

Spis treści

1. Sztuczna Inteligencja 9
 - 1.1. Kilka zdań o historii sztucznej inteligencji 11
 - 1.2. Wiedza - pozyskiwanie, reprezentacja i przetwarzanie 14
 - 1.3. Wnioskowanie w systemach komputerowych 15
 - 1.3.1. Dedukcja - wnioskowanie poprawne logicznie 15
 - 1.3.2. Indukcja - od szczegółu do ogółu 16
 - 1.3.3. Abdukcja - próba wyjaśniania przyczyn tego, co widzimy dookoła 17
2. Uczenie się maszyn - wybrane zagadnienia 19
 - 2.1. Definiowanie pojęcia uczenie 19
 - 2.2. Indukcja reguł - krótki przegląd 22
 - 2.3. Uczenie ze wzmocnieniem 25
 - 2.3.1. Definiowanie uczenia ze wzmocnieniem 25
 - 2.3.2. Q Learning 27
 - 2.3.3. Uczenie na podstawie różnic w czasie (ang. temporal difference learning) 28
 - 2.4. Sieci neuronowe - wielowarstwowy perceptron 29
 - 2.5. Algorytmy ewolucyjne 31
 - 2.5.1. Algorytmy genetyczne 31
 - 2.5.2. Programowanie genetyczne 34
 - 2.5.3. Strategie ewolucyjne 35
 - 2.5.4. Programowanie ewolucyjne 36
3. Gry logiczne 38
 - 3.1. Gry logiczne a sztuczna inteligencja 39
 - 3.2. Podejścia i problemy w tworzeniu 'inteligentnych graczy' 42
 - 3.2.1. Drzewo gry 42
 - 3.2.2. Heurystyka 44
 - 3.3. Przestrzeń stanów i algorytmy przeszukiwania jej 47
 - 3.3.1. Algorytm mini-max 47
 - 3.3.2. Algorytm ciec 48
 - 3.3.3. GLEM - Ogólny Liniowy Model Oceniania (Generalized Linear Evaluation Model) 49
 - 3.3.4. ProbCut 55
 - 3.3.5. Multi-ProbCut 56
4. Uczenie się maszyn a gry 58
 - 4.1. Uczenie w grach 58
 - 4.2. Program grający w warcaby Arthura Samuela 60
 - 4.3. Automatyczne odkrywanie cech 62
5. Kółko i krzyżyk - uczenie ewolucyjne 63
 - 5.1. Automatyczna indukcja podprogramów - wytwarzanie reprezentacji wyższego poziomu 63
 - 5.2. Zastosowanie proponowanego rozwiązania w grze kółko i krzyżyk 66
 - 5.3. Uczenie z rywalizacją a kółko i krzyżyk 68
 - 5.4. Ewolucje równoległe - koewolucja 71
 - 5.4.1. Dzielona funkcja przystosowania 71
 - 5.4.2. Dzielony przydział przeciwników 72
 - 5.4.3. Eksperymenty - kółko i krzyżyk i koewolucja 72
6. Backgammon - uczenie ze wzmocnieniem 75
 - 6.1. Zasady gry w backgammon 75
 - 6.2. Łączenie sieci neuronowych i uczenia na podstawie różnic w czasie 77

6.3.	TD-GAMMON	78
6.3.1.	Architektura programu	79
6.3.2.	Metoda trenowania	80
6.3.3.	Eksperymenty z TD-GAMMONem	80
6.3.4.	Efektywnosc stosowanej techniki uczenia	81
7.	Othello - heurystyczne przeszukiwanie przestrzeni stanów	84
7.1.	Gra othello	84
7.2.	Program LOGISTELLO	85
7.3.	Algorytm genetyczny w doborze wag dla funkcji oceniającej - program OTHELLO	87
7.3.1.	Zastosowane metody i rozwiązania	88
7.3.2.	Eksperymenty	90
8.	LOKI - gra w pokera	96
8.1.	Program LOKI	97
8.1.1.	Wersja gry TEXAS HOLD'EM	97
8.1.2.	Wymagania	98
8.1.3.	Architektura programu	99
8.1.4.	Generator trójek	100
8.1.5.	Tworzenie modeli przeciwnika (ang. opponent modeling)	104
8.2.	Skuteczność programu LOKI	106
9.	Warcaby - program CHECKERS	111
9.1.	Zasady gry	112
9.2.	Architektura programu CHECKERS	112
9.3.	Uczenie	115
9.3.1.	Trudności i propozycje ich rozwiązania	117
9.3.2.	Rola głębokości przeszukiwania podczas uczenia	119
9.4.	Możliwości programu CHECKERS	120
9.5.	Rezultaty	125
9.5.1.	Czy sieć się uczy?	125
9.5.2.	Jaki poziom gry reprezentuje sieć	126
9.5.3.	Wpływ parametrów	129
9.6.	Wnioski	130
10.	Podsumowanie	131
11.	Literatura	133

Od autorki

Tematyka tworzenia programów grających w różne gry budzi duże zainteresowanie młodzieży, co mam okazję obserwować w czasie prowadzenia zajęć z takich przedmiotów jak algorytmy genetyczne, czy metody uczenia się maszyn (*Machine Learning*). Tworzeniem programów grających – zwłaszcza uczących się grać, zajmuje się duża grupa pracowników naukowych w różnych ośrodkach, o czym świadczy na przykład specjalny numer renomowanego czasopisma *Artificial Intelligence*, Vol. 134, Issue 1-2, January 2002, wydawanego przez Elsevier Science B.V. Można tam znaleźć 12 artykułów poświęconych tej tematyce. Trudno jest znaleźć opracowanie polskojęzyczne na temat różnych możliwych podejść do uczenia komputera grać w takie gry, jak warcaby, go, backgammon, itp. Studenci na ogół znajdują interesującą ich literaturę w internecie lub w trudno dostępnych, angielskojęzycznych czasopismach naukowych. Dlatego też wydaje się, że niniejsza praca trafi w oczekiwania młodzieży i wypełni istniejącą lukę.

Nasze doświadczenia (w Zakładzie Informatyki na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej) w tym zakresie nie są duże, nie stanowimy zespołu badawczego realizującego taką właśnie tematykę. Są to raczej drobne prace przyczynkarskie, pozwalające weryfikować studentom ich własne pomysły. Zazwyczaj student realizując jakiś program grający – w kółko i krzyżyk na większej planszy, warcaby, czy inna gra, dochodzi do wniosku, że:

1. Zaprogramowanie takiej gry nie jest łatwym zadaniem, nawet, jeśli ograniczymy się do stosowania znanego z literatury algorytmu mini-max i ?? ciec. Pozostaje wtedy do rozwiązania problem, jak ocenić stan planszy. Jeśli nie korzysta się ze sprawdzonych już i dostępnych w literaturze funkcji oceniających, to droga dochodzenia do własnej funkcji, dającej w miarę zadowalające rezultaty, jest stosunkowo długa.
2. Jeśli już uda się dobrze zaprogramować funkcję oceniającą, to przy oddawaniu swojej pracy na zaliczenie student stwierdza ze zdziwieniem: „Ależ przecież ja włożyłem w ten program tylko swoją wiedzę, a on potrafi ze mną wygrywać i to mimo moich wysiłków, żeby mu na to nie pozwolić.”

Zdarza mi się też słyszeć przy oddawaniu pracy inne stwierdzenie, coś w rodzaju: „Cały mój wysiłek poszedł na marne, ten mój program jest całkiem głupi, nawet nowicjusz jest w stanie sobie z nim poradzić, mimo iż tak bardzo starałem się uczynić go mądrym. Ale przynajmniej przez cały semestr świetnie się bawiłem spędzając czas nad tym projektem.” Tak mówią głównie ci, którym nie udało się stworzyć dobrze grającego programu.

Biorąc pod uwagę powyższe, wydaje się, że warto spróbować ułatwić nieco dostęp do wiedzy z zakresu tworzenia gier metodami wykorzystującymi uczenie się maszyn. Nie jest moim zamiarem napisanie podręcznika wprowadzającego krok po kroku w zawilosci dziedziny, ale chciałabym pokazać, jak ciekawa, ale jednocześnie rozległa i trudna jest ta dziedzina. Możliwe, że praca ta zainspiruje do opracowywania własnych metod uczenia komputera grać – w zadaniu tym inwencja twórcy programu jest bardzo ważna, dlatego też taki przegląd metod zastosowanych do różnych gier może być inspirujący do własnej pracy. Przekazywana czytelnikowi praca przedstawia zarówno uznane osiągnięcia ze świata, jak też i prace powstałe w Zakładzie Informatyki Politechniki Wrocławskiej w ramach prowadzonych tu zajęć dydaktycznych.

Uczenie się maszyn (ang. *Machine Learning*) jest gałęzią szerszej dziedziny – sztucznej inteligencji (ang. *Artificial Intelligence*), która w całości budzi stosunkowo duże zainteresowanie młodzieży studiującej. Biorąc pod uwagę, że tworzeniem programów grających może się zainteresować wiele młodych osób, które o sztucznej inteligencji nie słyszały nic, w pierwszym rozdziale zamieszczone jest krótkie wprowadzenie w tę tematykę. Przedstawiono w nim definicje dziedziny (zasygnalizowano też pojawiające się dość silne kontrowersje wokół definiowania sztucznej inteligencji) oraz naszkicowano krótko historię jej rozwoju. Nie poświęcono wiele miejsca perspektywom jej rozwoju, zasygnalizowano je jedynie na rysunku. Uważny czytelnik znajdzie informacje gdzie i czego szukać, jeśli zechce poznać przyszłe kierunki rozwoju sztucznej inteligencji.

Sztuczna inteligencja nierozzerwalnie związana jest z wiedzą i jej przetwarzaniem – wnioskowaniem. Uczenie to głównie wnioskowanie indukcyjne, wydaje się, że i te tematykę warto krótko przedstawić, co też uczyniono w drugiej części pierwszego rozdziału.

W drugim rozdziale przedstawiono wybrane zagadnienia z maszynowego uczenia się. Czytelnik nie znajdzie tu kompendium wiedzy na temat algorytmów uczenia się maszyn, traktuje o tym książka P. Cichosza [25], która liczy blisko 900 stron a mimo to zawiera „Listę nieobecności” ([25], rozdział 14.2) – zdaniem autora, pomija ona niektóre metody, takie jak sieci neuronowe, algorytmy ewolucyjne czy zbiory przybliżone. Tym bardziej nie zakładamy w tej pracy wyczerpania tematu maszynowego uczenia się. Jednakże, aby niniejsza praca mogła mieć większe grono czytelników, po bardzo krótkim wprowadzeniu, opisano nieco

szerzej tylko te metody, które będą wykorzystywane w dalszych rozdziałach, poświęconych konkretnym przykładom programów grających.

Rozdział trzeci wprowadza w problematykę gier i przybliża podstawowe podejścia do tworzenia programów grających – drzewo gry, przestrzeń stanów, metody przeszukiwania, funkcje heurystyczne do oceny stanów planszy, itp. Przedstawione tu pojęcia są konieczne do zrozumienia samej idei programów i algorytmów grających.

W czwartym rozdziale skupiono się na połączeniu gier i maszynowego uczenia się. W zamierzeniu autorki ma on dać odpowiedź na pytanie, co można zautomatyzować tworząc program grający.

Kolejne rozdziały dotyczą programów grających w konkretne gry. I tak, rozdział piąty to kółko i krzyżak – jest tam pokazane wykorzystywanie algorytmów ewolucyjnych. Następna część pracy (rozdział szósty) to słynny TD-GAMMON – gra w backgammona i uczenie ze wzmocnieniem i z opóźnioną nagrodą (kara). Jest to niewątpliwie duże osiągnięcie w dziedzinie uczenia się maszyn, dlatego też trochę więcej miejsca jest temu poświęcone. Kolejny rozdział (siódmy) poświęcony jest komputerowym programom grającym w othello. Przedstawiono tu, poza literaturowymi przykładami, także nasze doświadczenia z szukaniem optymalnego zestawu wag dla funkcji oceniającej stan planszy. Zestaw cech zaczerpnięto z literatury, a do optymalizacji wag wykorzystano – z niezłym skutkiem – algorytm genetyczny. Rozdział ósmy przedstawia problematykę i podejścia do tworzenia komputerowego gracza w pokera. Jest to zadanie stosunkowo trudne, chodzi przecież o ocenę (i przewidzenie) najbardziej prawdopodobnego układu kart.

Rozdział dziewiąty przedstawia efekt pracy mojego magistranta, która zainspirowała mnie do napisania niniejszej pracy. Wykorzystując zgromadzoną wiedzę o różnych metodach automatycznego uczenia się w grach, ówczesny student, Artur Spirydowicz, opracował sposób pozwalający zastosować uczenie ze wzmocnieniem i opóźnionymi wypłatami (jak w backgammonie) do deterministycznej gry, jaka są warcaby. Zarówno sposób poradzenia sobie z tym problemem jak i uzyskane wyniki są dość interesujące, dlatego warto to rozwiązanie pokazać czytelnikom.

Oddając czytelnikowi te prace, życzę milej lektury.

Halina Kwasnicka