Zastosowanie Qlearningu w zachowaniu boidów

Jakub Łęcki, Marek Hering, Maciej Jabłoński

08.06.2020

Streszczenie

Tematem niniejszej pracy jest problem nauczenia boidów (w naszym przypadku są to ryby), aby poprzez prawidłowe poruszanie się maksymalizowały swój czas życia. W tym celu użyliśmy koncepcji qlearningu oraz algorytmu stada.

1 Boidy

1.1 Pochodzenie

Termin **boid** został stworzony przez Craiga Reynoldsa w 1987 roku jako określenie stworzenia wykazującego cechy stadne. Słowo boid wzięło się z uproszczenia terminu 'bird-like' jako odniesienie do ptaków formujących się w gromady.

1.2 Zasady zachowania

Okazuje się, że w świecie rzeczywistym wiele gatunków zwierząt łączących się w grupy wykazuje podobne własności. Patrząc na stada ptaków, ławice ryb, roje pszczół lub stada owiec można zauważyć, że każda z jednostek stosuje się do 3 podstawowych zasad:

1. Rozdzielność - osobnik nie lubi przebywać w tłoku, dlatego zachowuje dystans do swoich sąsiadów
2. Spójność - osobnik nie lubi przebywać w samotności, więc kieruje się ku najbliższym współstadnikom
3. Wyrównanie - osobnik porusza się w kierunku zbliżonym do kierunku otaczających członków stada

Łącząc te 3 proste zasady, boidy tworzą złożone i bardzo zorganizowane skupiska, które obserwujemy jako np. ławice ryb, które pozostają w płynnym, nieustannym ruchu.

2 Qlearning

Jest to jedna z technik szerokiej dziedziny uczenia maszynowego znanej jako "Uczenie ze wzmocnieniem" (ang. Reinforcement Learning). Opiera się na śledzeniu zachowania agentów oraz efektów, które owe akcje powodują. W tym celu używana jest tablica stanów-akcji zwykle nazywana jako **qtable**.

2.1 Qtable

Tablica ma wymiary m $\times$ n, gdzie:

- m - liczba możliwych stanów
- n - liczba akcji możliwych do wykonania

W każdej komórce $Q(s, a)$ znajduje się oczekiwana wartość nagrody jaką agent otrzyma będąc w stanie s i wykonawszy akcję a. Po wykonaniu akcji, agent przechodzi do kolejnego stanu s'. Agent będąc w stanie s będzie wybierać swoją kolejną akcję na podstawie polityki $argmax(Q(s))$, czyli wybranie akcji za którą teoretycznie otrzyma największą nagrodę.

2.2 Środowisko

Jest to zbiór agentów oraz dowolnych innych encji z którymi agent może w jakiś sposób oddziaływywać. Rolą środowiska jest wykonanie akcji wybranej przez agenta i ocenienie jak dobrze akcja została wybrana. W tym celu środowisko nadaje agentowi nagrody (i kary, jeśli nagroda jest ujemna). W kolejnych rundach skutkuje to stopniowym poprawianiem procesu wyboru akcji i agent zbiera coraz wyższe nagrody. Po wykonaniu kroku środowisko przekazuje do algorytmu uczenia zestaw danych:

- stan w którym był agent
- akcja jaką wykonał
- stan w którym znajduje się po wykonaniu akcji
- nagroda jaką otrzymał za przejście do kolejnego stanu

2.3 Proces uczenia

Aby agent wybierał z czasem coraz lepsze decyzje, wartości w Qtable muszą ulegać zmianie. Odbywa się to w oparciu o poniższe równanie:

$$Q'(s,a) \leftarrow Q(s,a) + \alpha \cdot \Big(r + \gamma \cdot \max_{\mathrm{a}} Q(s',a) - Q(s,a)\Big) \tag{1}$$

Gdzie s i a to stan i akcja przed jej wykonaniem, a s' to stan po wykonaniu akcji.

2.3.1 Współczynnik uczenia α

Wartość $\alpha \in <0,1>$ reguluje jak bardzo znacząca jest nowa informacja uzyskana w wyniku wykonania akcji w środowisku. Przy wartości 0 agent nie będzie się uczył niczego nowego, natomiast przy $\alpha = 1$ agent kompletnie zignoruje wiedzę dotychczasową i zastąpi ją nowymi danymi.

2.3.2 Współczynnik dyskontowania γ

Wartość $\gamma \in <0,1>$ określa ważność przyszłych nagród zdobywanych przez agenta. Wartość dążąca do zera zwiększy sugerowanie się pamięcią krótkotrwałą, natomiast do 1 pamięcią długotrwałą. Ważnym elementem jest, aby γ rzeczywiście zawierała się w przedziale $<0,1)$ ponieważ zapewnia to zbieżność wartości przewidywanej nagrody. Jeśli proces uczenia byłby nieskończony i $\gamma \geqslant 1$ (a nawet lekko poniżej) wartości nagród rosłyby nieustannie zaburzając proces.

2.3.3 Efekt

W wyniku przeprowadzenia odpowiedniej ilości kroków wartości nagród zbiegają się do końcowych, a w tabeli powstają zależności pomiędzy poszczególnymi stanami umożliwiające dotrzeć do nawiększej nagrody, dysponując jedynie obecnym stanem i przewidywaną nagrodą.

2.3.4 Polityka wspomagająca

Gdy podczas uczenia agent będzie słuchał się tylko tabeli qlearningu, może się zdarzyć że zadowoli się częściowo poprawnym rozwiązaniem, zamiast szukać rozwiązania dokładnego. Aby uniknąc takiej sytuacji proces wybierania akcji rozszerza się o politykę **epsilon greedy**, w której z pewnym prawdopodobieństwem wybierana jest losowa akcja zamiast wskazywanej przez tabelę. Wartość ϵ jest zmniejszana z czasem, aby agent coraz częściej stosował to czego się nauczył.