Laboratorium 11 - Spadek wzdłuż gradientu

Dawid Żak Szymon Hołysz

2025-06-18

Table of contents

Zadanie 1.

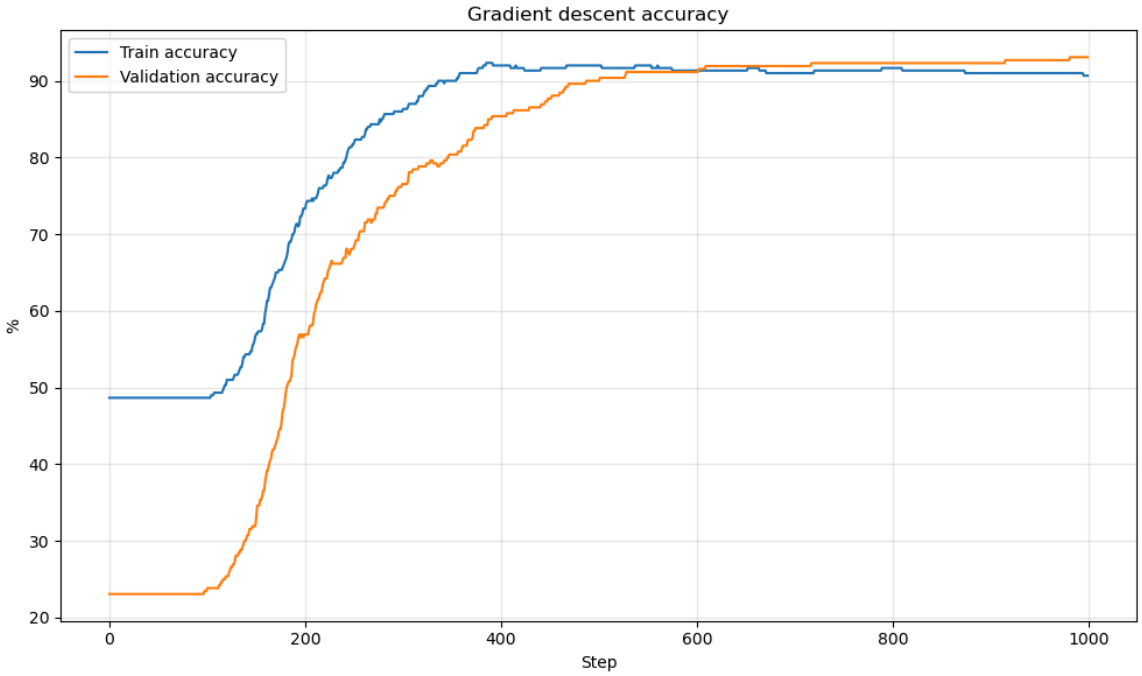
Rozwiąż ponownie problem predykcji typu nowotworu (laboratorium 2), używając metody spadku wzdłuż gradientu (ang. *gradient descent*). Stałą uczącą możesz wyznaczyć na podstawie najmniejszej i największej wartości własnej macierzy $A^T A$. Porównaj uzyskane rozwiązanie z metodą najmniejszych kwadratów, biorąc pod uwagę następujące kryteria:

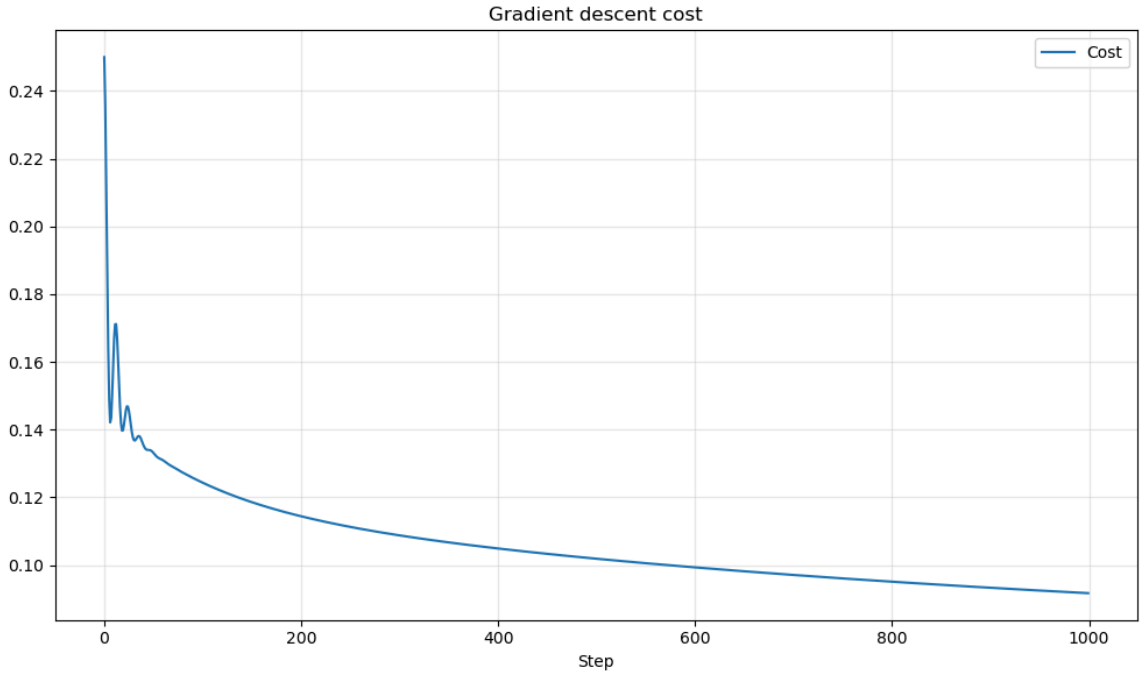
- Dokładność predykcji na zbiorze testowym
- Teoretyczną złożoność obliczeniową
- Czas obliczeń.

```
(300, 31) (300, 2) (260, 31) (260, 2)
```

```
Learning rate:  3.667777291369401e-08
Step   1    cost value:  0.25,  train accuracy: 48.67,  validation accuracy:
23.08
Step  50    cost value:  0.13,  train accuracy: 48.67,  validation accuracy:
23.08
Step 100    cost value:  0.12,  train accuracy: 48.67,  validation accuracy:
23.46
Step 150    cost value:  0.12,  train accuracy: 56.67,  validation accuracy:
31.92
Step 200    cost value:  0.11,  train accuracy: 73.33,  validation accuracy:
56.92
Step 250    cost value:  0.11,  train accuracy: 81.67,  validation accuracy:
68.08
Step 300    cost value:  0.11,  train accuracy: 86.00,  validation accuracy:
76.54
Step 350    cost value:  0.11,  train accuracy: 90.00,  validation accuracy:
80.38
Step 400    cost value:  0.10,  train accuracy: 92.00,  validation accuracy:
85.38
```

```
Step 450     cost value:   0.10,   train accuracy: 91.67,   validation accuracy:
87.69
Step 500     cost value:   0.10,   train accuracy: 92.00,   validation accuracy:
90.00
Step 550     cost value:   0.10,   train accuracy: 92.00,   validation accuracy:
91.15
Step 600     cost value:   0.10,   train accuracy: 91.33,   validation accuracy:
91.15
Step 650     cost value:   0.10,   train accuracy: 91.33,   validation accuracy:
91.92
Step 700     cost value:   0.10,   train accuracy: 91.00,   validation accuracy:
91.92
Step 750     cost value:   0.10,   train accuracy: 91.33,   validation accuracy:
92.31
Step 800     cost value:   0.10,   train accuracy: 91.67,   validation accuracy:
92.31
Step 850     cost value:   0.09,   train accuracy: 91.33,   validation accuracy:
92.31
Step 900     cost value:   0.09,   train accuracy: 91.00,   validation accuracy:
92.31
Step 950     cost value:   0.09,   train accuracy: 91.00,   validation accuracy:
92.69
Step 1000   cost value:  0.09,  train accuracy: 90.67,  validation accuracy: 93.08
```





Metodą spadku wzdłuż gradientu udało się uzyskać dokładność predykcji na zbiorze walidacyjnym na poziomie 93%. Jest to nieco mniejszy wynik o dokładności predykcji metodą najmniejszych kwadratów, która wynosiła 97%.

Porównując czasy wykonania, należy zwrócić uwagę na znaczną przewagę metody najmniejszych kwadratów, przy której czas rozwiązywania układu równań funkcją biblioteczną numpy.linalg.solve był krótszy niż 0.1 s, natomiast znajdywanie rozwiązania metodą gradient descent trwało 1.9 s.

Wynika to bezpośrednio ze złożoności obliczeniowej, która dla metody najmniejszych kwadratów wynosi $O(n^3)$, gdzie n to liczba parametrów, natomiast dla gradient descent - $O(ndk)$, gdzie n to liczba iteracji, d to liczba parametrów, a k to liczba punktów danych. Stąd gradient descent może sprawdzać się tylko dla dużych liczb parametrów.

Zadanie 2.

Należy wyznaczyć najkrótszą ścieżkę robota pomiędzy dwoma punktami $x^{(0)}$ i $x^{(n)}$. Problemem są przeszkody usytuowane na trasie robota, których należy unikać. Zadanie polega na minimalizacji funkcji kosztu, która sprowadza problem nieliniowej optymalizacji z ograniczeniami do problemu nieograniczonego optymalizacji.

Macierz $X \in \mathbb{R}^{(n+1)\times 2}$ opisuje ścieżkę złożoną z $n+1$ punktów $x^{(0)}, x^{(1)}, x^{(2)}, ..., x^{(n)}$. Każdy punkt $x^{(i)} \in \mathbb{R}^2$. Punkty początkowy i końcowy ścieżki, $x^{(0)}$ i $x^{(n)}$, są ustalone.

Punkty z przeszkodami (punkty o 2 współrzędnych), $r^{(j)}$ dane są w macierzy przeszkód $R \in \mathbb{R}^{k\times 2}$.

W celu optymalizacji ścieżki robota należy użyć metody największego spadku. Funkcja celu użyta do optymalizacji $F\big(x^{(0)}, x^{(1)}, ..., x^{(n)}\big)$ zdefiniowana jest jako:

$$F\big(x^{(0)}, x^{(1)}, ..., x^{(n)}\big) = \lambda_1 \sum_{i=0}^{n} \sum_{j=1}^{k} \frac{1}{\epsilon + \parallel x^{(i)} - r^{(j)} \parallel^2} + \lambda_2 \sum_{i=0}^{n-1} \parallel x^{(i+1)} - x^{(i)} \parallel^2$$

Symbole użyte we wzorze mają następujące znaczenie:

- Stałe λ_1 i λ_2 określają wpływ każdego członu wyrażenia na wartość $F(X)$.
  - λ_1 określa wagę składnika zapobiegającego zbytniemu zbliżaniu się do przeszkody
  - λ_2 określa wagę składnika zapobiegającego tworzeniu bardzo długich ścieżek
- n jest liczbą odcinków, a $n + 1$ liczbą punktów na trasie robota.
- k jest liczbą przeszkód, których robot musi unikać.
- Dodanie ϵ w mianowniku zapobiega dzieleniu przez zero.

1. Wyprowadź wyrażenie na gradient ∇F funkcji celu F względem $x^{(i)}$: $\nabla F = \left[\frac{\partial F}{\partial x^{(0)}}, ..., \frac{\partial F}{\partial x^{(n)}}\right]$. Wzór wyraź poprzez wektory $x^{(i)}$ i ich składowe, wektory $r^{(j)}$ i ich składowe, $\epsilon, \lambda_1, \lambda_2, n$ i k (niekoniecznie wszystkie). Wskazówka. $\frac{\partial \|z\|^2}{\partial z} = 2z$.
2. Opisz matematycznie i zaimplementuj kroki algorytmu największego spadku z przeszukiwaniem liniowym, który służy do minimalizacji funkcji celu F. Do przeszukiwania liniowego (ang. *line search*) użyj metody złotego podziału (ang. *golden section search*). W tym celu załóż, że F jest unimodalna (w rzeczywistości tak nie jest) i że można ustalić początkowy przedział, w którym znajduje się minimum.
3. Znajdź najkrótszą ścieżkę robota przy użyciu algorytmu zaimplementowanego w w poprzednim punkcie. Przyjmij następujące wartości parametrów:
   - $n = 20, k = 50$
   - $x^{(0)} = [0, 0]$, $x^{(n)} = [20, 20]$
   - $r^{(j)} \sim \mathcal{U}(0, 20) \times \mathcal{U}(0, 20)$
   - $\lambda_1 = \lambda_2 = 1$
   - $\epsilon = 10^{-13}$
   - liczba iteracji = 400

Ponieważ nie chcemy zmieniać położenia punktu początkowego i końcowego, $x^{(0)}, x^{(n)}$, wyzeruj gradient funkcji F względem tych punktów.

Obliczenia przeprowadź dla 5 różnych losowych inicjalizacji punktów wewnątrz ścieżki $x^{(1)}, ..., x^{(n-1)}$.

Narysuj przykładowy wykres wartości funkcji F w zależności od iteracji.

Zapewnij powtarzalność wyników, ustawiając wartość odpowiedniego ziarna.

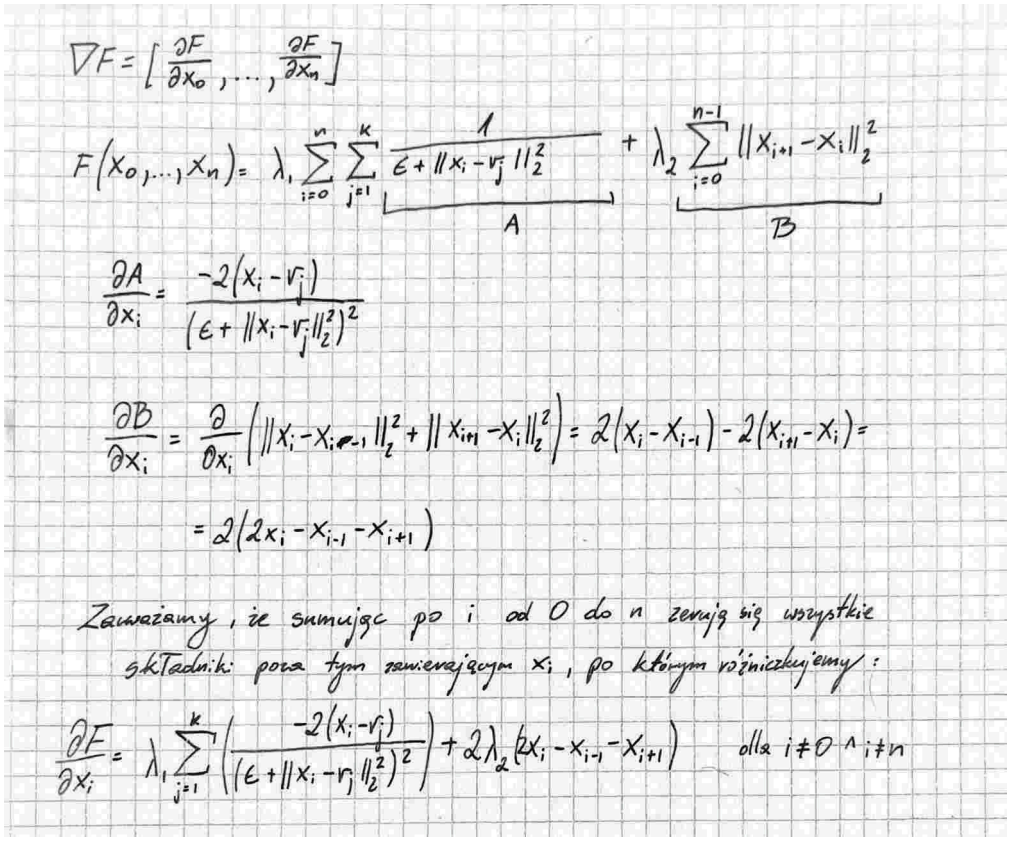


Figure 1: wyprowadzenie gradientu

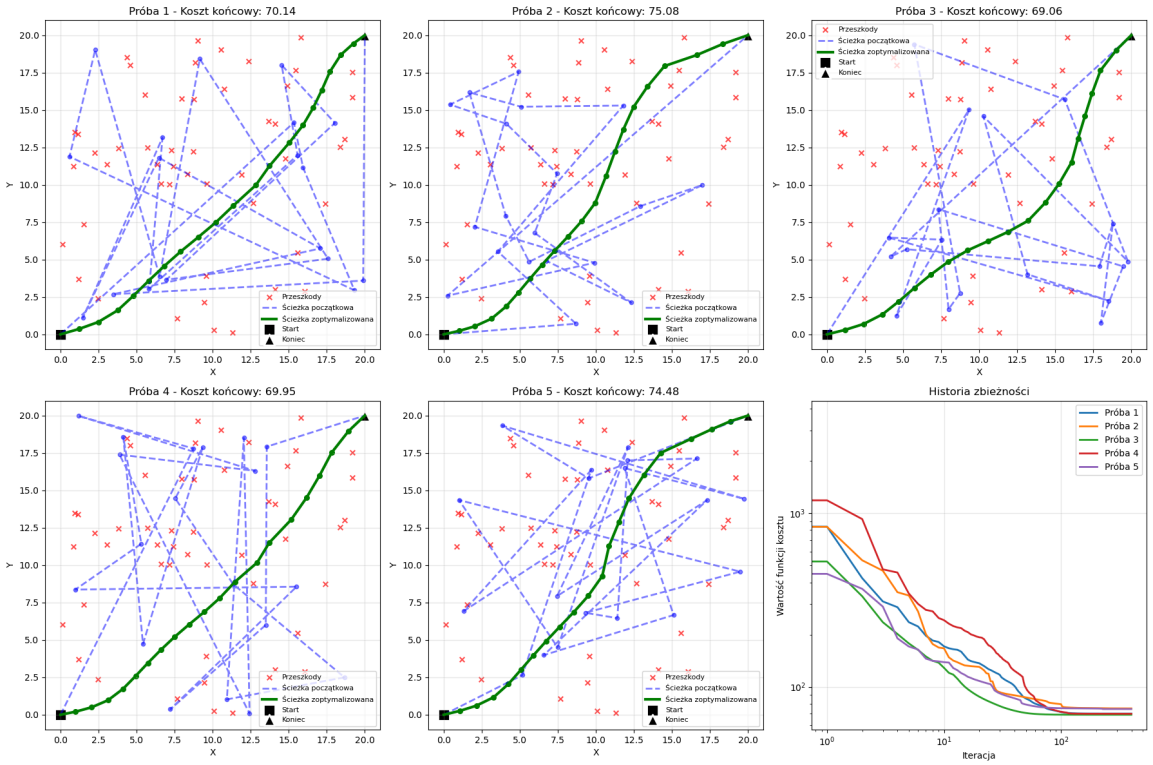
Po znalezieniu gradientu funkcji kosztu zaimplementujemy algorytm największego spadku: - inicjalizujemy punkty ścieżki i przeszkody - Powatarzamy poniższe kroki ustaloną liczbę razy (w naszym wypadku 400): - Obliczamy gradient ∇F w punkcie - Wyznaczamy optymalną wartość kroku za pomocą metody złotego podziału - Aktualizujemy punkty ścieżki

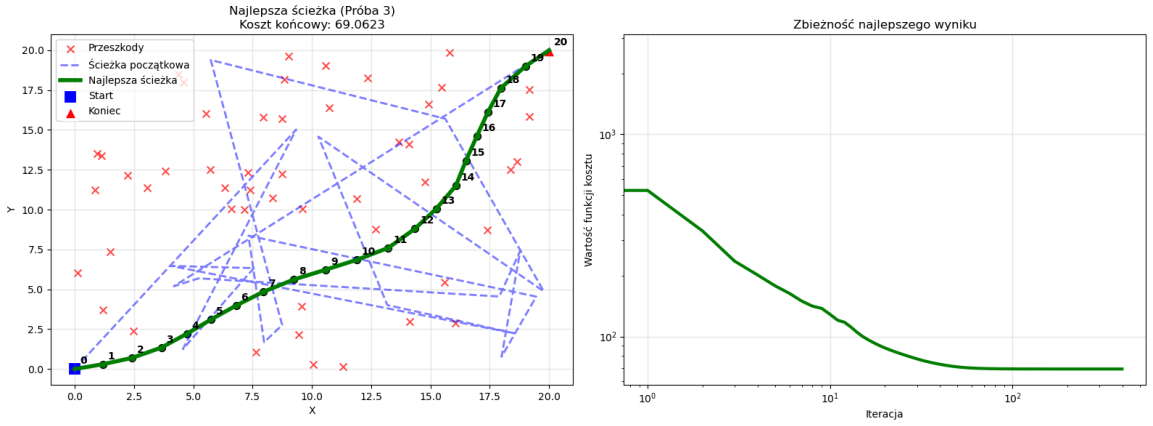
Metoda złotego podziału opiera się na wykorzystaniu szczególnych własności funkcji unimodalnej, to jest takiej, która na danym przedziale posiada dokładnie jedno minimum.

Ustalamy współczynnik t, $0 < t < 1$, a następnie powtarzamy poniższe operacje aż do uzyskania żądanej zbieżności: - Obliczamy długość przedziałów $d = t(b - a)$, - Przyjmujemy punkty $l = b - d$, $r = a + d$, - Porównujemy wartości funkcji i decydujemy, czy minimum znajduje się w przedziale lewym, czy prawym i ustawiamy odpowiednio a i b.

```
Koszty końcowe dla każdej próby:
Próba 1: 70.139501
Próba 2: 75.080737
Próba 3: 69.062293
Próba 4: 69.951910
Próba 5: 74.477471
```

Znalezione przez algorytm optymalne ścieżki ilustrują poniższe wykresy:






```
Najlepszy wynik z próby 3
Koszt końcowy: 69.062293
Liczba iteracji: 400
```