

[WSI] Laboratorium 4 – Behavioral cloning

Wykonał Andrii Gamalii, numer indeksu: 323665

Obliczenia

$$\sqrt{3,14 * 323665} \approx 1\,008,1211 \Rightarrow$$

- $x = 121 \bmod 3 = 1$

Stąd muszę rozwiązać zadanie z wykorzystaniem maszyny wektorów nośnych (SVM).

O programie

Python wersji 3.10.0 64-bit

W rozwiązaniu użyłem poniższych bibliotek zewnętrznych:

- numpy~1.23.5
- pygame~2.1.0
- scikit-learn~1.1.3

Dla każdego kierunku tworzę klasyfikator SVM o modelu liniowym.

Atrybuty

Zdefiniowałem 10 atrybutów:

1. obecny kierunek
2. długość węża (w blokach)
3. odległość od głowy do górnej ściany
4. odległość od głowy do dolnej ściany
5. odległość od głowy do lewej ściany
6. odległość od głowy do prawej ściany
7. czy jest jedzenie na prawo od głowy (tak = 1, nie = -1)
8. czy jest jedzenie na lewo od głowy (tak = 1, nie = -1)
9. czy jest jedzenie na górze od głowy (tak = 1, nie = -1)
10. czy jest jedzenie na dole od głowy (tak = 1, nie = -1)

Miara dokładności

Wziąłem zbiór danych długości 3000, z niego zbiory treningowy i testowy w proporcji 4:5 i zmierzyłem dla każdego kierunku miarę dokładności modeli. Uzyskałem takie wyniki:

- RIGHT – 0,825
- UP – 0,905
- DOWN – 0,8
- LEFT – 0,863

To znaczy, że model najlepiej się zachowuje, gdy ma wybrać kierunek wprawo i najgorzej jak w dół.

Odpowiedzi na pytania:

- 1) **Przeuczenie** występuje w przypadkach zbyt dużej liczby iteracji uczenia na odnośnie małym zbiorze danych. Albo jeśli dane są typowe (czyli jakieś sample powtarzają się zbyt często), albo jeśli dane są wyrafinowane, bez szumu. W takich przypadkach model zbyt szybko się dostosowuje do danych i traci swoją elastyczność.
- 2) **Niedouczenie** jest powodowane najczęściej małą ilością danych, ich zbytnią różnorodnością, sprzecznością w danych a także charakterem modelu (jak dobrze pasuje dla danego zadania). Przez co trudniej jest wyznaczyć parametry modelu.
- 3) W moim przypadku raczej występuje niedouczenie z powodu słabego modelu i czasami sprzecznych sampli w zbiorze danych.
- 4) Zminimalizować to niedouczenie można podbierając lepsze dane, w których jest wyraźniej widać pattern albo stosując silniejszy model (nie liniowy SVM, a np. z przekształceniem jądrowym).
- 5) Model nie gra tak dobrze jak ja, ponieważ 1) dysponuje wiedzą ograniczoną zdefiniowanymi atrybutami, 2) zbiór danych nie jest idealny, 3) model liniowy SVM jest ograniczony do przypadków klasyfikacji danych liniowo-separowalnych.

Badania:

Przebadłem wpływ hiperparametru liczby iteracji na miarę dokładności odpowiedzi modelu (`learning_rate=0.00001`, `lambda_param=0.0000001`). Takie dostałem wyniki:

liczba iteracji	1	10	100	1000	3000
miara dokładności	0,736	0,758	0,783	0,882	0,86

Można z pewnością powiedzieć, że im więcej model przechodzi iteracji trenowania, tym lepiej zgaduje odpowiedź. A także jest pewna liczba iteracji, po której nie ma sensu więcej trenować model, bo parametry jego funkcji zaczynają oscylować w okolicy pewnego punktu.

Przeprowadziłem 50 rozgrywek dla swojego modelu SVM z parametrami (`learning_rate=0,001`; `n_iters=1000`; `lambda_param=0,01`) i modelu SVC z biblioteki `sklearn` z domyślnymi parametrami. Średnie uzyskanych wyników:

- 2,74 – SVC z `sklearn`
- 2,42 – mój SVM

Stąd można wywnioskować, że zaimplementowany przeze mnie model zachowuje się dość nieźle w porównaniu do SVC z biblioteki. Wydaje mi się, że różnica w wynikach byłaby większa dla lepiej zdefiniowanych atrybutów i lepszego zbioru danych.