściowych:

- punkt startowy

- współczynnika uczenia / parametru kroku

Eksperymenty przeprowadzić dla funkcji z jednym minimum oraz dla funkcji z minimum lokalnym, czyli np. :

$$f(x) = x^2 + 3x + 8$$

$$f(x) = x^4 - 5x^2 - 3x$$

Nie trzeba implementować liczenia pochodnej z funkcji wejściowej - podajemy jako już znaną funkcję.

Przyjęte założenia oraz doprecyzowanie treści

W zadaniu nie zostało sprecyzowane *kryterium stopu* algorytmu. Typowe kryteria stopu dla metody gradientu prostego to: ograniczenie liczby iteracji, wartość gradientu bliska zeru lub brak poprawy funkcji celu. W swojej implementacji zastosowałem ostatnie z wyżej wymienionych kryteriów:

Kryterium stopu : $|f(x_{k+1}) - f(x_k)| < e$

gdzie *e* – założona dokładność. Na potrzeby eksperymentów przyjąłem *e* = 10^{-3}

Raport z przeprowadzonych eksperymentów

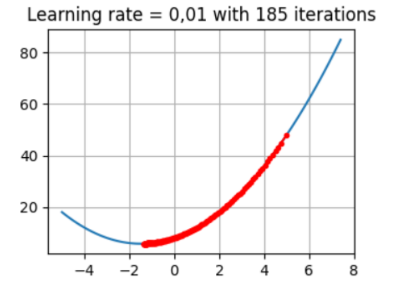
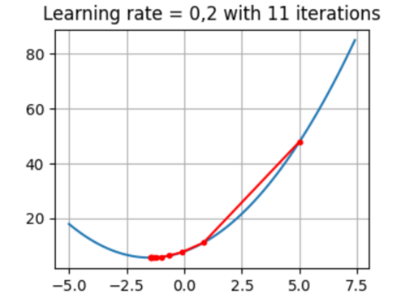
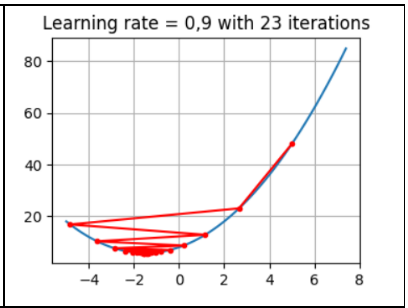
Funkcja kwadratowa z jednym minimum

Na początku zbadałem wpływ wartości *współczynnika uczenia* na wyznaczanie minimum.

Parametry eksperymentu: $f(x) = x^2 + 3x + 8$, *e* = 10^{-3}, punkt startowy *x = 5*

Przeprowadziłem kilka eksperymentów z różnymi parametrami algorytmu, lecz w sprawozdaniu zamieszczam jedynie te, które pozwalają na zaobserwowanie wpływu tychże parametrów na działanie *metody gradientu prostego.*

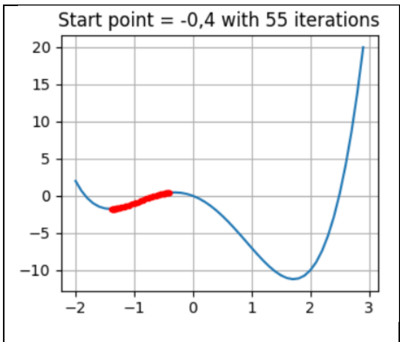
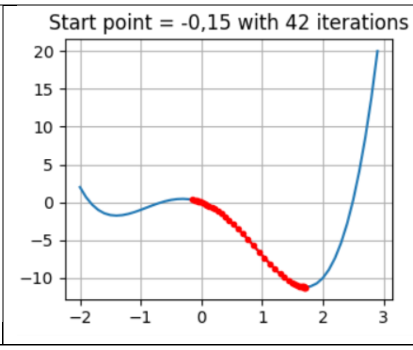
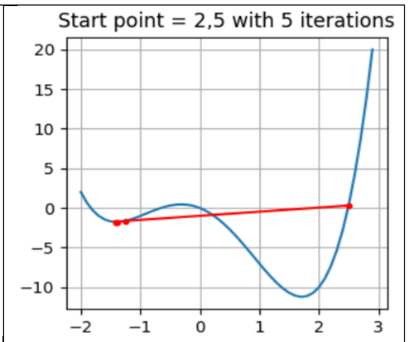
Pod każdym z wykresów zamieściłem wartość argumentu *x*, w którym funkcja osiąga minimum oraz zwięzłe obserwacje. Testowa funkcja kwadratowa osiąga swoje minimum dla *x* = -1,5.

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>x = - 1.35</td>
<td>x = -1,47</td>
<td>x = -1,53</td>
</tr>
<tr>
<td>Bardzo mały parametr kroku skutkuje dużą liczbą iteracji. Schodzenie po tzw. „minus gradiencie” jest wyraźnie zbyt wolne i czasochłonne.</td>
<td>20-krotne zwiększenie długości kroku pozwoliło na znaczne ograniczenie liczby iteracji. Widać wyraźnie, że pierwsze przesunięcie do kolejnego punktu x_{i+1} jest największe, a w kolejnych krokach maleje.</td>
<td>Przyjęcie zbyt dużej wartości learning rate może skutkować oscylacjami. Kolejne kroki mają zbyt dużą wartość, co skutkuje przeskakiwaniem rozwiązania (overshoot). Takie zachowanie algorytmu jest wysoce niepożądane.</td>
</tr>
</table>

<u>Funkcja stopnia czwartego z dwoma minimami lokalnymi</u>

Druga funkcja (wielomian czwartego rzędu) ma dwa minima lokalne. Na jej przykładzie postanowiłem sprawdzić działanie metody w zależności od wyboru punktu startowego.

Parametry eksperymentu: $f(x) = x^4 - 5x^2 - 3x$, $e = 10^{-3}$, *learning rate* = 0,01

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td colspan="2">W pierwszym przypadku jako punkt startowy wybrano x = -0,4. Algorytm zaczynając z tego miejsca znalazł minimum – niestety jedynie lokalne. Niewielkie przesunięcie punktu początkowego w prawo (do x = -0,15) poskutkowało znalezieniem minimum globalnego. Wniosek: należy być ostrożnym podczas wybierania punktu startowego w okolicach maksimum lokalnego.</td>
<td>W tym przypadku zwiększyłem learning rate do wartości 0,1. Chciałem przez to pokazać, że mimo rozpoczęcia procedury obok <u>minimum globalnego</u>, możemy przeskoczyć do <u>minimum lokalnego</u> przez zbyt dużą wartość kroku.</td>
</tr>
</table>

Wnioski i przemyślenia

- Dobranie zbyt dużej wartości *learning rate* skutkuje na ogół oscylacjami i *przeskakiwaniem* minimum.

- Dobór punktu startowego jest kluczowy dla funkcji z wieloma minimami lokalnymi. Aby częściowo rozwiązać ten problem można uruchomić algorytm kilkukrotnie dla różnych (randomowych) punktów startowych.