Obraz zawierający tekst, Czcionka, biały, czarne i białe

Opis wygenerowany automatycznie

Sprawozdanie

Wprowadzenie do sztucznej inteligencji

Ćwiczenie nr. 2

Mikołaj Bańkowski

Numer albumu 310408

prowadzący

Grzegorz Rypeść

Warszawa 2024

Spis treści

[1. Temat ćwiczenia 3](#_Toc161697564)

[2. Minima i maksima funkcji 4](#_Toc161697565)

[3. Co jest przestrzenią poszukiwań 11](#_Toc161697566)

[4. Czy jest potrzebny gradient? 11](#_Toc161697567)

[5. Jak odchylenie standardowe w szumie mutacji (σ) wpływa na wyniki (wartości 0.01, 0.1, 1, 10)? 12](#_Toc161697568)

[6. Jak liczba rodziców μ i dzieci λ wpływa na proces optymalizacji (wartości {1, 1}, {1, 16}, {16, 1}, {16,16}, {128, 512})? 16](#_Toc161697569)

[7. Proszę punkt startowy ES (dla μ , λ=(128 ,512) , σ=0.1) oraz algorytmu SGD z poprzedniego zadania ustawić na punkt (10,10) i porównać wyniki. 22](#_Toc161697570)

[7.1 Co i dlaczego można zaobserwować? 22](#_Toc161697571)

[7.2 Kiedy ES jest lepszym rozwiązaniem niż SGD? 23](#_Toc161697572)

[7.3 Który algorytm optymalizacji jest bardziej złożony obliczeniowo? 23](#_Toc161697573)

# 1. Temat ćwiczenia

Proszę znaleźć minima oraz maksima funkcji wykorzystując strategię ewolucyjną μ+λ (ES, ang. Evolution Strategy). Strategia ma dokonywać mutacji osobnika za pomocą dodania do niego szumu Gaussowskiego. Krzyżowanie ma być dokonane za pomocą interpolacji, tzn. wynikiem krzyżowania osobników o1 i o2 jest osobnik a o1+(1−a)o2 , gdzie a jest zmienną losową z rozkładu jednostajnego na przedziale [0;1] . Proszę samemu zaproponować strategię selekcji i eliminacji. Osobnik jest tutaj dwuwymiarowym wektorem, niech będzie on reprezentowany przez numpy array, albo torch tensor.

Odpowiedzieć na pytania:

1. Co jest przestrzenią poszukiwań?
2. Czy jest potrzebny gradient?
3. Jak odchylenie standardowe w szumie mutacji (σ) wpływa na wyniki (wartości 0.01, 0.1, 1, 10)?
4. Jak liczba rodziców μ i dzieci λ wpływa na proces optymalizacji (wartości {1, 1}, {1, 16}, {16, 1}, {16,16}, {128, 512})?

Proszę punkt startowy ES (dla μ , λ=(128 ,512) , σ=0.1) oraz algorytmu

SGD z poprzedniego zadania ustawić na punkt (10,10) i porównać wyniki.

Odpowiedzieć na pytania:

1. Co i dlaczego można zaobserwować?
2. Kiedy ES jest lepszym rozwiązaniem niż SGD?
3. Który algorytm optymalizacji jest bardziej złożony obliczeniowo?

# 2. Minima i maksima funkcji

Do wstępnej analizy badanej funkcji i jej wizualizacji, wykorzystano narzędzie [WolframAplha](https://www.wolframalpha.com/input?i2d=true&i=Divide%5B9xy%2CPower%5Be%2CPower%5Bx%2C2%5D%2B0.5x%2BPower%5By%2C2%5D%5D%5D)

Obraz zawierający diagram, origami, design, sztuka

Opis wygenerowany automatycznie przy średnim poziomie pewności

Obraz zawierający zrzut ekranu, krąg, Wielobarwność

Opis wygenerowany automatycznie

**Na podstawie analizy poniższych wykresów funkcji, można zauważyć, że funkcja posiada dwa minima i dwa maksima.**

Minima funkcji znajdują się w II i IV ćwiartce układu współrzędnych. Są to kolejno

min{f(x,y)} ≈ -2.43687 dla (x, y) ≈ (-0.84307, 0.707107)

min{f(x,y)} ≈ -1.19715 dla (x, y) ≈ (0.59307, -0.707107)

Maxima funkcji znajdują się w I i III ćwiartce układu współrzędnych. Są to kolejno

max{f(x,y)} ≈ 1.19715 dla (x, y) ≈ (0.59307, 0.707107)

max{f(x,y)} ≈ 2.43687 dla (x, y) ≈ (-0.84307, -0.707107)

**Minima funkcji znalezione za pomocą algorytm ewolucyjnego ES(μ+λ)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 10 | 100 | 0.1 | 200 | x = -0.8430703308792917 y = 0.7071067885974828 | -2.4368681615230177 |

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, diagram, zrzut ekranu, krąg

Opis wygenerowany automatycznie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 10 | 100 | 0.1 | 200 | x = 0.5930703224398627 y = -0.7071067812684363 | -1.197146871901824 |

Obraz zawierający tekst, zrzut ekranu, numer, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, diagram, zrzut ekranu, krąg

Opis wygenerowany automatycznie

**Maksima funkcji znalezione za pomocą algorytm ewolucyjnego ES(μ+λ)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione maksimum** |
| 10 | 100 | 0.1 | 200 | x = -0.8430703305330756 y = 0.7071068216406652 | 2.4368681615230097 |

Obraz zawierający tekst, diagram, zrzut ekranu, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, numer, linia

Opis wygenerowany automatycznie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione maksimum** |
| 10 | 100 | 0.1 | 200 | x = -0.8430703308792917 y = 0.7071067885974828 | -2.4368681615230177 |

Obraz zawierający tekst, diagram, krąg, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, numer

Opis wygenerowany automatycznie

Podczas szukania drugiego maksimum lokalnego – będąc bardzo blisko jednego z maksimum lokalnego algorytm ‘przeskoczył’ do drugiego maksimum globalnego. Można wywnioskować, że algorytm nie daje nam gwarancji, że znajdziemy wszystkie minima/maksima, ale dąży do znalezienia najlepszego optima.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione maksimum** |
| 10 | 100 | 0.1 | 200 | x = -0.8430703201291451 y = 0.7071067702216859 | 2.4368681615230168 |

Obraz zawierający tekst, diagram, zrzut ekranu, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

# 3. Co jest przestrzenią poszukiwań

# Przestrzeń poszukiwań w algorytmie λ + μ to obszar, w którym algorytm przeszukuje potencjalne rozwiązania. W przypadku algorytmu ewolucji strategii (ES) λ + μ przestrzeń poszukiwań obejmuje zakres możliwych wartości zmiennych decyzyjnych, które mogą być oceniane przez funkcję celu.

# W algorytmie λ + μ definiuje się dwie liczby: λ i μ. Parametr λ określa liczbę potomków, które są tworzone w każdym pokoleniu poprzez krzyżowanie i mutację. Parametr μ określa liczbę rodziców, którzy są wybierani z populacji i służą jako rodzice do generowania potomstwa.

# Przestrzeń poszukiwań w algorytmie λ + μ jest zazwyczaj definiowana przez ograniczenia problemu optymalizacyjnego, w tym zakresy wartości zmiennych decyzyjnych.

# W przypadku mojego kodu, przestrzenią poszukiwań jest obszar zdefiniowany przez poniższy fragment kodu. Z tego obszaru są

# Obraz zawierający tekst, zrzut ekranu, Czcionka Opis wygenerowany automatycznie

# Oznacza to, że przestrzeń poszukiwań dla zmiennych x i y mieści się w zakresie od -20.0 do 20.0. Algorytm będzie próbował znaleźć wartości x i y w tym przedziale, które minimalizują lub maksymalizują funkcję celu, w zależności od tego, czy szukamy minimum czy maksimum.

# Czy jest potrzebny gradient?

# W algorytmie ewolucji strategii ES(μ + λ) gradient nie jest potrzebny. Algorytm ten jest oparty na ewolucji populacji, w której generowane są nowe rozwiązania poprzez krzyżowanie, mutację i selekcję, a następnie oceniane są one na podstawie funkcji celu.

# ES(μ + λ) jest często wykorzystywany w problemach optymalizacji, w których gradient może być trudny do obliczenia lub nie jest dostępny. Zamiast tego, algorytm ten polega na przeszukiwaniu przestrzeni poszukiwań poprzez ewolucję populacji rozwiązań w kierunku optymalizacji funkcji celu. Dzięki temu, ES(μ + λ) może być skuteczny w przypadku funkcji celu nieliniowych, nieregularnych lub zawierających wiele lokalnych ekstremów, gdzie metody oparte na gradientach mogą nie być wystarczająco skuteczne.

# 5. Jak odchylenie standardowe w szumie mutacji (σ) wpływa na wyniki (wartości 0.01, 0.1, 1, 10)?

# Przestrzeń przeszukiwań

# 

**Minima funkcji znalezione za pomocą algorytm ewolucyjnego ES(μ+λ)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 10 | 100 | 0.01 | 2000 | x = 921.8740423793206 y = -8708.683560421592 | 0.0 |

# Obraz zawierający tekst, zrzut ekranu, linia, Wykres Opis wygenerowany automatycznie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 10 | 100 | 0.1 | 2000 | x = 7025.51025670431 y = -5897.360235172188 | 0.0 |

# Obraz zawierający tekst, zrzut ekranu, linia, diagram Opis wygenerowany automatycznie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 10 | 100 | 1 | 2000 | x = -0.8430703388621299 y = -0.7071067807915719 | -2.436868161523017 |

# Obraz zawierający tekst, zrzut ekranu, diagram, linia Opis wygenerowany automatycznie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 10 | 100 | 10 | 2000 | x = -0.843070332338862 y = -0.7071067785899547 | -2.436868161523017 |

# Obraz zawierający tekst, zrzut ekranu, diagram, linia Opis wygenerowany automatycznie

# Przestrzeń przeszukiwań

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Znalezione minimum** |
| 10 | 100 | 0.1 | 50000 | 2.4368681615230168 |

# Obraz zawierający tekst, zrzut ekranu, diagram, linia Opis wygenerowany automatycznie

# Silna mutacja to mała szansa trafienia w interesujący obszar, słaba mutacja to mały krok

# Zasięg mutacji zwiększa różnorodność populacji, a co za tym idzie zmniejsza ryzyko wpadnięcia w minimum lokalne oraz zwiększa szybkość znalezienia minimum dla funkcji. Z drugiej stromy, gdy jego wartość jest zbyt duża może utrudnić znalezienie minimum funkcji oraz zwiększyć liczbę iteracji potrzebnych do jego znalezienia

# Jak liczba rodziców μ i dzieci λ wpływa na proces optymalizacji (wartości {1, 1}, {1, 16}, {16, 1}, {16,16}, {128, 512})?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 1 | 1 | 0.1 | 200 | x = -0.7124067452599503 y = -0.8536002220308432 | -2.4362729706189845 |

# Obraz zawierający tekst, zrzut ekranu, diagram, linia Opis wygenerowany automatycznie

# Znaleziono jedno z minimum po wielu uruchomieniach algorytmu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 1 | 16 | 0.1 | 200 | x = -0.8430703235228891 y = -0.7071067889332481 | -2.436868161523017 |

# Obraz zawierający tekst, zrzut ekranu, linia, numer Opis wygenerowany automatycznie

# Znaleziono jedno z minimum po bardzo dużej ilości ponownego uruchomienia algorytmu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 16 | 1 | 0.1 | 200 | x = -0.8333078712098345 y = -0.576516880709629 | -2.349028928488879 |

# Obraz zawierający tekst, zrzut ekranu, diagram, Wykres Opis wygenerowany automatycznie

# Znaleziono jedno z minimum ale bardzo bardzo ciężko to szło, ciężko było znaleźć jakiekolwiek minimum, trzeba było wielokrotnie uruchomić algorytm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 16 | 16 | 0.1 | 200 | x = -0.8430703351073741  y = -0.7071067802526086 | -2.436868161523017 |

# Obraz zawierający tekst, zrzut ekranu, diagram, numer Opis wygenerowany automatycznie

# W tym przypadku praktycznie za każdym uruchomieniem algorytmu udawało się znaleźć minimum jedno z dwóch minimów

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **λ** | **μ** | **σ** | **Pokolenia** | **Punkt końcowy** | **Znalezione minimum** |
| 128 | 512 | 0.1 | 200 | x = 921.8740423793206 y = -8708.683560421592 | 0.0 |

# Obraz zawierający tekst, zrzut ekranu, diagram, linia Opis wygenerowany automatycznie

# Bezproblemowo za udawało się znaleźć minimum

# {1, 1}: Ta kombinacja może być niewystarczająca do skutecznej eksploracji przestrzeni rozwiązań, co może prowadzić do zatrzymania w lokalnych minimach lub nie znalezienia ich nawet

# {1, 16}: Ta kombinacja może prowadzić do bardziej intensywnej eksploatacji okolicy rodzica, co może pomóc w szybszym znalezieniu lokalnego minimum, ale ryzyko utknięcia w nim jest większe.

# {16, 1}: Ta kombinacja umożliwia lepszą eksplorację przestrzeni rozwiązań, ale może prowadzić do większego rozproszenia się wysiłków optymalizacyjnych.

# {16, 16}. Ta kombinacja zapewnia to równowagę między eksploracją a eksploatacją, co może pomóc w efektywnym przeszukiwaniu przestrzeni rozwiązań i unikaniu zatrzymywania się w lokalnych minimach.

# {128, 512}: Ta kombinacja ma dużą liczbę rodziców i dzieci. Wysokie wartości μ i λ mogą prowadzić do bardzo szerokiej eksploracji przestrzeni rozwiązań, co jest korzystne podczas szukania minimum i dużo łatwiej jest znaleźć szukane minimum.

# 7. Proszę punkt startowy ES (dla μ , λ=(128 ,512) , σ=0.1) oraz algorytmu SGD z poprzedniego zadania ustawić na punkt (10,10) i porównać wyniki.

## 7.1 Co i dlaczego można zaobserwować?

Obraz zawierający tekst, diagram, zrzut ekranu, krąg

Opis wygenerowany automatycznieObraz zawierający tekst, diagram, zrzut ekranu, Wykres

Opis wygenerowany automatycznie

Algorytm ewolucyjny znalazł minumum natomiast algorytm SGD nie znalazł. Spowodowane jest to tym, że gradient w punkcie (10,10) jest zerowy więc algorytm nie zadziałał

## 7.2 Kiedy ES jest lepszym rozwiązaniem niż SGD?

## Algorytm ewolucji strategii (ES) może być lepszym rozwiązaniem niż stochastyczny spadek gradientu (SGD) w sytuacjach, gdy:

## Brak dostępu do gradientu.

## Funkcja celu zawiera wiele lokalnych ekstremów.

## Funkcja celu jest złożona.

## Problem wymaga optymalizacji globalnej.

## 7.3 Który algorytm optymalizacji jest bardziej złożony obliczeniowo?

ES mu + lambda jest bardziej złożonym obliczeniowo algorytmem niż SGD.

W ES mu + lambda proces optymalizacji obejmuje generowanie, selekcję, krzyżowanie i mutację populacji osobników, co wymaga większej ilości obliczeń w porównaniu do SGD, który polega głównie na obliczeniach gradientu i aktualizacji wag.

Proces ewolucji populacji w ES mu + lambda może być bardziej złożony ze względu na wymagane operacje na dużych zbiorach danych oraz ewentualną konieczność dostosowania parametrów populacji.