Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Uczenie Maszynowe

Sprawozdanie z projektu

Kacper Marchlewicz, Przemysław Wyziński

Warszawa, 2023

Spis treści

1. Wstęp

1.1. Treść zadania

Zastosowanie uczenia ze wzmocnieniem do ustawienia prawdopodobieństwa krzyżowania i sposobu krzyżowania w algorytmie ewolucyjnym. Niech stanem będzie procent sukcesów (ile potomków było lepszych niż rodzice) oraz średnia odległość pomiędzy osobnikami w aktualnej populacji, a akcją wybór wskazanych parametrów algorytmu. Zarówno stany jak i akcje można zdyskretyzować. Funkcje do optymalizacji należy pobrać z benchmarku CEC2017, którego kod da się znaleźć w Pythonie, R i C. Przykład w Pythonie. Przed rozpoczęciem realizacji projektu proszę zapoznać się z zawartością strony.

1.2. Założenia i implementacja algorytmu ewolucyjnego

Utworzony przez nas algorytm ewolucyjny kolejno składa się z:

— selekcji turniejowej o rozmiarze 2
— krzyżowania: uśredniającego/wymieniającego
— mutacji
— sukcesji elitarnej

Początkowa populacja jest generowana losowo dla podanych ograniczeń kostkowych. Obliczenia dokonywane są do momentu przekroczenia limitu liczby iteracji, którego wartość zależy od liczby populacji i budżetu obliczeniowego. Algorytm pod koniec każdej iteracji oblicza średnią odległość pomiędzy osobnikami i procent sukcesów - zliczamy ile dzieci jest lepsze od obojga rodziców.

1.3. Założenia i implementacja algorytmu uczenia ze wzmocnieniem

Początkowo algorytm inicjuje macierz Q oraz licznik epok zerem. Następnie w każdej epoce wybiera losowy punkt początkowy, generuje populację i:

— wybór akcji na podstawie mapy (wykorzystujemy strategię zachłanną)
— wykonanie akcji - modyfikacja parametrów krzyżowania
— operacje algorytmu ewolucyjnego na populacji
— obliczenie następnych współrzędnych na mapie na podstawie stanu
— obliczenie nagrody
— aktualizacja nagród w mapie
— przejście do następnego stanu

Powyższe akcje wykonują się do końca każdej epoki, która kończy się gdy skończy się określona liczba iteracji. Jest nią maksymalna liczba generacji algorytmu ewolucyjnego.

Nagrodą jest procentu sukcesów (ile osobników na etapie sukcesji jest lepszych niż stara populacja). Stanem jest procent sukcesów (ile potomków było lepszych niż rodzice) oraz średnia odległość pomiędzy osobnikami w aktualnej populacji. Dyskretyzacje dokonaliśmy w formie doboru zakresów wartości, na podstawie których zwracana jest współrzędna. Przedziały te wyznaczyliśmy arbitralnie. Korzystaliśmy z strategii zachłannej z $\epsilon = 0{,}1$.

Po wykonaniu uczenia, algorytm ewolucyjny podejmuje decyzje o modyfikacji parametrów krzyżowania w zależności od stanu.

1.4. Struktura programowa projektu

Projekt składa się z następujących plików:

— model.py - zawiera implementację algorytmu ewolucyjnego
— ewo_calculating_params.py - skrypt służący do wywołania badań nad parametrami algorytmu ewolucyjnego
— qlearn.py - zawiera implementację q-learning połączonego z algorytmem ewolucyjnym
— train.py - skrypt służący do badań porównawczych działania algorytmu ewolucyjnego kierowanego przez q-learning i do wywołania badań nad parametrami qlearning

2. Prezentacja badań i wyników

2.1. Dobór parametrów algorytmu ewolucyjnego

Pierwszym krokiem po implementacji algorytmów było wyznaczenie odpowiednich wartości algorytmu ewolucyjnego. Badania przeprowadzaliśmy dla domyślnego układu:

— liczba populacji = 20
— rozmiar elity = 1
— siła mutacji = 1
— krzyżowanie uśredniające
— prawdopodobieństwo krzyżowania = 0,5
— maksymalna liczba wywołań funkcji celu = 10000

Kolejno badaliśmy zmiany jednego parametru. Algorytm szuka minimum funkcji f4, f5, f6 z benchmarku cec2017 w 10 wymiarach. Wyniki są podane dla 25 uruchomień programu.

Wpływ liczności populacji dla funkcji f4

liczba populacji	min.	max.	śr.	std.
5	400,95	408,34	403,94	1,94
10	400,34	408,34	404,35	1,84
15	400,34	469,76	405,97	9,09
20	400,34	469,76	406,28	10,02
30	**400,34**	**469,82**	**406,25**	**9,03**
40	400,34	471,13	407,28	11,51
100	400,34	471,34	407,35	11,61

Wpływ liczności populacji dla funkcji f5

liczba populacji	min.	max.	śr.	std.
5	553,96	738,11	610,29	44,84
10	538,36	738,11	601,88	45,01
15	520,63	738,11	590,61	43,78
20	520,63	738,11	588,51	39,72
30	520,63	738,11	582,18	38,61
40	517,95	738,11	576,36	38,05
100	**509,82**	**738,11**	**569,68**	**39,01**

Wpływ liczności populacji dla funkcji f6

liczba populacji	min.	max.	śr.	std.
5	620,10	719,89	662,54	26,41
10	619,27	719,89	659,12	23,33
15	618,01	719,89	652,94	22,51
20	606,22	719,89	648,21	22,80
30	602,98	719,89	643,99	23,31
40	**601,16**	**719,89**	**641,15**	**23,02**
100	601,16	719,89	637,58	23,20

Większa populacja zwiększa szanse na znalezienie lepszych rozwiązań,ale jednocześnie mniej iteracji w algorytmie, a co za tym idzie mniej zmian względem jej stanu początkowego. Potrzebny jest większy budżet ewaluacji, a to spowoduje większe wymagania obliczeniowe. Należy odpowiednio zbalansować rozmiar populacji do posiadanego budżetu. Mniejsze populacje będą bardziej się różnić od pierwotnego wyglądu. Odpowiedni rozmiar zależny od problemu pozwala znaleźć równowagę między eksploracją a wykorzystaniem zasobów.

Wpływ rozmiaru elity dla funkcji f4

rozmiar elity	min.	max.	śr.	std.
0	400,57	460,02	407,67	10,87
1	**400,32**	**460,02**	**406,64**	**8,16**
2	400,32	469,60	407,00	9,95
3	400,32	469,60	406,44	8.72
4	400,24	469,60	406,41	9,12

Wpływ rozmiaru elity dla funkcji f5

rozmiar elity	min.	max.	śr.	std.
0	522,95	612,88	562,97	22,73
1	522,95	650,83	565,17	27,81
2	522,95	650,83	566,28	26,22
3	**515,61**	**650,83**	**564,03**	**24.95**
4	515,61	650,83	564,15	23,87

Wpływ rozmiaru elity dla funkcji f6

rozmiar elity	min.	max.	śr.	std.
0	609,86	682,20	639,91	18,09
1	609,86	682,20	638,47	16,05
2	**602,91**	**682,20**	**637,76**	**16,73**
3	602,91	682,20	637,39	17.31
4	602,91	682,20	636,30	17,58

Wprowadzenie elitarności pomaga w szybszym osiąganiu wysokiej jakości rozwiązań, ale może ograniczać eksplorację przestrzeni rozwiązań. Czasem może wykonywać rzecz odwrotną - nowa populacja będzie w całości lepsza od starej, co spowoduje przechowywanie gorszych osobników. W niektórych przypadkach mały rozmiar elity może wystarczyć, podczas gdy w innych przypadkach większa liczba może być konieczna.

Wpływ siły mutacji dla funkcji f4

siła mutacji	min.	max.	śr.	std.
0,05	482,99	2246,94	1013,15	441,62
0,1	404,27	2246,94	811,65	393,71
0,5	400,08	2246,94	677,25	373,51
0,75	400,08	2246,94	611,09	343.32
1	**400,08**	**2246,94**	**570,87**	**317,53**
2	400,08	2246,94	543,30	296,35
5	400,08	2246,94	524,30	278,28
10	400,08	2246,94	526,06	262,66

Wpływ siły mutacji dla funkcji f5

siła mutacji	min.	max.	śr.	std.
0,05	523,88	690,04	561,17	33,33
0,1	521,91	690,04	559,95	28,39
0,5	521,91	690,04	561,25	28,58
0,75	**518,74**	**690,04**	**559,04**	**27.74**
1	518,74	690,04	559,17	26,65
2	517,75	690,04	556,22	25,78
5	517,75	690,04	553,01	25,18
10	517,75	690,04	551,22	24,09

Wpływ siły mutacji dla funkcji f6

siła mutacji	min.	max.	śr.	std.
0,05	625,37	691.19	651,17	15,83
0,1	617,31	691.19	649,24	15,61
0,5	616,32	691.19	650,09	16,27
0,75	607,05	691.19	647,12	17.01
1	**607,05**	**691.19**	**645,06**	**17,44**
2	601,54	691.19	640,39	20,06
5	601,54	691.19	635,57	22,01
10	601,54	691.19	632,74	21,91

Zbyt słaba mutacja może prowadzić do zbyt wolnego eksplorowania przestrzeni rozwiązań, podczas gdy zbyt silna mutacja może prowadzić do zbyt szybkiego rozprzestrzeniania się populacji w przestrzeni rozwiązań i utraty cennych informacji genetycznych. Optymalna wartość zależy od rozwiązywanej funkcji, gdyż należy znaleźć równowagę między eksploracją a eksploatacją.

Wpływ prawdopodobieństwa krzyżowania dla funkcji f4

prawd. krzyżowania	min.	max.	śr.	std.
0,05	400,54	416,19	405,78	3,10
0,1	400,54	416,19	405,81	2,56
0,2	400,39	472,33	407,24	9,89
0,3	**400,39**	**472,33**	**407,32**	**10.28**
0,4	400,39	472,33	407,89	12,20
0,5	400,39	472,33	407,89	12,38
0,6	400,39	472,33	407,82	12,53
0,7	400,20	472,33	407,77	12,46
0,8	400,20	472,33	407,69	12,52
0,9	400,20	472,33	407,48	11,91

Wpływ prawdopodobieństwa krzyżowania dla funkcji f5

prawd. krzyżowania	min.	max.	śr.	std.
0,05	546,50	694,70	594,19	34,80
0,1	538,40	694,70	585,03	31,93
0,2	517,55	694,70	580,25	30,17
0,3	517,55	694,70	575,78	30.60
0,4	517,55	694,70	573,86	29,19
0,5	517,55	694,70	573,46	28,58
0,6	**512,62**	**694,70**	**571,98**	**28,04**
0,7	512,62	694,70	570,77	27,54
0,8	512,62	694,70	570,13	26,93
0,9	512,62	694,70	568,73	26,66

Wpływ prawdopodobieństwa krzyżowania dla funkcji f6

prawd. krzyżowania	min.	max.	śr.	std.
0,05	605,55	693.81	646,35	20,12
0,1	605,55	693.81	645,85	20,51
0,2	605,55	693.81	644,34	19,61
0,3	601,85	696.73	643,20	20.67
0,4	601,07	696.73	641,77	20,25
0,5	**601,07**	**696.73**	**639,64**	**19,92**
0,6	601,07	696.73	638,90	20,14
0,7	601,07	696.73	638,55	19,70
0,8	601,07	696.73	637,54	19,31
0,9	601,07	696.73	637,34	18,94

Wartość prawdopodobieństwa powinna być dobrana odpowiednio. Zbyt wysokie może prowadzić do szybkiej utraty różnorodności genetycznej, podczas gdy zbyt niskie może prowadzić do utknięcia w lokalnych ekstremach.

Pogrubiliśmy wartości parametrów które uznaliśmy na odpowiednie dla danych funkcji. Staraliśmy się kierować równowagą pomiędzy eksploracją a eksploatacją.

2.2. Dobór parametrów algorytmu QLearn

Badaliśmy zmiany parametrów beta i gamma dla algorytmu QLearn. Program ustawiliśmy na 200 epizodów, następnie algorytm 25 razy rozpoczynał działanie w losowym miejscu na q-mapie. Wiersze - beta, kolumny - gamma. W komórce kolejno znajdują się : min, max, średnia odchylenie.

Wyniki dla funkcji f4

β/γ	0,1	0,25	0,5	0,75	0,9	**0,95**	1
0,1	400,19	400,17	400,22	400,29	400,34	400,25	400,30
	400,19	400,17	400,22	400,29	400,34	400,25	400,30
	400,19	400,17	400,22	400,29	400,34	400,25	400,30
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,25	400,32	400,37	400,30	400,38	400,39	400,28	400,24
	400,32	400,37	400,30	400,38	400,39	400,28	400,24
	400,32	400,37	400,30	400,38	400,39	400,28	400,24
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,5	400,34	400,26	400,18	400,32	400,33	400,32	400,40
	400,34	400,26	400,18	400,32	400,33	400,32	400,40
	400,34	400,26	400,18	400,32	400,33	400,32	400,40
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,75	400,24	400,32	400,19	400,38	400,36	400,41	400,38
	400,24	400,32	400,19	400,38	400,36	400,41	400,38
	400,24	400,32	400,19	400,38	400,36	400,41	400,38
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,9	400,34	400,30	400,27	400,26	400,28	**400,17**	400,38
	400,34	400,30	400,27	400,26	400,28	**400,17**	400,38
	400,34	400,30	400,27	400,26	400,28	**400,17**	400,38
	0,00	0,00	0,00	0,00	0,00	**0,00**	0,00
0,95	400,38	400,29	400,31	400,32	400,37	400,26	400,30
	400,38	400,29	400,31	400,32	400,37	400,26	400,30
	400,38	400,29	400,31	400,32	400,37	400,26	400,30
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	400,35	400,35	400,33	400,32	400,24	400,32	400,21
	400,35	400,35	400,33	400,32	400,24	400,32	400,21
	400,35	400,35	400,33	400,32	400,24	400,32	400,21
	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Wyniki dla funkcji f5

β/γ	**0,1**	0,25	0,5	0,75	0,9	0,95	1
0,1	504,71	505,54	504,24	504,51	506,41	503,46	505,89
	504,71	505,54	504,24	504,51	506,88	503,46	505,89
	504,71	505,54	504,24	504,51	506,62	503,46	505,89
	0,00	0,00	0,00	0,00	0,23	0,00	0,00
0,25	505,82	503,99	504,86	503,11	506,51	504,41	505,52
	505,82	503,99	504,86	503,11	506,51	504,41	505,52
	505,82	503,99	504,86	503,11	506,51	504,41	505,52
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,5	506,08	506,67	504,13	505,95	505,10	502,91	505,00
	506,08	506,67	504,13	505,95	505,10	502,91	505,00
	506,08	506,67	504,13	505,95	505,10	502,91	505,00
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,75	505,93	505,46	504,61	505,85	506,02	506,08	504,55
	505,93	505,46	504,61	505,85	506,02	506,08	504,55
	505,93	505,46	504,61	505,85	506,02	506,08	504,55
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,9	**502,13**	506,81	505,96	505,00	505,05	502,98	505,39
	502,13	506,81	505,96	505,00	505,05	502,98	505,39
	502,13	506,81	505,96	505,00	505,05	502,98	505,39
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,95	505,20	505,81	505,07	507,51	504,28	506,03	503,80
	505,20	505,81	505,07	507,51	504,28	506,03	503,80
	505,20	505,81	505,07	507,51	504,28	506,03	503,80
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	505,32	506,15	505,03	505,46	504,17	505,80	505,27
	505,32	506,15	505,03	505,46	504,17	506,89	505,27
	505,32	506,15	505,03	505,46	504,17	506,10	505,27
	0,00	0,00	0,00	0,00	0,00	0,49	0,00

Wyniki dla funkcji f6

β/γ	0,1	0,25	0,5	0,75	0,9	0,95	**1**
0,1	600,90	600,98	601,17	600,88	601,06	601,13	**600,76**
	600,90	600,98	601,17	600,88	601,06	601,13	**600,76**
	600,90	600,98	601,17	600,88	601,06	601,13	**600,76**
	0,00	0,00	0,01	0,00	0,00	0,00	**0,00**
0,25	600,86	601,13	601,06	601,12	601,10	601,12	600,97
	600,86	601,13	601,06	601,12	601,10	601,12	600,97
	600,86	601,13	601,06	601,12	601,10	601,12	600,97
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,5	600,84	600,96	600,93	601,22	600,98	600,83	601,11
	600,84	600,96	600,93	601,03	600,98	600,83	601,12
	600,84	600,96	600,93	601,10	600,98	600,83	601,11
	0,00	0,00	0,00	0,09	0,00	0,00	0,01
0,75	600,89	601,11	600,80	601,11	601,15	601,07	600,95
	600,89	601,11	600,80	601,11	601,15	601,07	600,95
	600,89	601,11	600,80	601,11	601,15	601,07	600,95
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,9	600,83	600,93	601,00	600,80	600,81	601,03	601,18
	600,83	600,93	601,05	600,80	600,81	601,03	601,18
	600,83	600,93	601,02	600,80	600,81	601,03	601,18
	0,00	0,00	0,02	0,00	0,00	0,00	0,00
0,95	600,97	600,82	601,06	601,04	601,17	601,02	600,99
	600,97	600,82	601,06	601,04	601,17	601,02	600,99
	600,97	600,82	601,06	601,04	601,17	601,02	600,99
	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	601,11	600,98	600,80	600,94	600,99	601,03	601,10
	601,11	600,98	600,80	600,94	601,02	601,03	601,10
	601,11	600,98	600,80	600,94	600,99	601,03	601,10
	0,00	0,00	0,00	0,00	0,01	0,00	0,00

Dobór bety i gammy jest ważnym elementem algorytmu. Przy niskiej wartości beta agent uczy się powoli, bardziej polega na wcześniejszych wynikach. Zwiększanie wartości beta prowadzi do szybkiego uczenia się, ale może powodować, większą podatność na zmienność w środowisku.

Wysoka wartość gamma oznacza długoterminową perspektywę i silnie uwzględnia przyszłe nagrody. Zmniejszanie współczynnika gamma powoduje koncentrację na natychmiastowych nagrodach, krótkoterminową perspektywę.

Pogrubiliśmy wartości parametrów które uznaliśmy na odpowiednie dla danych funkcji. Dobieraliśmy te, które osiągały najmniejszą wartość.

Chociaż zmiana parametrów beta i gamma wpływa na wyniki działania algorytmu QLearn, to nie jest to wpływ bardzo znaczący. Dla wszystkich przebadanych przez nas kombinacji bety i gammy algorytm zwracał lepsze wyniki od najlepszego algorytmu AE.

2.3. Porównanie działania algorytmów QLearn i AE

Po doborze najodpowiedniejszych parametrów rozpoczęliśmy porównanie działanie algorytmu ewolucyjnego w wersji klasycznej i wspomaganej przez algorytm Qlearn.

Funkcja f4

Parametry algorytmu ewolucyjnego:

— populacja = 30
— rozmiar elity = 1
— siła mutacji = 1
— prawdopodobieństwo krzyżowania = 0,3

Parametry algorytmu Qlearn:

— $\beta = 0{,}9$
— $\gamma = 0{,}96$

Wyniki:

algorytm	min.	max.	śr.	std.
AE	401,08	468,31	410,75	16,04
Qlearn	400,36	400,36	400,36	0,00

Funkcja f5

Parametry algorytmu ewolucyjnego:

— populacja = 100
— rozmiar elity = 3
— siła mutacji = 0,75
— prawdopodobieństwo krzyżowania = 0,6

Parametry algorytmu Qlearn:

— $\beta = 0{,}9$
— $\gamma = 0{,}1$

Wyniki:

algorytm	min.	max.	śr.	std.
AE	509,41	538,72	523,66	8,36
Qlearn	505,61	504,87	505,52	0,24

Funkcja f6

Parametry algorytmu ewolucyjnego:

— populacja = 40
— rozmiar elity = 2
— siła mutacji = 1
— prawdopodobieństwo krzyżowania = 0,5

Parametry algorytmu Qlearn:

— $\beta = 0{,}1$
— $\gamma = 1$

Wyniki:

algorytm	min.	max.	śr.	std.
AE	601,65	647,45	616,69	12,31
Qlearn	600,58	600,58	600,58	0,00

Wnioski:

Algorytm ewolucyjny wspomagany przez Qlearning jako bardziej elastyczny, może korzystać z różnych rodzajów krzyżowania i dostosowywać się do specyfiki problemu. Możliwość modyfikowania prawdopodobieństwa pozwala na balansowanie pomiędzy poszukiwaniem nowych rozwiązań a eksploatacją odkrytych. Jak widać po wynikach, udaje mu się znaleźć lepsze optima od czystego algorytmu ewolucyjnego, który za to jest bardziej efektywny w eksploracji przestrzeni rozwiązań. Do znalezienia globalnych minimów może być wymagana modyfikacja funkcji nagród w Qlearning, gdyż obecny system nagradza eksploatację.

Dzięki projektowi pogłębiliśmy wiedzę na temat algorytmów ewolucyjnych i uczenia ze wzmocnieniem. Poznaliśmy nowe podejście do tematu, które mogliśmy praktycznie zastosować. Problem dyskretyzacji stanu w celu utworzenia q mapy był dla nas całkowicie nowym zadaniem. Przekonaliśmy się, że przy długich obliczeniach programu warto jest zapisywać obecne postępy, na wypadek wystąpienia błędu.