

## Optimalizacja funkcji wielu zmiennych metodami gradientowymi

### 1. Cel ćwiczenia.

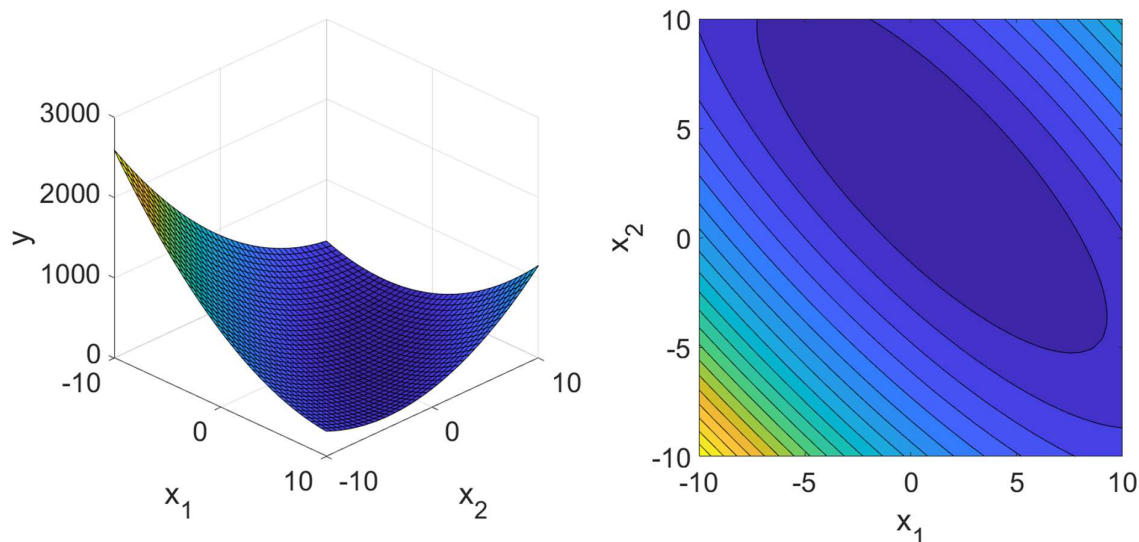
Celem ćwiczenia jest zapoznanie się z gradientowymi metodami optymalizacji poprzez ich implementację oraz wykorzystanie do wyznaczenia minimum podanej funkcji celu.

### 2. Testowa funkcja celu.

Funkcja celu dana jest wzorem:

$$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

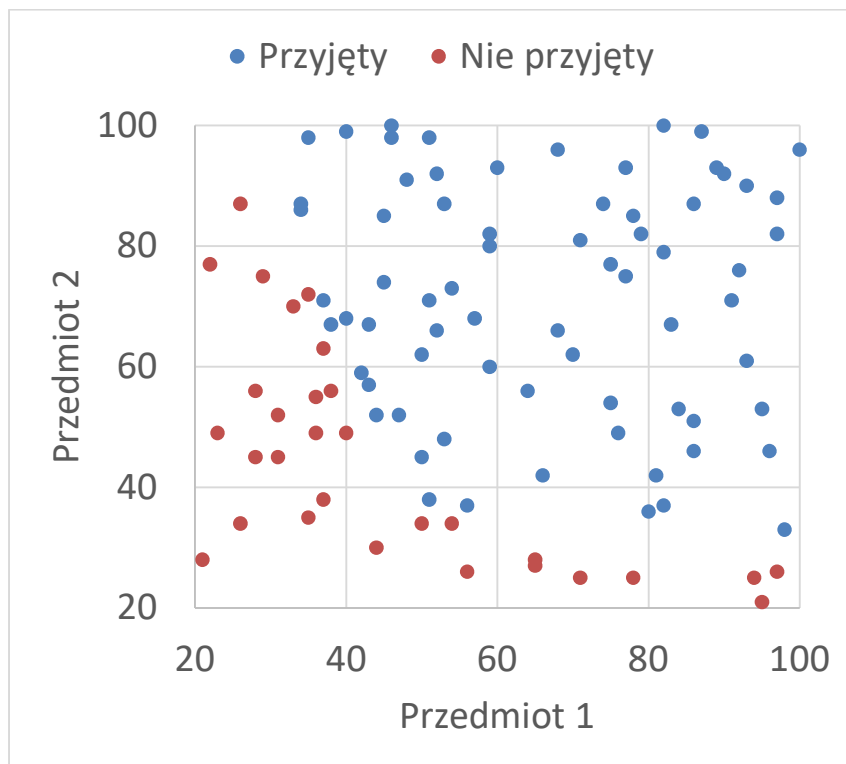
Jej wykres przedstawiony jest poniżej.



Punkt startowy powinien należeć do przedziału  $x_1^{(0)} \in [-10, 10]$ ,  $x_2^{(0)} \in [-10, 10]$ .

### 3. Problem rzeczywisty.

O przyjęciu na pewną uczelnię decydują oceny uzyskane z dwóch przedmiotów. Na poniższym rysunku przedstawiono przykładowe oceny wraz z decyzją o przyjęciu lub odrzuceniu kandydata.



Celem optymalizacji jest znalezienie optymalnych parametrów klasyfikatora, który na podstawie otrzymanych ocen, będzie w stanie przewidzieć, czy dana osoba zostanie przyjęta na uczelnię. Klasyfikator będzie wykorzystywał hipotezę postaci:

$$h_{\theta}(\mathbf{x}) = \frac{1}{1 + e^{-\theta^T \mathbf{x}}}$$

gdzie: wektor  $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$  zawiera szukane parametry klasyfikatora, wektor  $\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$  zawiera oceny z przedmiotu 1 ( $x_1$ ) oraz przedmiotu 2 ( $x_2$ ).

Klasyfikator wskaże, że  $i$ -ty kandydat zostanie przyjęty na uczelnię, jeżeli wartość hipotezy będzie większa lub równa 0,5, tj.:

$$h_{\theta}(\mathbf{x}^i) \geq 0,5 \Rightarrow y^i = 1$$

Dane uczące (przedstawione na rysunku) są dostępne w plikach XData.txt oraz YData.txt w postaci macierzy  $X$  oraz  $Y$ :

$$X = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m], \mathbf{x}^i = \begin{bmatrix} 1 \\ x_1^i \\ x_2^i \end{bmatrix}, i = 1, \dots, m$$

$$Y = [y^1, y^2, \dots, y^m], y^i = \{0,1\}, i = 1, \dots, m$$

gdzie: liczba danych  $m = 100$ ,  $y^i = 1$  oznacza, że  $i$ -ty kandydat został przyjęty,  $y^i = 0$  oznacza, że  $i$ -ty kandydat nie został przyjęty.

Poszukiwanie optymalnych parametrów klasyfikatora odbywa się poprzez minimalizację funkcji kosztu danej wzorem:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left( y^i \cdot \ln(h_{\theta}(\mathbf{x}^i)) + (1 - y^i) \cdot \ln(1 - h_{\theta}(\mathbf{x}^i)) \right)$$

Pochodne cząstkowe funkcji kosztu  $J(\theta)$  wynoszą:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(\mathbf{x}^i) - y^i) \cdot x_j^i \text{ dla } j = 1, \dots, n$$

gdzie:  $n$  – liczba współrzędnych wektora gradientu.

W celu sprawdzenia poprawności implementacji funkcji kosztu oraz jej gradientu, można obliczyć ich wartości dla wektora  $\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ . Poprawne wartości wynoszą w przybliżeniu:  $J(\theta) = 0,6931$  oraz

$$\nabla J(\theta) = \begin{bmatrix} -0,2 \\ -16,405 \\ -18,29 \end{bmatrix}.$$

#### 4. Algorytmy optymalizacji.

Do wyznaczenia minimum funkcji celu należy zastosować:

- dla testowej funkcji celu metody najszybszego spadku, gradientów sprzężonych oraz Newtona, każdą w wersji stałokrokowej i zmiennokrokowej (długości kroku w wersji stałokrokowej należy przyjąć równe 0,05 oraz 0,12, w wersji zmiennokrokowej długość kroku należy wyznaczyć metodą złotego podziału),
- dla problemu rzeczywistego metodę gradientów sprzężonych w wersji stałokrokowej (długości kroku należy przyjąć równe 0,01, 0,001 oraz 0,0001).

#### 5. Zadanie do samodzielnego wykonania.

##### a. Testowa funkcja celu.

Zadanie polega na wykonaniu 100 optymalizacji dla każdej długości kroku startując z losowego punktu. Wyniki należy zestawić pliku xlsx w tabeli 1. Wartości średnie należy przedstawić w tabeli 2. Dodatkowo, dla jednego wybranego punktu startowego należy narysować 6 wykresów. Na każdym, na wykres poziomicy należy nanieść rozwiązania optymalne uzyskane po każdej iteracji:

- wykres 1 – dla długości kroku równej 0,05 rozwiązania otrzymane każdą z metod,
- wykres 2 – dla długości kroku równej 0,12 rozwiązania otrzymane każdą z metod,
- wykres 3 – dla wersji zmiennokrokowej rozwiązania otrzymane każdą z metod,
- wykres 4 – dla metody najszybszego spadku rozwiązania otrzymane dla każdej długości kroku,
- wykres 5 – dla metody gradientów sprzężonych rozwiązania otrzymane dla każdej długości kroku,
- wykres 6 – dla metody Newtona rozwiązania otrzymane dla każdej długości kroku.

### b. Problem rzeczywisty.

Zadanie polega na wykonaniu jednej optymalizacji dla każdej długości kroku startując z punktu  $\theta^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ . Wyniki należy zestawić pliku xlsx w tabeli 3. W kolumnie  $P(\theta^*)$  należy podać jaki procent przypadków ze zbioru uczącego został poprawnie zaklasyfikowany, tj.:

$$P(\theta^*) = \frac{1}{m} \sum_{i=1}^m \begin{cases} 1 & \text{jeżeli } [h_{\theta}(\mathbf{x}^i)] = y^i \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

gdzie:  $[\cdot]$  oznacza zaokrąglenie liczby do najbliższej wartości całkowitej.

Dodatkowo, na wykres przedstawiający przyjęte/nie przyjęte osoby należy nanieść granicę klasyfikacji otrzymaną dla najlepszego przypadku (arkusz wykres). Granica klasyfikacji dana jest równaniem:

$$h_{\theta}(\mathbf{x}) = 0,5$$

W rozpatrywanym przypadku (przy założeniu, że  $\theta_2 \neq 0$ ) granica klasyfikacji to funkcja o równaniu:

$$x_2 = -\frac{\theta_0 + \theta_1 x_1}{\theta_2}$$

## 6. Sprawozdanie

Sprawozdanie powinno zostać przygotowane w formacie docx (lub doc) albo pdf i powinno zawierać parametry poszczególnych algorytmów, dyskusję wyników oraz wnioski. Dodatkowo, w sprawozdaniu należy umieścić kod zaimplementowanych metod, funkcję lab4 oraz funkcje wykorzystane do obliczenia funkcji celu, gradientu oraz hesjanu. Wyniki optymalizacji oraz wykresy należy przygotować w formacie xlsx (lub xls).

### Pseudokod metod gradientowych.

**Dane wejściowe:** punkt startowy  $\mathbf{x}^{(0)}$ , dokładność  $\epsilon > 0$ , maksymalna liczba wywołań funkcji celu  $N_{\max}$

```
1:  i = 0
2:  repeat
3:      wyznacz d(i)
4:      wyznacz h(i)
5:      x(i+1) = x(i) + h(i) · d(i)
6:      i = i + 1
7:      if f_calls > Nmax then
8:          return error
9:      end if
10: until ||x(i) - x(i-1)||2 < ε
11: return x* = x(i)
```

Kierunek  $d^{(i)}$  wyznacza się ze wzoru:

- dla metody najszybszego spadku:  $d^{(i)} = -\nabla f(x^{(i)})$
- dla metody gradientów sprzężonych:  $d^{(0)} = -\nabla f(x^{(0)})$   
 $d^{(i)} = -\nabla f(x^{(i)}) + \beta \cdot d^{(i-1)}$   
$$\beta = \frac{(\|\nabla f(x^{(i)})\|_2)^2}{(\|\nabla f(x^{(i-1)})\|_2)^2}$$
- dla metody Newtona:  $d^{(i)} = -H^{-1}(x^{(i)})\nabla f(x^{(i)})$

### Pseudokod metody złotego podziału.

**Dane wejściowe:** przedział poszukiwań  $[a, b]$ , dokładność obliczeń  $\varepsilon > 0$ , maksymalna liczba wywołań funkcji celu  $N_{\max}$

```
1:  i = 0
2:  α = (50,5 - 1) / 2
3:  a(0) = a, b(0) = b
4:  c(0) = b(0) - α(b(0) - a(0))
5:  d(0) = a(0) + α(b(0) - a(0))
6:  repeat
7:      if f(c(i)) < f(d(i)) then
8:          a(i+1) = a(i)
9:          b(i+1) = d(i)
10:         c(i+1) = b(i+1) - α(b(i+1) - a(i+1))
11:         d(i+1) = c(i)
12:     else
13:         a(i+1) = c(i)
14:         b(i+1) = b(i)
15:         c(i+1) = d(i)
16:         d(i+1) = a(i+1) + α(b(i+1) - a(i+1))
17:     end if
18:     i = i + 1
19:     if f_calls > N_max then
20:         return error
21:     end if
22: until b(i) - a(i) < ε
23: return x* = (a(i) + b(i)) / 2
```