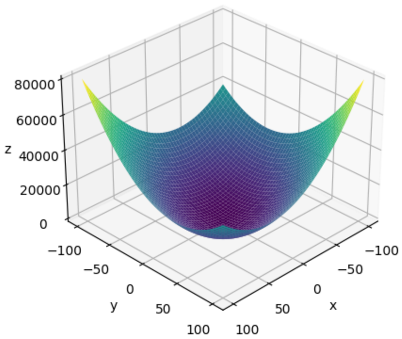
Praca domowa 2
Metoda gradientowa oraz
minimalizacja wybranej funkcji

Damian Jankowski s188597

22 kwietnia 2023

1 Wstęp

Tematem pracy domowej jest zapoznanie się z minimalizacją funkcji metodą gradientową. W tym celu wybrałem funkcję $f(x,y) = 3x^2 - 2xy + 3y^2 - 2x - 4y$. Poszukiwanie minimum funkcji zacząłem od wybrania losowego punktu startowego X_0 z zakresu $-100 \leq x \leq 100$ oraz $-100 \leq y \leq 100$.



Rysunek 1: Wykres funkcji $f(x,y)$

2 Zadanie

Do wyznaczenia minimum funkcji metodą gradientową trzeba użyć wzoru:

$$X_{k+1} = X_k - \alpha_k \nabla f(X_k) \tag{1}$$

gdzie α_k jest stałą kroku, a $\nabla f(X_k)$ jest gradientem funkcji $f(x,y)$ w punkcie X_k.

Gradient funkcji $f(x,y)$ to inaczej wektor pochodnych cząstkowych:

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{2}$$

W przypadku wybranej funkcji gradient ma postać:

$$\nabla f(x,y) = \begin{pmatrix} 6x - 2y - 2 \\ -2x + 6y - 4 \end{pmatrix} \tag{3}$$

Dla tego przykładu stała kroku $\alpha_k = 0.1$ natomiast warunek zakończenia to ilość iteracji $k \geq 25$

3 Przykładowe obliczenia

Dla przykładowego punktu startowego $X_0 = (1, 1)$ wyznaczenie 5 kolejnych punktów wygląda następująco:

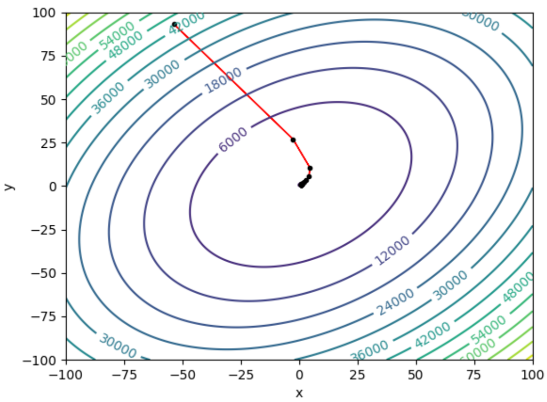
$$X_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 0.1 \begin{pmatrix} 6 - 2 - 2 \\ -2 + 6 - 4 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 1 \end{pmatrix}$$

$$X_2 = \begin{pmatrix} 0.8 \\ 1 \end{pmatrix} - 0.1 \begin{pmatrix} 6 \cdot 0.8 - 2 \cdot 1 - 2 \\ -2 \cdot 0.8 + 6 \cdot 1 - 4 \end{pmatrix} = \begin{pmatrix} 0.72 \\ 0.96 \end{pmatrix}$$

$$X_3 = \begin{pmatrix} 0.72 \\ 0.96 \end{pmatrix} - 0.1 \begin{pmatrix} 6 \cdot 0.72 - 2 \cdot 0.96 - 2 \\ -2 \cdot 0.72 + 6 \cdot 0.96 - 4 \end{pmatrix} = \begin{pmatrix} 0.68 \\ 0.928 \end{pmatrix}$$

$$X_4 = \begin{pmatrix} 0.68 \\ 0.928 \end{pmatrix} - 0.1 \begin{pmatrix} 6 \cdot 0.68 - 2 \cdot 0.928 - 2 \\ -2 \cdot 0.68 + 6 \cdot 0.928 - 4 \end{pmatrix} = \begin{pmatrix} 0.656 \\ 0.9072 \end{pmatrix}$$

$$X_5 = \begin{pmatrix} 0.656 \\ 0.9072 \end{pmatrix} - 0.1 \begin{pmatrix} 6 \cdot 0.656 - 2 \cdot 0.9072 - 2 \\ -2 \cdot 0.656 + 6 \cdot 0.9072 - 4 \end{pmatrix} = \begin{pmatrix} 0.644 \\ 0.8944 \end{pmatrix}$$



Rysunek 2: Rzut 2D funkcji $f(x, y)$ oraz kolejne kroki minimalizacji metodą gradientową dla wylosowanego punktu startowego

4 Kod programu

```
import numpy as np
import matplotlib.pyplot as plt


def gradient(x, y):
    grad = np.array([6 * x - 2 * y - 2, -2 * x + 6 * y - 4])
    return grad


def gradient_descent(gradient, w0, num_iter):
    w = w0
    w_hist = [w0]
    for i in range(num_iter):
        w = w - 0.1 * gradient(w[0], w[1])
        w_hist.append(w)
    return w_hist


x = np.linspace(-100, 100, 1000)
y = np.linspace(-100, 100, 1000)

X, Y = np.meshgrid(x, y)
```

```python
Z = 3 * X ** 2 - 2 * X * Y + 3 * Y ** 2 - 2 * X - 4 * Y

fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
ax.view_init(30, 45)
plt.savefig('function.png')
plt.show()

w_hist = gradient_descent(gradient, np.array([
    200 * np.random.rand() - 100,
    200 * np.random.rand() - 100,
]), 25)

cp = plt.contour(X, Y, Z, 15)
plt.clabel(cp, inline=True, fontsize=10)
plt.xlabel('x')
plt.ylabel('y')

for i in range(len(w_hist)):
    plt.plot(w_hist[i][0], w_hist[i][1], 'k.')
    if i < len(w_hist) - 1:
        plt.arrow(w_hist[i][0], w_hist[i][1], w_hist[i + 1][0] -
                  w_hist[i][0], w_hist[i + 1][1] - w_hist[i][1],
                  color='r', width=0.01)

plt.savefig('plot.png')
plt.show()
```