

SPRAWOZDANIE

z ćwiczenia nr 4

„Zagadnienie przeszukiwania i podstawowe podejścia do niego”

Przedmiot: Wprowadzenie do Sztucznej Inteligencji

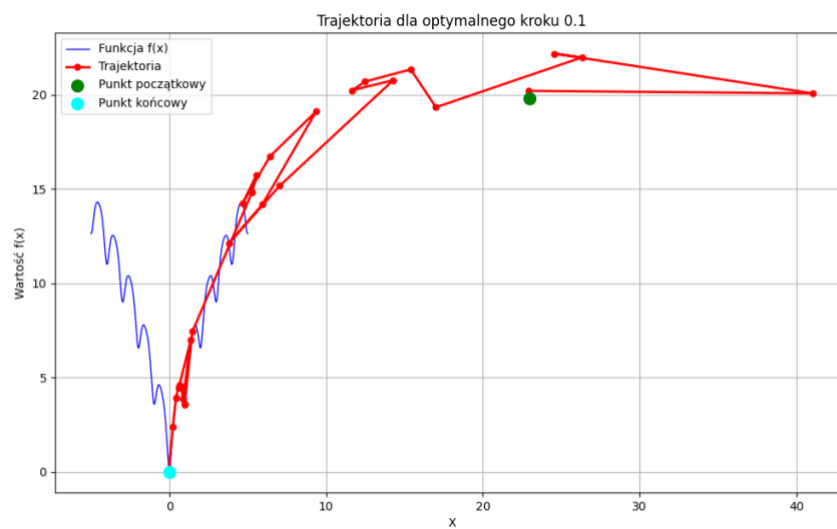
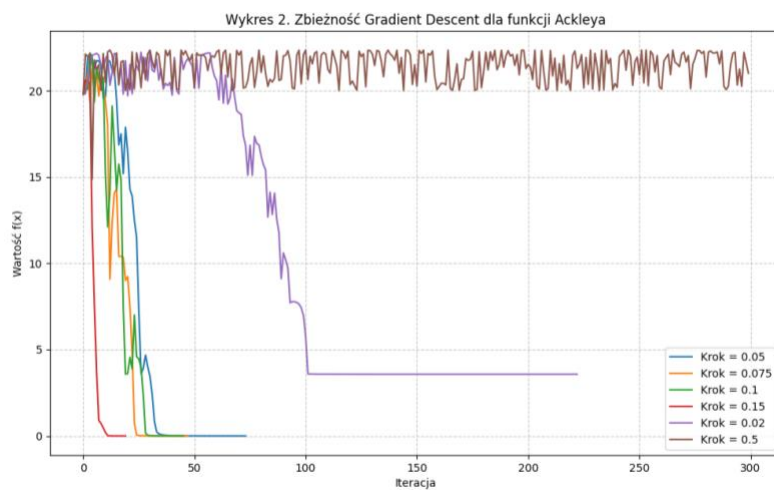
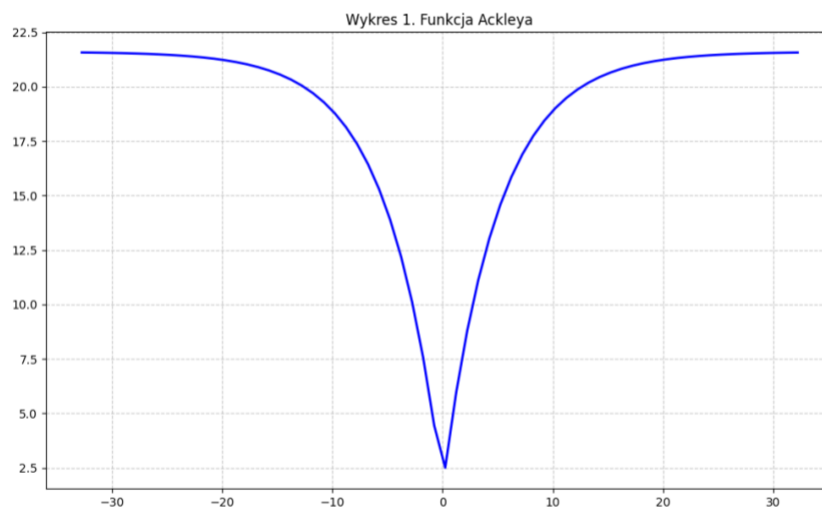
Imię i nazwisko: Viktoriia Nowotka

W trakcie ćwiczenia realizowano algorytm gradientu prostego, co polega na optymalizacji funkcji iteracyjnie. Podczas jednej iteracji wyliczano wartość gradientu funkcji celu i mnożono na współczynnik kroku (learning rate). W związku z tym, że funkcje trzeba minimalizować kolejny punkt uzyskujemy po wykonaniu różnicy.

Po realizacji algorytmu zostało zbadany wpływ wielkości kroku na działanie algorytmu.

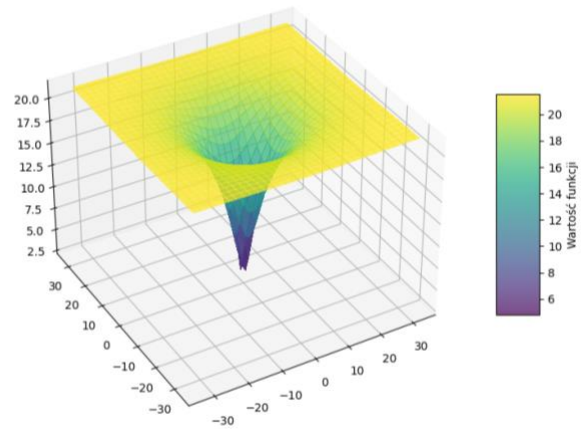
Początkowa lista wielkości kroku była następująca: $[0.005, 0.01, 0.015, 0.02, 0.05, 0.075, 0.09, 0.1, 0.15, 0.2, 0.25]$. Po testowym działaniu algorytmu zauważono, że dla funkcji Ackleja jednowymiarowej lepiej sprawdzają się mniejsze kroki, a dla dwuwymiarowej – większe. Dlatego w kolejnych wykresach dla każdej z funkcji rozpatrywano tylko lista kroków bardziej odpowiednich.

Funkcji Ackleya jednowymiarowa.

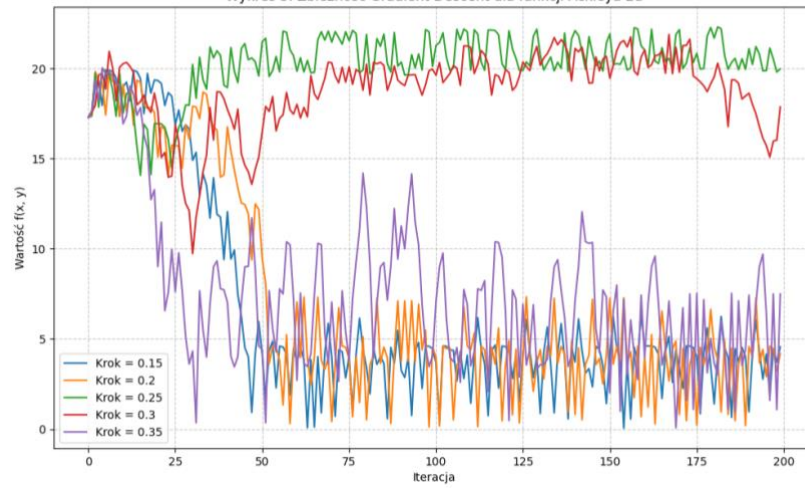


Funkcji Ackleya dwuwymiarowa.

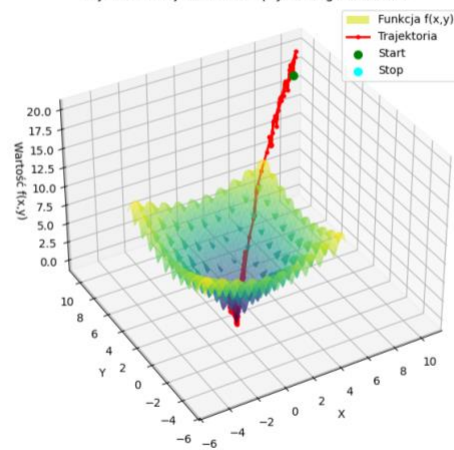
Wykres 4. Funkcja Ackleya 3D

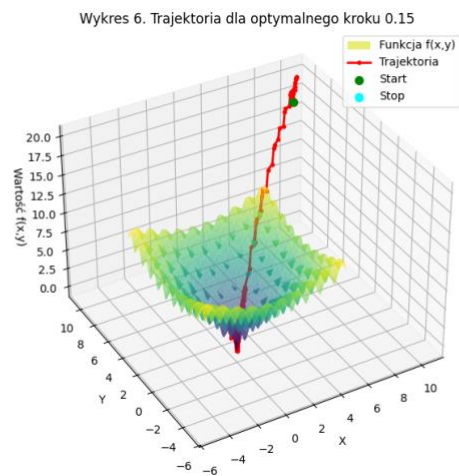
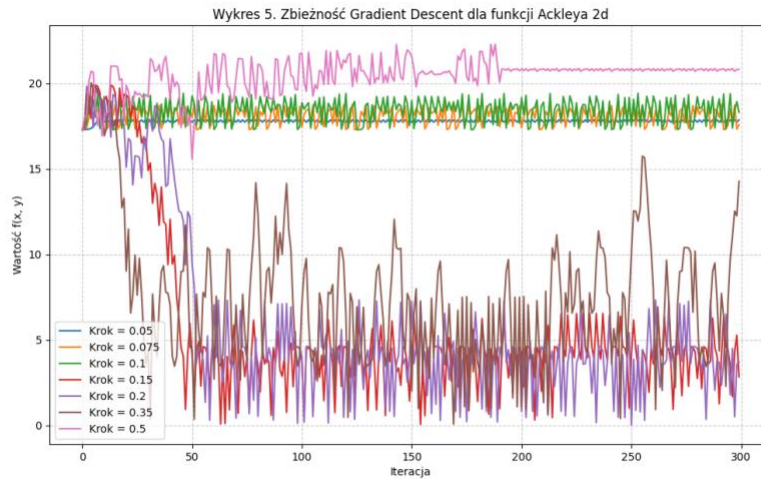


Wykres 5. Zbieżność Gradient Descent dla funkcji Ackleya 2d

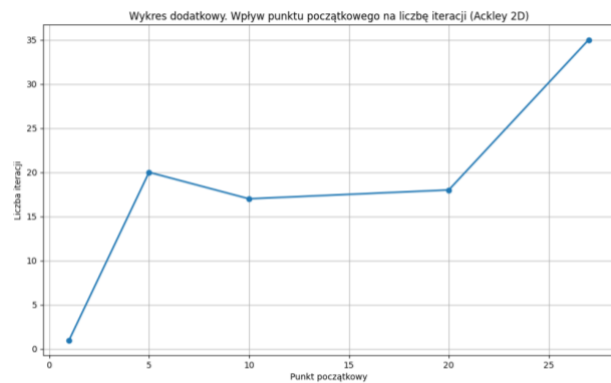
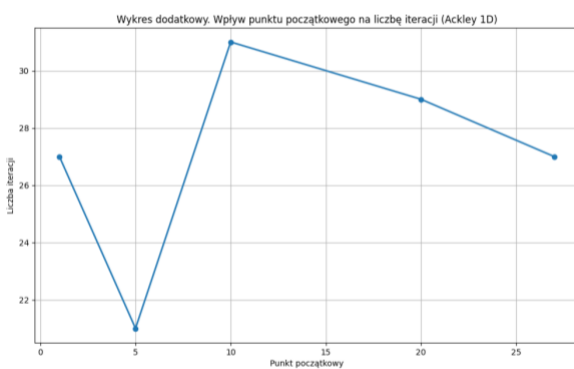


Wykres 6. Trajektoria dla optymalnego kroku 0.2





Wpływ punktu początkowego na liczbę iteracji dla znalezienia optimum



Wnioski

W trakcie wykonania ćwiczenia nr 4 zaimplementowano algorytm gradientu prostego. Zostało zbadano wpływ wielkości kroku dla optymalizacji funkcji Ackleya jedno i dwu wymiarowa.

Dla maksymalnej liczby 200-300 iteracji dla jednowymiarowej funkcji kroki w przedziale $[0,05; 0.15]$ są bardziej skuteczne. Większe kroki powodują chaotyczne krążenie w przestrzeni, wtedy jak mniejsze zbyt skupiają się na lokalnych minimum.

Dla dwuwymiarowej funkcji zakres kroku jest nieco innym: $[0,15; 0,35]$.

Na dwuwymiarowej funkcji celu bardziej jest widoczna zależność odległości punktu początkowego do optimum do ilości iteracji w algorytmie. W funkcji jednowymiarowej tej zależności nie obserwujemy, możliwe z powodu tego, że łatwej jest krążyć w lokalnym minimum, zanim wyskoczymy z niego.