

Sztuczna Inteligencja

Lista 5 – jedno zadanie do wyboru

Zadanie 1 (max. ocena: 3.5)

Zastosować klasyczny algorytm genetyczny do rozwiązania poniższego problemu maksymalizacji dochodu w produkcji czekolady.

Dany jest wytwórca czekolady dążący do maksymalizacji swego dochodu. Wytwórca decyduje o smaku czekolady, jej wielkości i obecności dodatków (orzechy, rodzynki itp.). Czekolada może być wytwarzana w czterech smakach: gorzka, mleczna, deserowa, biała; może zawierać lub nie dodatki, oraz może mieć jeden z dwóch rozmiarów (mała, duża). Wszystkie te parametry mogą wpłynąć zarówno na cenę czekolady, jak i na późniejszy popyt. Jakie czekolady powinien produkować wytwórca, by jego dochód (liczony jako iloczyn ceny i popytu) był maksymalny?

Założmy, że koszt wyprodukowania małej czekolady o dowolnym smaku jest stały i wynosi 10. Koszt dodatków do czekolady jest również stały i wynosi 5. Cena dużej czekolady jest dwukrotnie wyższa. Założmy ponadto, że dysponujemy badaniami określającymi zależności popytu na czekoladę od jej smaku. W umownych jednostkach popytu wyniki tych badań przedstawiają się następująco: Czekolada gorzka: 6, z dodatkami: 6, czekolada mleczna: 10, z dodatkami: 6, czekolada deserowa: 8, z dodatkami: 12, czekolada biała: 10, z dodatkami: 12. Ze względu na wyższą cenę, czekolady duże cieszą się dwukrotnie mniejszym zainteresowaniem.

Wskazówki:

1. Dysponując powyższym kompletem informacji, możemy rozwiązać problem za pomocą algorytmu genetycznego. Należy zastanowić się, co będzie "osobnikami" - będą nimi czekolady, a właściwie ich rodzaje. Każdemu rodzajowi czekolady (fenotypowi), odpowiadał będzie pewien zespół opisujących go cech (genów). Tymi cechami będą: smak (4 możliwości), ich wielkość (2 możliwości) i obecność dodatków (2 możliwości). Mamy więc łącznie 16 kombinacji cech czekolady. W klasycznym, najprostszym modelu, geny kodowane są za pomocą cyfr 0 i 1. W przypadku czekolady, każdy ich rodzaj będzie opisywany przez cztery takie cyfry: dwie opisujące ich smak, jedna obecność dodatków i jedna wielkość. Przyjmijmy następującą metodę kodowania cech: czekolada gorzka: 00, mleczna: 01, deserowa: 10, biała: 11, Nie ma dodatków: 0, są dodatki: 1. Czekolada mała: 0, czekolada duża: 1.

Np.: 0011 – czekolada gorzka z dodatkami, duża.

2. Konstruowanie funkcji przystosowania (fitness). Powinna to być funkcja przypisująca każdej kombinacji genów - każdemu rodzajowi czekolady - pewną nieujemną wielkość, tym większą, im bardziej dany osobnik nam odpowiada. Najbardziej naturalną miarą jakości czekolady jest dochód z jej sprzedaży - iloczyn popytu i ceny. Sposób eliminacji osobników słabszych określał będzie tzw. algorytm "koła ruletki": w fazie reprodukcji szanse na przeżycie będą proporcjonalne do wartości funkcji przystosowania.
3. Do poprawnego działania algorytmu genetycznego potrzeba jeszcze określić sposób działania operatorów genetycznych: mutacji i krzyżowania. W przypadku kodowania binarnego, możemy użyć mutacji polegającej na losowej zamianie wartości pojedynczego bitu na przeciwny:

Np.: 0 0 0 0 0 1 0 0

Operacja krzyżowania jest trochę bardziej złożona. Najpierw wybieramy losowo parę rodziców, następnie losujemy punkt przecięcia - jeden z trzech punktów pomiędzy genami - jednakowy

dla obojga rodziców, następnie konstruujemy potomka z początkowego fragmentu pierwszego rodzica i końcowego fragmentu drugiego. Finalnie potomek zastępuje jednego z rodziców.

Np.:

Rodzic 1: 0 0 0 | 0 Rodzic 2: 1 1 1 | 1

Potomek: 0 0 0 1

Krzyżowanie przeprowadzane będzie z prawdopodobieństwem 50%, mutacja: 5% (dotyczy to każdego bitu z osobna).

Działanie algorytmu (przykład): Ustalmy wielkość populacji na 4 osobniki. Populacja początkowa będzie składała się z osobników wybranych losowo:

Np.: 0111, 1000, 1000, 0100

Następnie powinniśmy zastosować operatory genetyczne. Jednak, ponieważ populacja składa się na razie z osobników losowych, nie ma to sensu - możemy przejść od razu do fazy reprodukcji. Faza reprodukcji zaczyna się od policzenia wartości funkcji przystosowania każdego osobnika. Następnie, aby móc zastosować algorytm “koła ruletki”, sumujemy wartości funkcji wszystkich osobników i wartości te dzielimy przez otrzymaną sumę - w ten sposób otrzymujemy procentowy “udział” każdego osobnika w sumie. Wartości te sumujemy następnie narastająco - tworzymy dystrybuantę rozkładu osobników.

Nr	Osobnik	Wartość przystosowania	% udział	suma udziałów, narastająco
1	0111	90	25%	25%
2	1000	80	23%	48%
3	1000	80	23%	71%
4	0100	100	29%	100%
	Suma:	350	100%	

Występujące w tabeli wartości fitness - funkcji przystosowania obliczymy na podstawie danych wejściowych: np. czekolada 0111 mają cenę 30 (gdyż ma dodatki i jest duża), a popyt na nią wynosi 3, tak więc zysk wyniesie $30 \times 3 = 90$. Aby wybrać osobniki do następnej populacji, losujemy cztery liczby z zakresu $[0,1)$ i w ostatniej rubryce sprawdzamy, któremu osobnikowi one odpowiadają. Wylosowaliśmy liczby: 0.22, 0.01, 0.15, 0.88. Odpowiadają one osobnikom: 1, 1, 1 i 4 (znajdujemy najmniejszą liczbę większą od wylosowanej i wybieramy odpowiadającego jej osobnika). Nowa populacja składa się zatem z osobników:

0111, 0111, 0111, 0100

Zgodnie z przewidywaniem, do następnej populacji dostały się osobniki najlepiej przystosowane. Po fazie reprodukcji przychodzi czas na krzyżowanie. Dla każdego osobnika losujemy więc (z prawdopodobieństwem 50%), czy zostanie na nim zastosowany operator krzyżowania. Załóżmy, że wylosowaliśmy osobnika 1 i 4. Losujemy następnie, z którymi osobnikami mają się one skrzyżować - wylosowaliśmy, że parą dla 1 będzie osobnik 2, a parą dla 4 będzie 1. W przypadku pierwszej pary wylosowaliśmy, że punktem krzyżowania będzie przerwa między trzecim i czwartym genem: Osobnik 1: 0 1 1 | 1 Osobnik 2: 0 1 1 | 1

Potomek: 0 1 1 1

Ponieważ osobniki 1 i 2 mają identyczny zestaw genów (opisują ten sam rodzaj czekolady), skrzyżowanie ich nie prowadzi do niczego nowego - jest to podstawowa cecha wszelkich operatorów krzyżowania. Inaczej będzie w drugim przypadku:

Osobnik 4: 0 1 0 | 0 Osobnik 1: 0 1 1 | 1

Potomek: 0 1 0 1

Wylosowaliśmy ten sam punkt krzyżowania, co poprzednio. Widzimy, że w populacji pojawił się nowy rodzaj czekolady: po osobniku 4 odziedziczył on brak dodatków, po osobniku 1 - wielkość. Po zastąpieniu pierwszych rodziców potomkami, populacja składa się z osobników:

0111, 0111, 0111, 0101

Nadszedł czas na mutacje. Zwykle jest to operator stosowany rzadziej, niż krzyżowanie - w naszym przypadku prawdopodobieństwo wynosi np. 5%, czyli zmutowany zostanie średnio mniej, niż jeden gen w całej populacji (liczącej 16 genów). Tym razem jednak, losując kolejno dla każdego bitu liczbę z przedziału [0,1), za siódmym razem wylosowaliśmy liczbę mniejszą, niż 0.05. Zmutowany zostanie więc trzeci gen drugiego osobnika:

0111, 0101, 0111, 0101

W efekcie mutacji powstał taki sam osobnik, jak po krzyżowaniu czwartego osobnika z pierwszym. Ostatecznie, po drugim cyklu ewolucji skład populacji przedstawia się jak powyżej. Wartości funkcji przystosowania wynoszą odpowiednio 90, 100, 90 i 100 - są więc średnio wyższe, niż początkowe. Po mutacji algorytm przechodzi do fazy reprodukcji, krzyżowania, mutacji, reprodukcji itd. Po kilku (pięciu? dziesięciu?) cyklach otrzymamy zestaw składający się z czekolady przynoszącej wytwórcy satysfakcjonujące zyski - może nawet będą wśród nich optymalne.

Uwaga: Przedstawione parametry pracy algorytmu genetycznego odpowiadają występującym w praktycznych zastosowaniach. Jedynie wielkość populacji została zmniejszona do 4, zwykle wynosi ona od kilkunastu do (w niektórych zastosowaniach) kilku tysięcy osobników. Poza tym prawdopodobieństwa mutacji i krzyżowania, a także sama konstrukcja operatorów genetycznych i sposób reprodukcji mogą być bez zmian używane w sytuacjach, gdy przestrzeń stanów jest na tyle duża, że uzasadnia stosowanie niekonwencjonalnych metod obliczeniowych.

Należy przygotować sprawozdanie z realizacji zadania z opisem sposobu realizacji algorytmu genetycznego, uzyskanymi wynikami i wnioskami.

Zadanie 2 (max. ocena: 5.0)

Opracować w dowolnym języku programowania program ilustrujący działanie klasycznego algorytmu genetycznego, ma on liczyć minimum dowolnej funkcji jednoargumentowej (podanej przez użytkownika) wykorzystując algorytm genetyczny. Założyć, że funkcja jest określona na przedziale $(-8,8)$. Uzasadnić wybór zastosowanych metod kodowania, mutacji, krzyżowania. Proces realizacji algorytmu przedstawić graficznie.

Należy przygotować sprawozdanie z realizacji zadania z opisem sposobu realizacji algorytmu genetycznego oraz uzyskanymi wynikami.

Wyniki należy uzyskać dla różnych parametrów algorytmu genetycznego (np. prawdopodobieństw mutacji), należy je porównać i przedstawić wnioski.

Zadanie 3

1. Zapoznać się z Matlab Fuzzy Logic Toolbox:
<http://www.mathworks.com/products/fuzzy-logic/>
2. Zaprojektować i zaimplementować rozmyty system wnioskujący wyznaczający wysokość napiwku dołączanego do rachunku w restauracji. Wysokość napiwku (wyjście systemu) ma być uzależniona od jakości obsługi oraz jakości jedzenia (wejść systemu). Klient dokonuje oceny tych kryteriów w skali od 0 do 8, możliwą wysokość napiwku przyjąć z zakresu od 5 do 25%. Zaproponować zmienne lingwistyczne opisujące wielkości wejściowe i wyjściową (np. niska, wysoka, zadawalająca, itp...), zbiory rozmyte (funkcje przynależności), bazę reguł rozmytych (np. „jeśli jakość jedzenia jest zadawalająca oraz jakość obsługi jest przeciętna to napiwek jest umiarkowany”). Przeprowadzić symulację działania systemu. (
3. Zaproponować i zaimplementować własny system wnioskowania rozmytego dla wybranego przez siebie problemu. Uzasadnić dobór parametrów i metod.
4. Przygotować sprawozdanie opisujące realizację pkt.3 – przedstawić wnioski.

Ocenie podlega 3 i 4 ((**max. Ocena: 4.0**))

UWAGI:

1. Termin deklaracji wyboru zadania – 19.01.2024
2. Termin realizacji – 26.01.2024