Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Wstęp do sztucznej inteligencji (laboratorium)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego nr 1, zadanie nr 1

Paweł Kutyła

Warszawa, 2023

Spis treści

1. Wstęp 2
1.1. Informacje elementarne 2
1.2. Wprowadzenie do działania algorytmu 2
2. Wizualizacja funkcji za pomocą narzędzia *c3d* 3
3. Gradient funkcji. 6
4. Wpływ punktu startowego na działanie programu 9

1. Wstęp

1.1. Informacje elementarne

Sprawozdanie przygotowywane w ramach projektów i ćwiczeń laboratoryjnych zostało opracowane w systemie LaTeX. Poniższe ćwiczenie zostało zrealizowane w języku Python `matlab`. Projekt realizowany w laboratorium numer 1 miał na celu wykazanie działania, implementację oraz analizę wyznaczonego odgórnie algorytmu, a jego implementacja znajduje się w pliku `pust_proj1.pdf`. Poniżej znajdują się krótkie opisy przybliżającę czytelnikowi treści zawarte w nieniejszym sprawozdaniu.

1.2. Wprowadzenie do działania algorytmu

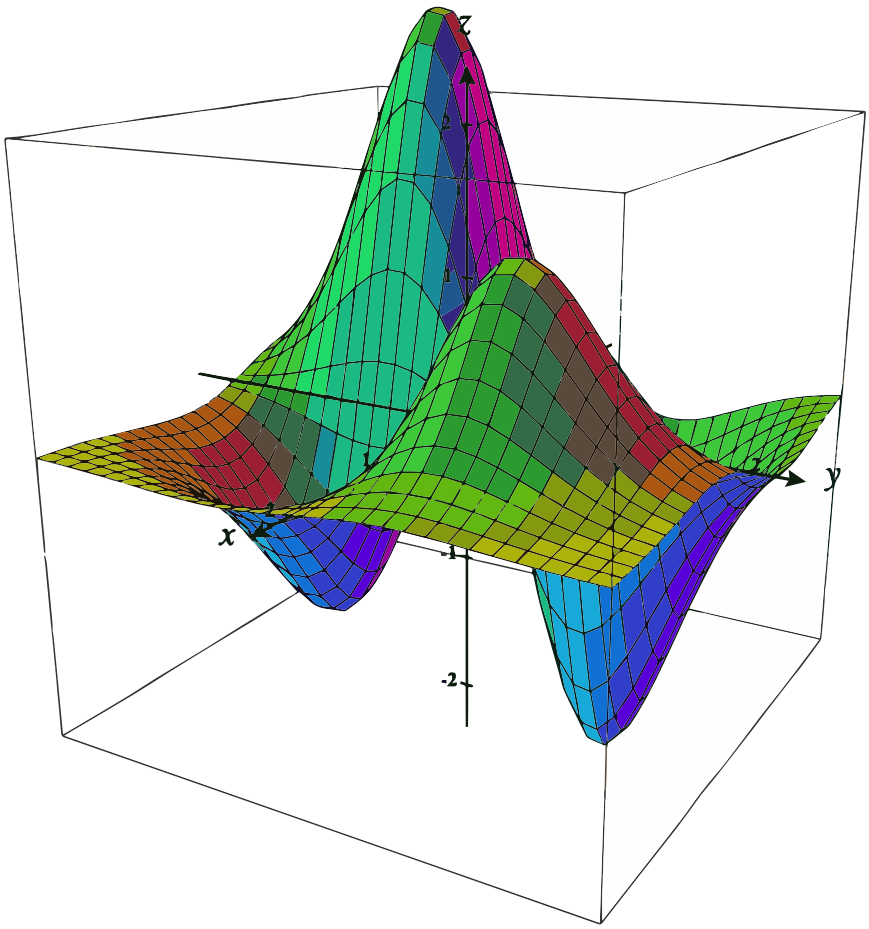
Algorytm *Steepest Gradient Descent* zwany w dalszej części sprawozdania SGD swoją jest metodą optymalizacyjną w analizie danych. Jego implementację jak i przykłady działania przedstawię w dalszej części sprawozdania.

2. Wizualizacja funkcji za pomocą narzędzia *c3d*

Pierwszym krokiem podjętym w ramach realizacji zadania laboratyjnego będzie wizualizacja funkcji wyrażonej w postaci:

$$f(x,y) = \frac{10xy}{e^{x^2+0.5x+y^2}} \tag{2.1}$$

Początkowa wizualizacja funkcji za pomocą narzędzia *c3d*. Następnie porównuje wyżej umiesz-



Rys. 2.1. Wizualizacja c3d

czoną funkcję z kodem wprowadzonym w języku Python. Poniżej zamieszczona jest wizualizacja w tak sposób utworzonej funkcji.

Dzięki porównaniu tych funkcji w prosty sposób możemy dokonać porównania, a tym samym ogólnej walidacji poprawności kodu zaimplementowanego na potrzeby laboratorium.

Równanie opisujące wyznaczanie kolejnych parametrów x i y zapisane za pomocą pseudokodu ma postać:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} := \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial}{\partial w_1}(w_1 + w_2 x_i - y_i)^2 \\ \frac{\partial}{\partial w_2}(w_1 + w_2 x_i - y_i)^2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} 2(w_1 + w_2 x_i - y_i) \\ 2x_i(w_1 + w_2 x_i - y_i) \end{bmatrix} \tag{2.2}$$

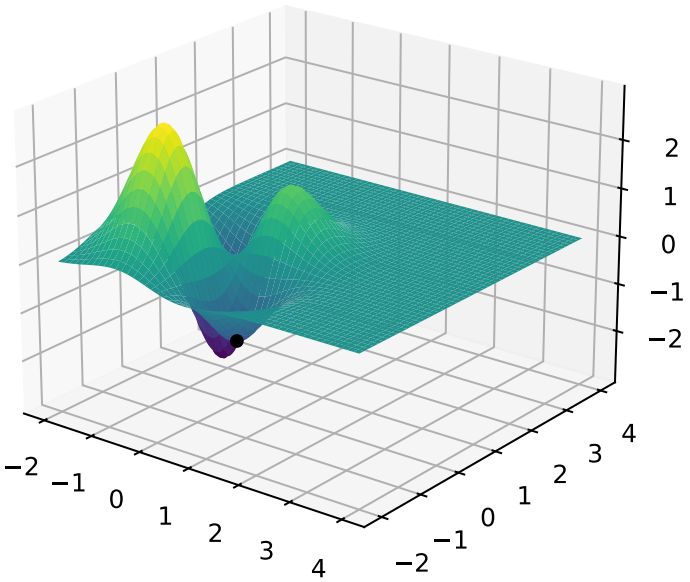
oraz

$$w \leftarrow w - \eta \nabla J(w) \tag{2.3}$$

W przypadku naszego problemu parametr wartości parametrów, których wielkości ulegają zmianie w trakcie iteracji są przedstawione są za pomocą symboli x i y. Pierwszym krokiem na drodze do zrealizowania zadania będzie zobrazowanie równań opisujących funkcję gradientu, a następnie wytłumaczenie jego znaczenia w procesie implementacji algorytmu SGD. Wzory opisujące gradient funkcji mają postać:

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial}{\partial x} f(x,y) \\ \frac{\partial}{\partial y} f(x,y) \end{bmatrix} = \begin{bmatrix} (-20x^2 - 5x + 10)y \cdot e^{(-x^2 - 0.5x - y^2)} \\ -10x(2y^2 - 1) \cdot e^{(-x^2 - 0.5x - y^2)} \end{bmatrix} \tag{2.4}$$

Całkowite wyprowadzenie wzoru zostalo przedstawione odręcznie poniżej:
Na samym końcu przytoczona zostanie definicja gradientu. Gradient jest wektorem składającym się z pochodnych cząstkowych funkcji. W naszym programie, dzięki wartości gradientu program poszukuje takich wartości parametrów idąc w kierunku gradientu lub w przeciwnym, co pozwala nam w ostateczności znaleźc takie wartości parametrów dla których wartość gradientu jest równa lub bardzo zbliżona 0 (funkcja ugrzęźnie na wypłaszczeniu)



Rys. 2.2. Zaimplementowana funkcja

3. Gradient funkcji.

$$f(x,y) = \frac{10xy}{e^{(x^2+0{,}5x+y^2)}}$$

$$f(x,y) = 10xy \cdot e^{(-x^2-0{,}5x-y^2)}$$

~~f(x,y)=~~

$$\frac{\partial f}{\partial x} = 10y\left(e^{(-x^2-0{,}5x-y^2)}\right) + (-2x-0{,}5)\cdot\left(e^{(-x^2-0{,}5x-y^2)}\right)$$

$$\frac{\partial f}{\partial x} = -10y \cdot e^{-x^2-0{,}5x-y^2}\left(2x^2+0{,}5x-1\right)$$

$$\frac{\partial f}{\partial x} = -20y\left(x^2-5x+10\right) \cdot e^{(-x^2-0{,}5x-y^2)}$$

$$\frac{\partial f}{\partial y} = 10x\left(e^{(-x^2-0{,}5x-y^2)}\right) + (-2y)\left(e^{(-x^2-0{,}5x-y^2)}\right)\cdot y$$

$$\frac{\partial f}{\partial y} = 10x \cdot e^{-x^2-0{,}5x-y^2}\left(1-2y^2\right)$$

$$\frac{\partial f}{\partial y} = -10x \cdot \left(2y^2-1\right) \cdot e^{(-x^2-0{,}5x-y^2)}$$

W kolejnym kroku dokonuje wyznaczenia gradientu funkcji. Dla przypomnienia kod funkcji realizowanej w ramach zadania przytoczono poniżej:

$$f(x,y) = \frac{10xy}{e^{x^2+0.5x+y^2}} \tag{3.1}$$

Równanie opisujące wyznaczanie kolejnych parametrów x i y zapisane za pomocą pseudokodu ma postać:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} := \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial}{\partial w_1}(w_1 + w_2 x_i - y_i)^2 \\ \frac{\partial}{\partial w_2}(w_1 + w_2 x_i - y_i)^2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} 2(w_1 + w_2 x_i - y_i) \\ 2x_i(w_1 + w_2 x_i - y_i) \end{bmatrix} \tag{3.2}$$

oraz

$$w \leftarrow w - \eta \nabla J(w) \tag{3.3}$$

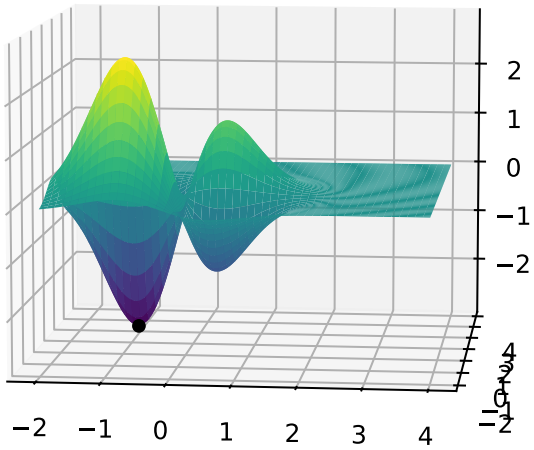
W przypadku naszego problemu parametr wartości parametrów, których wielkości ulegają zmianie w trakcie iteracji są przedstawione są za pomocą symboli x i y. Pierwszym krokiem na drodze do zrealizowania zadania będzie zobrazowanie równań opisujących funkcję gradientu, a następnie wytłumaczenie jego znaczenia w procesie implementacji algorytmu SGD. Wzory opisujące gradient funkcji mają postać:

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial}{\partial x} f(x,y) \\ \frac{\partial}{\partial y} f(x,y) \end{bmatrix} = \begin{bmatrix} (-20x^2 - 5x + 10)y \cdot e^{(-x^2-0.5x-y^2)} \\ -10x(2y^2 - 1) \cdot e^{(-x^2-0.5x-y^2)} \end{bmatrix} \tag{3.4}$$

Całkowite wyprowadzenie wzoru zostalo przedstawione odręcznie poniżej:
Na samym końcu przytoczona zostanie definicja gradientu. Gradient jest wektorem składającym się z pochodnych cząstkowych funkcji. W naszym programie, dzięki wartości gradientu program poszukuje takich wartości parametrów idąc w kierunku gradientu lub w przeciwnym, co pozwala nam w ostateczności znaleźc takie wartości parametrów dla których wartość gradientu jest równa lub bardzo zbliżona 0 (funkcja ugrzęźnie na wypłaszczeniu)

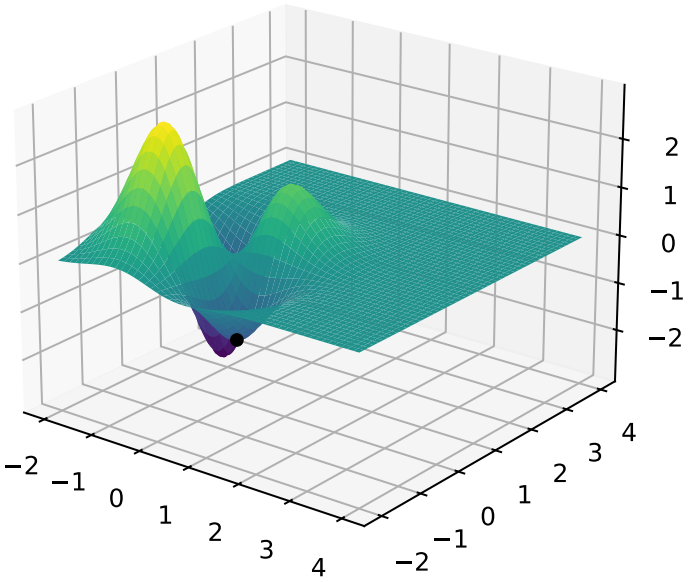
4. Wpływ punktu startowego na działanie programu

Wybór punktów startowych ma fundamentalny wpływ na dobór parametrów. Zbytnio oddalone punkty startowe mogą spowodować ugrzęźnięcie algorytmu SGD w miejscach, gdzie funkcja jest bardzo płaska. Zbyt bliskie punkty ekstremów skolei mogą spowodować lądować w zawsze do tego samego ekstremum, co może spowodować znaczne ograniczenie użyteczności algorytmu. Zwalczeniem tego problemu może być losowanie punktu startowego, jednak zbyt mały obszar wciąż spowoduje losowanie parametrów w wąskim przedziale, a tym samym ograniczenie funkcjonalności losowania.

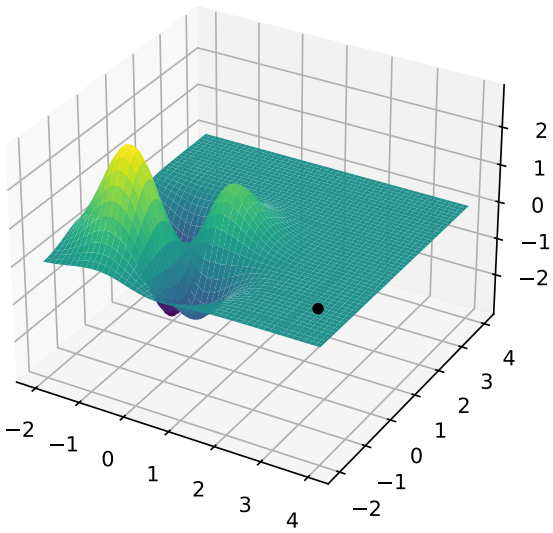


Rys. 4.1. Znalezione rozwiązanie :$(-0.84306831, 0.70710687)$ dla punktów startowych:$(-0.38, 0.92)$.

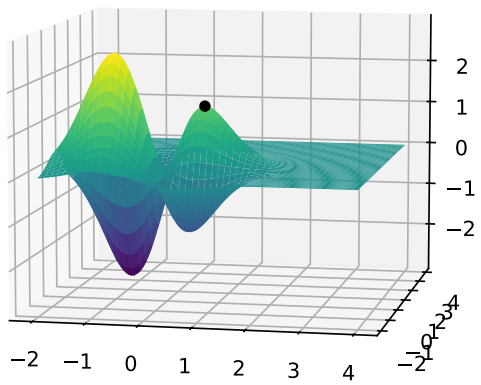
Podobnie możemy postąpić w przypadku szukania maximów: Powyższe przykłady dokładnie obrazują tezę wspomnianą wcześniej. Dla punktów startowych znajdujących się daleko od tych ekstremów funkcja utyka na wypłaszczeniach.



Rys. 4.2. Znalezione rozwiązanie :$(0.59307092, -0.70710326)$ dla punktów startowych:$(0.98, -0.41)$.



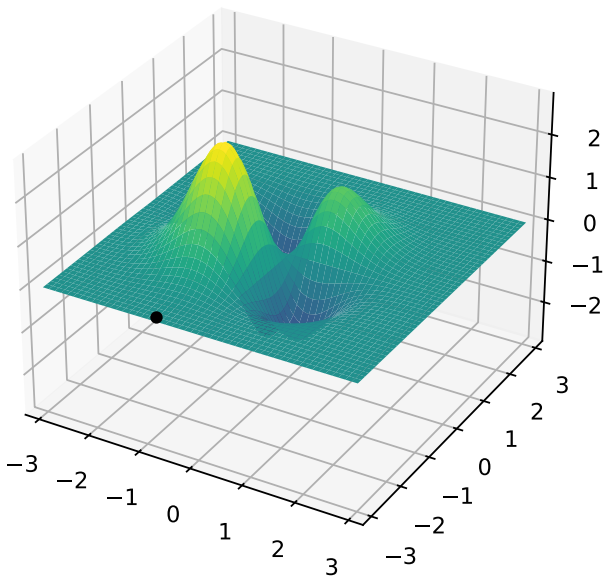
Rys. 4.3. Znalezione rozwiązanie:$(3.30516558, -0.84793013)$ dla punktów startowych:$(2.47, -0.79)$.



Rys. 4.4. Znalezione rozwiązanie $(0.59307031, 0.70711024)$ dla punktów startowych:$(-0.16, 1.46)$.

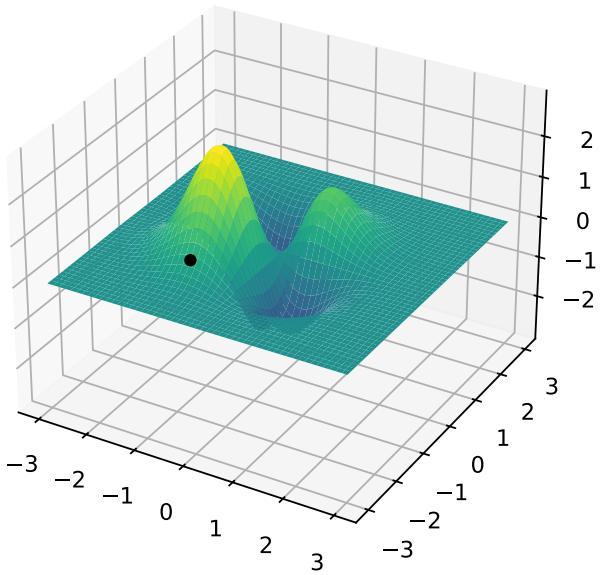
1. Wpływ parametru *learning rate* na optymalizację działania

Dobór odpowiednich wartości parameteru *learning rate* ma fundamentalny wpływ na działanie algorytmu. Zbyt mała wartośc tej zmiennej może spowodować, że zaimplementowany algorytm SGD będzie miał tendencję do grzęźnięcia w miejscach, gdzie funkcja będzie wystarczająco płaska. Kolejnym powodem, dla którego mała wartość parametru learning rate spowoduje bardzo wolne dążenie do naszych ekstremów, w efekcie zmuszając do wykorzystaniu bardzo wielu operacji w celu osiągnięcia pożądanych parametrów. Zbyt duża wartość tego parametru może spowodować skoki wobec rozwiązania jednak nigdy go nie osiągając, a w skrajnych przypadkach wypaczyć działanie algorytmu. Na potrzeby testowe ustalę stałą wartośc nasiona losowości *seed*, dla uzyskania losowych, ale powtarzalnych wyników. Zadanie rozważam dla punktów startowych $(-2.3300000000000143, -2.2200000000000166)$

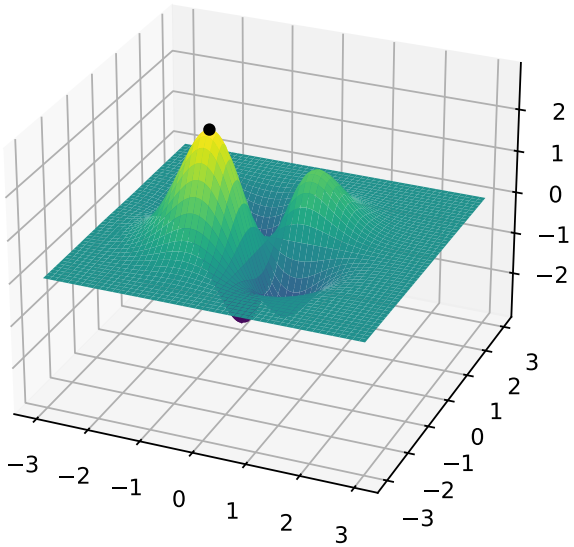


Rys. 1.1. Znalezione rozwiązanie :$(-0.84307033, -2.88370389)$ dla *learning rate* $= 0.5$

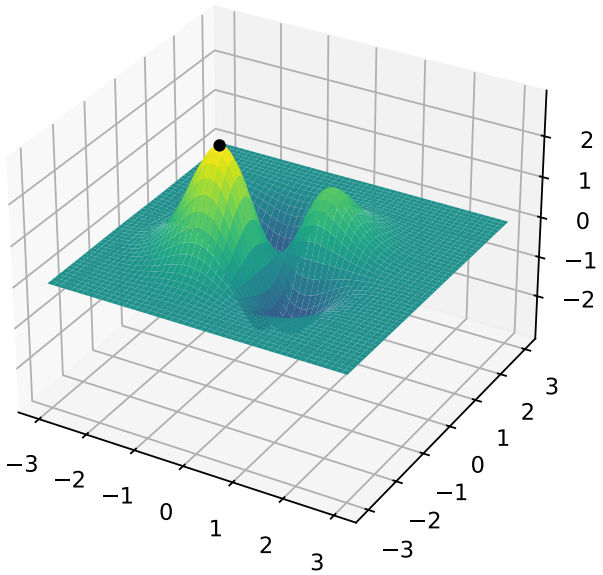
Jak możemy zauważyć podobne wyniku otrzymujemy zarówno dla dla parametru *learning rate* $= 0.1$ jak i *learning rate* $= 0.05$, jednak dla większej z tych wartości mamy mniej iteracji. Jednak dla większej dokładności, która jest kluczem w kolejnym zadaniu pozostanę przy tej mniejszej wartości.



Rys. 1.2. Znalezione rozwiązanie :$(-0.84307033, -1.72856491)$ dla *learning rate* $= 0.3$



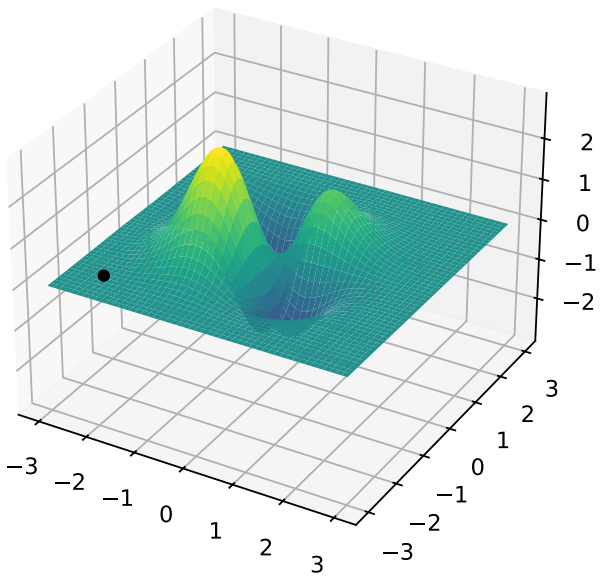
Rys. 1.3. Znalezione rozwiązanie :$(-0.84316976, -0.70710288)$ dla *learning rate* $= 0.1$



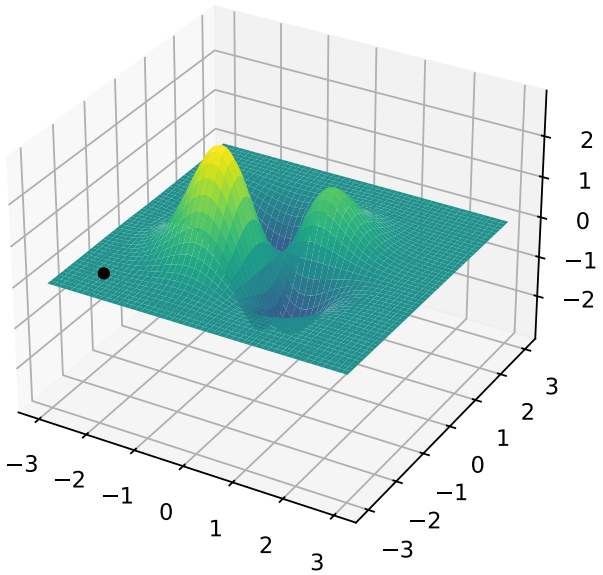
Rys. 1.4. Znalezione rozwiązanie :$(-0.84320853, -0.70712394)$ dla *learning rate* $= 0.05$

1. Analiza wpływu parametrów modelu na dokładność znajdowanych rozwiązań

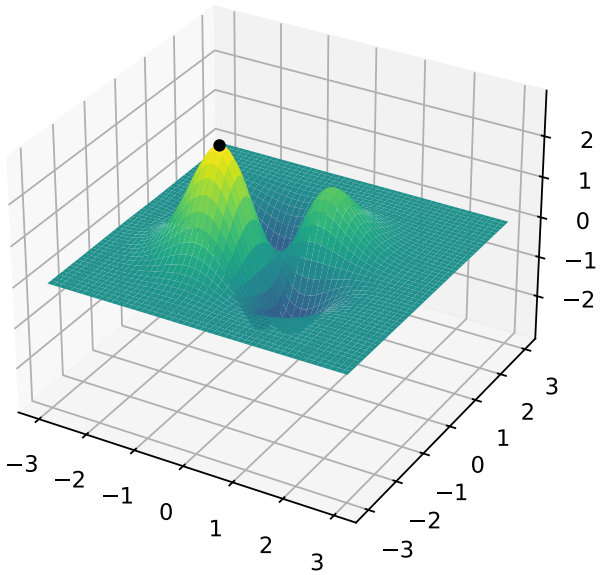
Ostatnim krokiem w wyznaczeniu poprawnego rozwiązania będzie analiza wpływu parametrów na dokładność znajdowanego rozwiązania. Pierwszym z nich, którego wpływ jest najbardziej oczywisty będzie δ. W najprostszym ujęciu będzie możemy to interpetować jako różnicę pomiędzy obecnyą wartością gradientu mutliplikowaną o *learning rate*. Gdy wartość ta wystarczająco mała program kończy działanie i zwraca wybrany punkt. Czym mniejsza wartośc tego δ tym dokładniejsza jest wartość otrzymywana w wyniku działania programu, jednak zwiększa to ilość iteracji. Podobnie wartośc parametrów *learning rate* ma pośredni wpływ na dokładnośc znajdowanych punktów, gdyż nieodpowiednio dobrany spowoduje znacznie pogorszoną skuteczność działania algorytmu podobnie jak w przypadku punktu początkowego. Zadanie rozważam dla punktów startowych: $(-2.3300000000000143, -2.2200000000000166)$



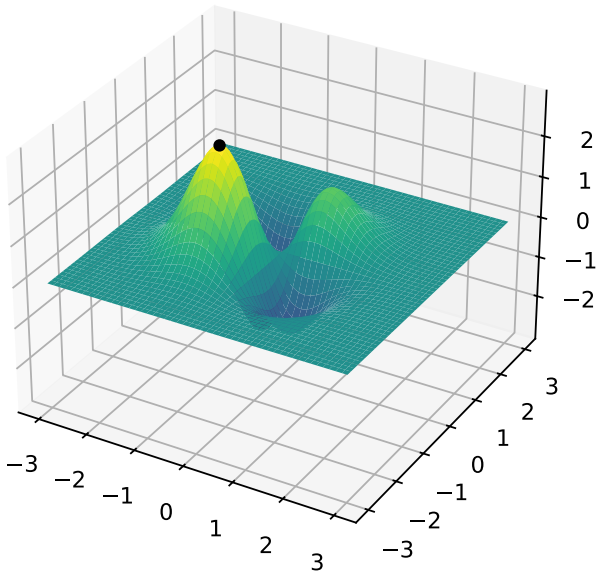
Rys. 1.1. Znalezione rozwiązanie :$(-2.33, -2.22)$ dla $\Delta = 0.01$



Rys. 1.2. Znalezione rozwiązanie :$(-2.33, -2.22)$ dla $\Delta = 0.005$



Rys. 1.3. Znalezione rozwiązanie :$(-0.84470838, -0.70749491)$ dla $\Delta = 0.001$



Rys. 1.4. Znalezione rozwiązanie :$(-0.84395346, -0.70728479)$ dla $\Delta = 0.0005$