

Algorytmy genetyczne i Sztuczne sieci neuronowe

Lista 1. - Wprowadzenie do algorytmu genetycznego i problemu plecakowego

Problem plecakowy (ang. *Knapsack Problem*) to klasyczny problem optymalizacyjny, w którym mamy plecak o określonej pojemności i zestaw przedmiotów, z których każdy ma przypisaną wartość oraz wagę. Celem jest wybranie takiego zestawu przedmiotów, aby maksymalizować sumę ich wartości, jednocześnie nie przekraczając pojemności plecaka.

Algorytm genetyczny (AG) to metoda inspirowana procesem ewolucji biologicznej, służąca do znajdowania przybliżonych rozwiązań problemów optymalizacyjnych. W AG używa się populacji rozwiązań, która ewoluuje z pokolenia na pokolenie, podlegając operacjom takim jak krzyżowanie, mutacja i selekcja, w celu znajdowania coraz lepszych rozwiązań.

1. Kluczowe zagadnienia:

- a. Problem plecakowy 0-1:
 - i. Każdy przedmiot może być albo wybrany, albo nie (0-1). Dlatego rozwiązania problemu można reprezentować jako wektor binarny, gdzie "1" oznacza, że przedmiot jest w plecaku, a "0" – że go nie ma.
 - ii. Funkcja celu: maksymalizacja sumy wartości przedmiotów przy jednoczesnym spełnieniu ograniczenia dotyczącego wagi.
- b. Algorytmy genetyczne – wprowadzenie:
 - i. Chromosom – zakodowane rozwiązanie problemu (w przypadku problemu plecakowego to wektor binarny).
 - ii. Populacja – zbiór chromosomów (rozwiązań), które podlegają ewolucji.
 - iii. Funkcja oceny (fitness function) – ocenia jakość rozwiązania (w naszym przypadku na podstawie wartości przedmiotów i spełnienia ograniczeń wagowych).
 - iv. Selekcja, krzyżowanie, mutacja – operatory genetyczne, które pozwalają ewoluować populacji.

2. Zadania praktyczne:

- a. Zadanie 1: Kodowanie problemu plecakowego
 - i. Pobierz zbiór danych:
https://drive.google.com/file/d/1LIDgekQWFsTtdIFG1fZrRtQKo6hvASi_/view?usp=sharing
 - ii. Zakoduj rozwiązanie problemu plecakowego jako chromosom (wektor binarny).
 - iii. Przykład: Dla zestawu 5 przedmiotów o wartościach [10, 20, 30, 40, 50] i wagach [1, 2, 3, 8, 7], chromosom [1, 0, 1, 0, 1] oznacza, że w plecaku znajdują się przedmioty 1, 3 i 5.
- b. Zadanie 2: Generowanie populacji początkowej

- i. Napisz funkcję, która generuje losową populację początkową.
- ii. Wskazówki:
 - 1. Populacja to zbiór chromosomów (rozwiązań).
 - 2. Populacja początkowa powinna składać się z określonej liczby losowo wygenerowanych rozwiązań.
- c. Zadanie 3: Funkcja oceny (fitness function) – wprowadzenie do dyskusji
 - i. Omów, jak mogłaby wyglądać funkcja oceny dla problemu plecakowego.
 - ii. Zastanów się, jak uwzględnić zarówno wartość przedmiotów, jak i ograniczenie wagowe.
 - iii. Wskazówki:
 - 1. Funkcja oceny powinna przyznawać wyższą wartość tym chromosomom, które mają większą wartość przedmiotów.
 - 2. Jeśli suma wag przedmiotów przekracza pojemność plecaka, można np. wprowadzić karę do funkcji oceny.

3. Warunki opracowywanych rozwiązań:

- a. Opracowane rozwiązania powinny być implementowane w języku Python. Dopuszczalne jest wykorzystanie innego języka programowania pod warunkiem uzyskania zgody prowadzącego.
- b. Podczas implementacji wolno posługiwać się bibliotekami do obliczeń numerycznych ogólnego przeznaczenia (np. NumPy, SciPy) oraz przetwarzania danych (np. Pandas).
- c. Podczas implementacji nie wolno posługiwać się dedykowanymi bibliotekami do tworzenia algorytmów genetycznych (np. PyGAD, DEAP itp.) oraz architektur sieci neuronowych (np. scikit-learn, PyTorch, Tensorflow, JAX itp.).