

# Przemysł 4.0

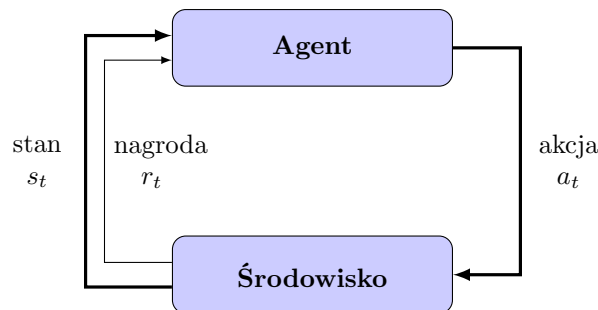
## Laboratorium 3

### *Uczenie ze wzmocnieniem*

prowadzący: *Dr inż. Radosław Idzikowski*

## 1 Wprowadzenie

Celem laboratorium jest zapoznanie się z uczeniem ze wzmocnieniem (ang. *Reinforcement Learning*, RL). RL służy do rozwiązywania problemów, które można opisać za pomocą agenta podejmującego decyzje w ramach środowiska (Rysunek 1). Decyzje podjęte przez agenta wiążą się z otrzymywaniem sygnału nagrody. W przeciwieństwie do innych typów uczenia maszynowego, agent otrzymuje pewną informację o “dobroci” podejmowanych decyzji. Nie jest to więc przypadek uczenia nienadzorowanego, gdzie nie ma zupełnie informacji o oczekiwanym wyjściu (w tym wypadku agent nie miałby dostępu do informacji o nagrodzie). Jednocześnie nie jest to przypadek uczenia nadzorowanego, gdzie przy treningu wykorzystuje się pary: wejście oraz przypisane oczekiwane wyjście (w tym wypadku agent trenowany nie byłby przez sygnał nagrody, ale przez przykłady idealnego zachowania w każdym z kroków decyzyjnych). Tym samym uczenie ze wzmocnieniem można stosować w przypadku środowisk, gdzie nieznanne są konkretne kroki konieczne do osiągnięcia zadanego celu, ale znany jest co najmniej sposób oceny finalnego rezultatu (lub co poprawi działanie samego agenta).



Rysunek 1: Schematyczne przedstawienie podstawowych pojęć w ramach RL: agent wchodzi w interakcję z środowiskiem poprzez podjęcie akcji  $a_t$ , ta natomiast prowadzi do zmiany stanu środowiska na  $s_t$  oraz otrzymaniem nagrody lub kary  $r_t$  przez agenta za podjętą akcję – proces ten następnie powtarza się w kolejnych iteracjach.

Za źródłem [1] z zalecanej literatury, w ramach RL program komputerowy odpowiedzialny za podejmowanie skomplikowanych decyzji w warunkach niepewności nazywany jest agentem (ang. *agent*). Zgodnie z zawartym w [1] przykładem: jeśli trenowany jest program do sterowania ramieniem robotycznym w celu podnoszenia przedmiotu to samo w sobie ramię nie stanowi części agenta (pomimo, że agent podejmuje decyzje na nie wpływające) – jedynie część kodu odpowiedzialna za podejmowanie decyzji określana jest jako agent.

Następnym podstawowym pojęciem jest środowisko (ang. *environment*) – wszystko co znajduje się poza agentem jest środowiskiem. Kontynuując przykład ramienia robotycznego z [1] – zarówno obiekt który ma być podniesiony, jak i podłoga na której obiekt się znajduje, tak samo

jak samo ramię jest częścią środowiska. Choć agent ma kontrolę nad ramieniem i może je poruszać to, to tym bardziej ze względu na istnienie zakłóceń oraz nieidealne odwzorowanie decyzji agenta na stan ramienia, ramię jest częścią środowiska.

Niestety, pełna informacja dotycząca stanu środowiska nie może być w praktyce wykorzystana (po części również dlatego, że nie cały stan środowiska ma znaczenie w ramach zadanego problemu). Przez co samo środowisko operacjonalizuje się w postaci zmiennych liczbowych je opisujących w postaci stanu (ang. *state*). Zbiór wszystkich możliwych stanów tworzy przestrzeń stanów (ang. *state space*). Kontynuując przykład ramienia robotycznego w ramach pojedynczego stanu można zawrzeć informację o dokładnej lokalizacji ramienia, prędkościami przegubów, kątów pomiędzy przegubami, lokalizację obiektu który ramię ma podnieść. Stanu środowiska które agent jest w stanie bezpośrednio odbierać w postaci danych wejściowych nazywana jest obserwacją (ang. *observation*) – co ważne informacja w ramach obserwacji może być niepełna lub być zniekształcona. Kontynuując przykład z [1], agent może nie mieć dostępu do bezpośrednich wartości lokalizacji czy prędkości ramienia robotycznego, a jedynie dostęp do wartości pikseli z obrazu kamery skierowanej na ramię robotyczne oraz obiekt który należy podnieść.

Formalny zapis problemu w ramach RL wymagałby wykorzystania *procesu decyzyjnego Markowa* – jednak nie będzie to konieczne w kontekście laboratoriów, zainteresowanych tematem zapraszam do literatury: [1, 2, 3].

## 2 Zadania

W ramach zajęć należy w zespołach wykonać następujące zadania:

1. Zapoznanie się z narzędziami `Google Colab`, `Gym`, `Stable-Baselines3` oraz `gnwrapper`.
2. Sprawdzenie różnych algorytmów w środowisku `Lunar Lander`.
3. Wizualizacja postępu procesu uczenia dla dowolnego innego środowiska i wizualizacja w formie wykresu.

Zadanie nr 1 nie jest na ocenę, ale jest obowiązkowe do zaliczenia zajęć. Za wykonanie zadania nr 2 jest ocena maksymalnie dostateczna plus (3.5). Za zadanie nr 3 jest maksymalnie +1.5 do oceny. Przypominam o przesłaniu linku do `google colab` z wykonanym zadaniem

## 3 Opis zadań

### 3.1 Zadanie 1

W ramach zadania należy zapoznać się z przygotowanym programem w `Google Colab` oraz z dokumentacją bibliotek. Proszę potestować różne parametry i następnie zobaczyć ich wpływ na proces uczenia w dowolnym środowisku.

### 3.2 Zadanie 3

Celem zadania jest skuteczne przeprowadzenie treningu w środowisku `Lunar Lander` z użyciem wybranego algorytmu do uczenia agenta. Jako efekt końcowy należy pokazać nauczonego model, który poprawnie ląduje w wyznaczonej strefie między flagami.

### 3.3 Zadanie 2

W zadaniu należy wybrać dowolne inne środowisko niż w zadaniu drugim. Ponownie należy sprawdzić działanie różnych algorytmów. Jednak w tym wypadku należy narysować wykres (preferowana biblioteka `matplotlib.pyplot`) z poziomu `Google Colab` z przebiegiem procesu uczenia w zależności od liczby epizodów dla wybranego algorytmu. Warto w pierwszej kolejności dostroić odpowiednie parametry.

```
1 import matplotlib.pyplot as plt
2 x = [i for i in range(1,11)]
3 y = [i**2 for i in range(1,11)]
4 plt.plot(x, y, label="Audio 1")
5 plt.legend()
6 plt.xlabel("x")
7 plt.ylabel("y")
8 plt.show()
```

Na powyższym listingu pokazano prosty sposób użycia biblioteki `matplotlib.pyplot`.

## Literatura

- [1] M. Morales, *Grokking deep reinforcement learning*, Manning Publications, 2020
- [2] R. Sutton and A. Barto, *Reinforcement Learning, second edition: An Introduction*, Adaptive Computation and Machine Learning series, MIT Press, 2018. [link](#)
- [3] Darmowy kurs na stronie Kaggle <https://www.kaggle.com/learn/intro-to-game-ai-and-reinforcement-learning>