

## Sztuczna inteligencja

### Ćwiczenia 4

(tydzień rozpoczynający się 23 maja)

Każde zadanie warte jest 1 punkt.

**Zadanie 1.** W grze Reversi pewne pola są *bezpieczne*, tzn. raz zajęte nie zmienia już w danej grze przynależności. Oczwistym przykładem są pola narożne. Jeżeli jednak o bezpieczeństwie pola będziemy mówić w kontekście konkretnej sytuacji na planszy, to pola narożne są potencjalnie niejednymi bezpiecznymi: przykładowo, jeżeli czarny ma zajęty narożnik, to bezpieczne dla czarnego są dwa przylegające do tego narożnika pola brzegowe.

- a) Jak wykorzystać wiedzę na temat bezpieczeństwa pola w tworzeniu heurystycznej funkcji oceniającej?
- b) Zdefiniuj precyzyjnie warunek bezpieczeństwa pola (w sposób umożliwiający napisanie programu sprawdzającego bezpieczeństwo pola).
- c) Zastanów się, jak w praktyce najlepiej sprawdzać bezpieczeństwo pola, tak żeby funkcja heurystyczna nie zwolniła zbyt wiele. Czy warto definiować jakąś podklasę pól bezpiecznych, które da się szybko wyznaczyć?

**Zadanie 2.** Gry karciane są dobrym przykładem sytuacji, w której nie tylko występuje element losowy (rozdawanie kart), ale również gracz musi podejmować decyzje w sytuacji, w której nie zna w pełni stanu gry (nie wie, co otrzymali inni gracze). Gdy programuje się takiego agenta, często używanym pomysłem jest wielokrotne losowanie możliwego stanu gry (czyli bieżącego układu kart), znajdowanie najlepszego ruchu w tym stanie (używając standardowych metod rozwiązywania gier z jawnym stanem) i ostateczny wybór ruchu, który był najlepszy w największej liczbie losowań.

Wyjaśnij następujące kwestie:

- a) Co może oznaczać: losowanie **możliwego** stanu?
- b) W jakich sytuacjach losowanie z poprzedniego punktu chcielibyśmy wykonywać przypisując stanom niejednakowe prawdopodobieństwa? Co można w ten sposób uzyskać i jakie to rodzi problemy? Uwaga dla osób mało grających w gry karciane: wiele gier ma element licytacji, w której gracze deklarują (często wielokrotnie) jaki wariant przyszłej rozgrywki wydaje im się odpowiedni.
- c) Jaki istotny aspekt gier karcianych jest pomijany w tym podejściu?

**Zadanie 3.** Rozważmy grę oszust (zob. Cheat\_(game) w Wikipedii). Gramy standardową talią 52 kart (asy są najniższymi kartami). Mamy  $k$  graczy, którym rozdane są wszystkie karty. Zaczyna rozdający, po czym gracze na zmianę „zagrywają” od 1 do 4 kart (przy czym zagrywają je koszulkami do góry, tak że nie są widoczne dla innych graczy). Celem jest pozbycie się wszystkich kart. Gracz zagrywając kartę (karty) mówi o ich wartości (na przykład mówi: zagrywam dwie siódemki). Można deklarować jedynie karty o równej wielkości, dodatkowo deklaracja powinna być starsza (bądź równa) deklaracji poprzedniego gracza, czyli na „dwie siódemki” można rzucić „trzy dziewiątki”, ale nie odwrotnie<sup>1</sup>. Gracze mogą kłamać odnośnie tego, co rzucili. Po każdej zagrywce jest faza sprawdzania, w której kolejni gracze mogą spasować lub sprawdzić zagrywającego (ta faza albo zawiera  $k - 1$  pasów, albo pewną liczbę pasów i pierwsze sprawdzenie). Jeżeli zagrywający kłamał, to zabiera wszystkie karty<sup>2</sup>. Jeżeli zagrywający powiedział prawdę, to karty bierze sprawdzający. W obu przypadkach kolejny gracz kontynuuje rozgrywkę (ponieważ stos jest pusty, może wyrzucić karty o dowolnej wysokości, oczywiście wyrzucając karty może skłamać o ich wartości).

Zaproponuj dwóch przykładowych prostych agentów, grających w tę grę (dozwolone są tylko takie idee, co do których masz przekonanie, że dadzą lepszą grę od agenta w pełni losowego).

**Zadanie 4.** Potestuj [playground.tensorflow.org](https://playground.tensorflow.org). Odpowiedz na pytania:

- a) Dla jakich zbiorów danych i jakich cech wystarcza 1 neuron do poprawnej klasyfikacji? (i dlaczego)
- b) Co dzieje się, gdy dla bardziej złożonych sieci damy zbyt dużą *Learning rate*?

<sup>1</sup>W starszeństwie nie ma znaczenia, ile kart rzucamy, jedynie to, jakie są ich wysokości

<sup>2</sup>Zasady gry tego nie precyzują, ale możemy przyjąć, że gracz zabiera karty zachowując ich kolejność i jeżeli ma odpowiednio dobrą pamięć, to może po fakcie sprawdzić, kto kłamał, a kto mówił prawdę w podczas budowy tego stosu.

- c) W którym zadaniu przydają się cechy  $\sin$  i  $\cos$ ?
- d) Dla każdego zbioru danych (oprócz spirali) powiedz, jaka najprostsza<sup>3</sup> sieć neuronowa korzystająca tylko z cech  $x_1$  i  $x_2$  poprawnie klasyfikuje ten zbiór danych.

**Zadanie 5.** Rozważamy sieć neuronową z prostą funkcją schodkową w roli  $\sigma$  (równą 1 dla liczb dodatnich, 0 w przeciwnym przypadku). Wejściami do tej sieci będą zera i jedynki, zatem sieć będzie obliczała jakąś funkcję boolowską.

- a) Podaj sieci neuronowe (złożone z jednego neurona) obliczające  $x \vee y$ ,  $x \wedge y$ ,  $\neg x$ .
- b) Podaj sieć neuronową dla  $x \text{ xor } y$
- c) Uzasadnij, że nie jest możliwa sieć z punktu b), która ma tylko 1 neuron

**Zadanie 6.** Rozważamy takie same sieci, jak w poprzednim zadaniu. Czy za pomocą sieci neuronowych można wyrazić dowolną funkcję boolowską? Jaka jest minimalna liczba warstw, która wystarcza (zakładamy, że neurony mogą mieć dowolną liczbę wejść).

**Zadanie 7.** Pokaż, że jeżeli graf MDP jest acykliczny, to (również dla  $\gamma = 1$ ) algorytm *Value Iteration* (Bellmana) znajduje optymalną politykę. Pokaż, jak znaleźć tę politykę w krótszym czasie.

**Zadanie 8.** ★ Co to jest nadracjonalność (superrationality)? (znajdź odpowiednie informacje w Internecie)

**Zadanie 9.** ★ Podobnie jak w poprzednim zadaniu, powinieneś posilkować się samodzielnie znalezionymi informacjami. Wyjaśnij, co to jest punkt równowagi Nasha. Opowiedz, na czym polega gra w Dylemat więźnia i jaki jest dla niej punkt równowagi Nasha. Jak twórca agenta grającego w tę grę mógłby wykorzystać to, że prawdziwe są następujące fakty:

- i) Jest wielu graczy, każdy gra w tę grę wiele razy, w różnych parach.
- ii) Każdy gracz przedstawia się przed rozgrywką swoim unikalnym identyfikatorem
- iii) Liczy się sumaryczny wynik wielu rozgrywek.

---

<sup>3</sup>Mająca najmniej warstw i (w drugiej kolejności) najmniej neuronów.